TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Georgia Kountioudi 184598IASM

Analysis, modelling and prediction of energy consumption in an office building

Master's thesis

Supervisor: Eduard Petlenkov, PhD

Tallinn 2022

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Georgia Kountioudi 184598IASM

Büroohoone energiatarbimise analüüs modelleerimine ja prognoos

magistritöö

Juhendaja: Eduard Petlenkov, PhD

Tallinn 2022

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Georgia Kountioudi

03.01.2022

Abstract

Prediction of energy consumption is crucial for the improvement of energy efficiency in buildings. An accurate forecast can provide valuable information to building operators and managers regarding the expected future consumption. This allows them to gain control over the current and future requirements of the buildings and make any necessary modifications to improve the overall efficiency while significantly reducing costs and environmental impact.

According to information from existing literature, energy consumption predictions are based on historical data which may include information about past energy consumption, weather conditions and geographical information. In addition, current studies, investigate the impact of building characteristics and occupants' behaviour in their models. For years, many researchers have been investigating different techniques that could be used to predict the energy consumption, but none considered superior to others, as each method may have different requirements regarding the data availability and problem.

In this thesis, we will try to analyse and build a prediction model for the energy consumption of an office building located in Portugal. The real-life data used for the research and development of the model were provided by R8 Technologies OU. The available data include datetime, temperature and relative humidity information for the period of 2017-2019. The historical energy consumption measurements are presented in an hourly manner.

Primary focus will be identifying the state-of-the-art methods used for short-term forecasting problems such as Support Vector Machine (SVR) and Artificial Neural Networks (ANNs). These methods were found to be among the most used methods in such projects, which set the basis in this research. In addition to above mentioned techniques, Random Forest (RF) method was used due to its ability to investigate the feature's importance and locate the most significant parameters/inputs. The purpose of this thesis is to train an accurate model that could be used to predict the hourly energy consumption of the targeted office building.

Models trained based on the 80% of the total available data and put into test for the remaining 20%. Common metrics used in regression problems were applied to evaluate model's performance. Metrics included mean absolute error (MAE), mean square error (MSE), mean absolute deviation (MAD), explained variance score and coefficient of determination (R^2 score). In addition to these metrics, the accuracy, based on the mean absolute percentage error (MAPE), was calculated for the models.

According to the outcomes presented below in this thesis, the most accurate prediction model was built with the use of the Random Forest method. Model reached a score of 94.32 % in terms of accuracy. In addition, the best model achieved a MAE of 4.19, MSE of 39.09, MAD of 2.72, and managed to reach the highest explained variance and R^2 score, compared to other models, of 0.93 during the evaluation process. The best results were obtained when training the RF model with the following inputs: temperature, relative humidity, year, month, day of the month, day of the week and the hour of the day.

This thesis is written in English and is 84 pages long, including 5 chapters, 26 figures and 25 tables.

List of abbreviations and terms

MAE	Mean Absolute Error
MSE	Mean Squared Error
MAD	Mean Absolute Deviation
MAPE	Mean Absolute Percentage Error
EU	European Union
CO2	Carbon Dioxide
AI	Artificial Intelligence
SVM	Support Vector Machine
ANN(s)	Artificial Neural Network
MLR	Multi Linear Regression
OLS	Ordinary Least Squares Regression
ARIMA	Autoregressive integrated moving average
DTs	Decision Trees
ML	Machine Learning
LS-SVM	Least Squares Support Vector Machines
MLR	Multilayer Regression
FF-MLP	Feed-forward Multilayer perceptron
RBF	Radial Basis Function
MLP	Multilayer perceptron
DL	Deep Learning
RF	Random Forest
GBT	Gradient Boost Machine
ELN	Elastic Net
ReLu	Rectified Linear Unit
GA	Genetic Algorithm
PSO	Particle Swarm Optimization
GLM	Generalized Linear Model
KNN	K-Nearest Neighbors
SVR	Support Vector Regression

VC	Vapnik-Chervonenkis theory
NN(s)	Neural Network(s)
FFNN	Feed Forward Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
NA	Not Available
HVAC systems	Heating, ventilation, and air conditioning systems
API	Application Programming Interface
ExV	Explain Variance Score

Table of contents

List of figures

List of tables

1 Introduction

Energy demand has increased over the years as the global need for energy is higher than ever. The increase of the energy consumption has become a major issue for the power operators, governments, building managers and end-users as it leads to huge environmental, economic, and social impact generating an absolute need for control. Essential part of the solution to the problem is the improvement of the energy efficiency in buildings as they are consider the largest source of urban energy consumption with nearly 40% of the total EU energy usage [1]. Moreover, optimization of the energy efficiency in the building sector is important to reach EU's targets to reduce the total energy consumption and CO2 emissions by 2050 [2]. Considering the existing high energy demand of the cities, a small increase in the building's efficiency can effectively reduce the total power usage while additional increase of efforts made towards generating more energy from additional renewable resources, could result in additional economic benefits arising from the delay of updating the current power infrastructure [3].

The concept of building benchmarking was created to investigate the energy efficiency of buildings and provide relative information based on the evaluation and comparison of a reference building against its peers. A benchmark is used as an indicative value of the total energy performance of all buildings within a peer group to a reference building [4]. All buildings used for comparison (group and reference), are required to have similar properties to allow proper evaluation and present trusted results. Benchmarking appear to be a powerful tool for the governments and the private sector towards managing the energy consumption, as they may provide realistic targets and means of improving the building's efficiency [5]. Based on prior knowledge of the energy consumption, building operators and managers can apply strategies to further improve the energy demand and supply management [6]. Thorough knowledge and understanding of the building's power need, enable operators' decision-making towards reducing building costs and improving the energy efficiency of buildings. In addition, analysing usage patterns and user's environmental preferences, can contribute on further evolution of the building's performance [7].

Building energy benchmarking could help stakeholders, establish strategies to further optimize their efforts and achieve future goals. Energy consumers could use benchmarking information to improve the efficiency of buildings and reduce costs by identifying the buildings' energy level after comparing it to similar buildings. On the other hand, policymakers could focus on reducing the energy consumption and environmental impact by establishing new policies and regulations reducing the poor-energy performance buildings and providing additional motives to work towards nearly zero-energy buildings [4]. In both cases, accurate prediction of the future energy consumption of a building is required to evaluate the buildings' performance after comparison against its peers.

1.1 Problem Statement

For several decades, researchers engaged on identifying and optimizing techniques that could establish a baseline for solving forecasting problems. Statistical methods, AI techniques or hybrids of the above compose a long list of approaches that could be used to solve forecasting problems. The wide variety of forecasting techniques investigated throughout the years, explains the difficulty presented upon selecting the appropriate technique for a problem, as there is no single method preferred for the prediction of the energy consumption.

Selecting a forecasting method for a problem, depends on several factors. Data availability, size, types, patterns, prediction horizon etc could enable a method to reach good and accurate results while lead others to failure. Literature review presents SVM, ANN, decision trees, and other statistical algorithms as common machine learning algorithms used for training forecasting models.

1.2 Aims and Objectives

Based on the above-mentioned challenges, this thesis aims to analyse and identify a forecasting technique that could solve this problem. The available data and operational requirements of the building are set by the stakeholders. Conducting thorough research on the existing literature will guide us through the project completion. The final aim of this thesis is presenting a prediction model that could accurately predict the energy consumption of an office building.

To achieve the above-mentioned aims, the below presented objectives need to be followed:

1) Research and gain deeper knowledge on short-term forecasting time series methods such as statistical and AI approaches, used for energy consumption.

2) Obtain and analyse the available dataset as given from the stakeholders. Perform Preliminary (exploratory) analysis and any required pre-processing on data.

3) Build, train and test sufficient models for the problem requirements.

4) Evaluate and compare trained models based on accuracy and other typical evaluation metrics, such as coefficient of determination and mean absolute error.

5) Apply further optimization to the best performing model to further improve the performance.

1.3 Thesis Organization

This thesis investigates the importance of prediction models in the energy sector and introduces the most common artificial intelligence techniques involved in the energy consumption forecast. Thesis is organized as follows: 1st Chapter introduces the thesis topic presenting the problem statement and intended aim and objectives. 2nd Chapter provides valuable information on the topic's background, analysing the existing literature and gaining a deeper understanding of the state-of-the-art methods used to solve similar problems. 3rd Chapter gives a short description of the working methodology followed including the necessary tools used to analyse and build the models, such as the selected programming language and required libraries. This chapter dives into the core of the working methodology with step-by-step explanation on concepts such as data pre-processing, feature importance and model training, with detailed description of the different forecasting techniques selected to solve this problem. 4th Chapter presents a detailed representation of the results achieved by different prediction methods and models including a comparison between the performance of the best models of each technique. Finally, 5th Chapter summarizes the efforts made towards providing a solution to the prediction problem and provides ideas and suggestions for future work and further possible improvements to the proposed models.

2 Background

Following this chapter, an investigation of the background of short-term forecasting will take place. The state-of-the-art forecasting techniques used on the existing literature, allows a deeper understanding of the approaches. Research focus on problems associated with energy forecast to provide additional information on the most common used techniques.

2.1 State-of-the-Art

Load forecasting is a well-known topic concerning research in the energy field for years. The variety of methods studied and proposed through the years is huge, starting with simple methods based in statistics until more complex methods such as ANNs which started to get more attention since 1990 [8].

One of the most recent papers reviewed, presented a holistic review of load forecasting in buildings, including different types of buildings, type and size of data, different forecasting horizons, geographical locations, and energy types. Additionally, feature parameters and their usage frequency were presented, by the authors, in prediction techniques since 1990 [9]. Authors categorized the main classes of the prediction methods as physical (white box), statistical, artificial intelligence and hybrid. 39% of the papers reviewed, were about commercial buildings, followed by 28% of residential. In 18% of papers, authors considered the energy consumption as a combination of all types of energy including heating, electricity etc, while in 35% authors considered only the electrical use of buildings. In 61% of the reviewed papers, authors used real data from which 33% were in hourly or sub-hourly time intervals. 39% of the data ranged between one month and one year, while 18% of data ranged above 2 years. The most common features used in studies included weather data such as temperature and relative humidity, followed by historical data of energy consumption and datetime. Finally, the most frequently used techniques were AI-based followed by hybrid and statistical.

Another study focused on data-driven machine learning techniques for building load forecasting is presented [10]. Research involved different types of buildings, size and type of datasets, prediction horizon, features, and prediction methods. Authors categorize the prediction models into two main classes, physical and data-driven (machine learning).

According to [10], 81% of the models were based on non-residential buildings, while 67% of the models used real data. Regarding the sample rate 57% were based on hourly time intervals. In 47% of the papers, authors considered the energy consumption as total (including heating, cooling, lighting etc). Typically used features included weather conditions, building characteristics, indoor environmental conditions and historical data of energy consumption and time. ANNs and SVM techniques, appear to cover 72% of the total existing research articles on building load forecasting with data-driven approaches. Research included papers until the year 2017, where 47% of the models used an ANN method, followed by SVM with 25%. The remaining 24% lies between other statistical algorithms such as MLR, OLS and ARIMA.

Another review based on data-driven approaches is presented in "A review of data-driven approaches for prediction and classification of building energy consumption" [11]. Authors reviewed methods such as ANN, SVM, DTs, genetic algorithm, and statistical regression in load forecasting for buildings. Results concluded that ANNs and SVM methods were the most common used methods in short-term predictions problems. Most popular features used in such methods were, weather conditions (temperature and relative humidity) and historical data of energy consumption. In addition, authors stated that statistical regression can perform well as a feature evaluation method.

Linear regression and fuzzy logic prediction capabilities were compared to ANNs on non-linear models [1]. Their feature selection was based on literature research, which revealed that most relevant features used in AI models were, weather conditions (temperature and relative humidity), time (hour of day and weekday or weekend type), and the building's occupancy condition as estimated from WiFi traffic data. These features were used in different combinations in prediction models. According to model results based on a one-month dataset in 15min intervals, ANNs and Fuzzy logic models reached higher accuracy compared to linear regression models when occupancy information was added to inputs such as weekday, hour and minute. Authors conclude that retraining models with additional data could improve the performance.

The authors of paper "Predicting future hourly residential electrical consumption: A machine learning case study" [12] tested different machine learning techniques on an hourly prediction model for the electrical use of a residential building. ML techniques included linear regression, ANN, SVM, and several models of ANNs, SVM and Fuzzy

clusters combined with expert systems. A combination of K-Folds and Cross Validation was used to identify the best parameters for the models. Model data obtained from sensors within one year period, included information such as weather conditions (temperature), time and past electrical consumption. Models were tested on different datasets and the best performing model was achieved with the LS-SVM method.

Authors of paper "Forecasting electricity consumption: A comparison of regression analysis, neural networks and least squares support vector machines" [13], did a comparison between multilinear regression, ANN and SVM load forecasting methods. Historical data between the period of 1970 and 2009 were used for modelling. Features contained information regarding the total amount of electricity generation, the installed load capacity, the total population of the country and the amount of subscribed energy users within. Interpolation method was used to handle the missing values of the dataset. Authors tested the MLR, LS-SVM, and FF-MLP forecasting models. On the SVM model, the best performing kernel was the RBF, while for ANN models, different learning algorithms were tested. Necessary data pre-processing required by NNs was used to scale all variables within the same range, a process called normalization. According to the final outcomes, authors identify that the LS-SVM model achieved higher performance.

Another case study for short-term load forecasting was completed for Portugal households in [14]. Data from 93 households were used in this study for the period of February 2000 until July 2001, including weekdays and weekends. Authors decided to use the MLP technique with a backpropagation learning algorithm for the training due to their capabilities in short-term predictions. Variables included information regarding the apartment area size, the number of occupants, the consumption used from electrical appliances and historical load data in hourly interval. Additional data pre-processing was required where outliers and missing values were removed from the dataset.

Deep learning algorithms were also used for the 24h ahead cooling load forecasting in [15]. DL algorithms were compared with the performance of most popular techniques found in literature such as MLR, RF, GBT, ELN and SVR. Data from one year period in 30 minutes intervals were used in the models and features included weather conditions (temperature and relative humidity), the cold-water temperature during supply and return and its flow rate, the cooling load, and additional time information such as month, day, hour, minutes, and the day type. Authors identify that ANNs models with *ReLU* (Rectified

Linear Unit) activation functions and additional dropout layers in their architecture, improved the overall prediction performance.

Another NN forecasting technique named LSTM, appeared in research papers of [16] [17] and [18]. Authors of first paper created a model for the week ahead forecast of the energy consumption and compared the performance to models with MLP, RF and SVM techniques. Results showed that LSTM achieved higher performance with a MAPE of less than 3%. The data used in the process were weather-based such as temperature, relative humidity, and wind speed as well as time-based such as hours and days of the week. RF feature importance method was used to identify the most important features for the model. LSTM model also achieved better performance compared to SVR, RF and MLP methods in "LSTM-based Short-term Load Forecasting for Building Electricity Consumption" [17], a paper that presented a model based on weather conditions (temperature and humidity) and hourly historical load data. In addition, authors of "Research on Power Load Forecasting Method Based on LSTM Model" [18], used load data to create a LSTM prediction model and proved that technique could effectively improve the accuracy in load forecasting.

In paper 'Time series forecasting for building energy consumption using weighted Support Vector Regression with differential evolution optimization technique' [19] authors used a weighed SVM technique to solve a time series energy consumption prediction problem. In addition to the SVR algorithm (with RBF kernel), they used a differential evolution technique to optimize and select the model's parameters. Models created to predict half-hourly energy consumption and daily. Available data included time (weekdays and time) information and the half-hourly measurements of the building's energy consumption for a period of one year. Different SVR models were developed using two datasets. First contained daily data of energy consumption for one year period and the second contained half-hourly data during a ten-days period. Results showed that both SVR models achieved higher accuracy compared to other methods such as GA and PSO.

Another SVM model was used for load forecasting in buildings in "Vector field-based support vector regression for building energy consumption prediction" [20]. Authors built a vector based SVR model using weather data and indoor environmental conditions. The data included information for one month period in an hourly manner during summertime.

Authors used a cross validation technique to evaluate the best parameters for the SVR model. Model achieved high accuracy with good generalization abilities.

Additional papers of recent efforts towards load forecasting, investigate the Random Forest method. Data for outdoor and indoor conditions of an office were collected by sensors and used in modelling from authors of paper "Comparison of model-based and data-driven approaches for modelling energy and comfort management systems, with a case study" [21]. Collected information included the indoor and outdoor temperature, humidity, the CO_2 concentration, and the energy consumption during the day based on hourly intervals. The proposed RF model manage to achieve high accuracy.

An RF model built to predict the energy consumption of 30 office buildings is presented in [22]. Authors used a dataset that included information between the years 2015 and 2017. Data included the weather conditions (temperature, relative humidity etc), building characteristics (height and total area), and datetime information such as month, hour of a day, weekend, and holidays. Authors proved that when RF is used in load forecasting models can reach high accuracy.

The capabilities of the RF models in load forecasting, were investigated and compared with the GLM and ANN methods by authors of "Ensemble learning models for short-term electricity demand forecasting" [23]. The proposed RF model built in 2020, managed to outperform the other models based on the two year and three-month data of electricity demand of a city. Previous attempt in 2019, was limited due to the available data at that time. After the collection of additional data in 2020, authors managed to revisit their approach and built a highly accurate RF model.

A comparison of the performance between the RF, KNN, SVM and MLR models for prediction of the energy demand is presented in "Demand Analysis of Energy Consumption in a Residential Apartment using Machine Learning" [24]. Results conclude that RF method managed to reach the highest accuracy compared to other techniques. Models used one year data of the daily energy consumption and building characteristics such as area, number of rooms and occupancy. In addition, information of indoor and outdoor weather conditions such as temperature, humidity, rainfall and windspeed were included in modelling.

2.2 Forecasting Principals

Forecasting is a method used for predicting future outputs and conditions based on historical past and/or present data. Few examples consist of the prediction of stock prices, sales of a company/product, supply and demand of a service/product/resource, infection rates etc.

There are three main types of forecasts based on the forecasting horizon [25]:

1. Short-term forecast – Required for scheduling by providing information within a small period starting from the next minutes until the next day (in an hourly or half-hourly manner)

2. Medium-term forecast – Required for operational planning by providing information for one day until a few months ahead (in an hourly or daily manner)

3. Long-term forecast – Supporting strategic decisions by providing information of over a year ahead.

To decide which forecasting technique to apply, a consideration of whether past data are available is required. Depending on the existence of data, forecasting approaches can be categorize as [25]:

1. Qualitative – There are no existing data or data appear to be irrelevant to the predictions

2. Quantitative – Data exist, and any past trends may also appear in future predictions.

Quantitative approach is applicable to the forecast problem since it is based on real data which are observed at specified time intervals. Furthermore, this establishes the problem as a time series problem. According to the problem specification, the forecast of the power consumption of an office building is required. The predictions should consider the energy consumption of the following day in 24-hour intervals.

Short-term predictions of energy consumption or as usually referred to as short-term load forecasts, can be achieved through various algorithms. Researchers categorized them into engineering, statistical and artificial intelligence methods [26]. Based on the literature

review, a decision to investigate the state-of-the-art AI methods, was made to solve the load prediction problem.

2.3 Artificial Intelligence Algorithms

Machine Learning is a subclass of Artificial Intelligence which helps machines learn from existing data in a way that imitates the way people learn. A variety of algorithms are used to execute sophisticated calculations and analysis on data to solve complex problems. To succeed in problem-solving, proper training of the machine is required. Training involves finding patterns in data. These patterns will be used by the system to solve a problem on unseen data. This process is referred to as testing.

ML has a very wide range of applications including fraud detection, machine translation, search engines, business intelligence, transportation, banking and insurance, medical diagnostics, marketing, forecasting etc [27].

ML algorithms can be classified into three categories based on their learning approach (Figure 1).

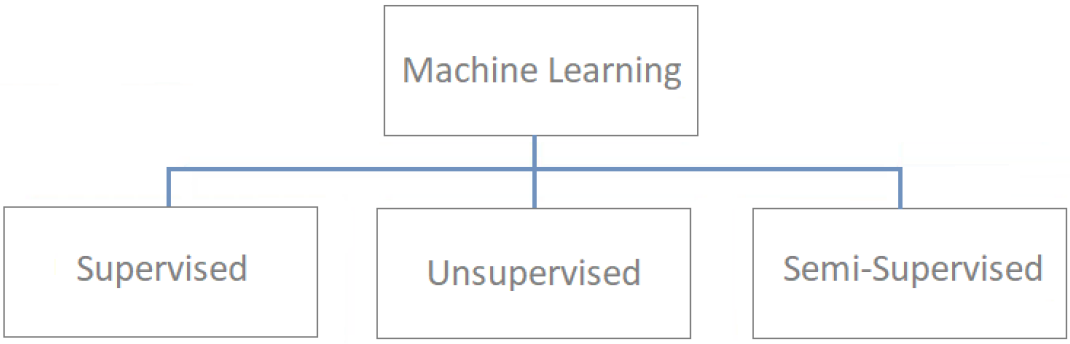


Figure 1. Machine learning approaches.

Supervised learning uses labelled datasets during training. The labelled input data will contain information of the predicted value, meaning it will contain the information about the question and the expected answer to a problem [28]. After the training, the algorithm will be able to recognise and predict the correct answers obtained by matching an input to an output.

Unsupervised learning uses unlabelled datasets. The algorithms use these data to identify patterns based on similarities or differences and makes connections/assumptions to classify the data [28].

Semi-supervised learning combines both techniques. Algorithm uses a small portion of labelled data for classification from a larger unlabelled dataset allowing supervised training with just a few labelled data.

An important subclass of ML called Deep Learning is often used dealing with sophisticated problems. DL simulates the human brain, enabling the system to create complex concepts upon simpler ones allowing completing the clustering or prediction of a data. Algorithms use the information obtained during training to learn and improve their outcomes. Key difference in DL algorithms compared to other ML methods, is that DL methods tend to require less extensive pre-processing in a problem in terms of domain expertise and allow automatic feature extraction, which enables them to locate the most important parameters of a dataset [29]. DL methods depend less in the expert analysis of the data by the people training them and more on their own capabilities to learn from the data and make decisions based on their knowledge to solve a problem such as load forecast. Neural networks are a popular type of the DL methods often used for prediction problems. This will be covered in detail later in the chapter.

2.3.1 Regression

Regression analysis is a statistical technique used to identify trends in data. The basic model of regression is called linear regression and it analyses the relation between two types of variables the dependent and independent (Figure 2). The dependent variable refers to the variable of our interest. This will be analysed, and a prediction will be made. The independent, describes the variables that influence the targeted/dependent variable. Linear regression can be expressed by an equation [25]:

$$y_t = \beta_0 + \beta_1 x_t + \varepsilon_t \tag{2.1}$$

where y is the dependent variable, x is the independent or explanatory variable, β_0 and β_1 are the coefficients of the intercept and the slope of a line and ε is an error describing the deviation from a straight line.

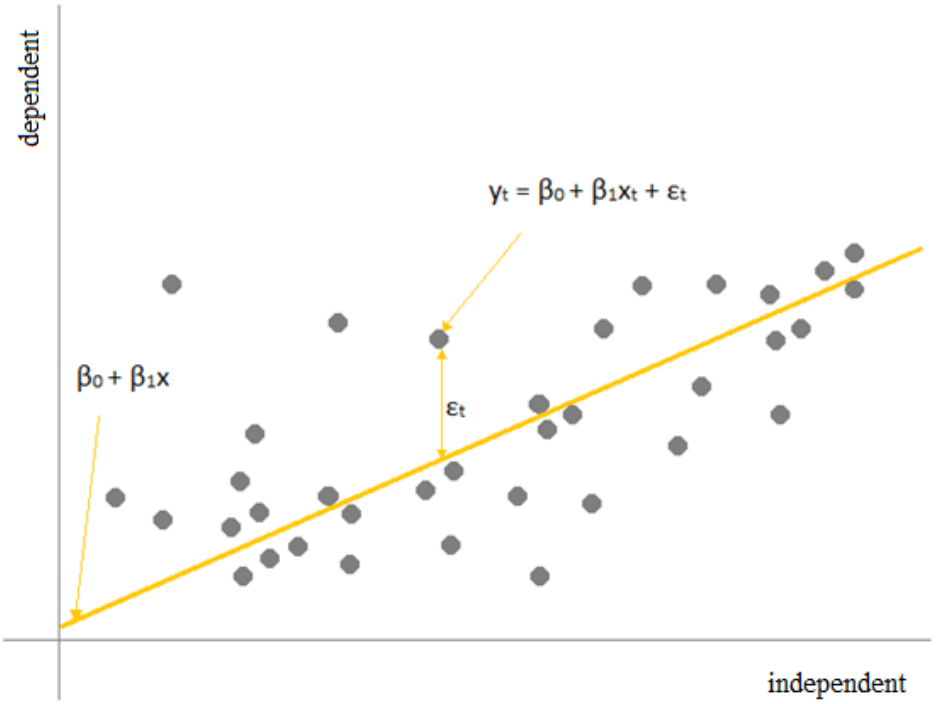


Figure 2. Example of linear regression problem.

In regression problems there is usually one dependent variable and one or more independent variables. When there is only one dependent variable, it's referred to as simple linear regression problem, if there are more than one then it's referred to as multiple linear regression problem. MLR function can be described as [25]:

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \ldots + \beta_n x_{n,t} + \varepsilon_t, \quad (2.2)$$

where y is the dependent variable, $x_1, \ldots, x_n$ are the independent variables, $\beta_0, \ldots, \beta_n$ are the coefficients and ε is the error describing the deviation from a straight line.

Linear regression though, is a supervised learning technique used in ML. Depending on the number of independent variables, simple linear regression or multiple linear regression is used to solve a problem such as load forecasting.

Linear regression is commonly used for the explanatory analysis between the target (dependent) and independent variables, identifying the correlation between them. This allows to locate and compare the most important features (independent variables) to solve a problem.

2.3.2 Decision Trees

Another supervised learning method called decision trees is used in classification and regression problems. The objective of this method is to reach a decision/solution based on a set of rules which containing information obtained through the training process. A simple example of DTs decision making in a classification problem, will be used to understand the processing of the algorithm (Figure 3).

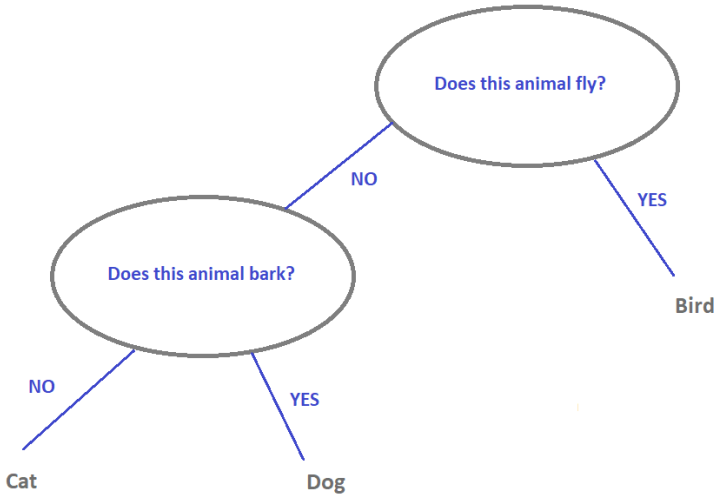


Figure 3. Example of decision trees classification problem.

The first node is called root. Every new data entry will pass to the root node of the tree first. In that node, the algorithm will review the data based on a known rule, in this case (Figure 13) the rule is set in the form of the question " Does this animal fly?" and according to the answer, it will propagate the data to the next node. This process continues until all data are compared and evaluated on each node rule (starting from the root) reaching the final node, called leaf. A leaf contains the final decision/answer to the problem. A similar logic is applied in regression problems. Algorithm decides the numerical value which refers to the answer/solution to a problem. An example of how relative humidity impacts the energy consumption value is presented (Figure 4).

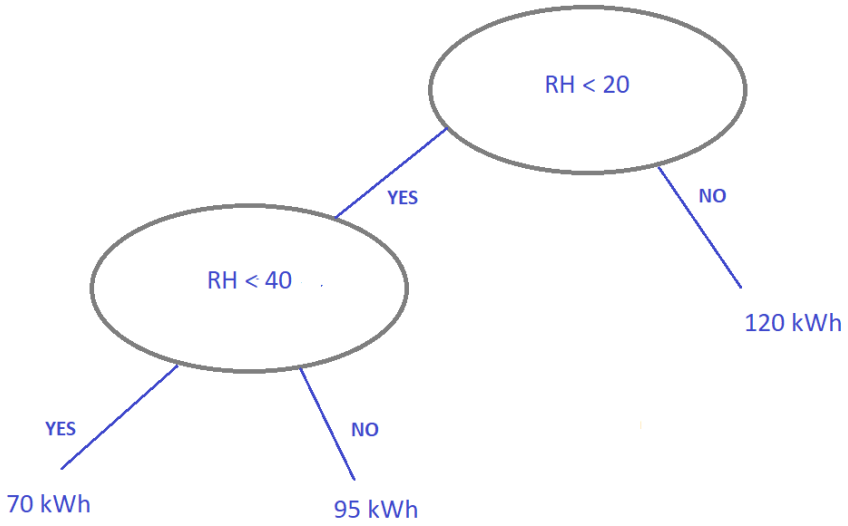


Figure 4. Example of decision trees regression problem.

Ensemble learning techniques, improve their solving solutions by building multiple models, such as DTs, which are used to "cast votes" resulting to an improved final answer/decision on a problem. Ensemble methods, usually achieve better results due to their ability to cover for errors occurring in decision making process [30]. Errors are described by bias and variance. Bias describes the algorithm's ability to identify and understand the connection between inputs and outputs, while variance describes the change of the algorithms' performance while training on unseen data. Ensemble models are more robust compared to the single models such as decision trees, as they tend to decrease the variance by increasing the bias, leading to an overall increase of their performance and accuracy [30].

Random Forest is a supervised ensemble learning technique built from multiple decision trees and used for both classification and regression problems. RF technique can be considered an improvement on the decision-making process of the DTs as with this technique the "winning" vote/decision will be based on many DTs. Each individual decision tree will provide an answer/solution to a problem such as load forecasting. Some answers may be the same and some may be different across the different DTs as they depend on each's DT training and learning capabilities, but all will be considered to identify the answer given more often. The answer gaining the most "votes" will be the final decision/answer to a classification problem, while on regression problem the decision comes from the mean value of the given decisions/answers.

2.3.3 Support Vector Machine

SVM is a supervised learning model used for classification or regression problems. When the SVM technique is used on regression problems, its's usually renamed to SVR [31]. The method is based on statistical frameworks known as VC theory which was developed by Vladimir Vapnik and Chervonenkis [32]. The aim of the SVM algorithm is to classify the data points to separate distinguish groups in a n-dimensional space. The simple classification example based on a 2-dimensional space (Figure 5) will be used to understand the way SVM technique works to arrange the data points into separate classes.

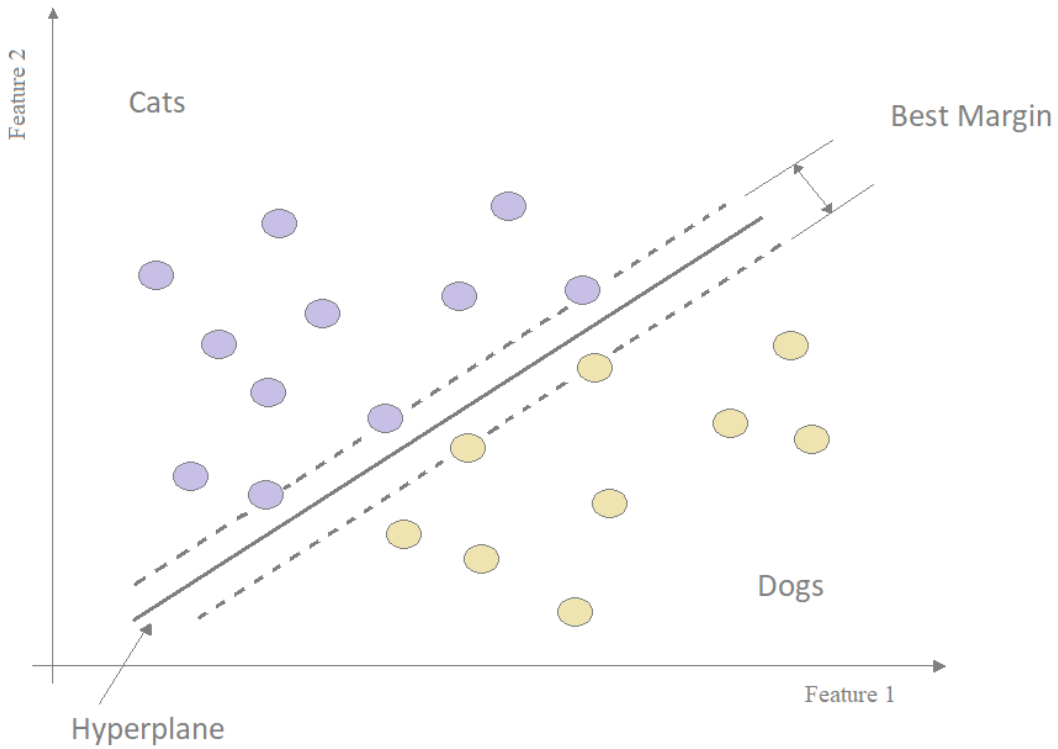


Figure 5. Example of support vector machine problem.

Cats and Dogs represent the two available classes, and the purple and yellow dots represent the data points. The purple dots represent a class of cats and the yellow ones a class of dogs. The straight line between them is called a hyperplane and is used to separate the dataset into those two different classes (Cats, Dogs). The two dashed lines left and right of the hyperplane are called support vectors and are used to mark the data points of each class which lie closer to the hyperplane. This creates a distinguished separation of the data points depending on the class/category they belong. SVM aim is to find the shortest possible distance, called margin, between the hyperplane and the support vectors

so when a new data point arrives, algorithm will be able to classify it in an available class based on the side it lands.

2.3.4 ANNs

As described earlier in this chapter, NNs are part of the DL class. They were initially inspired by the way human neurons work and process information. Humans make decisions based on information collected through different factors such as past knowledge, experience etc. The decision making is based on a complicated neuron system which applies different weights on each considerable factor. Similar to the human brain, ANN algorithms use nodes stacked in hidden layers to explore information on different factors and apply weights on them. Each node of a layer is linked with every node of the previous and next in order layers. Starting from the input layer, the information stored in the nodes are inputted to the nodes arranged in the next hidden layer after a weigh is applied on their interconnection. The nodes of the hidden layer will receive and then propagate this information to the next layer after applying a new weight on their connection. The process continues until the final node/nodes of hidden layer reach the output layer containing information of the final decision towards the problem. The weights are updated during training where the algorithm learns from located patterns in data after making comparisons of current task/problem with previously known ones, imitating the human brain. A representation of the ANN architecture is presented (Figure 6).

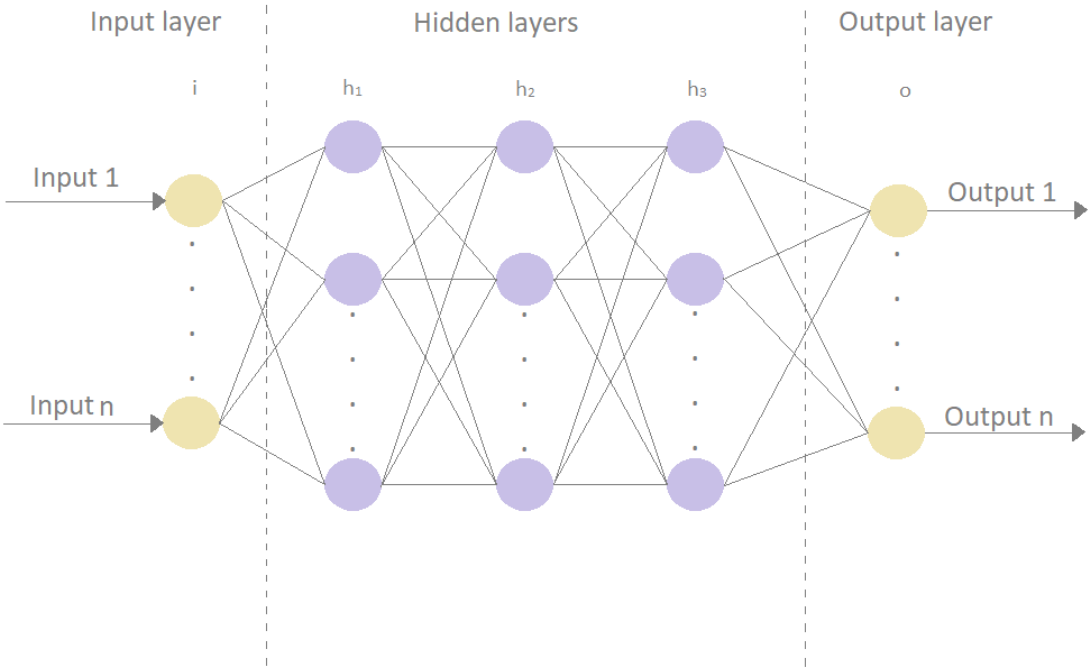


Figure 6. Artificial neural network architecture.

There is a large variety of ANN algorithms used for prediction problems. Algorithms can be divided into two main categories based on the way information pass through the network during the training, a process called propagation:

1. FeedForward - This is the simplest type of NNs, where information propagates in one single and forward direction from the input layer to the hidden layer and then to the output layer. A feedforward NN is also called Multilayer Perceptron (MLP).

2. FeedBackward – This is an extension of the FFNN algorithm, where feedback connections were included to the feedforward model allowing information from the output layer to reach previous layers. Feedback NNs are also called Recurrent NNs.

Backward propagation allows the network to modify the weights of the connected network nodes to further reduce the errors during the training process of an algorithm, often improving the results of simple feedback propagation networks. The errors can be defined by a wide variety of loss functions, such as MSE, MAE, MAPE etc. The selection of loss function and metrics will be covered in more detail in the chapter (3), Methodology and Workflow Implementation.

2.3.5 LSTMs

LSTM is a RNN type of technique which introduces an additional state cell in the RNN architecture. This cell is used as a type of memory, capable of learning long-term dependencies of a sequence in a forecasting problem. This addition, improved previous typical limitations and problems of RNNs, such as vanishing gradients and exploding gradients [33]. The problem of exploding gradients is used to describe the gradient decrease of the influence from an input of a hidden layer while passing back and forth the interconnected nodes. If this influence keeps decreasing through the RNN process it could reach a state where information disappears, creating the so-called problem of vanishing gradients.

RNNs are considered as state-of-the-art algorithms for sequential problems such as time series problems. They use loops to keep important information that could help on providing a solution to a problem. The use of a loop works as a small internal memory within the network. This memory enables the network to learn from the provided information of the current and previous inputs before taking a decision, such as predicting a value. Compared with more traditional feed forward NNs, recurrent NNs tend to be more powerful and robust due to their ability to "remember" past inputs.

Despite the RNNs ability to store important information of previous inputs, as explained above, there are limitations on how much information can be kept through the network. LSTMs were used to address such issues, as they can store and "remember" more information derived from long-term relationships between the nodes which allows the network to take better decisions based on current and past inputs.

A comparison between the traditional ANN, RNN and LSTM algorithm's structure and use of internal and long-term memory is presented (Figure 7).

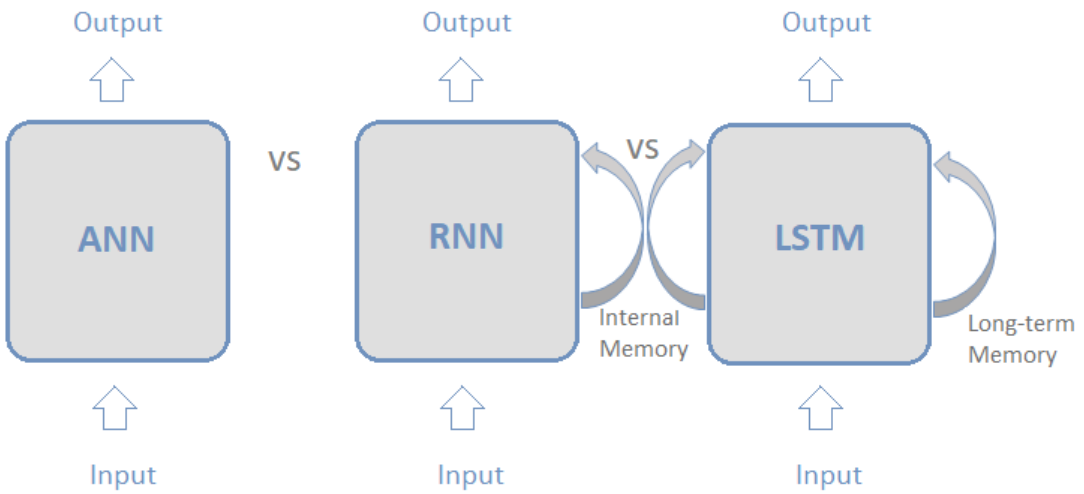


Figure 7. Structural comparison between ANN, RNN and LSTM.

A typical RNN architecture has the form of a chain, based on the way internal loops are used to propagate information from one layer to another (Figure 8).

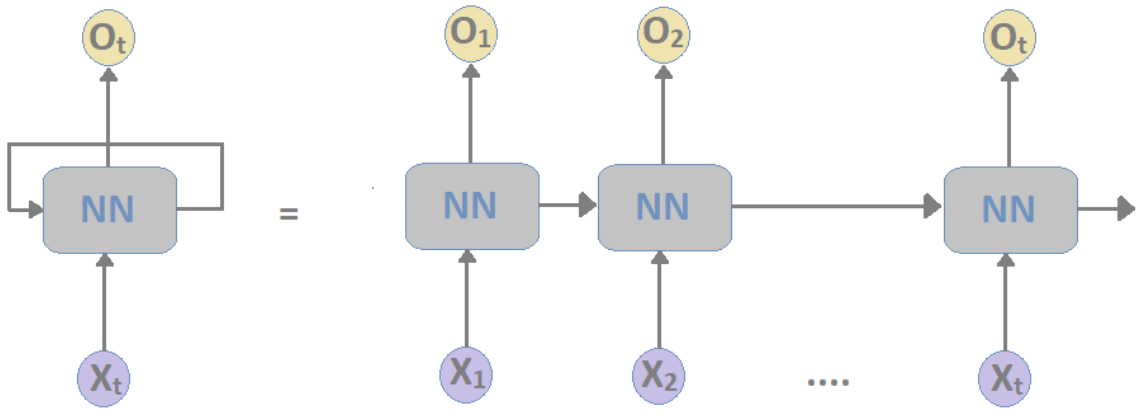


Figure 8. RNN architecture structure.

Within each of these repeating NN modules, there is usually a single layer with an activation function such as *tanh* (hyperbolic tangent). LSTMs have similar architecture with the RNNs, but instead of using single layer NNs as in RNNs, they use four. One of them uses *tanh* such as in RNNs and the other three use *sigmoid* (logistic function) as activation functions. All these are considered as decision gates and help the network remove or add information to the memory cell. The decision gates, allow the LSTM to control the state of the long-term memory cell on which the network is basing its

decision for a problem such as the prediction of the energy consumption, while avoiding issues such as vanishing and exploding gradients.

The above information can be seen in the comparison of the RNN and LSTM structures (Figure 9).

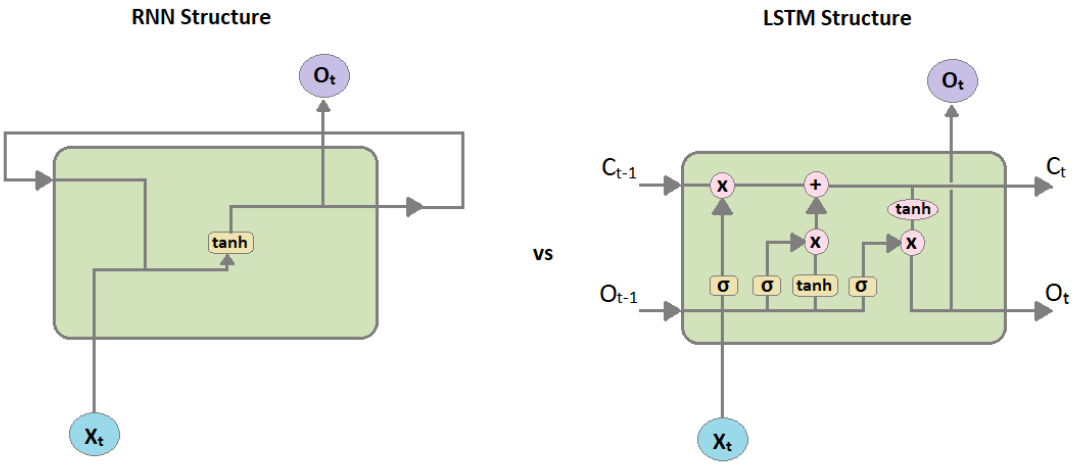


Figure 9. RNN vs LSTM internal gate structure.

The manipulation of the LSTM memory cell is achieved through an internal gate mechanism, which helps the network decide which information to keep and which to remove. The three main gates used are the following:

1. Forget gate removes information that are no longer needed by the network.
2. Input gate adds important information in the cell state for future consideration by the network.
3. Output gate adds additional information in the memory cell for additional consideration.

A small comparison between the forecasting techniques and their advantages and disadvantages [26] is presented (Table 1).

Table 1. Comparison between RF, SVR and ANNs.

Technique	Advantages	Disadvantages
RF	Simple, fast, less prone to overfitting	Less accurate
SVR	Highly accurate	Slow, Higher complexity, require big amount of data
ANNs	Highly accurate and fast, good on solving non-linear problems	High complexity and require big amount of data

RF, SVR and ANNs are the most common techniques used in forecasting energy consumption. Each technique has advantages and disadvantages and their performance depend on the data and problem presented. Methods appear to fit this forecasting problem, thus will be investigated further.

3 Methodology and Workflow Implementation

According to the literature review (chapter 2), load forecasting is identified as a popular problem among the research community. Many researchers tried to solve it by implementing different techniques during the past decades. This concludes that there is no single appropriate approach to solve such problems. Therefore, it is often for researchers to apply and test different methods and compare the performance of different models. The approach helped researchers identify the best performing method for their problems by comparing and evaluating their models through a common set of metrics, thus such approach will be followed for this problem.

A comparison of the most popular machine learning techniques in load prediction will be made. As identified by literature, the most common techniques were SVR and ANNs. In addition to these techniques, the RF method will be used due to its capability to identify the most important independent variables. A method referred to as feature importance or feature extraction [18], [22].

In this chapter, the steps of the working methodology will be described. The forecasting methods chosen for the problem will be discussed. The data preparation required before applying each method and the creation of the training and testing datasets required for the modelling process will be analysed. In addition, the tools used in this project will be presented.

Predictive analysis usually involves seven steps [34] summarized within four categories:

1. Problem Definition and Data Exploration
2. Data Preparation and Analysis
3. Modelling
4. Performance Analysis

Data exploration process starts by uploading data to a program/tool for analysis and identification of important information. Information involves data types, file format, number of rows and columns as well as information of the numerical representation of the data, such as the minimum and maximum value or the average. This information

allows a better understanding of the data and overall problem as during this process the dependent and independent variables and their properties will be identified.

Data preparation and analysis is a necessary step where the cleaning, preparing, and formatting of the data takes place. In this step, the investigation of redundancies that might appear between the target variable and the independent variables will be completed. Identification and correct handling of any missing values or outliers while making sure model does not overfit or underfit is important. Such issue could badly impact the prediction results. Moreover, data patterns and trends will be investigated to understand relationships between the variables and the target variable. Identifying variables with higher influence in the target variable will reveal the most important features for the model.

Overfitting is a problem where the prediction model performs extremely good during the training process but very poorly on new unseen data during the testing and validation process. This problem is caused due to the model memorizing the dataset instead of learning from it and identifying trends and patterns that could allow predictions. On the other hand, underfitting is a problem where the prediction model performs poorly on both training and testing phases. This means that the model was not able to complete its training and learn and identify trends and patterns in the data due to insufficient data.

During modelling, the selection of a prediction method takes place. Data will be fed into a selected model after the relationship of dependent and independent variables and the relation of independent variables with their peers is fully understood. In addition, prior understanding of the desired outcome through the modelling process is also mandatory to succeed.

Final step of the predictive analysis is the evaluation of the model's performance. Results based on a common set of metrics including the model's accuracy will be reviewed. A comparison of the performance between different models will be tested and evaluated as well as the model's overall efficiency. Chapter 4 will present in detail the results of the modelling process.

3.1 Problem Definition and Data Exploration

According to the problem definition, the final goal of this thesis is the creation of a prediction model for the energy consumption of an office building. The model will be built based on real data provided by R8 Technologies OU. The available dataset consists information of datetime in yyyy-mm-dd hh:mm:ss format for a three-year period from January 2017 until December 2019. Data information include weather conditions (temperature and relative humidity) and the historical energy consumption of the office building. Dataset consists of four columns and 26280 rows. There are three main data types identified in the dataset consisting of datetime, floats and integers.

Additional statistical information regarding the temperature, relative humidity and energy consumption columns is extracted from the dataset and presented in the Table 2:

Table 2. Data description.

	Temperature	Relative Humidity	Energy Consumption
count	26280.000000	26280.000000	25534.000000
mean	15.494787	76.883942	78.016871
std	6.446249	23.58409	23.395521
min	-4.000000	9.000000	0.000000
25 %	11.200000	58.000000	60.000000
50 %	15.100000	82.000000	72.250000
75 %	19.100000	100.000000	96.250000
max	40.400000	100.000000	182.750000

Count calculates the number of rows that include data information, meaning rows that are not empty. Mean represent the mean/average value of each column. It calculates the sum of all data points of each variable then divides the total with the number of rows. Min and max represent the minimum and maximum values of each variable. Std calculates the standard deviation of all data of each variable. This provides information of how scatter the data of each column are in relation to the mean value of the variable. Finally, the 25, 50 and 75 percent provide information about how many data values for each column are less than the 25^{th}, 50^{th} and 75^{th} percentiles respectively.

3.2 Data Preparation

Based on the dataset analysis, 746 missing values were identified in the available dataset. 743 of which created a big gap in the energy consumption column. Further investigation concluded that the energy consumption for the month of December of the year 2017 was

missing completely. In addition to this gap, three additional randomly placed points of data were identified missing again in the energy consumption column.

Through the visualization (Figure 10) of the energy consumption during the years of 2017 and 2019, a distinguished gap is formulated from the missing values.

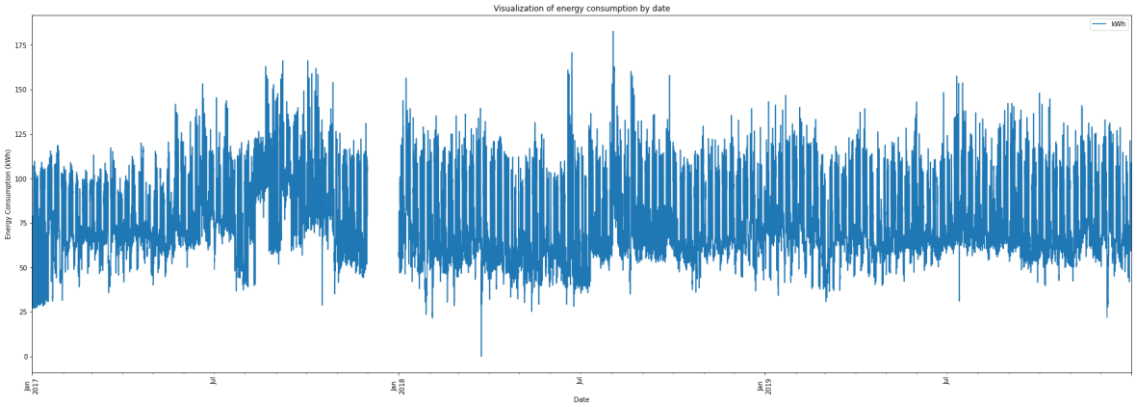


Figure 10. Energy consumption distribution.

Furthermore, a visualization of the monthly energy consumption during the year of 2017 is presented (Figure 11) to better understand the issue of missing so many data points on that period.

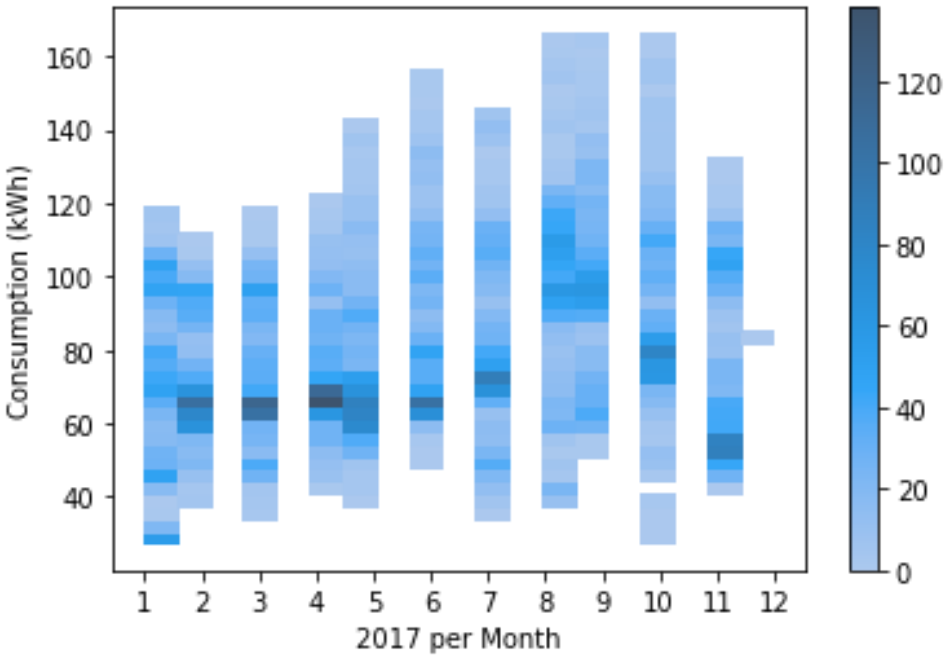


Figure 11. Energy consumption in 2017 with NA values.

To handle such a big gap in the column of energy consumption during December of 2017, simple application of a method such as removing the NA values from the dataset or

replacing with the mean or median value of the column would not be enough. Although these are popular methods for such issues, in this case the problem is that the gap is very big proportionally to the total amount of data possess. For example, if the mean replacement technique was selected, another issue would appear reducing the prediction capabilities of the model. This would be due to a big portion of data that would be left with the same information, in the form of a numerical value within the dataset. Moreover, since the problem is a time series problem, it is important to understand that the next value is based on the previous value. If proceeding with such simple replacement of the missing values for December, it would create a whole month filled with the same repeating value. Such process could be described as noise to the prediction model, making it difficult to understand real patterns from such similar data points.

A simple data removal though, would create a big gap in the time series period which would basically lose the sequence of hourly interval in the data. Although some forecasting methods (e.g., RF models) do not consider/understand such connections in data points of a time series problem, some others do (e.g., ANN). Considering the best approach could allow testing different methods and compare their forecasting capabilities, thus time series intervals should be kept as is.

To avoid such issues and considering that there is no specified uptrend or downtrend in the data pattern, the missing consumption values were replaced with the values of the consumption found in the next year for the same date. The data of December 2018 were used to replace the gap (Figure 12). Then for the remaining three random points, the linear interpolation method was used. This method calculates the missing data values, based on its neighbour data values (previous value and next value). In a case of such a small size of randomly placed missing data points though, any simple method would be valid (e.g., mean, median etc).

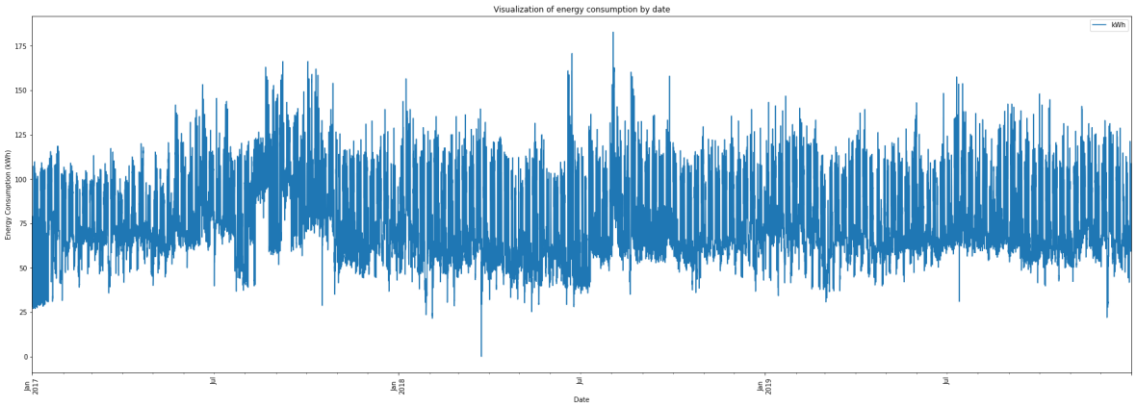


Figure 12. Energy consumption distribution after filling NA values.

Another data preparation technique described in this step of predictive analytics, is identifying and handling data outliers. Outliers refer to the points that are placed far away from the mean value of a variable. A usual rule of mean and standard deviation method identifies a value as an outlier if it lies three standard deviations away from the mean value of the variable. A typical practice when working with prediction problems is to remove such points from the dataset as long the amount to be removed would be small enough. In addition, the removal of such points should not impact the performance of the model.

Before applying such technique, additional investigation of the data should take place. Data should follow a normal distribution (Gaussian distribution) as this would provide additional information during the analysis. For example, some forecasting methods such as SVR or ANNs, require additional pre-processing for the data before starting the training process, such as data scaling. Chapter 3 will further cover the importance of such techniques, but for now it is important to consider the fact that such techniques assume that data observations fit the Gaussian distribution, thus the analysis and examination of this assumption is required. Gaussian distribution is also referred to as normal distribution.

Identifying the normal distribution of the data, could be achieved with different ways. A simple visualization (Figure 13) of the distribution of the energy consumption in a histogram can be used.

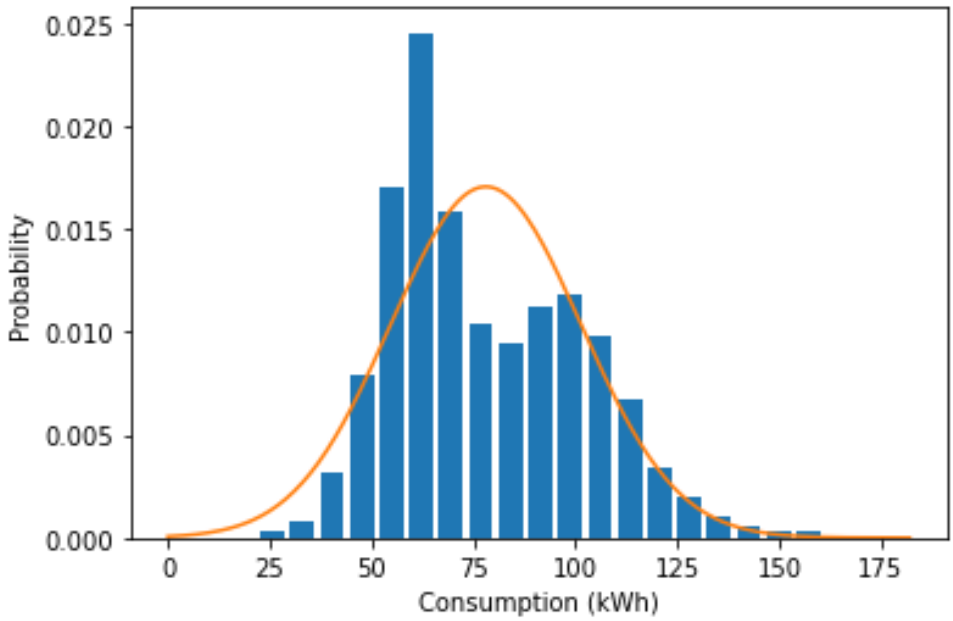


Figure 13. Energy consumption data distribution – histogram.

Results confirm the required assumption, as data follow a normal distribution where most of the data points lay close to the mean value of the energy consumption. The number decreases as we move further from the mean value. This is clearly described with the bell curve of the graph (Figure 13).

After confirming the normality of the data, process continues with the outlier's handling by removing the ones that lay beyond three standard deviations from the mean.

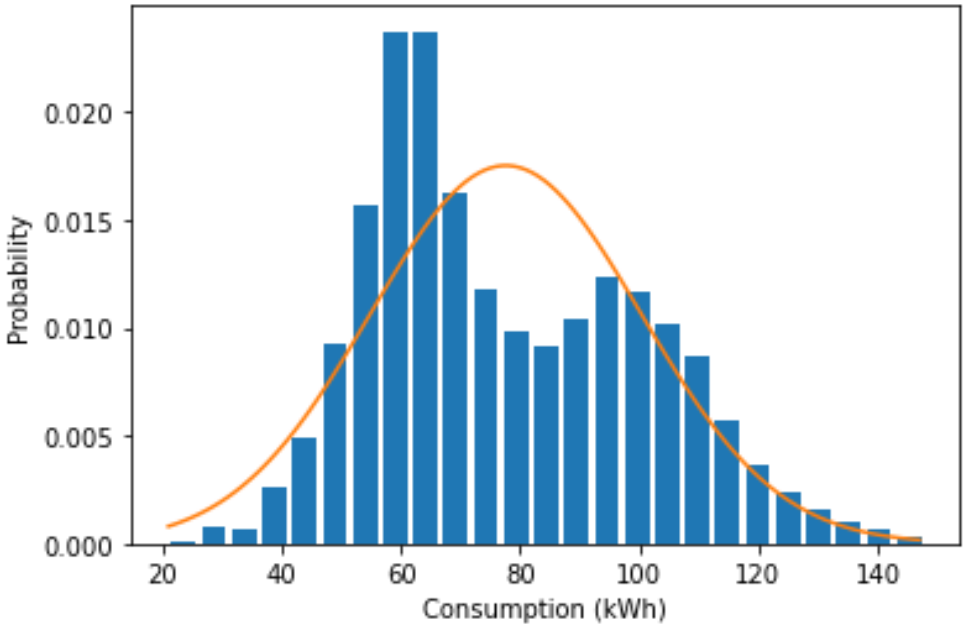


Figure 14. Energy consumption data distribution after outlier handling.

The data range of the energy consumption have changed (Figure 14), placing the data points closer to the mean while following the normal distribution.

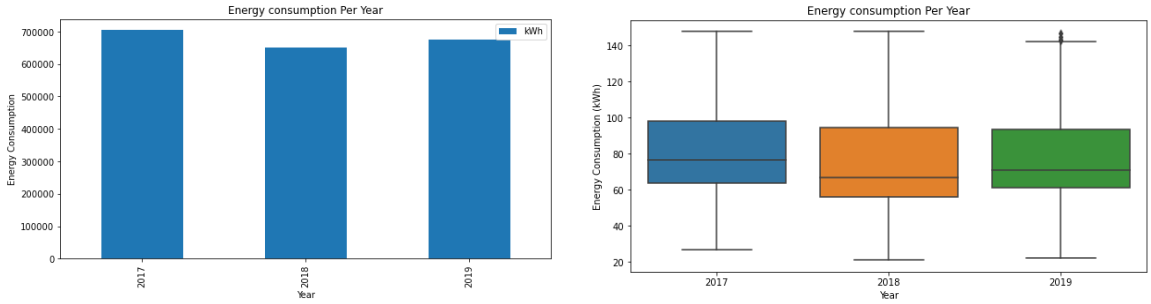
3.3 Data Analysis

Before starting the data analysis, additional information will need to be extracted from the available data of the dataset. As mentioned in the beginning of this chapter, data include four variables: temperature, relative humidity, energy consumption and the datetime information. Datetime column includes information about different parameters, such as year, month, and day. An important step of the data analysis process is to extract all the available information into separate columns and analyse the relationship between them and the target variable (consumption).

In addition, this analysis could provide valuable information for the model through additional investigation of data patterns. Information could be extracted based on the behaviour of a variable during a specified period. For example, investigating the behaviour of the energy consumption during the weekdays and the weekends or comparing the energy consumption during the months of a year. Based on the reasons explained above, information such as year, month, days of the month, days of the week, weeks of a year and the hours were extracted.

Data analysis is an important step. Through this process important trends and patterns are identified in the data, enabling additional knowledge and understanding for the problem and the possible solutions. During this step, different visualizations will be created to interpretate the results of the analysis, providing an overall better understanding of the data relationships and behaviours through graphs. Different graphs such as histograms and boxplots, will be used to provide information regarding the yearly, monthly, weekly, and daily energy consumption behaviour. This process will help on the investigation of the relationship between the energy consumption and the days of the week as well the time of a day. The understanding of how datetime variables influence the energy consumption is necessary as this is identified as target or dependent variable.

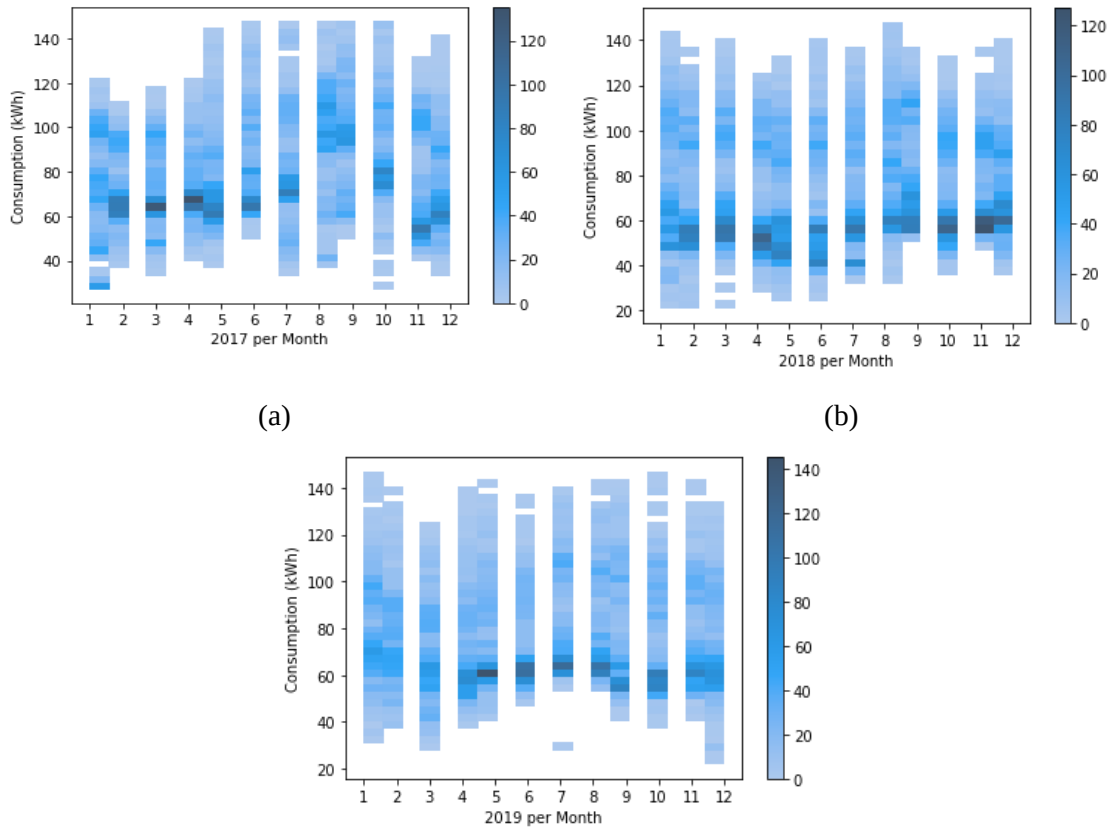
The yearly representation of the energy consumption is presented in the graphs (Figure 15).



(a) (b)

Figure 15. Energy consumption per year: (a) bar-plot, (b) boxplot.

According to the graphs (Figure 15), the energy consumption tends to be lower during the year of 2018 and higher during 2017. An additional monthly comparison for each year is presented (Figure 16).



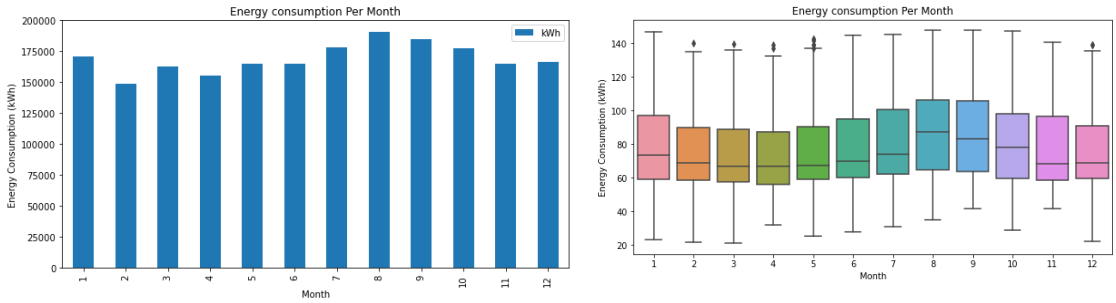
(a) (b)

(c)

Figure 16. Energy consumption per month based on the year.

The monthly energy consumption for the year of 2017 (Figure 16), conclude that during January and April, the energy consumption was lower. Although, such behaviour does not apply for the following years of 2018 and 2019, an assumption that within the specified range, lays the lower energy consumption for both months can be made. Specifically, April had the lowest consumption for 2018 and March for 2019, which allows the assumption of lower energy consumption during wintertime.

Additional analysis between the energy consumption and the months (Figure 17) confirms previous assumption of lower energy consumption during wintertime.

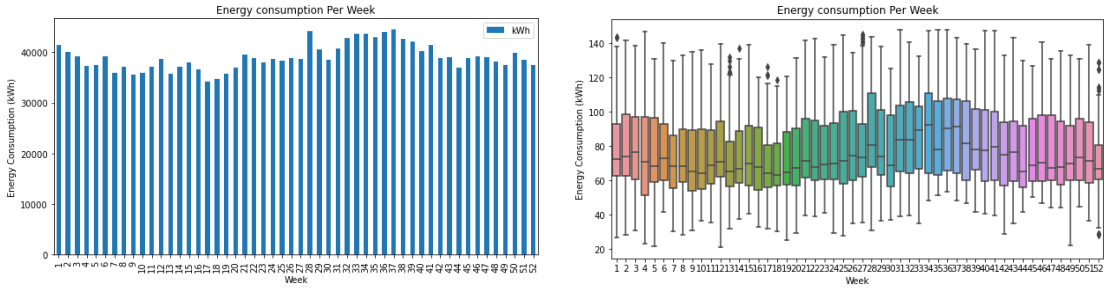


(a) (b)

Figure 17. Energy consumption per month.

Based on the graphs (Figure 17), the low energy consumption trend starts in February and ends in April. In addition, the peak of the energy consumption starts in July and continues until the beginning of autumn between September and October months. Furthermore, August is identified as the month with the highest energy consumption.

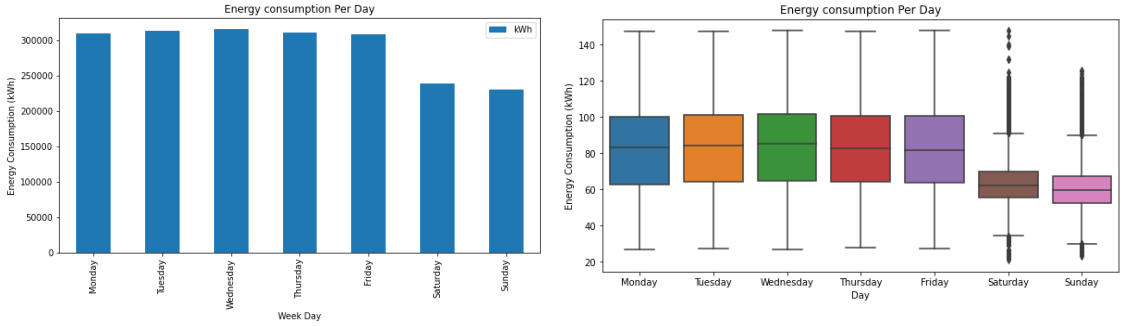
The assumption regarding the peak season of the energy consumption, is also confirmed by the weekly representation of energy consumption (Figure 18). Months with higher temperatures such as July and August have higher energy consumption through the year. In addition, since the office building is located in Portugal, a country located in the south of Europe, high temperature continues in autumn with September and October reaching high energy consumption.



(a) (b)

Figure 18. Energy consumption per week.

When analysing the energy consumption based on the day of the week (Figure 18), energy consumption appears to be very low during the weekend (Saturday and Sunday). This pattern can be explained by the type of the building and purpose of use. As mentioned in the problem definition (Chapter 1), data were collected from an office building. Assuming a typical usage behaviour of office buildings, people usually use the premises during office hours and working days, explain the overall low energy consumption during weekends.



(a) (b)

Figure 19. Energy consumption per day of week.

The office type and the building's purpose can also justify the results and information extracted through the data analysis between the energy consumption and the hours of a day. In the heatmap (Figure 19), energy consumption reaches a peak during the office hours were usually more people attend the office. Building's occupancy starts early in the morning at 07:00 and ends at night-time. The high energy consumption is identified between 09:00 and 17:00, where more people use the office.

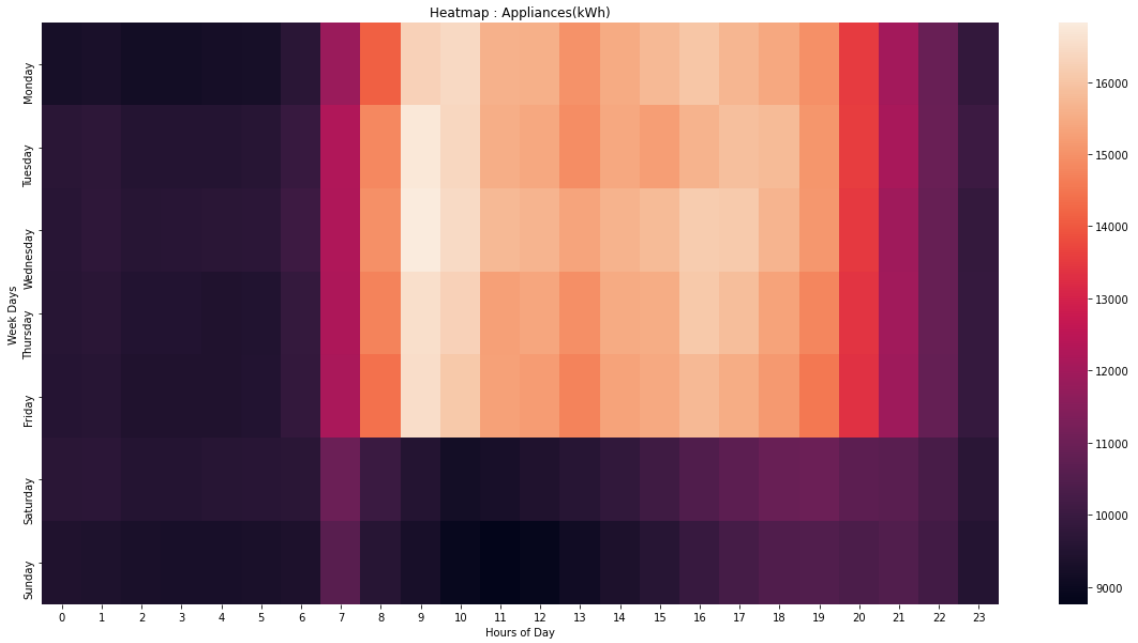
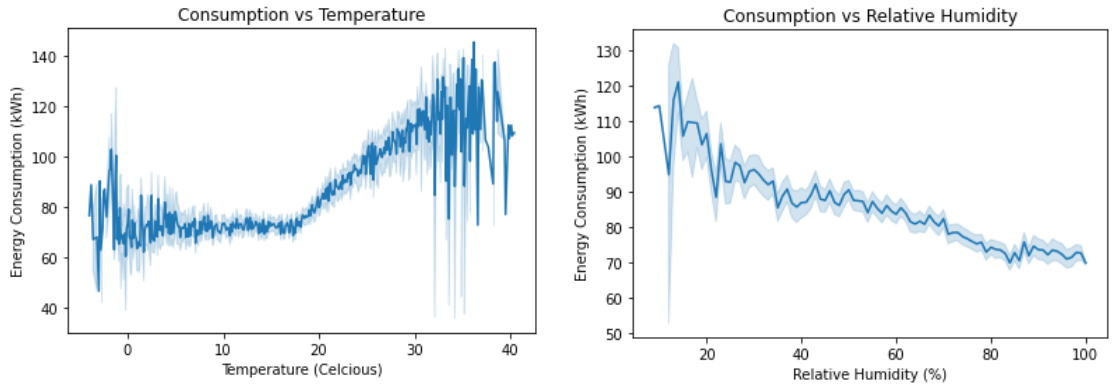


Figure 20. Energy consumption heatmap.

Another important step of the analysis is identifying the relationship between other variables starting with the available weather variables of temperature and relative humidity and then analysing their impact on the energy consumption. A justified connection between the variables is confirmed (Figure 21).



(a) (b)

Figure 21. Energy consumption vs temperature and vs relative humidity.

As relative humidity increases, energy consumption decreases, while energy consumption reaches the peak during higher temperatures (Figure 21). This can be explained based on the actions followed by the office users, where typically in high temperatures people tend to activate the HVAC systems to cool down the office temperature resulting in a high increase of the energy consumption.

Finally, to investigate further the relationships between the available variables (dependent and independent), a correlation table will be created (Figure 22). This can describe the linear correlation between two variables.

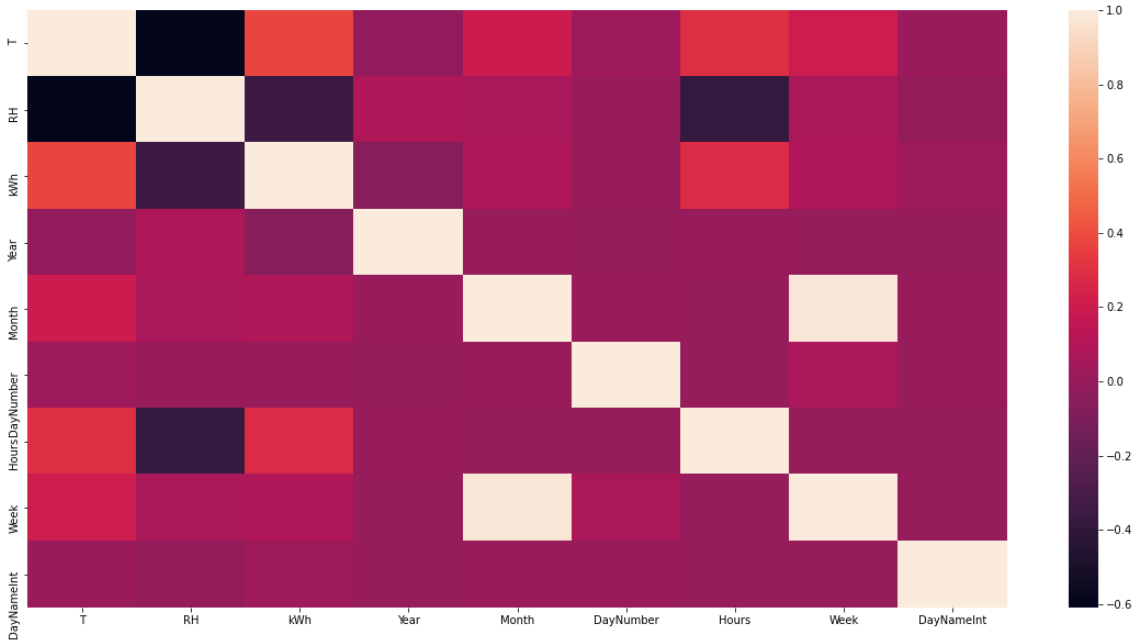


Figure 22. Correlation between variables.

There are two types of correlation, positive and negative. By positive we describe the correlation in the same direction, for example when a variable increase is associated with an increase of another variable. As negative, we describe the correlation in the opposite direction, where a variable increase is connected to the decrease of another. Value closer to 1.0 (or -1.0 for negative correlation), describes high correlation between two variables and higher effect on each other. Typically, variables with almost perfect correlation with a target value need to be left out during the modelling process as they could lead to poor modelling results, due to their high affect.

Based on the results of the correlation analysis (Figure 22), all independent variables are correlated with the energy consumption. None of the variables present very close correlation to 1.0 concluding that none should be left out during modelling. Almost perfect correlation close to 1.0, could harm the model's performance and lead to overfitting issues. In addition, a strong correlation appears between temperature (T) and humidity (RH) which can be explained by the meteorological dependency of these variables. The target variable (kWh) is positively correlated with the temperature and

negatively correlated with the humidity. The high correlation of the energy consumption and the temperature can be explained by the increase of air conditioners use during high temperatures. Also, the target variable (kWh) is positively correlated with the hours which can be explained from previous analysis with the heatmap (Figure 20). A strong relationship between the energy consumption and the time during the days of a week was observed there.

To get additional information about the relationships between the independent and dependent variables, an investigation of the features importance with the help of the RF method will be needed. The discovery of the variables that affect the energy consumption the most and the ones that have the less impact on it will be necessary. This information could be used during the modelling phase. This could allow the creation of more accurate models with less parameters but with higher influence on the target variable (energy consumption).

3.4 Tools and Software Overview

Jupyter notebook and python were used to analyse the data and built the prediction models. For the data analysis and proper structure of the data numpy and pandas libraries were used. Pandas is a powerful library used for data analysis it is fast and easy to use on time series data and provides high performance. Numpy is a fundamental library for python used in data science. It provides high performance analysis and manipulation of multidimensional arrays in a simple way of use. Additional libraries such as matplotlib, pyplot, and seaborn were used for data visualization purposes and data analysis. Then depending on the forecasting method and problem, appropriate packages of scikit-learn (scipy, sklearn) library for the data preparation (splitting the dataset into training and testing), model selection and evaluation of the regression models were selected. Moreover, the installation of a popular library for AI and ML problems named tensorflow provided by an API named keras, allowed the implementation of deep learning models, such as LSTMs.

3.5 Modelling

By this step, the necessary data pre-processing such as filling the missing values and removing the outliers has finish. Important information has been extracted with the help of charts during the data analysis. The next step is the creation of two new datasets, one for the training process and one for the testing of the model. This can be achieved by splitting the available data points of the original dataset between these two datasets.

The training dataset should contain the biggest proportion of the data information, which typical lays between 70 and 85 percent of the total data. During training, the selected forecasting algorithm will use the provided data samples, to learn and identify patterns and trends of the energy consumption. This is a critical step as based on the information collected through the training process, the model will base the future predictions.

Once the training of the model is completed, the testing process begins. The testing dataset contains the leftover data points, meaning the data points that were not included during the model training. The idea behind this process, is to have a percentage of known past energy consumption values hidden from the model. That way, the model based on the information extracted during the training, will have to predict the energy consumption for the new unseen data (testing dataset). The testing process is necessary for the evaluation of the prediction result and overall performance of the model. To evaluate the prediction and overall predicting capabilities of the model, a comparison between the predicted and the actual values of the energy consumption is necessary. Model evaluation can be achieved with various methods and metrics. Chapter 4 will thoroughly present the modelling results and overall evaluation process. During the state-of-the-art analysis (Chapter 2), different techniques were investigated. Further analysis of such techniques will be presented for this load forecasting problem.

3.5.1 Random Forest Model

RF method is a widely used technique with great results in similar forecasting problems. The modelling process will start with this method due to its ability to identify the most important variables in a dataset. This could help to further improve current analysis. Additional information on features importance, could help built and evaluate different models, based on different sets of features while searching for the best performing model.

The working methodology begins with the split of the data, where 80% of the available data points will be used for training purposes while the remaining 20% for testing. To split the dataset into train and test, the train-test-split method provided by the sklearn package was used. First model uses all available features which include information about the temperature, relative humidity, year, month, week of the year, day of the month, day of the week and hours. The target or dependent variable is the energy consumption. RF method does not require any additional data pre-processing technique such as normalization or standardization, allowing to continue and feed the data as is to the model.

Identifying the best tuning parameters for the model, requires the use of a technique named Randomized Search, on which a k-fold cross validation generator was applied. Cross-validation is a data splitting procedure. In RandomizedSearchCV, cross-validation determines how many passes of the generated data samples are done and evaluated until the search finds the best parameters for the model.

Randomized search is a commonly used technique used to try different combination sets of hyperparameters. Algorithm's process is completed by randomly selecting different sets of hyperparameters and comparing them to identify the best combination. Another technique, that could be used for identifying the appropriate tunning, is *Grid Search*. This method though require us to define all testing combinations and train the model. For example, if we wanted to check many options/values for each parameter, the method would check every possible combination, which would require too much time and system power to complete the process. The *Randomized Search* randomly selects the combinations while speeding up the process according to the selected number of iteration (referring to the number of total combinations to examine). To explore faster the wide range of values for each hyperparameter *Randomized Search* was selected. In addition, after testing different numbers of folds for the algorithm, results remained unchanged, thus a stratified k-fold validator was selected to split the data into two folds. K-fold splits the samples into two sets which both contain a similar amount of data samples.

Examples of the variables tested with the help of the *RandomizedSearch* to identify the best hyperparameters are presented (Table 3).

Table 3. Hyperparameters for RF.

Number of Estimators	Max Features	Max Depth	Min Split Samples	Min Leaf Samples	Bootstrap Samples
100	Auto	10	1	1	True
200	Sqrt	20	2	2	False
300	Log2	30	5	4	
400		40	10	10	
500		50	15	15	
600		60	20	20	
⋮		⋮	⋮	⋮	
2000		200	45	60	
				100	

The maximum features parameter refers to the number of features considered per split when algorithm divides the data into different subsets for each tree. The available options to test in this parameter are the following:

1. Auto, where algorithm considers the total number of features when splitting the data.

2. Sqrt, where algorithm considers the square root of the number of features when splitting the data.

3. Log2, where algorithm considers the logarithmic base 2 value of the total features when splitting the data.

After identifying the appropriate structure for the RF model, the performance of different models trained with various input combinations were tested (Table 4).

Table 4. RF models and features sets.

Model Number	Features
1	Temperature, Relative Humidity, Year, Month, Day of the Month, Day of the Week, Week of the Year, Hours
2	Temperature, Relative Humidity, Year, Month, Day of the Month, Day of the Week, Hours
3	Temperature, Relative Humidity, Month, Day of the Month, Day of the Week, Hours
4	Temperature, Relative Humidity, Year, Month, Day of the Month, Hours
5	Temperature, Relative Humidity, Month, Day of the Week, Hours
6	Temperature, Relative Humidity, Day of the Week, Hours
7	Temperature, Relative Humidity, Week of the Year, Day of the Week, Hours
8	Temperature, Day of the Week, Hours
9	Temperature, Relative Humidity, Month, Day of the Month, Day of the Week, Week of the Year, Hours

We trained a total of nine models using the random forest method. Models based on different inputs (Table 4) were put into test and their performance was evaluated based on a set of metrics commonly used in regression problems. Model results are presented in the next chapter.

3.5.2 Support Vector Machine Model

SVR modelling process, follows a similar approach as before with RF. Starting with the split of the data into train and test sets with the use of the train-test-split method, and dividing the dataset into 80% for the training and 20% for testing. First model will include all the available features. Randomized Search method with a k-fold cross validator will be applied to identify the best hyperparameters and tune the model. In total, nine different models, based on different input combinations, will be tested and their performance will be evaluated (Table 5).

Table 5. SVR models and features.

Model Number	Features
1	Temperature, Relative Humidity, Year, Month, Day of the Month, Day of the Week, Week of the Year, Hours
2	Temperature, Relative Humidity, Year, Month, Day of the Month, Day of the Week, Hours
3	Temperature, Relative Humidity, Month, Day of the Month, Day of the Week, Hours
4	Temperature, Relative Humidity, Year, Month, Day of the Month, Hours
5	Temperature, Relative Humidity, Month, Day of the Week, Hours
6	Temperature, Relative Humidity, Day of the Week, Hours
7	Temperature, Relative Humidity, Week of the Year, Day of the Week, Hours
8	Temperature, Day of the Week, Hours
9	Temperature, Relative Humidity, Month, Day of the Month, Day of the Week, Week of the Year, Hours

Before applying the methodology, additional pre-processing requirements of SVR models need to be considered. During the data preparation step, we mentioned that SVR models are subject to additional pre-processing before training. A data scaling technique such as data normalization or standardization, is required on the available variables (dependent and independent). Such methods are used to rescale the inputs and outputs within a specified range, either between the minimum and maximum values as allowed by data normalization or between the mean and standard deviation through the data

standardization. Keeping the same range means keeping the same importance level between the variables, feeding them as equals to the model.

After testing both techniques, normalization achieved slightly better results in the problem compared to standardization, thus MinMaxScaler was chosen. This is a common data normalization algorithm often used to normalize data by scaling them down on a specified range. In this problem, the specified range lays between the values of -1 and 1.

After applying data normalization in the input and output variables, process continued with splitting the dataset into train and test. Then with the help of the RandomizedSearchCV method provided by the sklearn library, a wide range of values for each hyperparameter was tested to identify the best performing combination for the SVR model.

Table 6. Hyperparameters combinations for SVR.

Kernel	Gamma	C
RBF	1e-4	1
	1e-3	1e-3
	0.01	0.01
	0.1	0.1
	0.2	10
	0.4	100
	0.6	1000
	0.9	2000
	10	5000
	40	10000
	100	

Only the RBF kernel was considered since this is the only kernel that can be used for regression problems. Other kernel options are usually used in classification problems, since originally the support vector machine method was created for classification purposes.

The Gamma parameter decides how curved will the line of the decision boundary be. For example, on a classification problem, where a model needs to decide if the next value belongs to the class of cats or dogs, the decision comes by identifying the best line that separates these classes, this is the decision boundary. With more classes available on a problem, the decision would come after identifying the best n-D hyperplane that separates the classes. Typical gamma values used in problems lay between 0.001 and 100, where

the higher the gamma value is the more curve the decision boundary gets, but range can change based on the dataset.

Depending on the data, sometimes it is not possible to perfectly separate the classes with one line, leaving some residual points of a class mixed with the other class and vice versa, which generates an error. The C parameter is used to control this error, where a low C value represents a low error. Typical values used for the parameter C also range between 0.001 and 100, but again value depends on the actual data used in each case. In any case, a low C value and error does not mean that the better the model will get on prediction.

3.5.3 ANN Models

Neural networks are considered as more complex techniques compared to the previously used RF and SVR, as they tend to have a sophisticated structure that is mimicking the human brain neuron structure.

NNs also require additional data pre-processing before the training of the model, but it is quite more complicated process compared to previous SVR technique. Modelling process also requires the data scaling on the inputs and outputs before the training. This is due to the nature of the algorithm, which as explained earlier in chapter two, is trying to learn how to map the inputs to the outputs during the training process. This process is completed through the update of the weight between two nodes placed in two interconnected layers. The model weights are first initialized to minor random values, which as training continues, are updated by a selected algorithm aiming to reduce the error between the expected and predicted result. This increases the importance of scaling down all inputs and outputs as otherwise, the unscaled variables could reduce the speed of the training (for unscaled features) or result to exploding gradients (for unscaled target variable), a common NN problem further explained in chapter two. Moreover, scaling differences between feature variables can result in unstable and poorly performing learning process for the model, as some features could include large values while others very small [35].

Normalization or standardization techniques could help NNs training process by scaling down the variables in a specified range. In this case, normalization technique was chosen for two reasons; due to the superior performance of this approach in previous models and due to the range required to scale down the data. Standardscaler, would not be able to

keep the range of [-1, 1] since this approach is based on the mean of the dataset divided by the standard deviation which according to the data, would result in different range. Range [-1, 1] is preferred for this forecasting problem since most values within are already set within the range, such as the sine and cosine of day. Sine and cosine represent the minimum and maximum values of the dataset after scaling.

An additional but very important process used by researchers to improve the training process of the neural networks, is feature engineering. This is a process where additional information is extracted by the original dataset. As explained, in the beginning of the data analysis chapter, such information was extracted from the datetime column. During the implementation of previous random forest and support vector regressor methods, the additional extracted information was used to further improve the models. However, the use of the same extracted information on the NN models, would not help to achieve such an improvement without prior consideration of an additional requirement of the technique. This requirement is a pre-processing technique that would allow datetime features to preserve their cyclical behaviour.

Datetime features, such as hours of the day or days of the week, follow a cyclical pattern. For example, all possible values that could be expressed through the hours of a day within a 24-hour interval, lay between the range of 00:00 and 23:00. The difference between the last hour of a day (23:00) and the first hour of that same day (00:00) is one hour. Such information needs to be encoded to the model, otherwise in the model would understand that the difference between them is 23, based on the calculated range of the row numbers as seen in Table 7.

Table 7. Example of cyclical pattern.

Row Number	Datetime
0	01/01/2017 00:00
1	01/01/2017 01:00
2	01/01/2017 02:00
3	01/01/2017 03:00
4	01/01/2017 04:00
⋮	⋮
20	01/01/2017 20:00
23	01/01/2017 23:00

A typical method to encode such information on NN models, is by extracting/converting the corresponding sine and cosine values of such features. Figure 23 presents an example, of how the time of the day could be represented in a graph after converting it to the

corresponding sine and cosine values. The range between -1 and 1 of axis y, is explained by the prior normalization of the variable as a conversion to the cosine and sine values was applied, setting the range between -1 and 1. This corresponds to one 0 to 2π cycle.

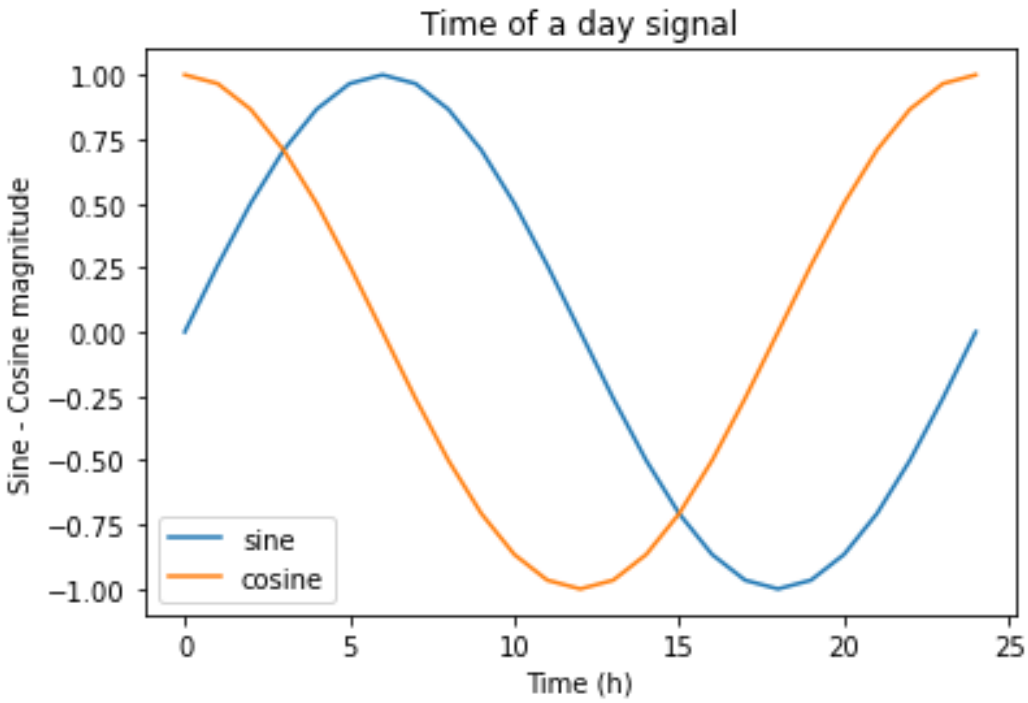


Figure 23. Time of the day representation after converting to sine and cosine.

The sine and cosine values of one day, represented by waves for the period of the 24 hours (Figure 23). In addition, to further improve the performance of the NN model, a review of the available information within the dataset and the features is required. The importance of the temperature and relative humidity is already confirmed by the analysis. Datetime variables should also be considered for this problem.

Considering the duplicated datetime information in the data, a decision to remove the unnecessary input of the duplicated variables has been made. New variables that maintain the require cyclical behaviour will be extracted and feed to the model, as explained in above process. The newly extracted information of sine and cosine values for the year and day were selected as new inputs since they contain hidden patterns and information of other datetime feature such as week of the year, day of the month and hour of the day.

Another important factor to investigate through the NN modelling process, is identifying the appropriate structure for the model. The simplest structure of a NN model, contains

one hidden layer between the input and output layers. The more the number of hidden layers increases, the more complex the model becomes since the amount of learning parameters increases. Usually, the term of the deep learning is used for the NN models that have more than one hidden layer, since those models need to complete a deeper journey following the specified hierarchy of selected algorithms in each layer until completing their training. The number of layers and nodes included on each layer, could vary and the only way to identify an appropriate value for the problem is through a trial-and-error process.

The modelling approach will be based in the following process: starting with the simplest structure of a NN model, layers and nodes will be added as long model improves. This can be achieved by optimizing (reducing) the training error through the training process. A similar approach will be followed for the selection of appropriate epochs and batch size numbers, as this could only achieved by a trial-and-error process. The batch size refers to the number of samples that will pass through the model at one time, while epoch, refers to a single propagation of all available data during the training process. Typically, a large batch size requires higher computational power but can provide faster training for the model. Thus, it is necessary to find the appropriate size that could achieve a fast and accurate model, considering that a very large batch size could impact the model's performance and lead to poor generalization. Generalization refers to the model's ability to perform well on new unseen data.

Selecting the appropriate activation function for the problem, requires testing. A comparison between the performance of commonly used functions for regression will be required. The activation function is a function used in each layer to define how the weighed total amount of an input is transformed into an output. This output is used by the inter-connected node of the next layer. Typically, all hidden layers use the same activation function, so considering the most common, a decision has been made to test *relu*, *sigmoid* and *tanh* in the layers.

The most popular function among these options, is *relu*, which has the simplest implementation. In addition, *relu* function is able to overcome common issues such as vanishing gradients that limit other activation functions [36]. Even though this function could perform well in the problem, a decision to test other options and compare their results has been made. Regarding the output layer, the selection of an activation function

is simple as there is only one appropriate option to output the numerical value of the predicted energy consumption, the linear function.

An additional function named optimizer is used to update the weights and other attributes such as learning rate, aiming to reduce the losses of the NN and achieve better predictions on the problem. There is a wide variation of optimizers to test. A comparison between the performance of the most popular options such as *rmsprop*, *adam* and *sgd* will be made to decide the best performing optimizer. *Adam* function is considered as the best optimizer due to its speed and effectiveness [37]. Two additional variations of *adam* named *adamax* and *nadam* will be tested for this problem.

When configuring a NN model, we usually refer to another hyperparameter called learning rate. This is usually considered as the most important parameter since it defines how much a model could change during the training and weigh update process. This value can be considered as the value that controls how fast a model can learn. Typically, the learning rate falls within the range of 0.0 and 1.0, with a default value of 0.001. Generally, a model that uses a small learning rate requires more training time, but is more able to find the optimal set of weights during the process [38]. On the other hand, a larger learning size speeds up the training process. For this problem, different sizes will be explored, and their performance will be compared.

During research, we came across a technique referred to as regularization which is used to avoid overfitting issues during modelling. Overfitting is a common issue where a model performs well in training but poorly on unseen data. Regularization can be achieved with the introduction of an additional layer called dropout. Authors of [15] used this technique and proved the improvement of their model. A dropout layer is used between two usual NN layers, such as Dense layers, to remove/drop a small percentage of randomly picked input data points during the training, helping the model generalize. For example, if two dense layers are used and a dropout layer with a rate of 0.2 is added between them, it means that there is a 20% probability of setting an input point of the first layer to zero, excluding it from consideration on the second layer when the information propagate. This technique typically reduces the performance of a model during the training but helps it perform better on testing.

For the modelling process, the tensorflow library provided from keras API will be used. Keras is a neural network library and tensorflow is a library used in machine learning. These libraries will allow to select and test important hyperparameters such as activation functions, possible optimizers and learning rates to further optimize and build NN models.

Unfortunately, the train-test-split method provided from sklearn used in other techniques to split the dataset to training and testing datasets, cannot be used in NN methods. The problem with this method, is that it does not allow to maintain the time series properties intact during training. This is because method picks data points in random time intervals when splitting the data into train and test. For time series regression problems, is critical to keep the correct order between the data as the value of the next point depends on the value of the previous data point. Using the built-in train-test-split method of sklearn, would cause the model to take random points and try to understand if there is any connection/pattern between them. Then model would try to predict the next values, which as we understand is not possible to work for the NNs.

To propagate the information to the model, a manual split of the dataset into train and test will be made. Before splitting the data, the implementation of the MinMaxScaler is required. Normalization is necessary, as with previous techniques, but instead of applying the scaler to all data (train and test), the scaler will be fit only on 80% which represents the training dataset. Then the scaler will be used to transform the whole dataset, including the testing data points. This is a typical process for ANNs to avoid overfitting issues [39]. Using the whole dataset, would allow the network to understand, after some time, the normalization process and learn the mean and standard deviation values of the dataset. This is due to the NNs abilities to search deeper and extract necessary information for their improvement. After learning such values, model would use them to increase its accuracy and overall performance but would not be able to perform good on unseen data since mean and standard deviation values would differ. Using the scaler only on the training dataset, allows the calculation of the mean and standard deviation or the minimum and maximum in this forecasting problem, only for those points. Then those are used to transform all the values in the required range of [-1, 1]. This helps avoid a situation where model memorizes information of the data, thus allowing it to perform better on new.

3.5.3.1 Multilayer Perceptron Model

MLP is a feedforward neural network used in classification and regression problems. Technique uses a supervised learning algorithm with the ability to learn non-linear functions. A typical structure of an MLP consists of one or more hidden layers between the usual input and output layers. Each layer consists of a group of neurons called nodes which are responsible to receive input information and compute an output. The nodes of a layer are inter-connected with all nodes of the next and previous layers. The input layer passes the feature/input information to the next hidden layer. Each interconnected node of the layer will be associated with a weight which value is based on the importance of its input information and its effect on the others. Nodes of hidden layers, compute the input information and transform them into outputs which then are passed as inputs to the next hidden layer. The output of each node is computed by the activation function of the layer, which calculates the weighted sum of the inputs connected to the node. Information propagates from one hidden layer to the next until the final output layer of the network is reached.

The training process of a MLP model starts with the feedforward propagation of the input information to the network. The calculated weights of the network are multiplied, and a bias is added on each layer to compute the final output of the model. Bias is described as a constant used to shift the activation function to better fit the data. After this process, a comparison of the predicted output of the model and the actual value of the dataset will be made. The error of the comparison will be calculated with the help of a loss function. Finally, based on the calculations of the loss function, information will be feed back to the network, a process referred to as backward propagation, to update the weights and reduce the errors. This process allows the model to learn better and improve its predictions.

Modelling approach starts with a simple model using a single hidden layer and the minimum number of nodes (equal to number of features + 1). Process continues by increasing the node number as long model improves. The same approach will be followed for the layers where the number of layers will keep increasing as long performance increases. Once the appropriate structure of the model is identified, different hyperparameters will be tested to locate the optimal tuning. Examples of parameters tested during the modelling process are presented (Table 8).

Table 8. Hyperparameters tunning on MLP – testing examples.

Number of Hidden Layers	Number of Nodes per Layer	Activation Function	Optimizer Function	Learning Rate	Epoch size	Batch Size
1	7	Relu	Rmsprop	0.0001	20	10
2	8	Sigmoid	Adam	0.001	50	12
3	9	Tanh	SGD	0.01	100	15
4	10		Adamax	0.1	500	20
	25		Nadam		1000	32
	30					35
	32					39
	40					40
	48					50
	50					70
	52					80
	100					100
	512					150

A short description of the advantages and disadvantages of different activation functions and present (Table 9). *Relu* transforms all positive values into a linear output and zero for all negative input values. *Sigmoid* and *tanh* are both nonlinear activation functions. *Sigmoid* transform the input values into a value between 0.0 and 1.0. All inputs larger than 1.0 will be transformed to an output of 1.0 while all negative inputs will be transformed to 0.0. *Tanh* uses a similar approach but can be considered as an improved *sigmoid* approach due to the range of the outputs the values which lay between -1.0 and 1.0.

Table 9. Comparison between activation functions.

Activation Function	Advantages	Disadvantages
Relu	Easy and fast to implement. No problem with vanishing gradients. Considered as the most popular function.	Dying r*elu* problem when learning rate is too large or when too many negative values exist.
Sigmoid	Good in probability predictions due to range (0.0, 1.0). Prevents jumps in the output values.	Vanishing gradients problem.
Tanh	Negative value presented with negative output.	Vanishing gradients problem.

A short comparison of the most common optimizers is presented (Table 10).

Table 10. Comparison between optimizers.

Activation Function	Advantages	Disadvantages
Rmsprop	Robust and good with small learning rates. Very popular and good at adapting.	Learning process could be slow.
Adam	Includes the advantages of rmsprop algorithm. Reduces the noise. Considered as the most popular method.	Tends to converge faster, in some cases.
SGD	Simple and fast implementation	High requirements in memory as it is updated frequently. Could be stuck at local minima or saddle point while trying to reach the best solution.

In addition to the parameters, the impact of the dropout layers will be investigated. The performance of different models will be tested, with 15, 20, 30 and 40 percent of their input data randomly removed during training.

3.5.3.2 LSTM Model

In previous chapters, LSTM technique was described as an improved RNN method. The technique's architecture makes it ideal for time series prediction problems. This explains

the popularity of the algorithm on similar forecasting problems. Researchers proved the superiority of the algorithm when dealing with time series data [16], thus explaining the decision to test it in this time series problem.

Feature scaling is a critical pre-processing step for the LSTM. MinMaxScaler will be used to scale down all variables between the range of [-1, 1] as before with previous forecasting methods. To avoid overfitting on the model, the same approach to previous MLP model, will be followed. The scaler was fitted only on 80% of the data (used for training purpose) enabling the calculation of the minimum and maximum values of the dataset to be based on training data. Then, using the information (min and max values) learned by the scaler, the transformation of the whole dataset within the range of [-1, 1] was achieved. Transformation included the remaining 20% previously reserved for testing.

After applying the MinMaxScaler, the data were split into train and test datasets. Typically, on NN models, the next modelling process would be training and identification of the appropriate structure of the model, but RNN methods, such as LSTM, have additional requirements regarding the shape of the data used as inputs.

These models, expect a three-dimensional input shape. The shape represents information of the following order: samples, timesteps, features. Samples define the length of the dataset, meaning the amount of data points included in the training dataset. Features define the number of features/input variables selected for training. Finally, timesteps is the representative value of the data sequence. In this forecasting problem, the data sequence is equal to 24 due to the interest towards the 24-hour load prediction. The use of such sequence will provide the model with the information such as how many data rows, should be considered for the prediction of the next value when requested.

The methodology behind it, is splitting the data into separate tables, each containing a time series sequence of 24 data rows, which represent the energy consumption of a day in 24-hour intervals. Starting with the first 24 rows (meaning the energy consumption within a day), process creates a new, separate table, where the values are added. This table is then added into another big table. Then process takes the next 24 rows, create another new table where it copies the values of the next day's consumption, and adds it into the big table. Process continues until all data are added into the big table, separated in a 24-

row manner. After reshaping the inputs into three-dimensions, data will be on the format of [sample, timesteps, features], as required by the LSTM method.

The next step is identifying the architecture of the model. A similar process to the MLP technique will be used, starting with the minimum number of layers and the minimum nodes (equal to the number of features + 1) then keep on increasing both as long model improves. When the error of the loss function stops decreasing, model will reach the optimal structure. Also, the performance of different epoch and batch sizes will be tested for the model.

A few examples of the hyperparameters tested to identify the best LSTM model structure and tunning is presented (Table 11).

Table 11. Hyperparameters tunning on LSTM – testing examples.

Number of Hidden Layers	Number of Nodes per Layers	Activation Function	Optimizer	Learning Rate	Epoch Size	Batch Size
1	7	Tanh	Rmsprop	0.0001	40	10
2	12	Sigmoid	Adam	0.001	100	15
3	20	Relu	Adamax	0.01	300	20
4	24		Nadam	0.1	1000	25
	30		SGD			32
	40					45
	60					40
	100					50
						60
						80
						100
						150

Moreover, to further improve the model, a decision to test the regularization technique was made. A comparison between the performance of models without regularization, and those with 15, 20, 30 and 40 percent dropout rate will be evaluated.

3.6 Evaluation Metrics and Loss Functions

Model evaluation is a crucial process of the modelling. Evaluation metrics allow the optimization of learning parameters of the models and define the most successful among peers. Process results in the identification of the best performing model that could be used for the forecasting of the energy consumption of an office building.

To achieve a fair and valuable evaluation of the models, a common set of metrics needs to be defined. These metrics will be used to compare the performance of different models. MAE, MSE, MAD, explained variance score, and the coefficient of determination (R^2) are common metrics for regression problems, thus a decision to use them has been made. On top of these metrics, the accuracy of each model will be compared. Accuracy will be calculated based on MAPE to compare the predicted values with the actual values of the dataset.

Mean absolute error (*MAE*) is the mean or average difference between the predicted value and the actual value. This metric calculates the mean value of the residuals in the dataset.

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |y_i - x_i| , \tag{3.1}$$

where y_i is the predicted value, x_i is the actual value and N defines the total amount of data points.

Mean squared error (MSE) defines the average of the squared difference between the predicted value and the actual value in the dataset. This metric calculates the variance of the residuals in a forecasting problem.

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 , \tag{3.2}$$

where y_i is the observed value, $\hat{y}$ is the predicted value and N defines the number of data points.

Mean absolute deviation (MAD) calculates the average distance between the predicted value and the actual value. It calculates the median over the absolute deviations from the median.

$$MAD = \frac{1}{N} \sum_{i=1}^{N} |x_i - m(X)| , \tag{3.3}$$

where m(X) is the average value of the dataset, x_i are the data values in the set and N defines the number of data points.

The explained variance measures the variation of the prediction value compared to the actual dataset. The higher the variance the better the prediction of the model.

$$explain\ variance\ (y, \hat{y}) = 1 - \frac{Var\{y-\hat{y}\}}{Var\{y\}}\ , \tag{3.4}$$

where $\hat{y}$ is the estimated output target value, y is the correct target output value, and Var is the variance, the square of standard deviation.

The coefficient of determination (or R^2) can be considered as the percentage of correct predictions returned by a model. This metric calculates the variation amount that can be explained by the model, meaning how many data points managed to fall within the line set by the regression equation. This means that they were predicted correctly. The higher the coefficient gets, the better prediction of the model.

$$R^2 = 1 - \frac{\Sigma(y_i-\hat{y})^2}{\Sigma(y_i-\bar{y})^2}\ , \tag{3.5}$$

where y_i is the actual value, $\hat{y}$ is the predicted value and $\bar{y}$ is the mean of all actual values in the dataset.

The accuracy of the model will be calculated based on a function that converts the average value of the error (MAPE) to a percentage amount. For example, if a MAPE of 0.02 is reached by the model, then model would be 98% accurate.

MAPE is defined as the mean or average of the absolute percentage errors of the predictions. Error is defined as the residual number after the subtraction of the actual value from the predicted value. This can express the accuracy of the model as a percentage.

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{x_i-y_i}{x_i}\right| \tag{3.6}$$

where x_i is the actual value, y_i is the predicted value, and N defines the number of data points.

3.7 Issues during workflow implementation

Before continuing to the next chapter and the presentation of the modelling and prediction results, a brief description of a few issues that occurred during the implementation of this project will be discussed. Starting with the required pre-processing techniques for the handling of the big gap of December month during 2017 in the dataset. In the beginning of modelling, a decision to handle such issue was made. Solution included the implementation of a simple replacement method such as mean, median, interpolation etc. During many efforts to identify the best performing technique among them, many models with different replacement methods were tested, but all seemed to perform poorly.

The use of such method created a big set of data points with the same values, which was considered as noise to the prediction models. This was reducing their overall ability to accurately predict new values of the energy consumption. After identifying the issue, a decision to implement a different technique and replace the missing values of the energy consumption of that month with the available values of the next year's consumption during the same period, was made. This approach helped models reach higher performance.

Another issue caused NN models perform extremely poorly during the initial efforts. Initial models reached extremely high errors in terms of MSE, MAE, MAD etc. Poor performance resulted after selecting a wrong splitting approach for the dataset. Many models were built and optimized after splitting the dataset with the train-test-split method, provided by the sklearn library. Unfortunately, a lot of time was lost while testing different techniques and approaches that could optimize the performance of NN models, before figuring out that the wrong splitting method was selected. Instead of the pre-built train-test-split sklearn method, the manual split of the data into train and test datasets was required. The problem of the pre-built function was that it was selecting random points during the splitting of the data, which caused changes in the time series sequence.

Data separation was not the only problem during NN modelling. Unfortunately, another important aspect of the method was missed during pre-processing which was keeping the cyclical format of the datetime data. After poor modelling results and additional research, the importance of the cyclical nature of such data was identified. An approach that could

provide such information to the model was used and concluded to improved results. Models presented increase in accuracy while minimizing the prediction errors.

4 Results and Analysis

4.1 Features Importance

Random Forest can be used to identify important features within a dataset. When applying such technique valuable information regarding the available variables can be collected. In this forecasting problem, obtained information identified the variables with the higher impact on the target variable (energy consumption) (Figure 24).

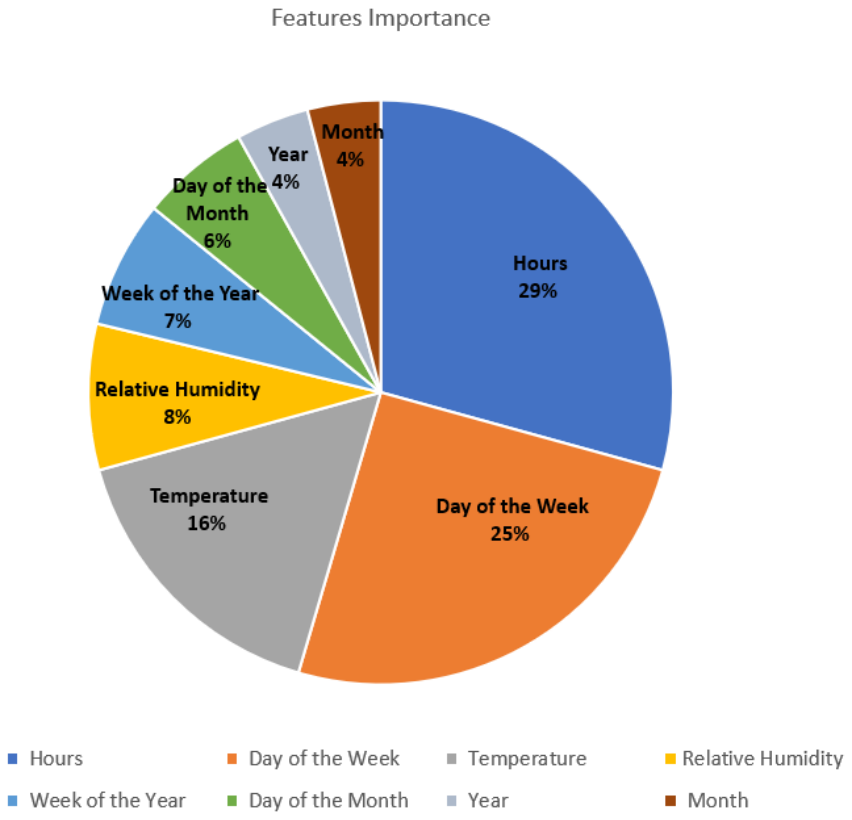


Figure 24. Features Importance.

The most important features were the hours, day of the week and the temperature, followed by the relative humidity and the week of the year (Figure 24). The least important features for the energy consumption variable were the month and year.

4.2 Modelling Results

4.2.1 Random Forest Model Results

After applying the randomized search with the cross-validation technique, the best tuning parameters and architecture of the RF model was identified. The best RF structure is described below:

Number of estimators or Number of trees: 500

Maximum depth of the trees: 70

Minimum number of samples required at leaf node: 1

Minimum number of samples required to split an internal node: 2

Number of features/inputs: log2

Bootstrap data samples used while building the trees: False

If bootstrap parameter is set to false, then the full dataset is used to build each tree.

After building several models, based on different sets of features (Table 4), their performance on new data was evaluated. This was achieved with the use of the testing dataset. Testing results were compared with the results reached during model training.

Table 12. RF model results.

RF Model Number	Training Results					Testing Results					Accuracy
	MAE	MSE	MAD	ExV	R^2	MAE	MSE	MAD	ExV	R^2	
1	1.56	5.42	1.02	0.99	0.99	4.19	39.09	2.72	0.93	0.93	94.32%
2	1.77	6.67	1.18	0.99	0.99	4.75	48.47	3.11	0.91	0.91	93.45%
3	2.32	10.53	1.64	0.98	0.98	6.28	77.52	4.55	0.85	0.85	91.22%
4	3.21	21.16	2.07	0.96	0.96	8.72	158.77	5.45	0.70	0.70	87.62%
5	2.91	16.63	2.04	0.97	0.97	7.84	118.75	5.60	0.78	0.78	88.85%
6	3.61	26.10	2.51	0.95	0.95	9.13	161.31	6.51	0.70	0.70	86.89%
7	2.60	13.13	1.85	0.97	0.97	7.09	96.86	5.22	0.82	0.82	89.96%
8	5.02	49.48	3.59	0.90	0.9	10.35	197.00	7.66	0.63	0.63	85.19%
9	2.19	9.37	1.55	0.98	0.98	5.91	68.49	4.23	0.87	0.87	91.78%

The best RF model managed to reach 94.32 % accuracy. During the evaluation process, the model managed to reach a MAE of 4.19, which means there is a small enough difference between the predicted values and the actual value of the energy consumption

of the test dataset. Considering this represents the average error of the prediction. For example, in some cases where the actual value of the energy consumption was 60 kWh, the predicted value was between 56 and 64 kWh, which can be considered as good prediction. Moreover, the model reached a MAD of 2.72, a value lower than the value of MAE, as it calculates the median of all errors of the predictions.

The high R^2 score of the method, is a percentage representation of the calculated difference between the MSE and the residuals, which can be defined as the value of the difference between the predicted and real value of the energy consumption. This is used to identify how close the prediction is to the actual value.

Based on the results (Table 12), most models had good prediction capabilities. The best performing model (Table 12: number 1) used temperature, relative humidity, year, month, day of the month, day of the week and the hour of the day as inputs. Despite the results of the feature extraction (Figure 24), the model that used the most important features (Table 12: model number 8), was the lowest performing. It reached 85.19% accuracy, and an R^2 score of only 0.63 during testing. In addition, model reached the highest errors among its peers.

4.2.2 SVR Model Results

After testing a wide range of parameter combinations, results of RandomizedSearchCV technique concluded the following hyperparameters as best for a SVR model:

Kernel: RBF

Gamma: 10

C: 1000

Nine SVR models based on different dataset combinations were created (Table 5). Models were evaluated based on a common set of metrics. Results achieved during the training and testing process were documented (Table 13).

Table 13. SVR model results.

SVR Model Number	Training Results					Testing Results					Accuracy
	MAE	MSE	MAD	ExV	R^2	MAE	MSE	MAD	ExV	R^2	
1	0.05	0.00	0.05	0.88	0.88	0.06	0.01	0.05	0.84	0.84	89.56%
2	0.05	0.00	0.05	0.88	0.88	0.06	0.01	0.05	0.83	0.83	89.25%
3	0.06	0.00	0.06	0.85	0.85	0.07	0.01	0.06	0.75	0.75	87.24%
4	0.07	0.01	0.07	0.81	0.81	0.09	0.01	0.07	0.54	0.54	84.04%
5	0.06	0.01	0.06	0.82	0.82	0.07	0.01	0.06	0.71	0.71	86.72%
6	0.07	0.01	0.05	0.75	0.75	0.07	0.01	0.06	0.70	0.70	86.75%
7	0.06	0.01	0.06	0.82	0.82	0.07	0.01	0.06	0.73	0.73	86.86%
8	0.07	0.01	0.05	0.73	0.73	0.07	0.01	0.05	0.72	0.72	87.19%
9	0.06	0.00	0.06	0.86	0.86	0.07	0.01	0.06	0.78	0.78	87.92%

Based on the results (Table 13), model number 1 managed to reach the highest performance while minimizing the errors of prediction. Model achieved an accuracy of 89.56% while keeping the MSE, MAE and MAD low. Model inputs included temperature, relative humidity, year, month, day of the month, day of the week, week of year and the hour of the day.

In addition, a comparison between the normalization and standardization techniques was carried out for the best performing model and results are presented (Table 14).

Table 14. Comparison of normalization and standardization techniques on SVR model.

Scaling Technique	Training Results					Testing Results					Accuracy
	MAE	MSE	MAD	ExV	R^2	MAE	MSE	MAD	ExV	R^2	
Normalization	0.05	0.00	0.05	0.88	0.88	0.06	0.01	0.05	0.84	0.84	89.56%
Standardization	0.09	0.01	0.10	0.99	0.99	0.42	0.34	0.30	0.66	0.66	86.64%

Results concluded the superiority of the normalization technique (Table 14). The model that used this method reached higher accuracy and managed to further decrease the errors. Standardization resulted in higher prediction errors probably due to the nature of the dataset where considering the values of the variables, the standard deviation was quite high. Based on the results (Table 14), the decision to apply data normalization to NN models was made.

4.2.3 MLP Model Results

After testing different number of nodes, hidden layers, batch sizes and epochs, the best performing structure for the model was identified. Based on the performance and overall

ability to minimize the prediction errors during training, the best model structure is the following:

Nodes number: 7

Hidden layers: 1

Based on a comparison between different optimizers (Table 15), *adam* optimizer was identified as the best performing option for the MLP model.

Table 15. Optimizer's comparison in MLP.

Optimizer	Training Results					Testing Results					Accuracy
	MAE	MSE	MAD	ExV	R^2	MAE	MSE	MAD	ExV	R^2	
Adam	0.11	0.02	0.09	0.40	0.40	0.10	0.02	0.07	0.45	0.45	73.17%
Rmsprop	0.11	0.02	0.09	0.40	0.39	0.10	0.02	0.08	0.38	0.38	71.21%
SGD	0.12	0.02	0.09	0.33	0.33	0.10	0.02	0.08	0.38	0.36	69.86%
Adamax	0.11	0.02	0.09	0.38	0.38	0.10	0.02	0.07	0.41	0.41	72.72%
Nadam	0.11	0.02	0.08	0.42	0.42	0.10	0.02	0.07	0.42	0.42	72.03%

Different activation functions were tested and according to the results (Table 16), *relu* managed to achieve slightly better performance on the MLP model.

Table 16. Activation function's comparison in MLP.

Activation Function	Training Results					Testing Results					Accuracy
	MAE	MSE	MAD	ExV	R^2	MAE	MSE	MAD	ExV	R^2	
Relu	0.11	0.02	0.09	0.40	0.40	0.10	0.02	0.07	0.45	0.45	73.17%
Sigmoid	0.11	0.02	0.09	0.39	0.39	0.10	0.02	0.07	0.43	0.43	73.07%
Tanh	0.11	0.02	0.08	0.41	0.41	0.10	0.02	0.07	0.42	0.42	72.55%

Learning rates were tested. Based on the results (Table 17), 0.001 achieved higher performance.

Table 17. Learning rate's comparison in MLP.

Learning Rate	Training Results					Testing Results					Accuracy
	MAE	MSE	MAD	ExV	R^2	MAE	MSE	MAD	ExV	R^2	
0.1	0.12	0.03	0.09	0.31	0.17	0.13	0.03	0.10	0.35	0.05	59.41%
0.01	0.11	0.02	0.09	0.41	0.40	0.10	0.02	0.08	0.43	0.42	71.89%
0.001	0.11	0.02	0.09	0.36	0.36	0.10	0.02	0.06	0.40	0.40	72.13%
0.0001	0.11	0.02	0.09	0.37	0.37	0.10	0.02	0.08	0.41	0.40	71.05%

In addition, higher learning rate resulted in the worst model performance with an accuracy below 60% and underfitting issues as identified by the poor results of R^2 during training and testing process (Table 17).

Different batch sizes were tested in the model. Results concluded that a batch size equal to 100 reached the highest performance. A few indicative examples of batch size testing are presented (Table 18).

Table 18. Batch size comparison in MLP.

Batch Size	Training Results					Testing Results					Accuracy
	MAE	MSE	MAD	ExV	R^2	MAE	MSE	MAD	ExV	R^2	
20	0.11	0.02	0.09	0.38	0.38	0.10	0.02	0.07	0.40	0.38	71.32%
39	0.11	0.02	0.09	0.36	0.36	0.10	0.02	0.06	0.40	0.40	72.13%
50	0.11	0.02	0.09	0.39	0.38	0.10	0.02	0.07	0.40	0.40	72.17%
100	0.11	0.02	0.09	0.41	0.41	0.09	0.02	0.06	0.45	0.44	73.30%

Regularization technique was tested on the model. A comparison between different dropout rates concluded that use of 15% dropout rate, increased the model's performance (Table 19).

Table 19. Comparison of regularization method in MLP model.

Dropout layer	Training Results					Testing Results					Accuracy
	MAE	MSE	MAD	ExV	R^2	MAE	MSE	MAD	ExV	R^2	
None	0.11	0.02	0.09	0.41	0.41	0.11	0.02	0.08	0.38	0.38	70.36%
15%	0.11	0.02	0.09	0.40	0.40	0.10	0.02	0.07	0.45	0.45	73.17%
20%	0.11	0.02	0.09	0.39	0.39	0.10	0.02	0.08	0.44	0.44	73.07%
30%	0.11	0.02	0.09	0.37	0.37	0.10	0.02	0.07	0.42	0.42	73.05%
40%	0.12	0.02	0.10	0.36	0.36	0.10	0.02	0.08	0.42	0.42	71.90%

According to the presented modelling results, the best performing model used a simple structure with a single hidden layer, with *relu* as activation function, between the input and output layers. The model was trained with *adam* as optimizer and 0.001 learning rate, for epoch and batch sizes equal to 100. Finally, regularization technique managed to further improve the MLP model's performance. Despite the efforts, none of the models managed to outperform the previous forecasting methods of RF and SVR. The accuracy of the models laid between 69% and 74% while none of the models managed to reach a R^2 score higher than 0.45 during the testing process.

4.2.4 LSTM Model Results

Different number of nodes, hidden layers, batch sizes and epochs were tested during modelling. Finally, the best performing structure for the model concluded as follows:

Nodes number: 7

Hidden layers: 3

Activation functions were tested and according to the results (Table 20), *relu* achieved the highest performance on the LSTM model.

Table 20. Activation function's comparison in LSTM.

Activation Function	Training Results					Testing Results					Accuracy
	MAE	MSE	MAD	ExV	R^2	MAE	MSE	MAD	ExV	R^2	
Relu	0.11	0.02	0.10	0.39	0.38	0.10	0.02	0.07	0.42	0.42	58.59%
Sigmoid	0.11	0.02	0.09	0.36	0.36	0.10	0.02	0.06	0.40	0.39	57.13%
Tanh	0.11	0.02	0.08	0.42	0.42	0.10	0.02	0.06	0.43	0.43	56.02%

Different optimizers were tested (Table 21) and *adamax* was identified as best performing.

Table 21. Optimizer's comparison in LSTM.

Optimizer	Training Results					Testing Results					Accuracy
	MAE	MSE	MAD	ExV	R^2	MAE	MSE	MAD	ExV	R^2	
Adam	0.11	0.02	0.09	0.39	0.38	0.10	0.02	0.07	0.44	0.43	59.13%
Rmsprop	0.11	0.02	0.09	0.41	0.40	0.10	0.02	0.07	0.42	0.41	55.34%
SGD	0.15	0.03	0.14	0.02	0.02	0.15	0.03	0.13	0.00	0.00	62.70%
Adamax	0.12	0.02	0.10	0.34	0.31	0.11	0.02	0.08	0.38	0.36	61.45%
Nadam	0.11	0.02	0.08	0.40	0.40	0.10	0.02	0.06	0.43	0.43	56.11%

Learning rates were tested. Based on the results (Table 22), the learning rate of 0.0001 achieved higher accuracy.

Table 22. Learning rate's comparison in LSTM.

Learning Rate	Training Results					Testing Results					Accuracy
	MAE	MSE	MAD	ExV	R^2	MAE	MSE	MAD	ExV	R^2	
0.1	0.15	0.03	0.14	0.00	0.00	0.15	0.03	0.13	0.00	0.00	62.86%
0.01	0.11	0.02	0.09	0.39	0.39	0.10	0.02	0.08	0.41	0.41	58.70%
0.001	0.12	0.02	0.10	0.32	0.32	0.11	0.02	0.09	0.36	0.36	59.19%
0.0001	0.14	0.03	0.12	0.16	0.12	0.13	0.03	0.11	0.18	0.14	64.89%

Different regularization techniques were tested in the LSTM models. A review between no dropout layers and layers with 15, 20, 30 and 40 percent dropout rates are presented in Table 23.

Table 23. Comparison of regularization method in LSTM model.

Dropout layer	Training Results					Testing Results					Accuracy
	MAE	MSE	MAD	ExV	R^2	MAE	MSE	MAD	ExV	R^2	
None	0.11	0.02	0.08	0.35	0.35	0.10	0.02	0.07	0.38	0.37	55.30%
15%	0.12	0.02	0.09	0.34	0.33	0.10	0.02	0.07	0.38	0.38	58.31%
20%	0.12	0.02	0.10	0.32	0.29	0.11	0.02	0.08	0.35	0.34	60.78%
30%	0.13	0.03	0.11	0.24	0.20	0.12	0.02	0.10	0.27	0.25	63.03%
40%	0.14	0.03	0.12	0.16	0.12	0.13	0.03	0.11	0.18	0.14	64.89%

According to the results (Table 23), the regularization technique with a dropout rate of 20% achieved higher accuracy. The model with 40% of dropout rate presented the lowest performance based on accuracy.

According to the modelling results, the best performing model used 3 hidden layers with 7 nodes on each. The best performing activation function was *relu.* Model used an *adamax* optimizer and a 0.0001 learning rate, for 40 epochs and a batch size of 100. The regularization technique improved the model's performance.

4.3 Forecasting Models Comparison

For this forecasting problem, four state-of-the-art techniques were selected for further investigation. Techniques were tested and their performance was evaluated. A quick comparison based on the accuracy of the forecasting techniques is presented (Table 24).

Table 24. Methods comparison based on accuracy.

Forecasting Method	Accuracy
Random Forest	94.32 %
SVR	89.56 %
MLP	73.17 %
LSTM	64.89 %

According to the results (Table 24), RF model reached the highest accuracy with a score of 94.32%. The SVR model followed reaching a score of 89.56% in accuracy. The lowest performing methods were based on the ANNs architecture. Even though NNs were among the most popular techniques used in forecasting, their capabilities in modelling were limited due to low amount of data available. Comparing the two NN techniques, results concluded that the simpler method of the multilayer perceptron, achieved better results than LSTM, reaching 73.17% in accuracy. LSTM model achieved only 63.16%.

A graphical representation of the predictions achieved with the best random forest model is presented (Figure 24).

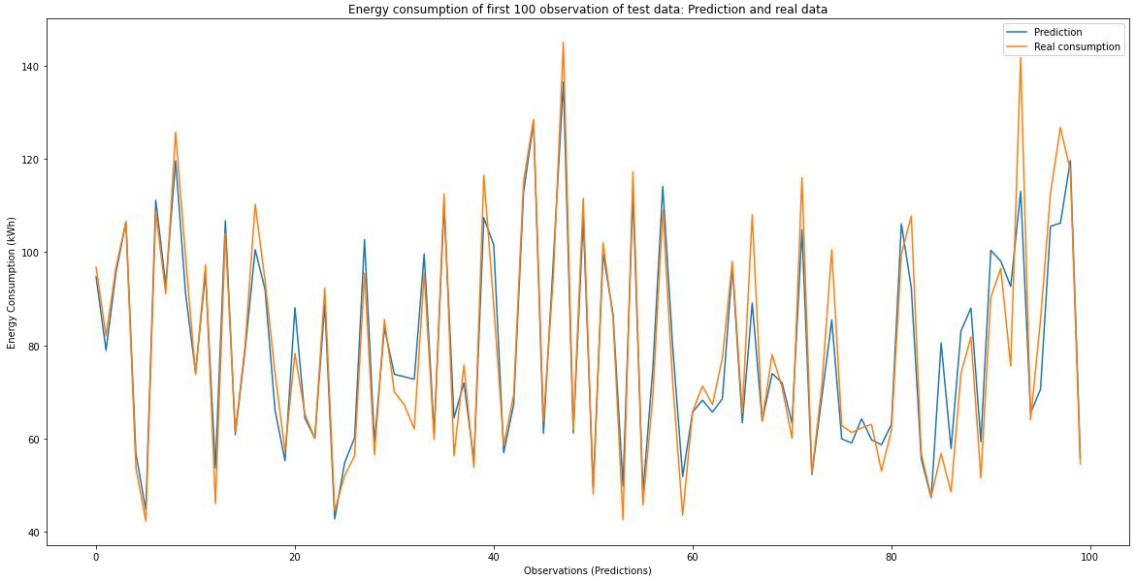


Figure 25. Prediction results of the best RF model.

Based on the results (Figure 25), the predicted values were overall very close to the actual values provided from the dataset. There were some noticeable points where the predicted energy consumption was a bit far from the actual value of the energy consumption. This can also be explained by the error results of the model. Despite that, most predictions lay within an acceptable range, proving that this model can be used for the prediction of the energy consumption of the specified office building. Few points identified with quite a high difference between the prediction and actual value. Those points fall over a 20-kWh difference, but overall predictions are considered accurate as most predictions fall close to the actual values of the energy consumption. The only exception comes from a single peak point where the highest difference between the actual and predicted value was close to 40 kWh. This concludes that there could be limited cases where prediction of energy consumption for the office will be quite off from actual.

In addition, a graphical representation of the predictions achieved with the best support vector regression model is presented (Figure 26), to allow a comparison between the best two forecasting techniques.

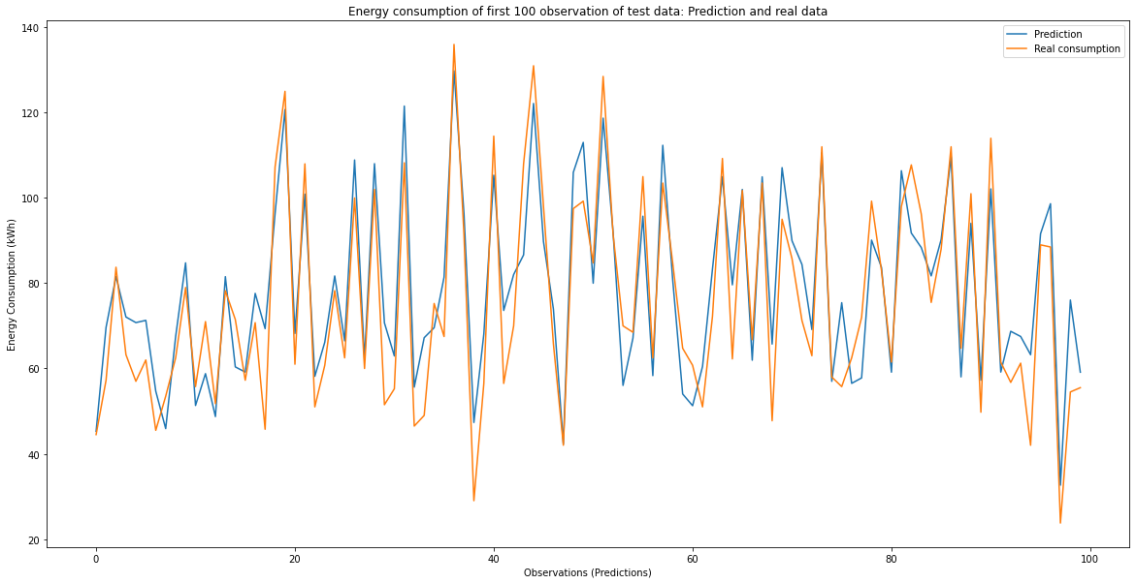


Figure 26. Prediction results of the best SVR model.

The predictions (Figure 26) achieved with the second-best forecasting technique, were unable to reach higher accuracy than the RF technique. Graph clearly shows more predicted values falling far from the actual values of the energy consumption provided by the dataset. This proves that support vector regression could not reach the performance of the RF technique in this forecasting problem.

In addition, a detailed comparison of the performance of the forecasting methods used in the problem is presented (Table 25).

Table 25. Comparison of forecasting methods.

Method	Training Results					Testing Results					Accuracy
	MAE	MSE	MAD	ExV	R^2	MAE	MSE	MAD	ExV	R^2	
RF	1.56	5.42	1.02	0.99	0.99	4.19	39.09	2.72	0.93	0.93	94.32 %
SVR	0.05	0.00	0.05	0.88	0.88	0.06	0.01	0.05	0.84	0.84	89.56 %
MLP	0.11	0.02	0.09	0.40	0.40	0.10	0.02	0.07	0.45	0.45	73.17 %
LSTM	0.14	0.03	0.12	0.16	0.12	0.13	0.03	0.11	0.18	0.14	64.89 %

According to the achieved results (Table 25), the best performing forecasting technique for the problem is the random forest, followed by the support vector regression. Unfortunately, the more complex ANNs methods, despite their popularity and overall great abilities on prediction problems, as identified through literature, lacked on performance. The best performing MLP model reached an R^2 of 0.45 while the best performing LSTM model was below 0.20. The poor performance can be explained by the limitation of the dataset since the available data were for a generally short period.

NNs have high requirements in terms of data size to perform good, thus additional years of data during training, could achieve higher performance.

4.4 Results Summary

According to the results (Chapter 4), random forest proved to be superior to other techniques. Overall, most RF models, achieved higher performance than the best models of SVR, ANN and LSTM. The best RF model managed to reach an accuracy of 94.32%, with 4.19 MAE, 39.09 MSE, 2.72 MAD and an explained variance and R^2 scores of 0.93. SVR was the second-best forecasting technique for this problem, reaching an accuracy of 89.56%, with 0.06 MAE, 0.01 MSE, 0.05 MAD and an explained variance and R^2 scores of 0.84. ANN methods could not provide a successful prediction model with the best MLP model reaching an accuracy of 73.17% with 0.10 MAE, 0.02 MSE, 0.07 MAD and 0.45 on R^2 and explained variance, while LSTM achieved only 64.89 % in accuracy, 0.13 MAE, 0.03 MSE, 0.11 MAD and 0.14 on R^2 and 0.18 explained variance scores. The main reason of poorer performance of NN compared to other techniques was the small size of data used during the training of the models. Typically, NN models require big datasets to extract important patterns and achieve accurate predictions. More data lead to more information propagated to the model which could result to higher accuracy in model predictions.

5 Summary

5.1 Conclusion

Load forecasting is essential for the improvement of the energy efficiency of buildings. Many researchers have tried to analyse the important factors affecting the energy consumption to identify a benchmark that could be used to evaluate building's performance. Important information could be extracted from the evaluation of a building, enabling different stakeholders taking decisions to further improve their energy efficiency and overall environmental and financial impact, following the latest regulations as established by the EU commission [2]. Based on the state-of-the-art forecasting methods for the energy consumption, we tried to identify and build a prediction model that could provide accurate predictions for a building office in Portugal. After analysis of the historical three-year old data of hourly energy consumption and the most common weather conditions, the most important factors were identified. Results concluded that the hour of the day, the day of the week and the temperature were the most important variables for this forecasting problem. This assumption was based on the analysis of the features correlation and the feature extraction method, as provided from the RF technique applied in the problem. Despite these results (Figure 24), the best performing model built, was based on the following inputs: temperature, relative humidity, year, month, day of the month, day of the week and the hour of the day. Several state-of-the-art techniques were used in modelling. Their results were analysed and compared. Random Forest, achieved the best performance in terms of accuracy, explained variance score and R^2. The proposed model managed to reach an accuracy of 94.32% and 0.93 coefficient of determination and explained variance score.

Additional techniques such as SVR, MLP and LSTM models, were analysed and compared to the RF model, but none reached higher performance in terms of accuracy or R^2 score. The second-best model was based on the SVR technique. The models reached an accuracy of 89.56% and 0.84 in the coefficient of determination and explained variance score. Even though SVR method did not outperform the RF technique in terms of accuracy, explained variance or R^2 scores, it managed to further reduce the prediction errors. Model reached a MAE of 0.06, MSE of 0.01 and MAD of 0.05 during model evaluation. Best RF model reached a MAE of 4.19, MSE of 39.09 and MAD of 2.72

during testing. The most complex ANN prediction techniques failed to achieve higher performance. The best MLP reached only 73.17% in accuracy, and 0.45 in R^2 and explained variance score. The best LSTM model achieved only 64.89% in accuracy with R^2 of 0.14 and explained variance of 0.18. Considering the poor results of NN methods, we identified that additional data could lead to an improvement of the performance. This is concluded based on the overall high data requirements of such methods.

Overall, data analysis and modelling process provided acceptable results. The prediction models could be used for the accurate prediction of the hourly energy consumption of the office, considering the work on the topic successful.

5.2 Considerations for future work

According to the results of the data analysis and the results achieved through various forecasting methods, such as RF, SVR and NNs, future work could be based on re-training the models with higher amount of data. This could help especially the more complex techniques such as neural networks. NNs have high data requirements, so additional years of data could result into better trained models and overall, more accurate performance of these models. In addition, other statistical methods could be considered for investigation in the problem. Based on this analysis, algorithms with less complexity seem to perform better. During research, Arima models were identified as good performing techniques for time series problems and have been widely used in the past as well as in more current projects [40]. In addition, such methods have provided good results when combined with other techniques such as RF [6], SVR [41] and NNs [42], thus further consideration of such method could lead to good results in this problem. Future work could start with a comparison of the best performing RF and SVR models with other Arima models in the problem, then investigate the potentials of combining these methods with Arima technique and compare their performance.

References

[1] Henrique Pombeiro, Rodolfo Santos, Paulo Carreira, Carlos Silva, and João M.C. Sousa, 'Comparative assessment of low-complexity models to predict electricity consumption in an institutional building: Linear regression vs. fuzzy modeling vs. neural networks', *Journals & Books*, vol. 146, pp. 141–151, Jul. 2017, doi: https://doi.org/10.1016/j.enbuild.2017.04.032.

[2] 'Energy performance of buildings directive'. [Online]. Available: https://ec.europa.eu/energy/topics/energy-efficiency/energy-efficient-buildings/energy-performance-buildings-directive_en

[3] Jonathan Roth, Benjamin Lim, Rishee K. Jain, and Dian Grueneich, 'Examining the feasibility of using open data to benchmark building energy usage in cities: A data science and policy perspective', *April 2020*, doi: https://doi.org/10.1016/j.enpol.2020.111327.

[4] Hye Gi Kim and Sun Sook Kim, 'Development of Energy Benchmarks for Office Buildings Using the National Energy Consumption Database', *20 February 2020*, p. 18, doi: https://doi.org/10.3390/en13040950.

[5] Nelson Isaiah Mukwaya and Peter Okidi-Lating, 'Benchmarking Energy Efficiency of Commercial Office Buildings in Kampala', May 2014. doi: http://dx.doi.org/10.15242/IIE.E0514065.

[6] Beiyan Jiang, Zhijin Cheng, Qianting Hao, and Nan Ma, 'A Building Energy Consumption Prediction Method Based on Random Forest and ARMA', presented at the 2018 Chinese Automation Congress (CAC), Xi'an, China, Jan. 2019. doi: 10.1109/CAC.2018.8623540.

[7] Mahnameh Taheri and Parag Rastogi, 'Benchmarking Building Energy Consumption Using Efficiency Factors', Sep. 2019. doi: 10.26868/25222708.2019.210575.

[8] Eugene A. Feinberg and Dora Genethliou, 'Load Forecasting Chapter', in *Load Forecasting Chapter - Part of the Power Electronics and Power Systems book series (PEPS)*, Springer, Boston, MA, 2005. [Online]. Available: https://doi.org/10.1007/0-387-23471-3_12

[9] Qingyao Qiao, Akilu Yunusa-Kaltungo, and Rodger E. Edwards, 'Towards developing a systematic knowledge trend for building energy consumption prediction', *Journal of Building Engineering*, vol. 35, Mar. 2021, doi: https://doi.org/10.1016/j.jobe.2020.101967.

[10] Kadir Amasyali and Nora M. El-Gohary, 'A review of data-driven building energy consumption prediction studies', *Renewable and Sustainable Energy Reviews*, vol. 81, no. Part 1, pp. 1192–1205, Jan. 2018, doi: https://doi.org/10.1016/j.rser.2017.04.095.

[11] Yixuan Wei *et al.*, 'A review of data-driven approaches for prediction and classification of building energy consumption', *Renewable and Sustainable Energy Reviews*, vol. 82, pp. 1027–1047, Feb. 2018, doi: https://doi.org/10.1016/j.rser.2017.09.108.

[12] Richard E .Edwards, Joshua New, and Lynne E. Parker, 'Predicting future hourly residential electrical consumption: A machine learning case study', *Energy and Buildings*, vol. 49, pp. 591–603, Jun. 2012, doi: https://doi.org/10.1016/j.enbuild.2012.03.010.

[13] Fazil Kaytez, M. Cengiz Taplamacioglu, Ertugrul Cam, and Firat Hardalac, 'Forecasting electricity consumption: A comparison of regression analysis, neural networks and least squares support vector machines', *International Journal of Electrical Power & Energy Systems*, vol. 67, pp. 431–438, May 2015, doi: https://doi.org/10.1016/j.ijepes.2014.12.036.
[14] Filipe Rodrigues, Carlos Cardeira, and J.M.F. Calado, 'The Daily and Hourly Energy Consumption and Load Forecasting Using Artificial Neural Network Method: A Case Study Using a Set of 93 Households in Portugal', *Energy Procedia*, vol. 62, pp. 220–229, 2014, doi: https://doi.org/10.1016/j.egypro.2014.12.383.
[15] Cheng Fan, Fu Xiao, and Yang Zhao, 'A short-term building cooling load prediction method using deep learning algorithms', *Applied Energy*, vol. 195, Jun. 2017, doi: https://doi.org/10.1016/j.apenergy.2017.03.064.
[16] Jiantao Zhao and Jingchang Huang, 'Application and research of short-term Power load forecasting based on LSTM', *2020 International Conference on Information Science, Parallel and Distributed Systems (ISPDS)*, Nov. 2020, doi: 10.1109/ISPDS51347.2020.00050.
[17] Xin Wang, Fang Fang, Xiaoning Zhang, Yajuan Liu, Le Wei, and Yang Shi, 'LSTM-based Short-term Load Forecasting for Building Electricity Consumption', *2019 IEEE 28th International Symposium on Industrial Electronics (ISIE)*, Aug. 2019, doi: 10.1109/ISIE.2019.8781349.
[18] Can Cui, Ming He, Fangchun Di, Yi Lu, Yuhan Dai, and Fengyi Lv, 'Research on Power Load Forecasting Method Based on LSTM Model', *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, Jul. 2020, doi: 10.1109/ITOEC49072.2020.9141684.
[19] Fan Zhang, Chirag Deb, Siew Eang Lee, Junjing Yang, and Kwok Wei Shah, 'Time series forecasting for building energy consumption using weighted Support Vector Regression with differential evolution optimization technique', *Energy and Buildings*, vol. 126, pp. 94–103, Aug. 2016, doi: https://doi.org/10.1016/j.enbuild.2016.05.028.
[20] Hai Zhong, Jiajun Wang, Hongjie Jia, Yunfei Mu, and Shilei Lv, 'Vector field-based support vector regression for building energy consumption prediction', *Applied Energy*, vol. 242, May 2019, doi: https://doi.org/10.1016/j.apenergy.2019.03.078.
[21] Yamuna Maccarana, Angela Panza, Gabriele Maroni, and Luca Sarto, 'Comparison of model-based and data-driven approaches for modeling energy and comfort management systems, with a case study', *2019 IEEE International Conference on Environment and Electrical Engineering and 2019 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)*, Aug. 2019, doi: 10.1109/EEEIC.2019.8783703.
[22] Haoxiang Li, Qi Zhou, Jing Tian, and Xiaoyu Lin, 'Energy Demand Forecasting for an Office Building Based on Random Forests', *2020 IEEE 4th Conference on Energy Internet and Energy System Integration (EI2)*, Feb. 2021, doi: 10.1109/EI250167.2020.9347021.
[23] Ahmed Ghareeb, Hussein Al-bayaty, Qubad Haseeb, and Mohammed Zeinalabideen, 'Ensemble learning models for short-term electricity demand forecasting', *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)*, Jan. 2021, doi: 10.1109/ICDABI51230.2020.9325623.

[24] Halima Haque, Adrish Kumar Chowdhury, M. Nasfikur Rahman Khan, and Md. Abdur Razzak, 'Demand Analysis of Energy Consumption in a Residential Apartment using Machine Learning', *2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, May 2021, doi: 10.1109/IEMTRONICS52119.2021.9422593.
[25] Robin John Hyndman and George Athanasopoulos, *Forecasting: Principles and Practice*. [Online]. Available: Hyndman, R., & Athanasopoulos, G. (2021). Forecasting: Principles and Practice. (3rd ed.) OTexts. https://otexts.com/fpp3/
[26] A.S. Ahmad *et al.*, 'A review on applications of ANN and SVM for building electrical energy consumption forecasting', *Renewable and Sustainable Energy Reviews*, vol. 33, pp. 102–109, May 2014, doi: https://doi.org/10.1016/j.rser.2014.01.069.
[27] Neha Sharma, Reecha Sharma, and Neeru Jindal, 'Machine Learning and Deep Learning Applications-A Vision', *Global Transitions Proceedings*, vol. 2, no. 1, pp. 24–28, Jun. 2021, doi: https://doi.org/10.1016/j.gltp.2021.01.004.
[28] Devin Soni, 'Supervised vs. Unsupervised Learning', Mar. 22, 2018. https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d
[29] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*. MIT Press, 2016. [Online]. Available: https://www.deeplearningbook.org/
[30] Jason Brownlee, 'Why Use Ensemble Learning?', Oct. 26, 2020. https://machinelearningmastery.com/why-use-ensemble-learning/
[31] Ashwin Raj, 'Unlocking the True Power of Support Vector Regression', Oct. 03, 2020. https://towardsdatascience.com/unlocking-the-true-power-of-support-vector-regression-847fd123a4a0
[32] Wikipedia contributors, 'Support-vector machine'. Wikipedia, The Free Encyclopedia. Accessed: Oct. 08, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Support-vector_machine&oldid=1048841631
[33] Kurtis Pykes, 'The Vanishing/Exploding Gradient Problem in Deep Neural Networks', May 17, 2020. https://towardsdatascience.com/the-vanishing-exploding-gradient-problem-in-deep-neural-networks-191358470c11
[34] Jair Ribeiro and Jair Ribeiro, 'What is Predictive Analytics, and how can you use it today?', Dec. 04, 2020. https://towardsdatascience.com/what-is-predictive-analytics-dc6db9759936
[35] Jason Brownlee, 'How to use Data Scaling Improve Deep Learning Model Stability and Performance', Feb. 04, 2019. https://machinelearningmastery.com/how-to-improve-neural-network-stability-and-modeling-performance-with-data-scaling/
[36] Jason Brownlee, 'How to Choose an Activation Function for Deep Learning', Jan. 18, 2021. https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/
[37] Sanket Doshi, 'Various Optimization Algorithms For Training Neural Network', Jan. 13, 2019. https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6
[38] Jason Brownlee, 'How to Configure the Learning Rate When Training Deep Learning Neural Networks', Jan. 23, 2019. https://machinelearningmastery.com/learning-rate-for-deep-learning-neural-networks/
[39] Jason Brownlee, 'Time Series Forecasting with the Long Short-Term Memory Network in Python', Aug. 28, 2020. https://machinelearningmastery.com/time-series-forecasting-long-short-term-memory-network-python/

[40]Meftah Elsaraiti, Gama Ali, Hmeda Musbah, Adel Merabet, and Timothy Little, ‘Time Series Analysis of Electricity Consumption Forecasting Using ARIMA Model’, *2021 IEEE Green Technologies Conference (GreenTech)*, Jun. 2021, doi: 10.1109/GreenTech48523.2021.00049.

[41] S Karthika, Vijaya Margaret, and K. Balaraman, ‘Hybrid short term load forecasting using ARIMA-SVM’, *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*, Jan. 2018, doi: 10.1109/IPACT.2017.8245060.

[42] Lingling Tang, Yulin Yi, and Yuexing Peng, ‘An ensemble deep learning model for short-term load forecasting based on ARIMA and LSTM’, *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, Nov. 2019, doi: 10.1109/SmartGridComm.2019.8909756.

Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis[1]

I Georgia Kountioudi

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis “Analysis, modelling and prediction of energy consumption in an office building” , supervised by Professor Eduard Petlenkov, PhD.
   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

03.01.2022

1 The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.