BRUNEL UNIVERSITY LONDON

DOCTORAL THESIS

DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING

Artificial Intelligence for Optimisation and Demand Side Response in Built Environment

Supervisor:
Dr QingPing Yang

Co-Supervisor:
Dr. Valerie N. Livina
(National Physical Laboratory)

Co-Supervisor:
Lee Stokes
(Mitie Energy)

Author:
José Joaquín MESA JIMÉNEZ

January 25, 2021

Abstract

In recent years, data analytics and machine learning have become important for generating insights and creating competitive advantages across many industries. However, despite recent developments in analytics and machine learning technologies, as well as accessible tools to perform the implementation of such technologies, built environment is still a largely unexplored field, where many engineering operations remain manual. Recent advances in building management systems and data engineering have provided vast amount of data streams of all types of sensors around the built environment: comfort variables, assets, security installations, meteorological measurements, electricity demand, etc. This creates a wide range of opportunities to explore data and extract real value for various operational purposes. The aim of this Thesis is to develop a series of tools related for control and optimisation of built environment, from demand side response events prediction to building operations management. Equipped with the results of this work, building managers will be more prepared to respond to energy demand events, organise energy resources more efficiently and to acquire a proactive approach to system failures and system errors tractability. The pilot results of the thesis have been successfully implemented in industrial applications.

Acknowledgements

I would like to thank the Department for Business, Energy and Industrial Strategy of the United Kingdom and the College of Engineering, Design and Physical Sciences of Brunel University London for funding this research. Also a special thanks to my supervisory team at BUL. Thanks to Dr. Valerie Livina for her continuous support, vast experience in research and always wise and useful advice, to Lee Stokes for being my business mentor, for his vast creativity and his practical approach to business problems, and to Dr. QingPing Yang for his support in different events at BUL and for this opportunity. Without them, this work would just not be possible. Last but not least, I would like to thank Alicia, for her continuous support and understanding of my (sometimes) stressful situations and for always, always being there.

Contents

List of Figures

List of Tables

Nomenclature

ACF Auto-Correlation Function

AHC Agglomerative Hierarchical Clustering

AHU Air Handling Unit

ANFIS Adaptive Neural Fuzzy Inference System

ANN Artificial Neural Network

APAR Air handling unit Performance Assessment Rules

APCS Adaptive Predictive Control Strategies

ARIMA Autoregressive Integrated Moving Average

BFGS Broyden–Fletcher–Goldfarb–Shanno

BMS Building Management System

BOO Build Own Operate

BOT Build Operate Transfer

BPTT Back Propagation Through Time

CNN Convolutional Neural Network

COBYLA Constrained Optimisation By Linear Approximation

CPP Critical Peak Pricing

DFA Detrended Fluctuation Analysis

DNN Deep Neural Network

DSR Demand Side Response

DTW Dynamic Time Warping

ELM Extreme Learning Machine

EMA Exponential Moving Average

EOF Empirical Orthogonal Functions

FCU Fan Coil Unit

FDD Fault Detection and Diagnosis

FE Forecasted Error

GHG Greenhouse Gas Emissions

GRNN General Regression Neural-Network

GRU Gated Recurrent Unit

HDA Historical Data Augmentation

HI Health Indicator

HLSE Hierarchical Label Set Expansion

HVAC Heating, Ventilation, and Air Conditioning

ID3 Iterative Dichotomiser 3

INDO Initial Demand Out-Turn

IoT Internet of Things

IPP Independent Power Producer

KNN K-Nearest Neighbors

L-BFGS-B Limited memory Broyden Fletcher Goldfarb Shanno Box

LSSVM Least Square Support Vector Machine

LSTM Long Short-Term Memory

MA Moving Average

MAE Mean Absolute Error

MARS Multivariate Adaptive Regression Spline

MaxEnt Maximum-Entropy classification

MCMC Markov Chain Monte Carlo

MLE Maximum Likelihood Estimation

MLR Multiple Linear Regression

MTS Multivariate Time Series

NARX Recurrent nonlinear autoregressive exogenous

NB Naive Bayes

NDF National Demand Forecast

NLP Natural Language Processing

NN Neural Network

OAT Outside Air Temperature

PCA Principal Component Analysis

PPA Power Purchase Agreement

PR-curve Precision-Recall curve

PS Power Spectrum

R1 First Reconciliation

R2 Second Reconciliation

RBF Radial Basis Function

RES Renewable Energy Sources

RMSE Root Mean Square Error

RNN Recurrent Neural Network

SARIMA Seasonal Autoregressive Integrated Moving Average

SAT Supply Air Temperature

SD Settled Demand

SF Settlement Final

SLSQP Sequential Least Squares Programming

SVM Support Vector Machines

TF-IDF Term Frequency-Inverse Document Frequency

TNC Truncated Newton

TNUoS Transmission Network Use of System

ToU Time of Use

TSDF Transmission Demand Forecast

TSVM Transductive Support Vector Machine

VAE Variational Auto Encoder

VAERE Variational Auto Encoder Reconstruction Error

XMTC Extreme Multi-label Text Classification

Chapter 1

Introduction

With the rise of the Internet of Things (IoT) technologies and the recent applications to managing all aspects of buildings, there is a huge potential in computational solutions. Machine learning solutions in the built environment, demand side response or electricity purchase through power purchase agreements, are still a largely unexplored field. Due to the amount of data generated from sensors or available information on energy systems, the requirements for analytics and automation become increasingly complex and, therefore, requiring complex solutions. The aim of this thesis is to develop a series of solutions that encompass the automation of the built environment: Demand Side Response (DSR), Power Purchase Agreements (PPA) and Building Management Systems (BMS) optimisation. These three areas are the main building blocks of this thesis. The aim is to enhance smart building capabilities by creating an unique framework for built environment optimisation utilising machine learning. As the built environment comprises the different areas mentioned earlier, the work has been subdivided into several tasks that compose this framework. The main tasks comprising these areas are outlined as follows:

- To correctly detect the peaks of highest demand of the year at a national level, so further action is being taken in response to DSR programs.
- To efficiently choose and combine renewable energy assets for a building or for sets of buildings, thus reducing the risks associated with the uncertainty and variable generation patterns to a minimum and promoting the use of renewable energy sources.
- To reduce operational times by using a dynamic approach. The motivation of this task is the lack of documentation problem with respect to HVAC equipment and sensors installed, which makes errors tractability and assets linkage very difficult in daily building operations.
- To reduce HVAC maintenance costs by adopting a proactive approach to HVAC failures instead of a reactive approach by acting before the failure happens.
- To automate this manual work for decreasing mobilisation time, as the current time spent for building mobilisation to an analytics platform requires a manual process of several days of manual tagging.

Another challenge is the lack of data provided to achieve these goals. Not enough data is normally provided for industrial applications in BMS. In comparison to other studies, where sufficient data from specific HVAC systems is available, the data used here is limited, e.g. only room temperature sensors are used instead of internal parameters of air handling units, which is not provided. This adds an extra challenge to the

problem that needs to be dealt with.

The motivation for each of these challenges to be tackled within the context of building management systems are outlined as follows:

First, for DSR interventions, a method for detecting and forecasting events of high energy demand is proposed, which are managed at national level in DSR programmes, such as the UK Triads. The methodology consists of two stages: load forecasting with Long-Short-Term-Memory neural network and dynamic filtering of the potential highest electricity demand peaks by using the exponential moving average. The methodology is validated on real data of a UK building management system.

Second, once the periods of highest demand for DSR programs have been identified to generate energy savings, the problem of the choice of the optimum combination of renewable sources is tackled. Due to decrease of the electricity prices from solar and wind generators, and the European Union regulations for decarbonisation of the economy, more than 40% of the most important Fortune 500 companies in the world now have targets related to green energy. This is one of the main reasons why multi-technology Power Purchase Agreements (PPAs) are becoming increasingly important. However, there are risks associated with the uncertainty and variable generation patterns in wind speed and solar irradiation. Besides, there are not many approaches to predict wind and solar generation in the forecasting horizon required by PPAs, which is usually of several years. Long-term wind and solar energy generation forecasts are proposed, followed by cost optimisation in energy generation scenarios.

Third, when the DSR opportunities and optimal combination of renewable generation assets have been considered for optimal resource allocation and energy savings opportunities, optimisation of building operations are considered using building sensors. In large buildings, linking heating, cooling or ventilation systems between themselves and to physical spaces is a very time-consuming task that requires highly skilled engineering knowledge, as all these systems are interconnected and they have a certain influence to each other (ventilation systems are often connected to heating and cooling), which often makes task of locating the sources of error or anomalies very time consuming and difficult as they are performed manually. A different approach would be to work out relationships and equipment linkage from time series data provided by the sensors, thus inferring equipment links from which anomalies can be traced back to the source more easily. A data-based solution is proposed to obtain equipment relationships based on cross-correlations to relate Air Handling Units (AHUs) to their respective areas of operation. A methodology is also proposed to identify whether or not to trust correlations based on the difference between supply and return temperature. A case study is presented based on a large building with 16 AHU systems.

Fourth, internal temperature sensors are used also for operational optimisation, but applied to early signals detection. In the context of sensor data generated by BMS, early warning signals are still an unexplored topic. The early detection of anomalies can help prevent malfunctions of key parts of a heating, cooling and air conditioning (HVAC) system that may lead to a range of BMS problems, from energy waste to fatal errors in the worst case. Early warning signals of BMS sensor data were analysed for early failure detection. Variance, lag-1 autocorrelation function (ACF1), power spectrum (PS) and variational autoencoder (VAE) techniques are applied to both univariate and multivariate scenarios.

Fifth, once the internal operational aspects of the building have been considered, the next step is the on-boarding speed. When working with building management systems, one faces two problems. On the one hand, each Building Management System (BMS) manufacturer has proprietary data structures and architecture. On the other hand, there is no standard for the naming conventions of the sensors or equipment within these structures by default. Translating building sensor points into a naming standard is a very time-consuming task that requires highly skilled engineering knowledge. This should be done using each item's name in the BMS (points) into the clean structured format (labels). The necessity of this manual step in the process is slowing the proliferation of IoT integration with existing BMS and causing large costs to companies during the BMS roll out. Medium to large buildings have between 200 and 1000 points. Manually translating these points into labels (the tagging process) takes around 8 hours for every 100 points. AI based series of multi-label classification methodologies are proposed to translate existing BMS points into formatted labels automatically. The methods are compared using several accuracy metrics and then the best performing algorithm is chosen.

This thesis considers to first tackle the external aspects of the building, in relation to DSR and renewable resources optimisation in a real case scenario, as PPA contracts have a length of 10 to 15 years and DSR events can be predicted every winter. Then, internal and operational aspects of the building have been considered, as this is a process of continuous improvement within the building, as more cases for internal optimisation can be considered by using the results of this work as a solid background. The scheme of the above for the industrial application can be seen in Figure 1.1.

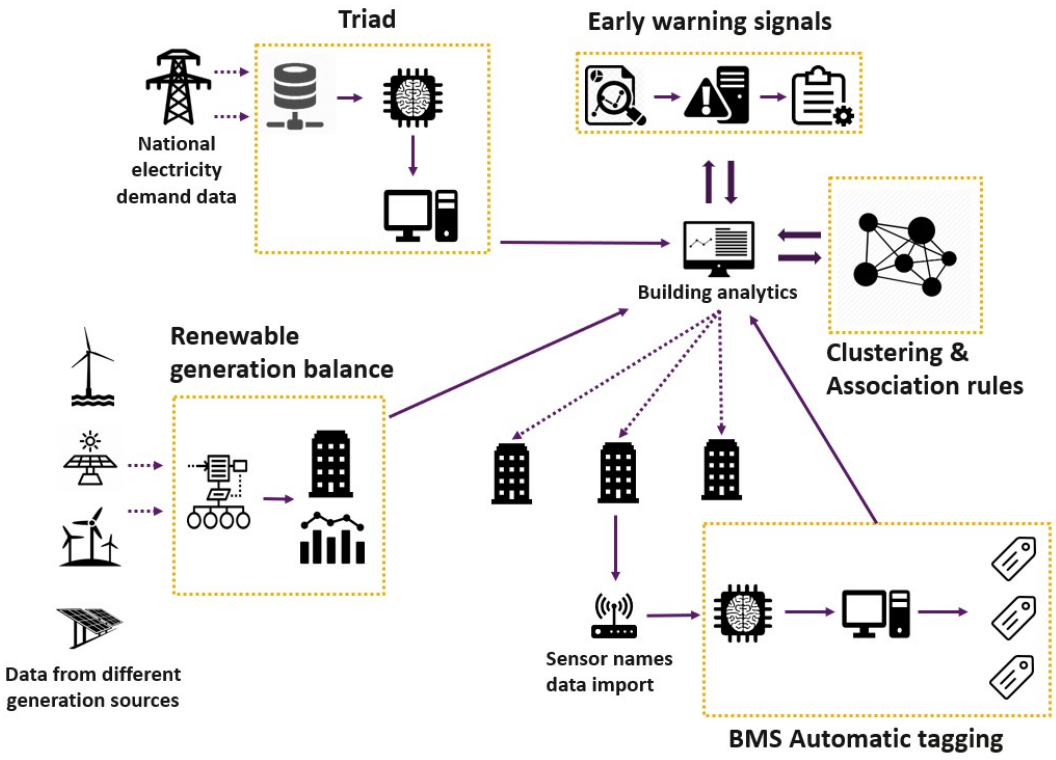


Figure 1.1: Schematic of industrial application for the Thesis.

As can be seen in Figure 1.1 Triad, which is the specific case study for DSR events, extracts national electricity demand data and stores it in a data base, before being run in an

application whose results are input to the building analytics platform. The PPA functionality gathers data from different providers manually, before performing forecasting and optimal asset combination. Early warning signals and BMS clustering use data directly from the building analytics platform, in order to generate signals as independent applications. The automatic tagging problem imports the names from a database remotely, which are then processed and classified before building mobilisation.

In general, this works includes a series of machine learning applications in the unexplored area of smart buildings. In this thesis, these applications solve common problems of control optimisation of the built environment. Equipped with the results of this work, building managers will be more prepared to respond to and organise energy resources more efficiently and to acquire a proactive approach to system failures and errors tractability, with a successful implementation in industrial applications. This thesis considers both internal and external aspects of the building. The external aspects are related to optimisation of energy resources and demand side response, and internal aspects refer to building sensor data that measures temperature and other factors related to policy comfort, HVAC systems or electricity consumption (meter data).

The main contributions to knowledge of this Thesis in terms of demand side response, power purchase agreements and building management systems can be summarised as:

- Development of a deep learning system for detecting and forecasting events of high energy demand at a national level for general DSR interventions. With $RRMSE = 2.23\%$ achieved in LSTM and being able to predict Triads with a fixed level of risk.
- Development of methods for long-term wind and solar generation forecast and a linear programming-based method for optimal combination of renewable assets with no weather data available.
- Methods for obtaining relationships between HVAC assets and their respective areas of influence in large facilities through clustering time series analysis from temperature sensor data. Results show a 70% of correct association of Air Handling Units (AHUs) with their respective areas of operation in a large facility with 16 AHUs and a correct Fan Coil Unit (FCU)-room association.
- Development of early warning signals for temperature sensor data that detects anomalies up to 32 hours prior to failure.
- An automatic method for translating building sensor points into a haystack naming standard that uses multi-stage text classification and that achieves a 90% of true positives. Also, the addition of a confidence formula to create a boundary for detecting false positives has been developed.

This thesis is organised as follows: In Chapter 2, an analysis of the current state of research is presented. Chapter 3 describes the methodology used for the experiments. This Chapter is divided in four main parts: the first one focused on time series forecasting and estimation, describing the techniques used in the DSR intervention and PPA assessment. The second one focuses on time series clustering, the third one in Early Warning Signals (EWS) methodologies and the fourth one is focused on text classification methodologies. The design of experiments for the five problems mentioned above, description of the problem structure and the data characteristics involved on each of them, as well as the results are outlined in Chapter 4. Finally conclusions, limitations and further work are presented in Chapter 5.

Chapter 2

Literature review

In recent years, smart building efforts were focused on cutting costs by streamlining building operations such as air conditioning and lighting. However, despite of the recent developments in analytics and machine learning, very little effort has been made in exploring these capabilities in buildings, despite of the vast amount of data available nowadays from building management systems. Managing building operations is critical to reducing annual building operating costs as mentioned in older works such as (Snoonian, 2003), however, there is a whole range of methodologies to apply to this field that have not been explored, even to the date. Recently, Internet of Things and cloud computing took part in the boost of smart buildings capabilities (Plageras et al., 2018), which gives plenty of room for improvement on analytic tools to enhance such capabilities.

In another classic of the smart buildings literature, Snoonian (2003) points out that smart buildings are the basic building blocks for smart cities, and explores several aspects of the buildings to be considered for smart building capabilities, such as demand side management capabilities, micro renewables, micro energy storage and electricity-price based consumption controller. This work considers that demand side response capabilities of buildings should be considered within the smart buildings tool set: for consumer electricity prices on the one hand, and for grid frequency and demand balance on the other.

In more recent works, new technologies are incorporated to smart buildings to better balance supply and demand by decentralised mechanisms. Such as the use of blockchain technologies in Van Cutsem et al. (2020), which presented a decentralised framework to manage the electrical consumption in a community of Smart-Buildings and local Renewable Energy Sources (RES). Froufe et al. (2020) identifies 11 drivers which foster buildings to be smarter emerged and evolved, which were divided in three categories: users (health, well-being and meeting expectations), owners (improvement of the cost-benefit ratio) and environment (reduction of consumption emissions and improvement of the interaction of the building with the environment). The work presented in this thesis aims to cover these three areas. Another work that aims to reduce consumption is done by De Paola et al. (2020), which presents a novel hybrid intelligent architecture which exploits heterogeneous devices pervasively deployed in the environment to reduce the energy consumption of buildings. Studies that use of IoT in smart buildings have recently increased due to the many potential applications in this area, as it is the case of Casado-Vara et al. (2020), which uses a technique named IoT slicing that combines complex networks and clusters in order to reduce algorithm input errors and improve the monitoring and control of a smart building, to transform temperature data so it can be used as an input for monitoring and control algorithms in smart buildings to optimise their performance.

Other IoT applications in smart buildings can be found, such as Carli et al. (2020) which considers an IoT based architecture for model predictive control of HVAC systems that optimizes the indoor thermal comfort and the energy consumption.

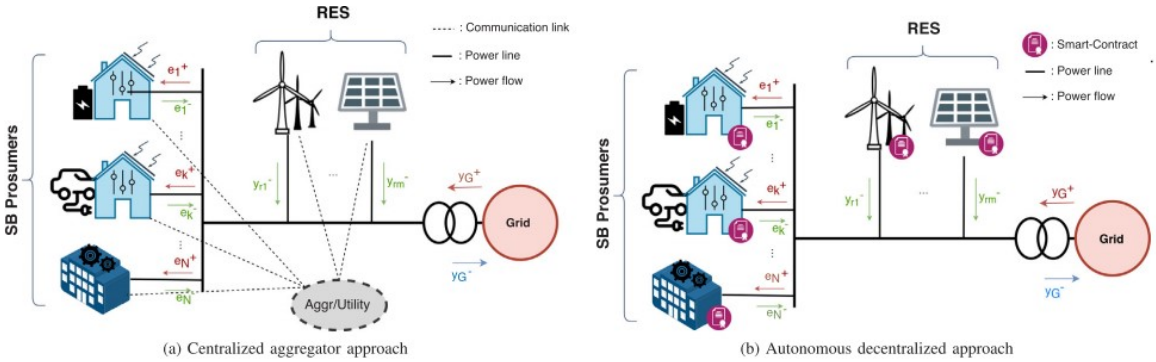


Figure 2.1: Smart-Buildings community with local RES: structure, communication, and power flows (Van Cutsem et al., 2020)

The consideration of heating, ventilation and air conditioning systems as a key part of smart buildings is also included in this work. According to (Weng et al., 2012), in the US buildings consume a 70% of the total electricity generated. They discussed the importance of actuation in making buildings more energy efficient through the development of several mechanisms for actuating the HVAC system and controlling plugs based on occupancy. More recent works focused on HVAC systems such as Gholamzadehmir et al. (2020) which acknowledges that the highest amount of energy consumption occurs in buildings, and reviews advanced control strategies and their impact on buildings and technical systems with respect to energy/cost saving. The scheme of application through a cloud platform can be seen in Figure 2.2. Other works consider forecasting internal space temperatures accurately for HVAC systems control, such as Mtibaa et al. (2020), which considers long short-term memory networks to predict indoor air temperature based on direct multi-step prediction. Yu et al. (2020) uses an HVAC control algorithm is proposed to solve the Markov game based on multi-agent deep reinforcement learning with attention mechanism, which does not require any prior knowledge of uncertain parameters and can operate without considering thermal dynamics models.

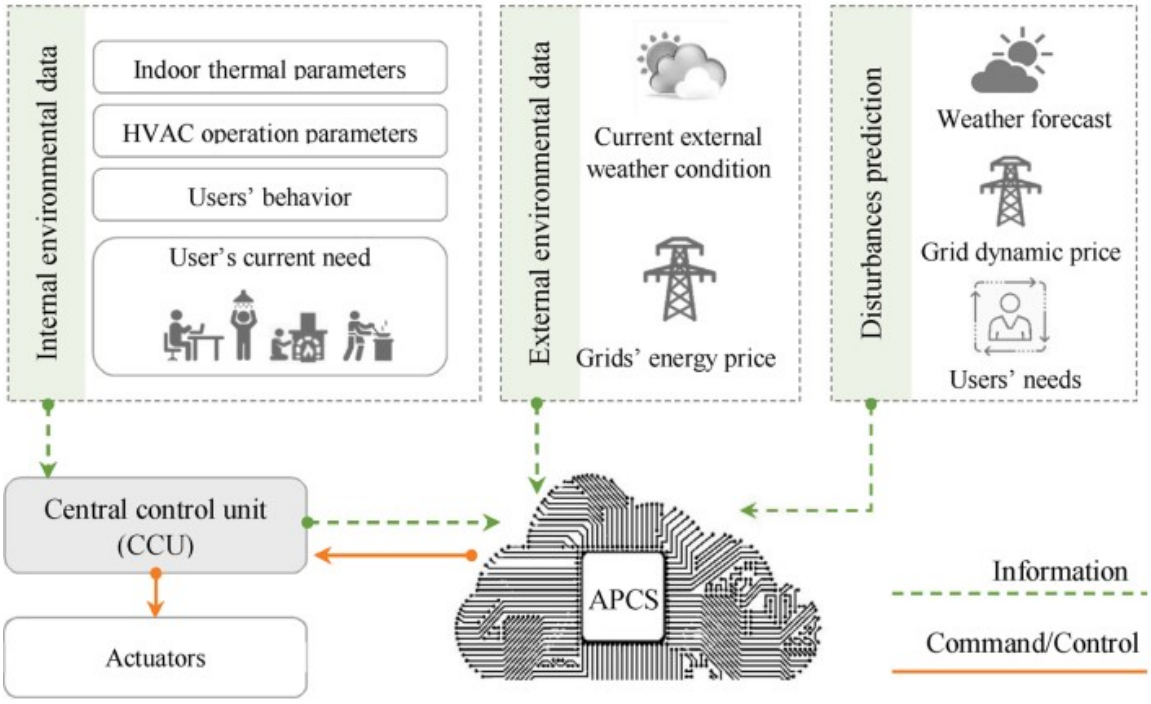


Figure 2.2: Application of Adaptive Predictive Control Strategies (APCS) as a supervisory control system through a cloud platform (Gholamzadehmir et al., 2020)

This work considers a series of machine learning techniques for both energy systems (forecasting techniques and software for demand side response and power purchase agreements) and HVAC, which also includes early warning signals, text classification for efficient building mobilisation and automatic sensor group clustering to identify links between sensor data, equipment and their physical areas of influence.

2.1 AI for demand, wind and solar generation forecast

2.1.1 DR interventions and load forecasting

The changes in energy policy aim to substantially increase renewable energy generation and reduce carbon emissions. Addressing the growing energy demand, aging infrastructure and intermittency of renewable energy requires an efficient forecasting methodology to predict periods of peak energy demand. Long-term power load forecasting at national level is an important basis for Demand Side Response (DSR) planning, which aims to reduce the need for last-minute energy generation from non-renewable sources.

In the energy sector, Demand Side Response is meant to substantially reduce the need for investment in peak generation. This is done by minimising consumption at times of high demand. With the goal of adding stability to the system, demand response lowers the need for coal and gas-fired spinning reserves. This reduces carbon emissions because most power plants burn fuel/coal continuously in order to supply power at short notice and thus reduces anthropogenic impact and decreases the need for local network investments. Demand side response refers to "voluntary changes by consumers of their electricity use pattern" (ELEXON, 2018), either in response to changes in the price of electricity over time or through incentive payments.

Reducing electricity demand peaks is a key issue for DSR programs and for reduction

of carbon emissions, as less power will have to be generated by coal and gas. In Faruqui et al. (2010), 15 recent assessments of residential dynamic pricing programs are surveyed, most conducted in the US after the year 2000. According to their survey, Time-of-Use (ToU) tariffs induce a reduction in peak consumption that ranges from 3% to 6% and Critical Peak Pricing (CPP) has the effect of decreasing peak usage by between 13% and 20%. In a more recent study, Li et al. (2020b) investigates UK consumers heterogeneous engagement in demand-side response, focusing on the activities that constitute the major part of domestic energy consumption. The findings can potentially help policy makers to focus on more targeted DSR plans and improve existing models that consider these factors. The variety of DSR programs has been increasing in Europe over the past years. However, the number of systems specifically oriented to national DSR programs in the scientific literature is lacking.

There is a rich variety of methodologies for peak load forecasting, and there have been significant improvements in time series forecasting due to the increase of the computer capacity which has led to new computational methods such as Machine Learning and other AI approaches. In Kouroupetroglou et al. (2017), in the comparison of machine learning models for short-term load forecasting in the Greek electric grid, six machine learning methods are compared: Support Vector Machines (SVM), K-Nearest Neighbors (KNN), random forests, Neural Networks (NN), xgboost and model trees. This is very relevant to the present research due to its load forecasting methods used at a national level. Four experiments were performed in order to minimize the error of prediction accuracy. The results of these experiments show that, overall, model trees performed better in terms of prediction error, followed by xgboost and SVM. In another comparative study, (Al-Musaylh et al., 2018), three methodologies are compared for electricity demand forecasting: Multivariate Adaptive Regression Spline (MARS), Autoregressive Integrated Moving Average (ARIMA) and SVM. The results of the study show that, in terms of statistical metrics, MARS model yielded the most accurate results for 1/2-hours and 1-hour forecasts, whereas the SVR models were better for a 24 h horizon and the ARIMA model's performance was lower for all forecasting horizons as it generated very high forecast errors.

Another commonly used approach for load forecast are Artificial Neural Networks (ANNs). ANNs are composed of a network of processing nodes (or neurons), which perform numerical transformations and are interconnected in a specific order so different weights are assigned to give importance to different factors through training the network. According to Chen et al. (1992), ANNs are well-known for being able to forecast the outputs of nonlinear datasets, to efficiently perform different simultaneous tasks. There are several studies for load forecasting using ANNs, such as the comparative study in Kandananond (2011), in which three methodologies, ARIMA, ANN and Multiple Linear Regression (MLR) were deployed to forecast the electricity demand in Thailand. The results showed that based on the historical data and on the error measurement the ANN model was superior to the other two. In Filik et al. (2011), mathematical models and neural networks to forecast the long-term electricity demand in Turkey are compared. Some short-term load forecasting studies combine ANNs with other methods, such as Saini et al. (2002) which is an ANN based peak load forecasting using Levenberg-Marquardt and quasi-Newton methods. Also, González-Romera et al. (2008) focused on the periodic behavior of consumption for forecasting the Spanish monthly electricity demand, in which the trend of electricity demand was predicted using an ANN combined with Fourier series. There are novel alternative methods that have been compared to more traditional ANN approaches, such as Singh et al. (2018), whose study integrates an evolutionary

approach with ANNs for short-term load forecast, based on "follow the leader" behavior (Fernandez et al., 2011). This hybrid approach is compared with four other variations of ANNs, showing that they are outperformed by the "follow the leader" hybrid approach in terms of predictive accuracy measures. The emerging class of ANN, extreme learning machine (ELM) plays an important role for this purpose in Li et al. (2016), because it is invoked to predict the hourly load of the next day and it improves its performance significantly. In more recent studies, ANNs are less frequently used due to the new techniques that outperform ANNs, however different variations of this method are appearing recently, such as Aly (2020), which uses bootstrap aggregating based ensemble ANN for short term load forecasting, or hybrid methods such as the support vector regression and ANNs, also for short-term load forecasting, in Abad et al. (2020).

Deep Neural Networks (DNNs) are also used for the purpose of load forecasting. DNNs are ANNs with several hidden layers, adding complexity to its structure. He (2017) studies one day ahead forecasting of hourly loads based on deep networks. The study of Hamedmoghadam et al. (2018) has a more specific goal, to use DNNs to predict the monthly electricity demand in Australia based on time series of consumption rates as well as socioeconomic and environmental factors. In a more recent study, Sideratos et al. (2020) combines attributes from ensemble forecasting, ANNs and deep learning architectures to predict the hourly load for the next seven days. Also Lai et al. (2020) presents a novel load forecasting method known as deep neural network and Historical Data Augmentation (HDA), which uses historical data augmentation to enhance regression by the DNNs.

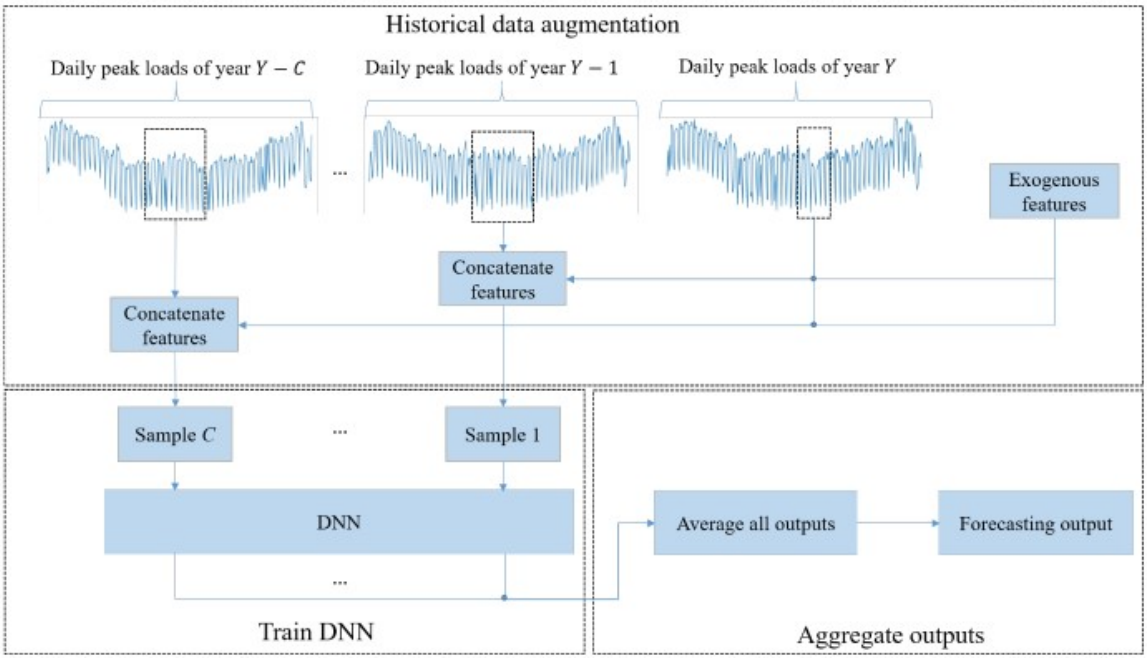


Figure 2.3: Framework of the DNN-based regression model and HDA (Lai et al., 2020)

Other methodologies such as Radial Basis Function (RBF) have been used to address the problem of load forecasting, (Yun et al., 2008; Liu et al., 2017a; Khwaja et al., 2017). The study of Yun et al. (2008) combines the RBF neural network with the Adaptive Neural Fuzzy Inference System (ANFIS) to adjust the prediction by taking into account the real-time electricity price. Khwaja et al. (2017) compares three different versions of RBF to predict electricity load. In the area of short-term load forecasting, Cao et al. (2015) ad-

dressed this problem by using ARIMA model and similar day method for intraday load forecasting. For very short-term load forecasting, Qingle et al. (2010) also proposed an ANN-based predictor that takes the load values of the current and previous time steps as the input to predict the load value at the coming step.

SVMs are also very relevant for load forecasting. This is shown in Chen et al. (2004) and Hong (2009), as well as more recent modified SVM versions, which are combined with other methods in order to achieve a better accuracy. This is the case of Daut et al. (2017) for load forecasting method using a combined Least Square SVM (LSSVM) and modified artificial bee colony (ABC-LSSVM), which proved to have a better performance than the standard ABC-LSSVM and LSSVM. Another example of modified SVM for load forecasting is shown in Liu et al. (2017a), which uses the sperm whale algorithm and wavelet least square support vector machine with DWT-IR for feature selection. Also, more recent versions of this algorithm combined with other techniques produce satisfactory results, as it is the case of Feng et al. (2019), which combines the dragonfly algorithm and SVMs to produce short term load forecasting of offshore oil field microgrids. For short-term load forecasting, improved SVM with extreme machine learning is used, using xgboost and decision trees as a hybrid feature selector is proposed in Ahmad et al. (2020). This is illustrated in Figure 2.4

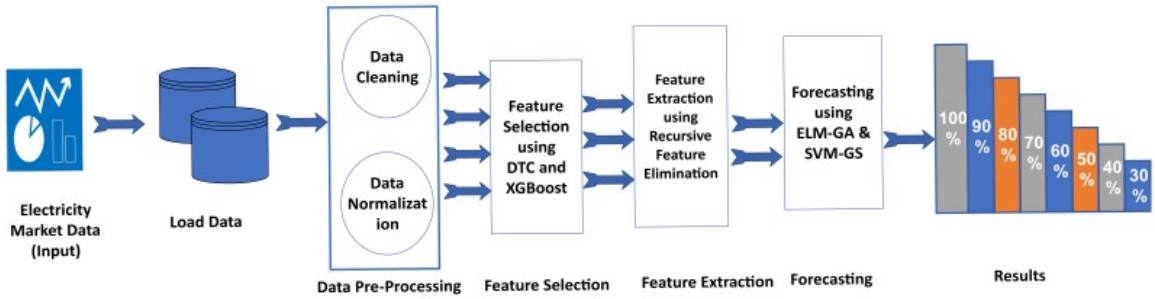


Figure 2.4: Model proposed by Ahmad et al. (2020)

Recurrent Neural Networks (RNNs) are very popular in the scientific literature as they can work on sequences of arbitrary length. More particularly, in Bianchi et al. (2017), a comparative study of short-term load forecast is performed by using different classes of RNNs, and although there is not a specific RNN model that outperforms the others in every prediction problem, it shows that LSTM and Gated Recurrent Units (GRUs) achieve outstanding results in many sequence learning problems. LSTM, together with GRUs, present no vanishing/exploding gradient problem. This has been proven in Zheng et al. (2017) which shows that LSTM outperforms traditional forecasting methods in the short-term electric load forecasting. They compare its performance with other methods such as Seasonal Autoregressive Integrated Moving Average model (SARIMA), a nonlinear autoregressive neural network model with exogenous inputs (NARX), SVM and NNETAR, a feed-forward neural network model for univariate time series forecasting with a single hidden layer and lagged inputs. Some other studies combine these methodologies, such as Tian et al. (2018), which uses a deep neural network model for short-term load forecast based on LSTM and Convolutional Neural Network (CNN), achieving the lowest error in comparison to the other algorithms tested. Kong et al. (2019) performs short-term load residential load forecasting using an LSTM Recurrent Neural Network showing a Mean Absolute Percentage Error (MAPE) between 1.5% to 35%, depending on the household. There are other several publications about LSTM for speech recognition, sentiment analysis and autonomous driving systems (Graves et al., 2013; Wang et al., 2016; Xu et al.,

2016). The abundance of more recent studies in this field proves that these techniques are still useful in this field. One example of this is shown in Bouktif et al. (2020), which shows multi-sequence LSTM-RNN deep learning and meta-heuristics for electric load forecasting. A short-term load forecasting algorithm using RNNs with input attention mechanism and hidden connection mechanism is shown in Zhang et al. (2020), whose results prove better than others for experiments in two datasets. A typical architecture of RNNs with a typical attention mechanism is shown in Figure 2.5. A similar approach which uses sequence to sequence RNNs and attention is Sehovac et al. (2020).

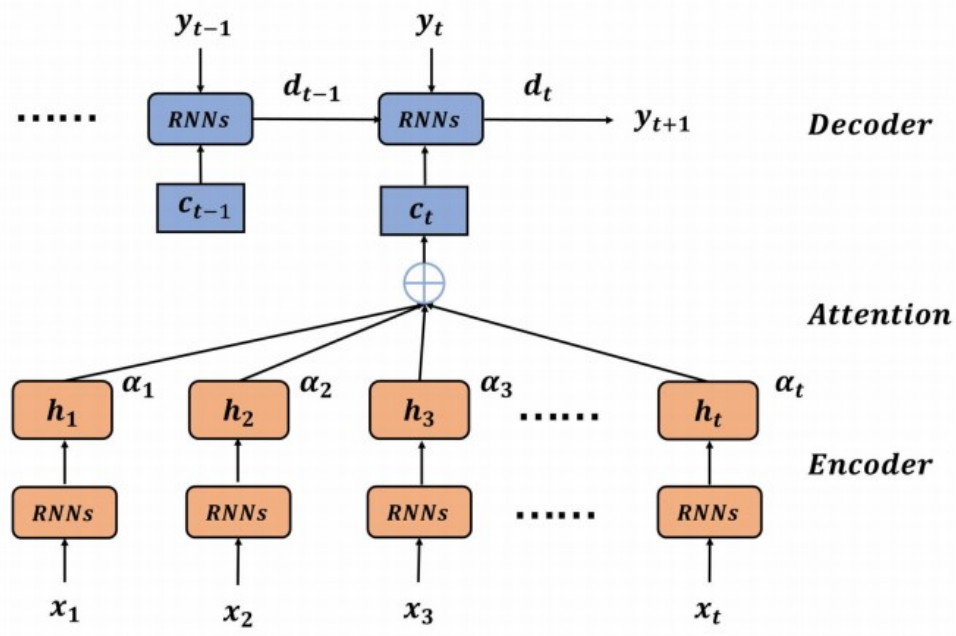


Figure 2.5: Architecture of RNNs with typical attention mechanism (Zhang et al., 2020)

2.1.2 Triad background

Triad forecasting is a matter of great interest for businesses, as this is an event that costs a significant amount of money, specially to those with higher number of infrastructural objects (banking, retail, telecommunications). Triads are the three half hour periods of peak power demand across the National Grid in a year (from November to February). These three points are used to calibrate the system costs, which are passed on to industry. The aim of the Triad system is to incentivise industry and users to help smooth out peaks in energy demand during the winter, especially in cold snaps (ELEXON, 2018).

According to Newbery (2011), the Triad charging system encourages demand reduction at these peak hours and hence signals the need for less generation and transmission. ELEXON provides a forecast for the UK electricity demand and energy managers, and businesses rely on this information (publicly available) to know when a Triad is going to happen. However, this information is incomplete and inaccurate as the demand values that ELEXON seeks to forecast are not the ones which Triad is calculated against. The model proposed in this work creates a better decision making framework because calling Triads implicates switching off equipment. Some companies cannot handle the disruption internally, as they need to run fuel generators, and this leads to considerable expenses.

TNuoS charges, which cover the costs of operating transmission networks, may represent around 5% of the bill. These fees are revised annually and forecast for 5 years ahead. The 2017 forecast published by National grid (Grid, 2018), shows the value of Triad growing from an average of £44 (≈57.36$) per kW to £59 (≈76.90$) per kW used during peak times. This forecast can be seen in Table 2.1.

The charge varies across 14 zones and is based on user's average half-hourly demand over three Triad periods taking place every winter season (ibid.). Because of economic interests for companies, most of the current Triad forecast systems are not publicly available.

In Marmaras et al. (2017) the electricity demand of each building on an actual Triad peak date and time was predicted successfully, and an overall forecasting accuracy of 97.6% was demonstrated for the considered buildings. Marmaras' model uses data from three different sources at various stages to predict the most probable half-hour of the day when the Triad could occur. These are data from National Grid, weather data and historical energy consumption; and its training set consist of historical data from 1990. This work, however, only validates the effectiveness of Triad forecasting using one year of data, not necessarily that the same model would work for longer periods and therefore, not offering a flexible framework when any changes (such as new policies) occur.

As changes happen very often in this field, ideally some parameters should be regulated. A single standalone system that works for every Triad season without having to do any modifications is difficult to develop. Algorithm validation is also not easy to

Table 2.1: Forecast of Triad fees (£/kW) forecast from season 2018/2019 to 2022/2023

Season / Region	**18/19**	**19/20**	**20/21**	**21/22**	**22/23**
Nothern Scotland	18.35	21.69	27.94	27.19	28.81
Southern Scotland	25.13	29.17	33.99	35.13	37.92
Nothern	36.92	41.50	44.31	48.63	51.20
Noth West	43.87	48.40	51.01	55.79	58.73
Yorkshire	43.83	48.47	51.15	56.25	59.31
Noth Wales & Mersey	45.43	49.97	52.66	57.27	60.70
East Midlands	47.39	52.26	55.59	60.83	64.07
Midlands	48.85	53.50	56.26	61.71	65.19
Eastern	49.37	54.33	57.91	63.17	66.51
South Wales	46.78	50.89	54.18	59.64	64.01
South East	52.52	57.11	60.34	65.51	68.72
London	54.84	60.21	63.92	69.30	72.89
Southern	53.80	58.55	61.67	66.74	69.94
South Western	53.86	57.30	60.08	63.71	67.73

because of the two reasons: data availability and constant changes in the patterns of the training data. This is why a certain degree of flexibility is proposed that the user can tune according to the degree of risk that can be afforded.

2.1.3 PPAs and renewable energy generation forecasting

Energy is the backbone of our economy, with a lot of emphasis being given to electrification under an agenda which is focused on climate change, sustainability and security of supply. Consumers, investors and politicians have influenced energy intensive organisations to find more efficient and sustainable ways to meet their electricity requirements.

According to Fulbright et al. (2016), more than 40% of Fortune 500 companies now have targets related to renewable energy procurement, energy efficiency or cutting Greenhouse Gas Emissions (GHG). One example is Google, whose target in 2010 was to achieve 100% renewable energy by 2017 for their global operations, including data centers and offices (Hölzle, 2016).

As defined in Jenkins et al. (1999), a Power Purchase Agreement (PPA) is a long-term contract between an Independent Power Producer (IPP) and an off-taker, usually an energy intensive organisations or a utility company. PPAs are seen as a hedging tool by many organisations, as they offer an opportunity for energy buyers to achieve price certainty and at the same time meet their sustainability objectives.

Mature renewable technologies were price competitive in 2020 and could offer prices for their intermittent output at all-time lows. The intermittency of renewable technologies is seen as one of the main challenges in renewable PPAs. Many corporates are reluctant to be exposed to this risk, despite the fact that it constitutes only a small fraction of the total value of a PPA.

In physically settled PPAs there is usually a Balancing Responsible Party that undertakes the balancing tasks, which are necessary to achieve effective hedging via PPAs. The balancing responsibilities should reflect three types of risks, which are related to variable generation patterns, (Hedges et al., 2019):

- **Balancing risk**: Risk associated to the exposure of power system costs that arise when an asset's forecast generation is different from its actual generation. This risk is related to the imbalance cost, therefore the more an asset contributes to the power system's imbalance, the higher the cost.
- **Shape or profile risk**: This risk is related to the variability of wind speed or solar irradiation, and is independent of the total volume generated by the asset.
- **Volume risk**: It captures the variable generation of an asset over a certain period of time. This can be related to deviations in the long-term, such as higher than expected wind speed or lower levels of irradiation due to abnormal weather conditions.

Risks can be mitigated depending on the structure of the PPA contract itself (Brindley, 2019), but also by achieving the best possible forecast for both wind and solar generation. A long-term scale forecasting horizon is considered, of at least one year, for wind and solar generation.

Wind generation forecasting has always been of interest for energy community, as estimating wind generation forecast error has an influence in sizing reserves and also in shape-balancing risk (Gil et al., 2010; Constantinescu et al., 2009; Lowery et al., 2012; Mauch et al., 2013). Wilczak et al. (2015) points out the importance of wind forecast, but their aim is to improve the accuracy of short-term wind forecast. In another example, Solari et al. (2012) uses geostrophic wind data to forecast wind speed for port safety, but the maximum horizon achieved is in the order of days. Similarly, Cheng et al. (2017) aims to forecast wind speed using anemometer data, but the forecast horizon is short-term. Also Lange et al. (2006) outlines several techniques for wind forecast, but it follows a similar forecast horizon. Many wind forecast horizons are oriented to achieve short-term accurate results (Huang et al., 2011; Akçay et al., 2017; Bossanyi, 1985), even though what is considered in this field as long-term does not extend to more than 120 hours, as is the case of Barbounis et al. (2006). A case of wind speed long-term forecast is found in Azad et al. (2014), where a series of Neural Network (NN) methodologies are used to forecast wind speed in a period of 6 months by using historic wind speed pattern data. Other predictions are short-term (Negnevitsky et al., 2006; Shi et al., 2013; Pinson et al., 2009; Juban et al., 2007). More recent studies such as Li et al. (2020a) shows a combination of SVM and an improved dragonfly algorithm that shows better prediction performance compared with back propagation NNs and Gaussian process. Another novel application on this is shown in Sun et al. (2020), which uses a combination of variational model decomposition and convolutional LSTM and shows better performance in comparison to similar variations including variational model decomposition, too.

The interest is in estimating electricity generation mid- and long-term, in particular, more within several intervals during the year than in a forecast for the next period. Matos et al. (2010) and Pinson et al. (2007) use probabilistic modeling and Markov Chain Monte Carlo (MCMC) to generate the wind samples, but they do so for a time horizon of maximum 50 hours. There are many studies based on Bayesian methodologies to model short-term wind forecast (Jiang et al., 2013; Bracale et al., 2015; McLean Sloughter et al., 2013), achieving very good results. As mentioned before, many wind models use meteorological data (wind speed, solar irradiation) to forecast generation, but the inconvenience of this is that the forecast goes as far as the meteorological model goes, and this is usually short-termIn more recent works, Chen et al. (2020) designs a novel medium-term wind power forecasting based on multi-resolution multi-learner ensemble adaptive model selection with good results, but the maximum forecasting horizon is only 18 hours. The scheme of this model can be seen below in Figure 2.6. This model first transforms the data into multi-resolution dataset, then 16 submodels are created from 4 base learners, which then uses adaptive model selection and finally support vector regression is used as a high level ensemble model with the selected sub-models from the previous part.

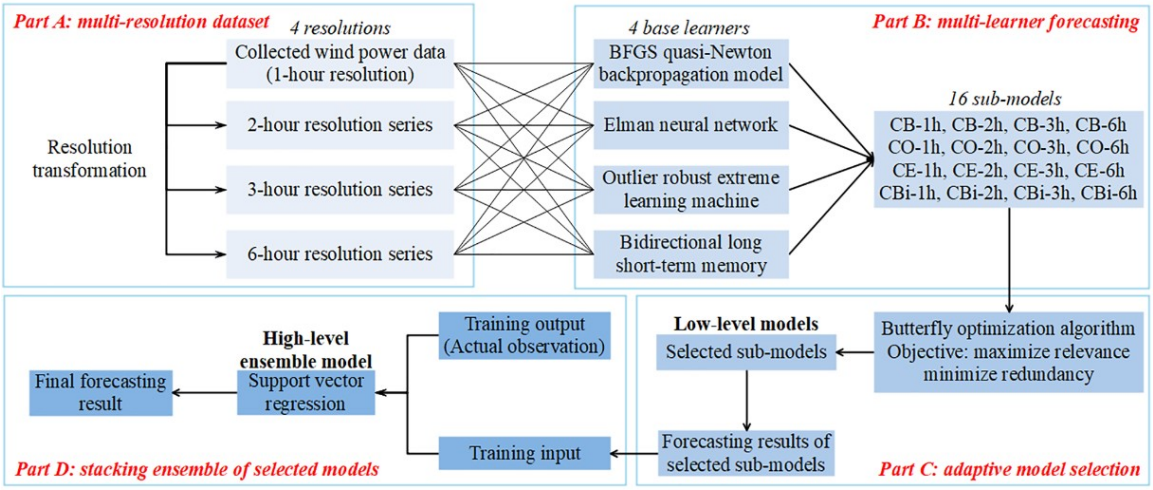


Figure 2.6: Framework of the model proposed by Chen et al. (2020)

Concerning solar forecast, the challenge is less difficult in comparison to wind forecast. This is due to the hours of daylight which are known. There are climatological factors that influence solar generation, such as cloudiness (Reikard, 2009; Heinemann et al., 2006; Sfetsos et al., 2000; Perez et al., 2010). One of the most popular approaches is NN-based models, such as Gensler et al. (2016), which uses autoencoders and Long Short-Term Memory (LSTM) NNs to perform the forecast, or using Artificial Neural Networks for this purpose, as in Chen et al. (2011) and Abuella et al. (2015). When looking at the forecasting period, similar to wind, most of the predictions are focused in the short term (Urquhart et al., 2015; Golestaneh et al., 2016; Bacher et al., 2009), therefore they cannot be used for the purpose of PPA forecast. Even more recent models create more sophisticated methods that achieve very good results, they still do not reach the forecasting horizon that is needed for this problem. It also brings the work of Abdel-Nasser et al. (2019), which uses LSTM-RNN to produce the forecasts and achieves a good result. Another interesting model that uses a similar approach for day-ahead solar forecasting is Li et al. (2019), which outperforms other compared models such as LSTM, SVM, RBF and Back Propagation Through Time (BPTT). The scheme of this model is shown in Figure 2.7

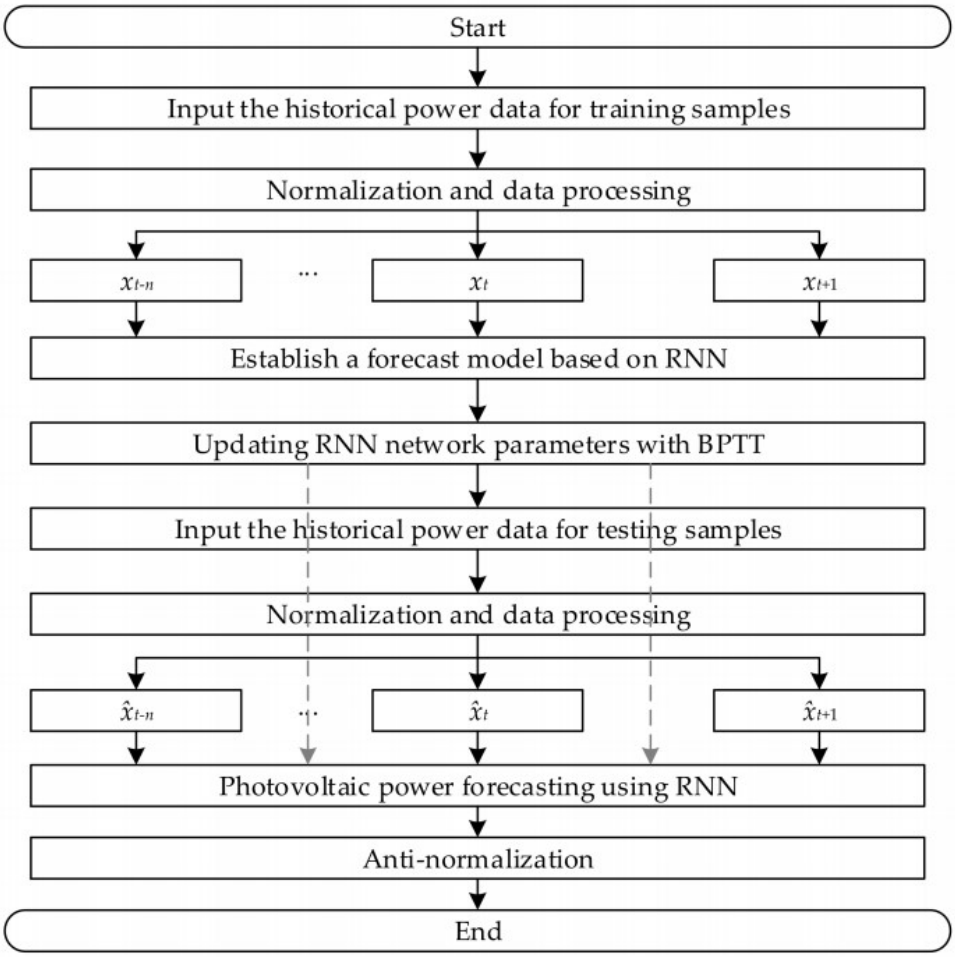


Figure 2.7: Flowchart for PV generation power forecasting (Li et al., 2019)

The goal of this work is to produce long-term probabilistic forecasts of wind and solar energy generation, with time horizon of one year, taking into account the above three types of risks in order to find an optimal match of the forecasts with respect to the target consumption profile.

2.2 AI applications in BMSs

2.2.1 HVAC systems and time series classification

Building Management Systems (BMS) present clear advantages for energy control such as identifying locations of potential energy waste for energy optimisation, decreasing equipment operating cost, providing indoor environmental safety and comfort through Heating, Ventilation, and Air Conditioning (HVAC) systems control, as well as controls of water consumption, elevators, etc. Over the past few years, a lot of efforts have been put to control the three main important aspects of the building: energy consumption, security and comfort. The schematic in Figure 2.8 represents the a high level overview of the built environment and role of BMS control signals and information flow, which also takes into account the supply of energy and data from the grid for demand response events. This figure shows the main building blocks of BMSs, in relation to the whole smart building spectrum:

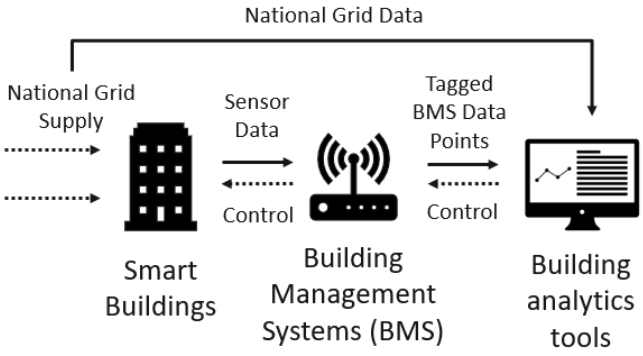


Figure 2.8: Schematic of BMS

The literature has been reviewed in two main parts in this section: the first one is dedicated to supervised and unsupervised HVAC equipment linkage for fault detection and diagnosis, as the main purpose of linking HVAC equipment is to trace back system failures and to detect system anomalies. The second one to time series clustering, as these are the kind of methodologies used to solve this problem.

When mobilising a set of building sensors into an analytics platform, BMS points are translated into a naming standard such as Haystack (Quan et al., 2003) or a more unified metadata schema such as Brick (Balaji et al., 2016), so that the analytics platform can recognise them, and sensor points can be programmed into rules for energy consumption, systems linkage, etc, which requires a metadata framework to form a link between point types, physical spaces and the linkage of HVAC components. One of these methods is the framework for metadata normalisation Plaster in Koh et al. (2018), which requires a certain level of human supervision such as knowing the point type, location and relationship with other equipment parts, specially in large facilities. A methodology that works out relationships between HVAC points based on sensor data may help to reduce human interaction when building this framework. However, it may happen that the company or individuals who want to perform this either do not have the resources or skill set to translate BMS points into such standards. BMS systems transfer sensor data to analytics platforms to be used for building optimisation.

Ventilation is one of the major areas of electricity consumption. In large industrial facilities, Air Handling Units (AHUs) are key consumption points. As represented in Figure 2.10, several units are also involved in AHUs, such as electric fans, humidifiers (in some AHUs), heating and cooling, which interact with other systems, such as boilers, cooling systems, etc. Therefore, controlling AHU's parameters means to control a significant part of electricity consumption, as this is a point where other systems converge. The goal of this work is to infer relationships between AHUs and both building areas and other HVAC parts for large manufacturing facilities using only time series data from the installed BMS sensors.

Previous studies have presented novel methodologies to infer relationships between HVAC components of large commercial buildings such as Pritoni et al. (2015), that utilises perturbations of subsystem variables to reveal correct associations with a 76% success. The authors state that statistical methods are not good for this purpose, however they only use correlations between variables and they don't test more complex statistical methodologies. Koh et al. (2016) uses a series of supervised learning methodologies to infer point type from sensor data, as well as to control perturbations to verify relationships between HVAC system parts. Gao et al. (2018) compares different supervised learning

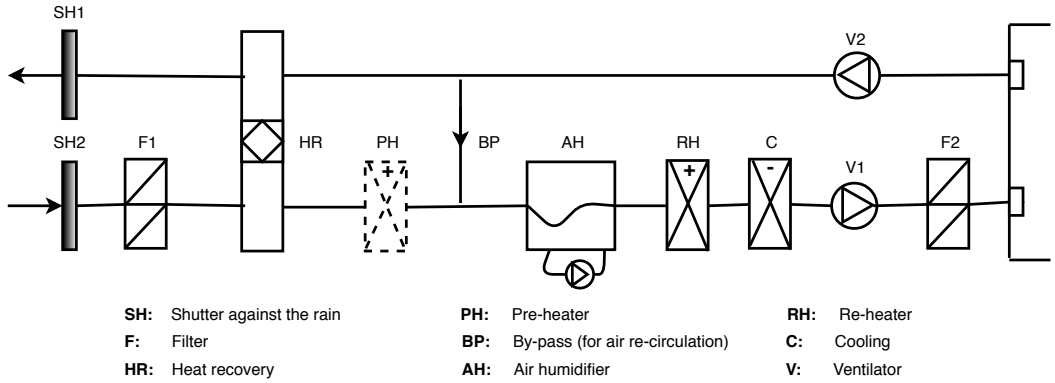


Figure 2.10: Schematic of AHU

methodologies, inferring equipment characteristics from time series features. The case study presented in our work proposes a system with old infrastructure where neither documentation nor prior knowledge is available, therefore a supervised learning classification approach would not be feasible. Inputs and outputs are known but, in order to verify such outputs, verification from experienced engineers has been necessary in order to compensate the lack of documentation, which has been a work that lasted several weeks. Similarly, Park et al. (2018) use supervised classification to infer AHU-VAV links by first extracting statistical features from the data and then applying random forests for each VAV. This is shown below in Figure 2.9, where the highest value of cross-correlation is used to determine the relationship between VAV damper position and supply air duct pressure for each AHU to infer the correct relationships. A study offering relationship between equipment parts according to physical spaces and with minimal intervention is presented in Li et al. (2017), which converts time series into frequency domain with short-time Fourier transformation operator, that contains implicit information about changes in sensor patterns. Then it wraps them in time dimension by using dynamic time warping and a predefined time wrapping function.

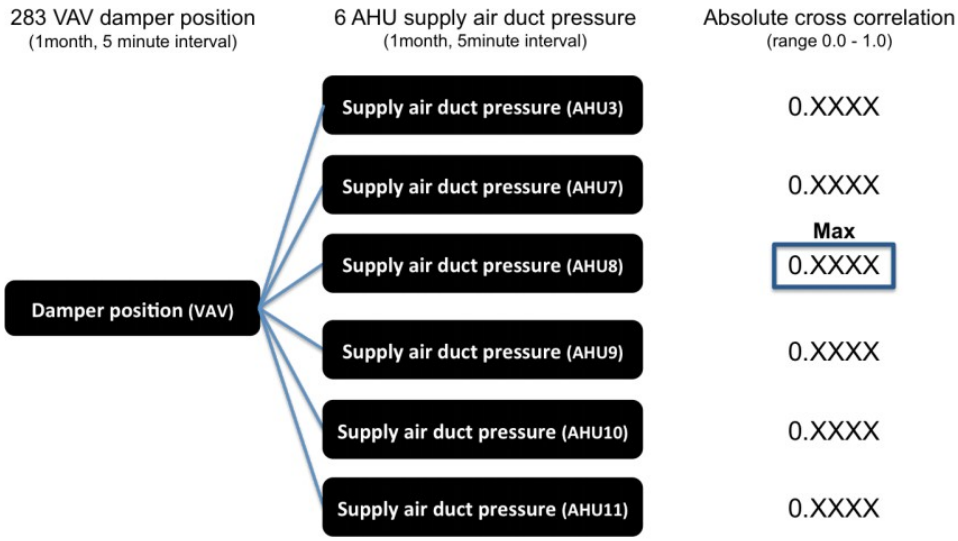


Figure 2.9: Architecture of RNNs with typical attention mechanism by Park et al. (2018)

The methodology applied in this study could be useful for diagnosis of system errors, which are defined after establishing normal working conditions of the system according

to the clusters. When there is a significant number of AHUs and the temperature gets outside a comfort policy, the origin of this failure can be very difficult to trace back. For AHU fault diagnosis some previous work has been reported in the literature. Lee et al. (1996) describes the application of Artificial Neural Networks (ANNs) to the problem of fault diagnosis in an AHU by using residuals of system variables to quantify the dominant symptoms of fault modes of operation. Following the same approach, Lee et al. (2004) proposed AHU subsystem level fault detection using a General Regression Neural-Network (GRNN), residual generation and fault detection and diagnosis. A novel feature extraction technique to extract temperature and power associated features from high-dimensional and unstructured terminal unit data is presented in Dey et al. (2020b), to diagnose faulty HVAC in an automatic and remote manner. Air handling unit Performance Assessment Rules (APAR) was used by Schein et al. (2006). They utilised control signals to determine the mode of operation of the AHU. A subset of expert rules which correspond to that mode is then evaluated to determine whether a fault exists. In the review of fault detection and diagnosis methodologies carried by (Yu et al., 2014), various Fault Detection and Diagnosis (FDD) techniques are described to illustrate the use of evaluation standard parameters for improving the performance of AHUs. They divide FDDs into three main categories, namely analytical-based methods, knowledge-based methods, and data-driven methods. In a more recent study, Dey et al. (2020a) proposes a method that employs sequential two-state clustering to identify abnormal behaviour of the fan coil unit. Some other recent studies on HVAC systems fault detection and diagnosis can be seen in Dey et al. (2020b), Stopps et al. (2019) and Yang et al. (2020). The last one presents very interesting findings about a model predictive control system with an adaptive machine-learning-based building model for building automation and control applications, achieving 58.5% reduction of cooling energy in the office and 36.7% reduction of the electricity consumption of the air-conditioning system. The schematic of the ANN with the dynamic recurrent nonlinear autoregressive exogenous (NARX) architecture as a part of this model, which adds delayed weights as inputs, is shown in Figure 2.11.

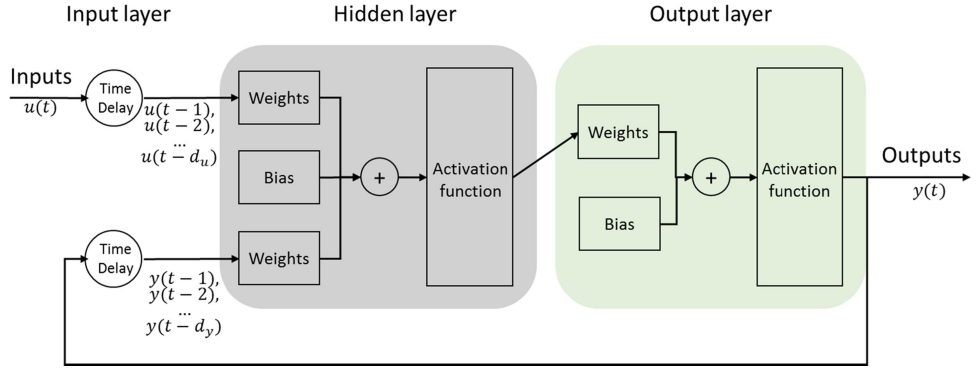


Figure 2.11: A schematic diagram of the NARX ANN for modeling dynamic systems by Yang et al. (2020)

The methodologies for detecting failures of AHUs have the specificity of using either control signals or the internal parameters of the AHU itself. In large facilities, linking different physical spaces with their correspondent control systems is needed to detect the sources of deviation from the prescribed conditions. There is growing demand to understand and extract value from sensor data, especially in large spaces where the amount of AHUs and of time series data provided by different sensors can create confusion when looking for links between different equipment units. So the real problem to solve is to

add clarity to equipment-spaces linkage and therefore, to create real value from sensor data.

This sensor data linkage is done by studying similarities between the time series data, and by clustering them based on these similarities. Aghabozorgi et al. (2015) states that finding the clusters of time series can be advantageous in different domains for anomaly and discord detection, recognising dynamic changes in time series, prediction and recommendation, and pattern discovery. The problem of this study fits into the pattern detection category, as the aim is to detect similarities between time series to identify links between assets.

One of the widely used metrics for time series similarities is Dynamic Time Warping (DTW). One of the pioneer works, Berndt et al. (1994) describes experiments with this dynamic programming approach to the problem of pattern detection. Rakthanmanon et al. (2012) demonstrates that DTW could be used for mining massive data sets faster than with Euclidean distance. Jeong et al. (2011) designs an approach that penalizes points with higher difference between a reference point and a testing point in order to prevent incorrect distance caused by outliers. This methodology is explained more in detail in the methodology section, Eq. 3.23.

Some other distance metrics have proven successful for pattern detection in time series data. Using integrated periodogram distance, Caiado et al. (2006) presents a metric based on different dependency measures to classify time series as stationary or non-stationary. Simulation results proved that the logarithm of the normalized periodogram and the metric based on the autocorrelation coefficients can distinguish ARMA and ARIMA models, which does not happen with the classical Euclidean distance. Lasso-based approaches are also widely used for time series grouping. As an example, Chan et al. (2014) proposed a two-step lasso procedure for multiple change-point estimation in time series. Tucker et al. (2001) decomposes high dimension Multivariate Time Series (MTS) into smaller dimension MTS which are relatively independent of one another, based on correlation between the variables. This methodology is explained more in detail in the methodology section, Eq. 3.27.

In recent years, several studies have been done in the field of time series clustering, such as Bode et al. (2019), which presents a time series clustering approach for building automation and control systems. This work uses unsupervised machine learning algorithms to improve supervised classification by adding more robust features compared to manual selection. Bandara et al. (2020) compares a set of recurrent neural networks on groups of similar time series for clustering, showing that long-short-term memory neural networks present a good result for this purpose as well. Other recent works on time series clustering methodologies and applications can be seen in Alonso et al. (2019) and Khiali et al. (2019).

2.2.2 Early Warning Signals analysis

The main problem in the state-of-the-art BMSs is that they tend to use a reactive management approach instead of a proactive one. The system failure is detected once it has happened, thus causing a disruption in services and forcing engineers to temporarily shut down some of the equipment in order to fix the failure. This work proposes a proactive approach to early detection of BMS failures, aiming at both Early Warning Signals (EWS) and forecast of time series sensor data.

There are several studies of EWS in dynamical systems such as climatic variables that can be applied to our case study due to its nature as dynamical system. An example is Prettyman et al. (2018), which uses lag-1 autocorrelation (ACF1), the Detrended Fluctuation Analysis (DFA) exponent and Power Spectrum (PS) in tropical cyclon data. The comparison of the different indicators for this study can be seen in Figure 2.12. For the multivariate tropical cyclone data case, Prettyman et al. (2019) uses Empirical Orthogonal Functions (EOF) for dimensionality reduction prior to applying ACF1, DFA and PS. They also study the possibility of using the Jacobian matrix eigenvalues of the system as tipping point indicator. In the same area of application, (Lenton et al., 2012; Livina et al., 2007) use ACF and DFA to detect climate tipping points. In (Livina et al., 2010) a novel statistical method is applied, which is the method of potentials Livina et al. (2013), for EWS to analyse the changing number of climate states during the last 60 kyr. The method detects the changes between states by estimating the probability density of the recorded time series. In the predictive maintenance field, Livina et al. (2019) uses DFA and ACF to detect anomalies in electronic components commonly used in applications of the automotive and aviation industries.

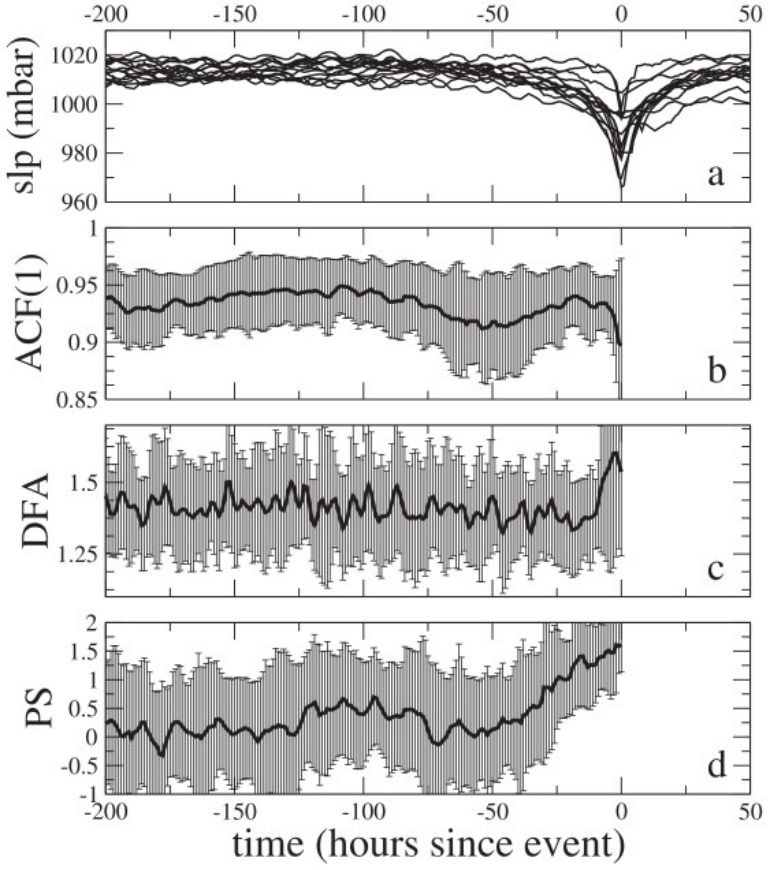


Figure 2.12: Comparison of ACF(1), DFA and PS indicators applied to sea-level pressure data (Prettyman et al., 2018)

Previous publications prove that EWS have a wide range of applications other than climate pattern change analysis. Also, early detection of failures for critical system components in general as explained in Qiu et al. (2015), which uses deep belief NNs for feature extraction and hidden state analysis of hidden Markov model to estimate the operating status of a compressor unit. Another popular area of application is economics, such as the analysis of banking system collapse. Squartini et al. (2013) studies early warning

signals of topological collapse in interbank networks. An earlier work in the similar field, Davis et al. (2008) compares early warning systems for banking crises, right after the financial crises of 2007. Another popular area of application of EWS is credit risk diagnosis, for example, Lu et al. (2013) uses a different and interesting approach. They apply computational linguistic text mining to extract and quantify relevant Chinese financial news to develop EWS for financial distress. Also, there are several applications in biology, with a wide variety of applications within the field such as Rogers et al. (2018), which aims to detect EWS of tree mortality in boreal North America using multi-scale satellite data, using the normalised difference vegetation index.

Variational Autoencoders (VAE) are becoming increasingly popular for failure detection. In Andrews et al. (2016) they use an autoencoder residual vector error magnitude. This method is tested on several images datasets, concluding that it is a valid methodology for failure selection, as the hidden layer representation is capable of characterising the fundamental attributes of of the system within normal conditions, therefore to measure the deviation from "normal functioning". A scheme for the VAE is shown in Figure 2.13. Following a similar method in a manufacturing field, Wen et al. (2018) use VAE for degradation assessment of the ball screw. The assessment is done using the Variational Autoencoder Reconstruction Error (VAERE) and it demonstrates the progressive degradation of this component. The scheme of this approach can be seen in Figure 2.14, where HI stands for health indicator.

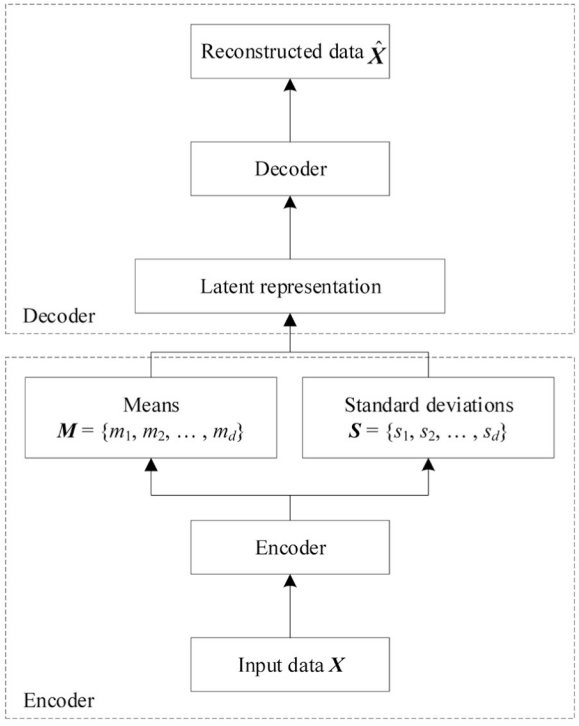


Figure 2.13: VAE scheme by Wen et al. (2018)

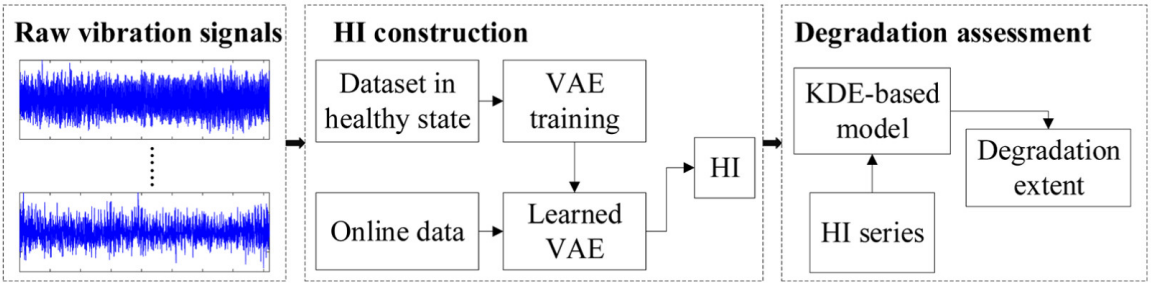


Figure 2.14: Framework for the degradation assessment method for the ball screw (Wen et al., 2018)

2.2.3 Automatic text classification

Text classification

The purpose of our work is to apply several machine learning methods for text classification in the context of Building Management Systems (BMSs). The mobilisation of a site implies the translation of different elements that are used in analysis platforms with the purposes of detecting failures of internal systems (heating, cooling), along with controlling areas of major electricity consumption and potential savings.

Most medium-to-large buildings have installed BMS which can provide valuable data to any IoT implementation. This data includes the operational states of existing equipment in the building and occupancy comfort parameters for the installed sensors. Automatic text classification is usually done by extracting features from the text document. This is a semi-supervised machine learning problem since the classes are pre-defined. In this work, a generic strategy for text classification is followed as defined in Dalal et al. (2011), which has the following steps: training set of text documents, pre-processing, feature extraction, machine learning model selection, train classifier and test classifier. The scheme of these steps can be seen below in Figure 2.15.

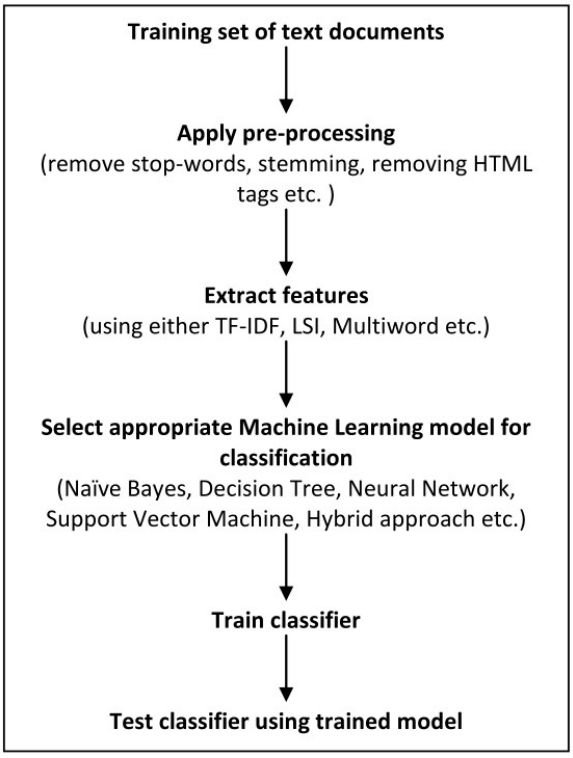


Figure 2.15: Generic strategy for text classification by Dalal et al. (2011)

Bayesian methodology is widely applied in text classification. Singh et al. (2019) compares models for Naive Bayes (NB) text classification, and finds that multi-variate Bernoulli model performs well with small vocabulary sizes, while the multinomial model performs better at larger vocabulary sizes. Liu et al. (2002) combines the Expectation Maximisation (EM) algorithm with the NB classification method using only partial information, one class of labeled documents and a set of mixed documents, showing extremely accurate results under certain class restrictions. In Chai et al. (2002), Bayesian online perceptron and Gaussian processes have been implemented and tested, showing that their performance is comparable to that of Support Vector Machines (SVMs). Liu et al. (2019) uses parallel naive Bayes algorithm for large-scale Chinese text classification. More recent examples for this algorithm for text classification can be seen, for example in Ranjitha et al. (2020), where Naive Bayes is used for text classification, which lead to an advantage in terms of characteristic dialect processing. Another recent work can be found in Le et al. (2019), where this technique is also combined with sentiment lexicon. The effectiveness of this technique is further enhanced with the use of a dictionary as an input source and a document preparation process which improves the accuracy to 98.2%.

Decision trees also play an important role in text classification. Harrag et al. (2009) use decision trees applied to text categorisation and classification. Random forest is an ensemble learning version of decision trees, as it constructs a multitude of decision trees at training time and outputs the class that is made of every of the contained classes. In terms of performance, Ali et al. (2012) shows that the random forest gives better results than dcision trees for the same number of attributes in large medical datasets. With respect to text classification, Akinyelu et al. (2014) uses random forests for content-based phishing detection, which yields a very high classification accuracy. Xu et al. (2012) presents an improved random forest algorithm by simultaneously employing a new feature weighting method and the tree selection methods to categorise text documents. As a result, the algorithm can effectively reduce the upper bound of the generalisation error and improve

classification performance.

SVMs are very popular for text classification. Joachims (1998) and Joachims (2001) show that SVMs are appropriate for this task, and it outperforms other algorithms. Also Tong et al. (2001) introduce an algorithm for performing active learning with SVMs, i.e. an algorithm for choosing which instances of data to request next for the training stage. Alsaleem (2011) shows that SVMs outperforms NB. In Sun et al. (2009), a comparative study on the strategies addressing imbalanced text classification using SVM classifiers is described. They evaluated 10 methods on 3 benchmark datasets using area under the PR-curve as the performance metric, finding that the standard SVM learnt the best decision surface in most test cases. In more recent studies on SVM for text classification, Gopi et al. (2020) classifies tweets data based on polarity using improved RBF kernel on SVM, which outperforms other SVM-RBF classifier and models. Kurnia et al. (2020) classifies user comments using *word2vec* embedding and SVM classifier. They classify comments from social media about mobile networks applications, achieving a 79.5% accuracy. Another recent methodology on SVM for text classification can be seen in Zhang et al. (2019), where text is represented mathematically by vector space model, and the classifier is trained to classify the text based on the principle of SVM. The framework of the SVM for this classification system can be seen in Figure 2.16. Another interesting work can be seen in Wang et al. (2019), combines Char Convolutional Neural Networks with SVM, to obtain the emotional tendencies of users reviews. Chatterjee et al. (2019) uses multi-class classification using SVM and one-vs-rest, which divides a multi-class classification problem into one binary classification per class. On top of that, this is enhanced by using multi-threading and CUDA.

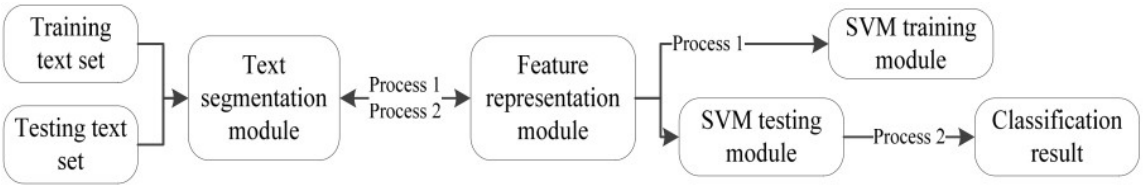


Figure 2.16: Framework of SVM Chinese text classification system by Zhang et al. (2019)

Logistic regression also provides good results in text classification. The study of Genkin et al. (2007) uses lasso logistic regression, which provides state-of-the-art text categorisation while producing sparse and thus efficient models. In the same way, Ifrim et al. (2008) present a coordinate-wise gradient ascent technique for learning logistic regression in the space of all *n-gram* sequences (contiguous sequence of n items from a given sample of text or speech) in the training data. They use several datasets, interestingly including a Chinese language dataset among them. A modified logistic regression for positive and unlabeled learning is applied by Jaskie et al. (2019), who introduce a new modified logistic regression with a variable upper bound that provides a better theoretical solution for the proposed problem. A comparison between Bayes classification and logistic regression is studied in tweets categorisation in Prabhat et al. (2017), showing after training that logistic regression gives a 10.1% more accurate and 4.34% more precise than the Bayes algorithm. Logistic regression and its variations are popular for sentiment analysis, as can be seen in recent studies such as Ramadhan et al. (2017) which studies tweets sentiment analysis by extracting the features first, then transforming the list of features into binary form and transformed again used Tf-idf method before being classified using logistic regression. This is very relevant for our work, as tweets have a character limitation, so this would prove that logistic regression is suitable for clas-

sification using text that is shorter than usual. Following a similar line of work, Rane et al. (2018) compares several methodologies, including logistic regression, SVM, Naive Bayes, AdaBoost, among others, for sentiment classification of Tweeter data for an US airline service analysis. Results show that logistic regression shows a good score with a F-measure of 81.9%, but it is outperformed by random forests in the first place, with a 86.5% F-measure. This creates a solid base for the BMS text classification case explored in this Thesis, as several of these methods are used in this work for short text classification. Also, for another work of tweet sentiment analysis, Hasanli et al. (2019) compared logistic regression, Naive Bayes and SVM to detect sentiment polarity. Logistic regression and SVM show a better performance if bag-of-words is used for the pre-processing, and Naive Bayes performs better if term frequency - inverse document frequency is used.

Deep learning has been increasingly gaining popularity and the literature provides examples of using some of these methods for text classification as well. Liu et al. (2017b) presents the first attempt at applying deep learning to extreme multi-label text classification (XMTC), with a family of new Convolutional Neural Network (CNN) models which are tailored for multi-label classification. Zhang et al. (2015) constructed several large-scale datasets to show that character-level convolutional networks could achieve state-of-the-art or competitive results for text classification. In Lai et al. (2015) they introduce a recurrent convolutional neural network for text classification without human-designed features, showing that the proposed method outperforms the state-of-the-art methods on several datasets, particularly on document-level datasets. More recent works on text classification can be seen for example in Elnagar et al. (2020). Here, the authors perform Arabic text classification using deep learning models. Results show that attention-Gated Recurrent Units (GRUs) achieves a top performance of 96.94% by using the dataset NADiA, which is the largest dataset of Arabic documents. Another example is given by Yao et al. (2019), where a novel text classification method termed text graph convolutional networks is used with the purpose of text classification. Results of this work shows that this promising methodology outperforms other state-of-the-art deep NNs such as CNN, LSTM and others that are not NNs, such as logistic regression. The algorithms are run in various datasets, such as news or movie reviews, with a large corpus. Gargiulo et al. (2019) also uses DNNs for hierarchical extreme multi-label text classification. They describe a methodology named Hierarchical Label Set Expansion (HLSE) used to regularize the data labels, evaluating the methodologies on the PubMed scientific articles collection, proving the usefulness of the proposed HLSE methodology. The graphical representation of the DNN model used for this work is shown in Figure 2.17. Deep learning methodologies proved unmistakably useful for text classification and they should be considered, as the literature suggests, for large corpus text where a lot of features need to be processed to perform the classification. For the purpose of this problem, whose length of the text is the real challenge (11 words maximum per label), DNNs are not considered. Although they can be considered for further research.

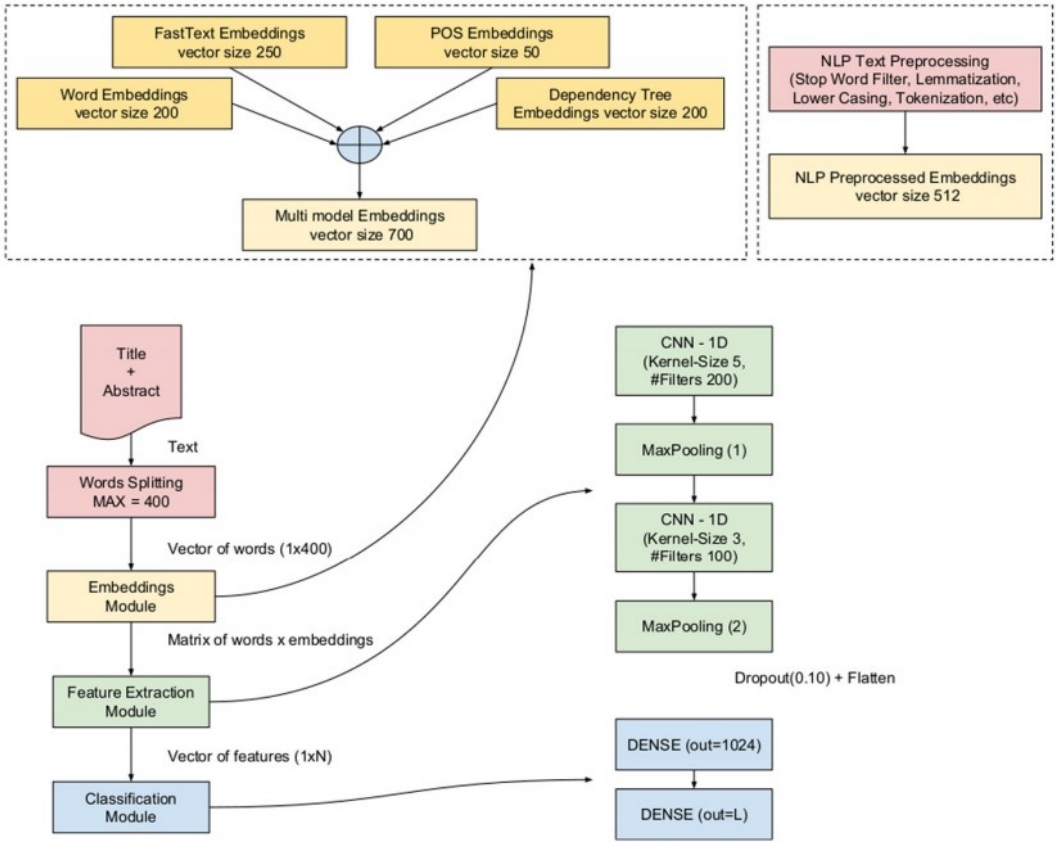


Figure 2.17: Graphical representation of the DNN models used b Gargiulo et al. (2019)

Text processing and categorisation

In recent years, there has been a lot of progress in natural language modeling and representation. Natural Language Processing (NLP) is of major interest in research as it represents the core business of Internet companies today. Goodman (2001) defines language modeling as the art of determining the probability of a sequence of words and introduces the n-grams. Assuming that similar words appear in similar contexts, Brown et al. (1992) used counts of classes, which leads to generalisation, therefore better performance on novel data. Bag-of-words model is one of the most popular representation methods, whose statistical framework is explained in Zhang et al. (2010). It consists of the sum of one-hot codes, ignoring the order of the words, but it can be extended to bag-of-Ngrams to capture local ordering of words. *Term Frequency - Inverse Document Frequency* (TF-IDF) is another common technique that evaluates how important a word is to a document in a collection of corpus proportionally to the number of times it appears in the document.

In a more advanced version of text modeling there are word vectors, also known as embedding. Each word is represented by a real valued vector in N-dimensional space (usually $N = 50 - 1000$). These representations manage to capture many degrees of textual similarity. Mikolov et al. (2013b) shown that word vectors capture many linguistic properties (gender, tense, plurality, even semantic concepts). Mikolov et al. (2013c) presents two novel architectures for computing continuous vector representations of words, and they measure the quality of these representations in a word similarity task. This work introduces Word2Vec, which uses a NN model to learn associations from a large corpus of text. The representation of this architecture is shown in Figure 2.18. In this representation, the CBOW architecture predicts the current word based on the context,

and the Skip-gram predicts surrounding words given the current word. Mikolov et al. (2013a) explores the previously mentioned similarities in languages for translation, reaching above 90% accuracy for the most confident translations. Although for the purpose of this problem such a deep language representation is not needed, this progress in NLP may be useful for future BMS automatic tagging improvements. In a more recent study, Miaschi et al. (2020) studied the linguistic knowledge implicitly encoded in the internal representations of BERT, a contextual language model (Devlin et al., 2019), in comparison to a contextual-independent one (Word2Vec). The findings reveal that contextual-independent model, the sum works best for obtaining sentence representations and for the contextual-dependent one, the mean works best.

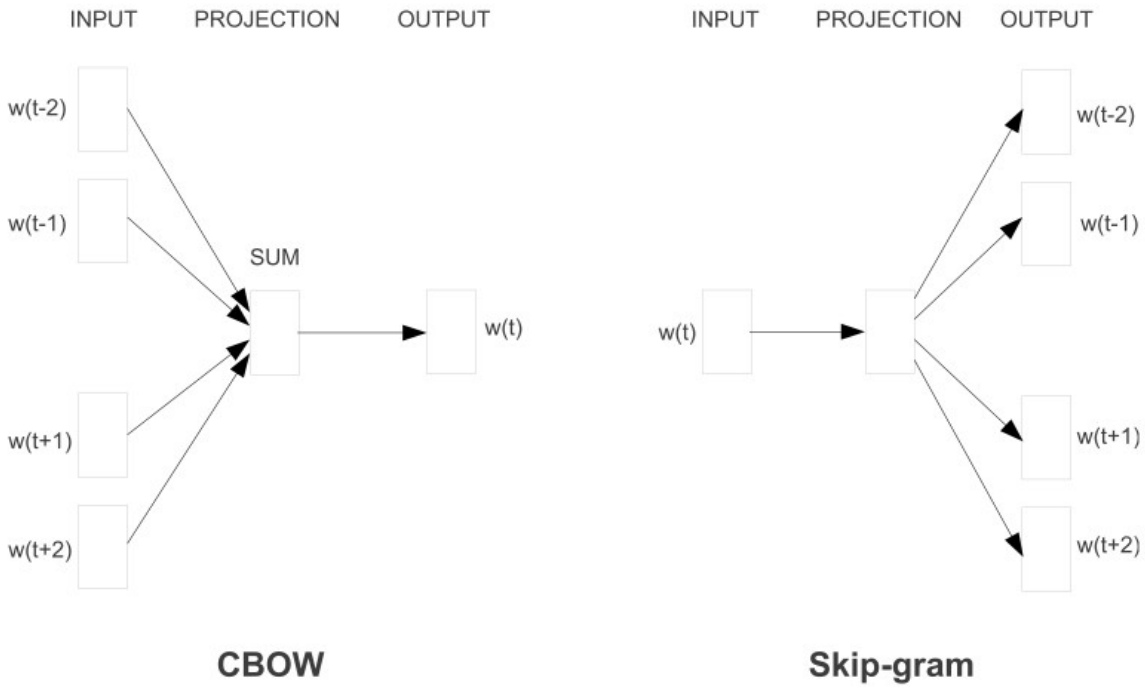


Figure 2.18: Model representation for vector representation by Mikolov et al. (2013c)

For the part of text classification for automatic BMS tagging, text classification to create a system that tags BMS sensor data automatically is performed. Several methods for text classification are compared, by following the generic strategy of Dalal et al. (2011) for solving these type of problems. Bag-of-words model for feature extraction is applied prior to the classification. In section 3.5, the methods to be used for text classification are introduced.

Chapter 3

Methodology

In this chapter, several methods are described that can be divided in three main categories: those related to time series forecasting and estimation, clustering or time series similarity measure methods; machine learning methods for text classification; and early warning signal methods.

3.1 Time series modeling and forecasting

In section 3.1.1, a background for LSTM neural networks is given. In Section 4, the performance of this model has been compared with ANN, SVM, random forests, Bayesian regression and the mean-only model. The reason is that these models proved to perform well for previous works in load forecasting and therefore, they are compared with LSTM. Another reason is that LSTM is a methodology that has proven give a good performance for time series forecasting, due to its proven capability of capturing both long and short-term seasonality, such as yearly and weekly patterns, which are found in the national energy demand data of this study. The basics for the other models are explained in different sub-sections within Section 3, as they are used for other challenges in this work.

The coming parts of Section 3.1 describe the methodologies used for wind and solar power forecasting.

3.1.1 LSTM and exponential moving averages

Long-Short Term Memory (Hochreiter et al., 1997) has proven to be a useful method for time series analysis of records with several factors correlated with the output. This method can provide a good working system for the UK national electricity demand forecast, and an effective way to ensure that the system addresses correlations in that data.

LSTM cells, which are LSTM units comparable to neurons in ANNs, manage two state vectors and for performance reasons they are separate (ibid.). The scheme of a single cell is illustrated in Figure 3.1:

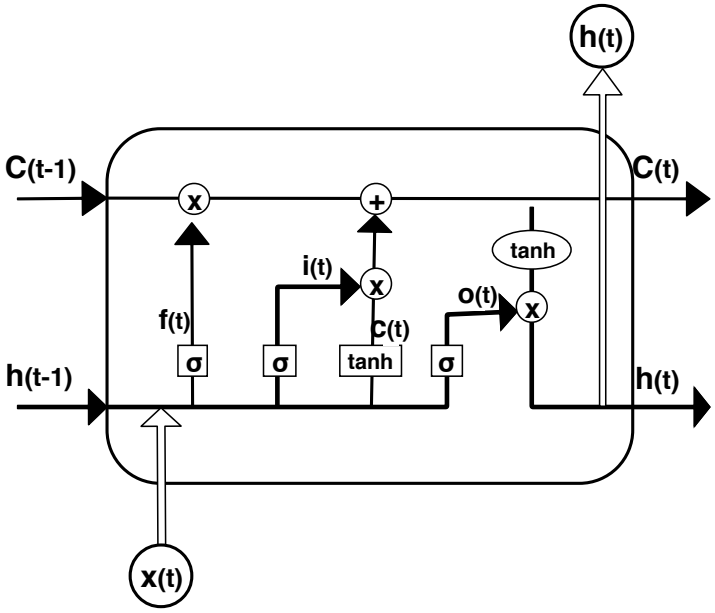


Figure 3.1: LSTM schematic (Hochreiter et al., 1997).

The state of the cell is represented by two vectors: $\mathbf{h}_{(t)}$ and $\mathbf{c}_{(t)}$. Vector $\mathbf{h}_{(t)}$ can be interpreted as the short-term state and $\mathbf{c}_{(t)}$ as the long-term state.

The current input vector $\mathbf{x}_{(t)}$ and the previous short-term state $\mathbf{h}_{(t-1)}$ are fed to four different fully connected gates. They serve different purposes:

- The main gate is the one that outputs $\mathbf{c}_{(t)}$. It has the usual role of analysing the current inputs $\mathbf{x}_{(t)}$ and the previous short-term state $\mathbf{h}_{(t-1)}$.
- The *forget gate* (controlled by $\mathbf{f}_{(t)}$) controls which part of the long-term state should be erased.
- The *input gate* (controlled by $\mathbf{i}_{(t)}$) controls which parts of $\mathbf{c}_{(t)}$ should be added to the long-term state.
- The *output gate* (controlled by $\mathbf{o}_{(t)}$) controls which part of the long-term state should be read and output at this time step (both to $\mathbf{h}_{(t)}$) and $\mathbf{c}_{(t)}$.

Function σ represents the logistic transformation after a fully connected NN set. The key idea is that the network can learn what to store in the long-term state, what to throw away, and what to read from it. The long-term state transverses the network from left to right, it goes through a *forget gate*, dropping some memories, and it adds new memories through the addition operation. After that it is copied and processed through the *tanh* function, whose result is filtered by the *output gate*. This produces the short-term state $\mathbf{h}_{(t)}$.

Equations 3.1 - 3.6 summarise how to compute the cell long-term state, and its output at each time step for a single instance (Geron, 2017):

$$\mathbf{i}_{(t)} = \sigma(\mathbf{W}_{xi} \cdot \mathbf{x}_t + \mathbf{W}_{hi} \cdot \mathbf{h}_{t-1} + \mathbf{b}_i), \tag{3.1}$$

$$\mathbf{f}_{(t)} = \sigma(\mathbf{W}_{xf} \cdot \mathbf{x}_t + \mathbf{W}_{hf} \cdot \mathbf{h}_{t-1} + \mathbf{b}_f), \tag{3.2}$$

$$\mathbf{o}_{(t)} = \sigma(\mathbf{W}_{xo} \cdot \mathbf{x}_t + \mathbf{W}_{ho} \cdot \mathbf{h}_{t-1} + \mathbf{b}_o), \tag{3.3}$$

$$\mathbf{c}_{(t)} = \mathbf{g}_{(t)} = \tanh(\mathbf{W}_{xg} \cdot \mathbf{x}_t + \mathbf{W}_{hg} \cdot \mathbf{h}_{t-1} + \mathbf{b}_g), \tag{3.4}$$

$$\mathbf{c}_{(t)} = \mathbf{f}_{(t)} \otimes \mathbf{c}_{(t-1)} + \mathbf{i}_{(t)} \otimes \mathbf{g}_{(t)}, \tag{3.5}$$

$$\mathbf{y}_{(t)} = \mathbf{h}_{(t)} = \mathbf{o}_{(t)} \otimes \tanh(\mathbf{c}_{(t)}), \tag{3.6}$$

where:

- $\mathbf{W}_{xi}$, $\mathbf{W}_{xf}$, $\mathbf{W}_{xo}$, $\mathbf{W}_{xg}$ are the weight matrices of each of the four gates for their connection to the input vector $\mathbf{x}_t$.
- $\mathbf{W}_{hi}$, $\mathbf{W}_{hf}$, $\mathbf{W}_{ho}$ and $\mathbf{W}_{hg}$ are the weight matrices of each of the four gates for their connection to the previous shot-term state $\mathbf{h}_{t-1}$.
- $\mathbf{b}_i$, $\mathbf{b}_f$, $\mathbf{b}_o$ and $\mathbf{b}_g$ are the bias terms for each of the four gates.
- $\otimes$ represents element-wise vector multiplication.

In order to achieve a more accurate result, parameter tuning is performed. These parameters are the number of years of data used for training, the number of cells and the number of epochs. This will also be discussed later on in section 4.1.

Exponential Moving Average

Exponential Moving Average (EMA) is a modified version of the simple Moving Average (MA), i.e. a type of moving average with more weight given to the latest data. The EMA works as a classifier in this case, generating binary signals, 1 when the peak is over the EMA, 0 when it is below it. The EMA is defined as follows:

$$S_t = \alpha y_{t-1} + (1 - \alpha) S_{t-1}, \tag{3.7}$$

where:

- S_t: value of the EMA for $t = now$.
- α: smoothing constant. When α is close to 1, dampening is quick and when α is close to 0, dampening is slow.
- y_{t-1}: actual observation for $t - 1$.
- S_{t-1}: value of the EMA for $t - 1$.

As a filter, EMA acts as a physical boundary, choosing as peak load any value above it. This filter value will be a certain percentage multiplied by EMA, depending on the level of risk that the user can afford to take, and that will be applied according to each DSR intervention. The system scheme, as well as the final values chosen after the calibration stage, are shown in Figure 4.6 for the specific case study of Triad peaks forecasting.

Model evaluation

A range of statistical error criteria is used. The LSTM forecasting method is to be compared with other machine learning techniques, therefore several accuracy evaluation metrics are defined. As accuracy estimate, the Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and the relative error (%) based on MAE and RMSE (MAPE and RRMSE) are used (Mohanad et al., 2018):

$$RMSE = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(y_j - \hat{y}_j)^2}, \tag{3.8}$$

$$RRMSE = 100 * \frac{\sqrt{\frac{1}{N}\sum_{j=1}^{N}(y_j - \hat{y}_j)^2}}{\bar{y_j}}, \tag{3.9}$$

$$MAE = \frac{1}{N}\sum_{j=1}^{N}|y_j - \hat{y}_j| \tag{3.10}$$

$$MAPE = 100 * \frac{1}{N}\sum_{j=1}^{N}\left|\frac{y_j - \hat{y}_j}{y_j}\right|, \tag{3.11}$$

- N: Total number of values;
- y_j: actual (observed) value to compare the forecast with;
- $\hat{y}_j$: forecasted value, output of the LSTM;
- $\bar{y_j}$: average of the array of observed values.

3.1.2 Wind generation estimation

For wind and solar energy generation, the forecasting horizon goal is of several years, although one year of data is available to test the results. As a part of this Thesis, only previous power generation time series data has been used to create the forecasting models. Meteorological data could have been used also for this purpose that could have improved these results, but the forecasting horizon is very long term and existing forecast on some parameters such as wind speed, solar radiation, cloud cover parameters can be available for short-term forecasting, but not for a forecasting horizon of several years. As part of this work, it is assumed that historical records already contain implicit seasonal information for each asset, including how location affects the specific outputs for each.

As the novelty of this work is to generate very long-term forecasting, Bayesian modeling is used to model the time-series and then Markov Chain Monte Carlo for the simulations of long-term wind power generation. In the literature, this method proved to be the one which provided a longer-term forecast (Matos et al., 2010), therefore it has been decided to explore this methodology to create a different model and to generate a forecast of a year. Another reason for using Bayesian modeling is the flexibility it for modeling it offers. The model can be trained to automatically identify switchpoints within the year for more accuracy.

Next, some basic concepts such as the Bayes theorem and Markov Chain Monte Carlo are introduced. Then the specific model for wind forecasting for this work is described and last, the stochastic model for solar power forecasting is defined.

Bayes theorem

Bayesian inference is the process of deducing properties about probability distribution from the observed data by using the Bayes theorem (Bayes, 1763). First, a prior distribution is defined, by estimating the histogram of the observed dataset. The prior distribution is used as the belief, then the prior parameters are updated according to the Bayes theorem:

$$P(A|X) = \frac{P(X|A)P(A)}{P(X)}. \tag{3.12}$$

$P(A)$ refers to the prior distribution, $P(X|A)$ is the likelihood distribution, and $P(A|X)$ refers to the updated posterior probability.

Markov Chain Monte Carlo

The main idea of Markov chain (Carlo, 2004) is to have an estimate state $\mathbf{x}$ that begins as an arbitrary value, further develops and eventually becomes a sample from a certain objective distribution $p(\mathbf{x})$. Markov chains are sampled from some distribution $q^{(t)}(x)$, where t denotes the number of time steps. At the beginning, the sampling algorithm departs from some distribution $q^{(0)}$ that arbitrarily initialises x for each chain. Then $q^{(t)}$ is influenced by all previous Markov chain steps. The goal is for $q^{(t)}(x)$ to eventually converge to $p(x)$.

According to Goodfellow et al. (2016), when a single Markov chain state x is updated to a new state x', the probability of a single state transitioning to state x' is given by

$$q^{(t+1)}(x') = \sum_x q^{(t)}(x)T(x'|x), \tag{3.13}$$

where $T(x'|x)$ is the transition distribution that specifies the probability that a random update will go to state x' when departing from state x. T can be re-writen using a matrix $\mathbf{A}$, defined so that $A_{i,j} = T(\mathbf{x}' = i|\mathbf{x} = j)$, and the expression can be re-defined according to Carlo (2004) as

$$\mathbf{v}^{(t)} = \mathbf{A}\mathbf{v}^{(t-1)}. \tag{3.14}$$

Model of wind energy generation

For the purpose of wind generation estimation, Bayesian modeling is used. It is known that wind patterns are very difficult to predict, therefore this methodology needs to handle some degree of uncertainty. We rely solely on historical wind generation data to estimate wind generation. The fact of not using any climatological information adds an extra difficulty to the problem, as patterns have to be inferred from historical data.

The first step taken is to detect changes on the wind generation distribution. Then, for every separation or period, the model has been fit individually using a mixture model (Reynolds, 2009). The high level overview of the process can be seen in Figure 3.2.

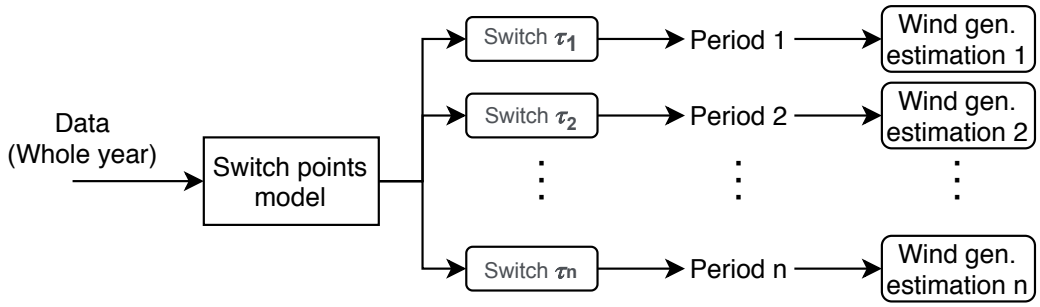


Figure 3.2: Overview of the wind generation model. One-year data is used for estimation of switch points. A different model is then applied separately to every period of data to get estimates for the period.

The first assumption is that for every period, the wind energy generation patterns have a particular probability distribution. Then the model assigns the switchpoints where changes in the arguments of such probability distribution occur. Switch points are then included in a discrete Poisson distribution, whose parameters have been defined according to the switch function implemented in the python library PyMC3 (Salvatier et al., 2016).

The schematic of the switch point detection model can be seen in Figure 3.3. Here, the index priors for the different switch points (or τs) have been defined with uniform distribution. It is assumed that wind generation data fits a beta distribution. The reason to select beta distribution is that it is very versatile: it can take either an "U" shape or just decrease from the left or increase towards the right side of the histogram. Beta distribution supports values between 0 and 1, therefore wind generation data has to be re-scaled within that range.

Once the data has been trained and the switch points have been obtained, the data is separated into the different subsets and train every period individually. The year 2018 is used as the training period, therefore the timeline with the parameters would be as shown in Figure 3.4.

For every sub-period, a mixture model is used composed of three beta distributions, whose parameters α and β are defined by uniform distribution with values between 0 and 10. The weights of each distribution have been defined by a Dirichlet distribution with $\alpha = 1$, under the condition that the sum of all of them is equal to 1.

The wind generation histogram accumulates more values near the minimum and the maximum, as observed in Figure 3.5 where switch points have already been applied. But it also can be observed that it may accumulate points around the centre of the histogram. This is the reason for choosing more than one distribution as a prior.

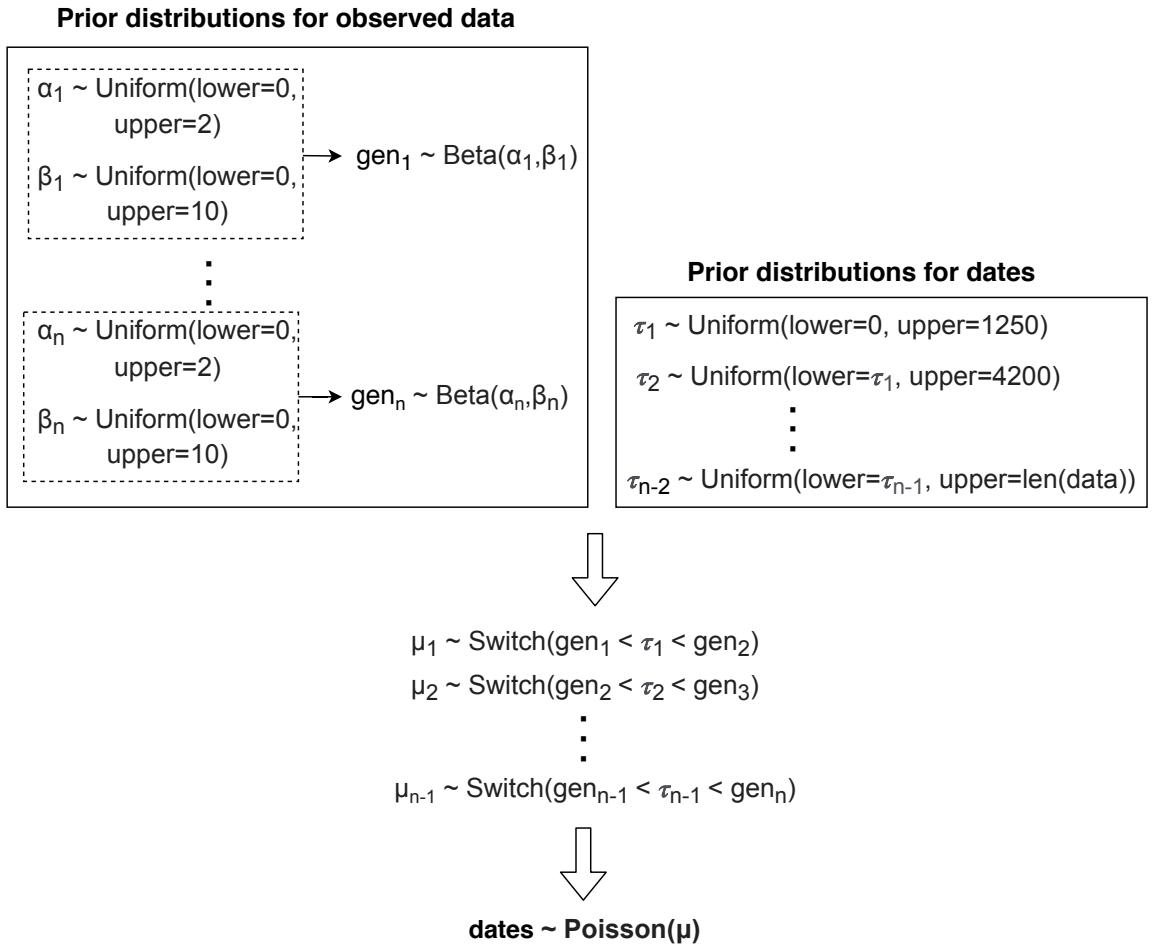


Figure 3.3: Switch-point model description. Priors for the beta distributions are defined by an uniform distribution between 0 to 10 for α and β. τ_n are defined by an uniform distribution too, where the lower boundary is the previous tau distribution to keep every limit consecutive to each other, and the upper one is restricted up to two months with respect to the previous upper limit, approximately. *gen* are different wind generation subsets, which are optimal after model is trained with observed data.

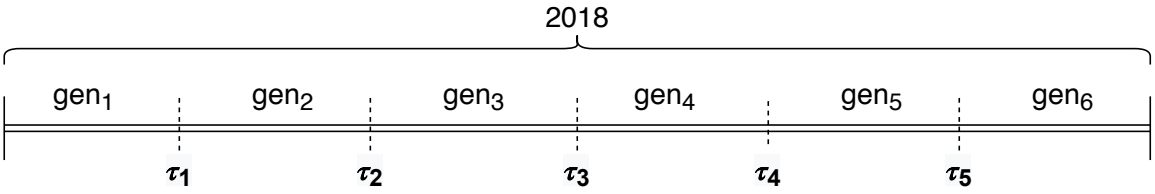


Figure 3.4: The timeline defining the estimation of wind generation for year 2018.

As can be seen in Figure 3.5, wind generation patterns oscillate between the maximum and the minimum, with a significant amount of values on both extremes. This pattern is more obvious during winter and spring, as the value moves to the centre (closer to minimum generation) in summer. After the summer period, the pattern starts inclining again towards maximum generation values.

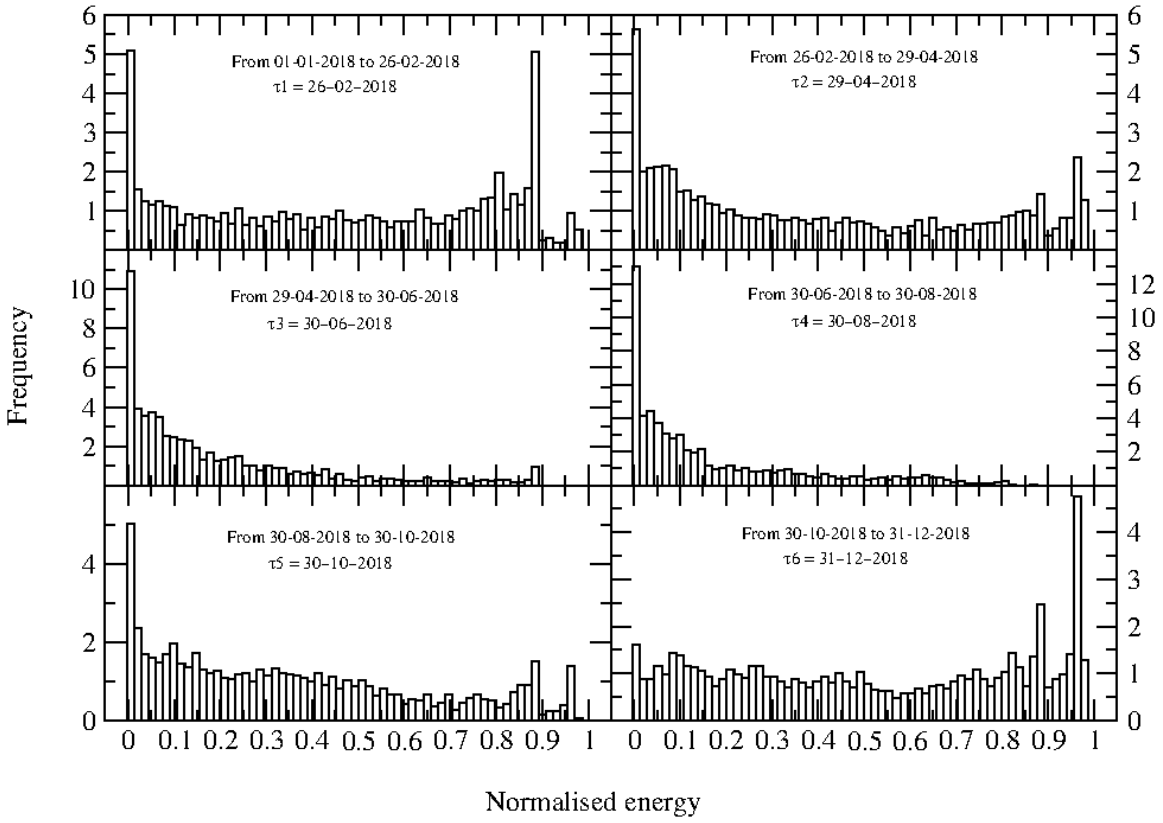


Figure 3.5: Histograms of wind generation for every switch point separation. The energy data has been normalised.

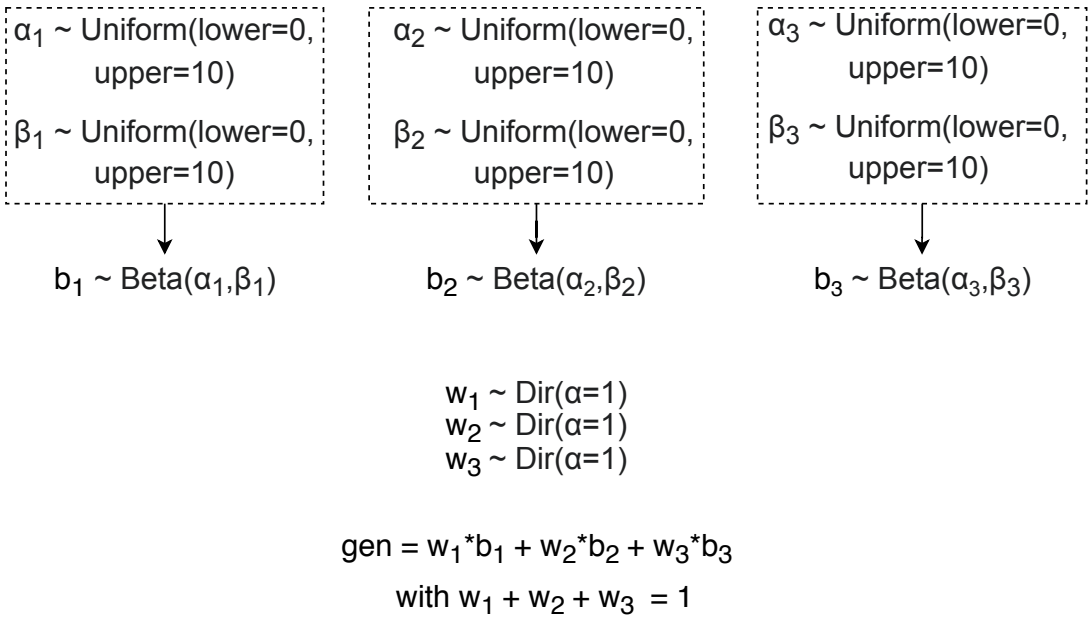


Figure 3.6: Mixture model for every period formed by three beta distributions, whose priors are defined by uniform distribution between 0 and 10 for both α and β. *Dir* is the Dirichlet distribution used to fit the weights.

3.1.3 Model of solar energy generation

A similar argument used for wind generation forecast, is used to decide the model for solar forecasting. Although in this case an added goal is to eliminate the random component of the very long-term forecast, as there is a very clear daily pattern associated with daylight hours. For this reason, a stochastic model has been designed which considers daily patterns and yearly seasonality. The parameters of the stochastic model are trained by using one year of solar power generation data.

A procedure is developed that relies on historic generation data, in order to extrapolate the generation to the following years according to a certain yearly decay of PV panels. The annual solar panel degradation can be assumed as 1% (Stahley, 2019). As meteorological data is not contemplated, a noise component to approximate random generation patterns is needed. For this purpose, red noise is generated (Gilman et al., 1963), with a scaling exponent $\beta = 0.9$, mean value= 1, standard deviation= 0.5, to create multiplicative noise.

The algorithm starts by calculating the rolling average of 30 half-hour periods. Next, the minimum and maximum of such rolling averages are estimated. Then, the lengths of the vectors for number of days and number of hours of light are defined, as to define two main trends is needed: a seasonal and an intraday. For both, a sinusoidal wave is used. Diurnal shape is introduced into the seasonal shape, with the maximum generation peaking at midday. Red noise is introduced after this, obtaining a red noise vector of the same shape as the training data and multiplying it, element-wise, by the previously obtained vector. Rate of decay of panel is added after that, depending on the number of years considered for the estimation. The definition of the model is as follows:

Let $\boldsymbol{S} = [S_0, ..., S_n]$ be the $n = 30$ period rolling average vector of solar generation, and S_{min}, S_{max} the minimum and maximum values of $\boldsymbol{S}$ where:

$$\overline{S}_t = \overline{S}_{t-1} + \frac{1}{n}(S_t - S_{t-(n-1)}), \text{ with } \overline{S}_t = \frac{1}{n}\sum_{i=0}^{n-1} S_{t-i}. \tag{3.15}$$

Now, let $\boldsymbol{t_s}$ be a vector of length equal number of days in the year and $\boldsymbol{t_d}$ be a vector of length equal Number of half-hour periods in a day, both uniformly sampled from $[0, \pi]$. M is also defined as the number of points in our training data. The seasonal component of the model is defined as:

$$y(t) = \sin(t) S_{max}, \text{ with } t \in \boldsymbol{t_s}, \tag{3.16}$$

Seasonal component is defined as the part of the stochastic model that models the seasonal behavior (winter, summer, etc.):

$$f_{seasonal}(t) = \begin{cases} S_{min} & \text{if } y(t) \leq S_{min}, \\ y(t) & \text{if } y(t) > S_{min}. \end{cases}$$

Now, $\mathbf{z}$ is defined as a vector of zeros of length M, to which the day periods are added according to:

$$Solar(k) = \mathbf{z}(k) + f_{seasonal}(t)\sin(\mathbf{t}_d), \tag{3.17}$$

for $24 < k + 48 < M, 0 < t <$ number of days in a year. As mentioned before, red noise is added $\eta(\beta = 0.9, mean = 1, sd = 0.5, size = m)$ and rate of decay of 1% defined here as $\mathbf{D}(i)$. Now, 3.17 can be re-defined as:

$$Solar(k) = (\mathbf{z}(k) + f_{seasonal}(t)\sin(\mathbf{t}_d)) \cdot \mathbf{D}(i) \cdot \eta(i), \tag{3.18}$$

for $24 < k < M - 24$, with $k - 24$ multiple of 48; $0 < t <$ number of days in a year; $0 < i < M$.

3.2 Optimisation of energy use

3.2.1 Minimisation of costs subject to shape

Price of newly built assets is prioritised over shape profile in the optimisation problem, as in a real case scenario a client would seek to fulfill electricity demand by paying the minimum price possible. In real case scenario there are other less quantifiable parameters that would influence a buyers decision, but in this study the focus is on the key parameter, which is the price of the asset:

$$\begin{aligned}
\text{minimize} \quad & \sum_{m=1}^{M} \beta_m * Ps_m * \vec{S}_m + \sum_{k=1}^{K} \gamma_k * Pw_k * \vec{W}_k, \\
\text{subject to} \quad & \sum_{m=1}^{M} \beta_m * \vec{S}_m + \sum_{k=1}^{K} \gamma_k * \vec{W}_k \geq \vec{C}, \\
& \sum_{m=1}^{M} \beta_m + \sum_{k=1}^{K} \gamma_k \geq 1, \\
& m = 1, ..., M; k = 1, ..., K, \\
& \\
& \beta_b = 0, 1; \; \forall \, b \in 1, ..., M, \\
& \gamma_c = 0, 1; \; \forall \, c \in 1, ..., K,
\end{aligned} \tag{3.19}$$

where Ps and Pw are kWh prices for every solar and wind generation asset; $\vec{S}$ and $\vec{W}$ are the vectors containing generation data for solar and wind, respectively; $\vec{C}$ is the target consumption profile; and β and γ are the binary decision variables for each generation source (*'on'* or *'off'*). Therefore, this is a linear programming minimisation problem, whose goal is to minimise the cost of energy generation for both assets, subject to difference between the aggregated generation sources and the target consumption profile and having to choose at least one asset, in order to find the optimal combination. This optimisation problem has been implemented in Python, using *optimize* method from *SciPy v1.4.1* library (Jones et al., 2001–).

3.2.2 Minimisation of shape profile without cost

A real case scenario involves costs. However, for further work on this area, it would be useful to experiment with an optimisation problem that only takes into account shapes with no cost, which also can be used in different scenarios. In order to test this minimisation problem, existing shapes are considered without forecast from three types of assets: solar, wind and landfill gas generation.

$$
\begin{aligned}
\text{minimize} \quad & \sum_{n=1}^{N} \alpha_n * \vec{LF}_n + \sum_{m=1}^{M} \beta_m * \vec{S}_m + \sum_{k=1}^{K} \gamma_k * \vec{W}_k - \vec{C}, \\
\text{subject to} \quad & \sum_{n=1}^{N} \alpha_n + \sum_{m=1}^{M} \beta_m + \sum_{k=1}^{K} \gamma_k \geq 1, \\
& n = 1, ..., N; m = 1, ..., M; k = 1, ..., K; \\
& \\
& \alpha_a = 0, 1; \forall\, a \in 1, ..., N \\
& \beta_b = 0, 1; \forall\, b \in 1, ..., M \\
& \gamma_c = 0, 1; \forall\, c \in 1, ..., K
\end{aligned}
\tag{3.20}
$$

where $\vec{LF}$, $\vec{S}$ and $\vec{W}$ are the vectors containing generation data for landfill gas, solar and wind, respectively; $\vec{C}$ is the target consumption profile; and α, β and γ are the binary decision variables for each generation source. This is a linear programming problem, whose goal is to minimise the difference between the aggregated generation sources and the target consumption profile. This optimisation problem has been implemented in Python, using *optimize* method from *SciPy v1.4.1* library (Jones et al., 2001–).

To find an optimal solution different solvers are used. A commonly used one is Limited memory Broyden-Fletcher Goldfarb–Shanno (L-BFGS-B) algorithm (Zhu et al., 1997) is a limited-memory algorithm for solving large nonlinear optimisation problems subject to simple bounds to variables, which is the main difference with respect to BFGS (Mokhtari et al., 2014). This solves unconstrained nonlinear optimization problems. The method is intended for problems on which Hessian matrix is difficult to obtain. The Truncated Newton (TNC) group of algorithms (Grippo et al., 1989) are designed for optimising non-linear functions with a large number of independent variables. The difference with the previously mentioned BFGS methods is that TNC algorithms do not use Hessian matrix. This method is normally truncated to a limited number of iterations. Constrained Optimisation By Linear Approximation (COBYLA) (Conn et al., 1997) is a numerical optimisation method for constrained problems on which the derivative of the objective function is not known. This iterative process uses the information obtained in the previous iteration for the next iteration yielding to continuous solution improvement. Sequential Least Squares Programming (SLSQP) optimizer (Kraft, 1988) uses the Han–Powell quasi–Newton method with a BFGS update of the B–matrix and an L1–test function in the step–length algorithm. The Nelder-Mead method (Powell, 1973), also known as downhill simplex method, is a numerical direct search method in a multidimensional space, which uses the vertices of a polytope of n dimensions to approximate to a local optimum, therefore does not guarantee global optimality. The Powell method (Powell, 1964) does not take derivatives, therefore it does not need to be differentiable. This method minimises the function by a bi-directional search along each search vector, iterating an arbitrary number of times until no significant improvement is made.

L-BFGS-B, TNC, COBYLA or SLSQP algorithms have in common that they support constraints. TNC-based algorithms (TNC, BFGS) are more robust and therefore would provide a solution where in other cases the optimal solution fails to converge. Similar happens with other gradient-based methods such as SLSQP. However, they require more computer memory for being a gradient-based method (Venter, 2010) and therefore they can be less suitable for very large datasets. Non-gradient based methods such as COBYLA are generally quicker and can find the optimum within the linear approximation of the objective that it creates (Wendorff et al., 2016) but it is more likely to fail in finding a global optimum if the problem is too complex. In general all the techniques are

a trade-off between computational weight and complexity. The choice of most suitable optimisation technique is investigated for the problem.

3.3 Clustering methodologies

The aim is to partition a set of data sets into homogeneous groups or clusters using machine learning techniques. Partition is performed in such a way that the objects in the same cluster are more similar to each other than to the objects in other clusters. Several time series data clustering metrics are tested, namely correlations, dynamic time warping and integrate periodogram distance. These metrics are used to perform agglomerative hierarchical clustering and then a different clustering methodology is tested: graphical lasso.

Apart from the examples exposed in the BMS clustering methodology in Section 2.2.1, another example on how hierarchical clustering works on sensor data is Shukla et al. (2020), which uses hierarchical clustering with LSTM to detect outliers on time series sensor data. The result of the clustering can be seen in Figure 3.7, which uses euclidean distance as distance measure for the time series clustering. For this work several distance metrics, that proved to separate the clusters correctly in several scenarios, are compared for the clustering.

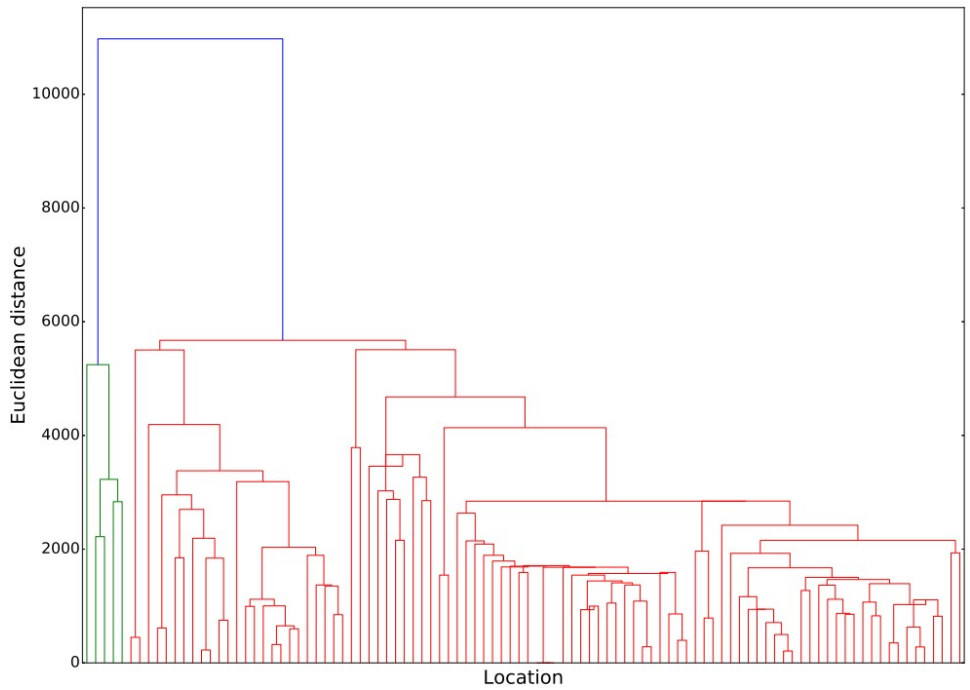


Figure 3.7: Dendrogram explaining cluster formation in sensors (Shukla et al., 2020)

For this reason, hierarchical clustering with different distance metrics should be a good approach for BMS temperature sensor data clustering, too.

3.3.1 Graphical lasso

In Friedman et al. (2008), the problem of estimating sparse graphs, graphs with only few edges, by a lasso penalty to the inverse covariance matrix is considered. Let $X_1, ..., X_n$ be independent and identically distributed $N_p(\mu, \Sigma)$ and being the estimated precision

matrix, which is the inverse matrix of the covariance matrix, denoted $\Omega \equiv \Sigma^{-1}$. The following optimisation problem is considered, according to Friedman et al. (2008):

$$\text{Objective:} \quad \hat{\Omega}_\lambda = argmin_{\Omega \in S_+^p} \{tr(S\Omega) - \log det(\Omega) + ||\Omega||_1\}, \tag{3.21}$$

where tr is the trace, $\lambda > 0$, and the penalisation parameter $||\Omega||_1 = \sum_{i,j} |\Omega_{i,j}|$ is the L_1-norm of Σ (lasso regularisation parameter). This problem maximises the penalized log-likelihood with respect to Ω, so the nodes are not fully connected and the connections kept on each cluster have useful information concerning the relationships of time series in each cluster. This happens because lasso regularisation parameter shrinks the less important features coefficient to zero, removing less meaningful coefficients. The algorithm employed to solve this problem is the GLasso algorithm, which is explained in Friedman et al. (ibid.).

With this estimate of the inverse of the correlation matrix, the partial independence relationship is obtained. If two features are independent conditionally of the others, the corresponding coefficient in the inverse of the covariance matrix would be zero, as it learns independence relations from the data, instead of being a distance measure itself between time series.

According to the authors of the implementation package in *"scikit-learn"*, the search for the optimal penalization parameter is done on an iteratively refined grid: first the cross-validated scores on a grid are computed, then a new refined grid is centered around the maximum, and so on Pedregosa et al. (2011). One of the challenges here is that the solvers can fail to converge to a well-conditioned estimate. The corresponding values of α then come out as missing values, but the optimum may be close to these missing values.

3.3.2 Agglomerative hierarchical clustering

Agglomerative Hierarchical Clustering (AHC) has a long history, especially in taxonomy or classificatory systems, and phylogenetics (Lance et al., 1967; Ward Jr, 1963). Further studies generalised this algorithm (Sibson, 1973), and have further developed and improved (Defays, 1977; Jain et al., 1988).

Based on the definition given in Zaki et al. (2014), the goal of hierarchical clustering is to create a sequence of nested partitions or clusters, which can be conveniently visualised via a tree or hierarchy of clusters, also called the cluster dendrogram. In AHC, it starts with each of the n points in a separate cluster, and then merges the two closest clusters until all points are members of the same cluster. Algorithm 1 shows this procedure, with $\mathbf{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, where $\mathbf{x}_i \in \mathbb{R}^d$, a clustering $\zeta = \{C_1, \ldots, C_k\}$ is a partition of $\mathbf{D}$.

Algorithm 1: Agglomerative Hierarchical Clustering (D,k), reproduced from Zaki et al. (2014):

$\zeta \leftarrow \{C_i = \mathbf{x}_i\} \, |\mathbf{x}_i \in \mathbf{D}|$; `// Each time series is in a separate cluster initially`
$\Delta \leftarrow \{\delta(\mathbf{x}_i, \mathbf{x}_j) : \mathbf{x}_i, \mathbf{x}_j \in \mathbf{D}\}$; `// Compute matrix with distances`
repeat
  Find the closest pair of clusters $C_i, C_j \in \zeta$;
  $C_{ij} \leftarrow C_i \cup C_j$; `// Merge the clusters`
  $\zeta \leftarrow (\zeta \setminus \{C_i, C_j\}) \cup \{C_{ij}\}$; `// Update the clustering`
  Update distance matrix Δ to reflect new clustering;
until $|\zeta| = k$;

Various distance measures between time series can be used for clustering: Pearson's correlation coefficient, dynamic time warping and integrated periodogram distance.

Pearson's correlation similarity

Let $x = [x_1 \, x_2 \, ... \, x_L]^T$ and $y = [y_1 \, y_2 \, ... \, y_L]^T$ be two zero-mean real-valued random vectors of length L. As described in Benesty et al. (2009), the Pearson's correlation coefficient between x and y is

$$\rho^2(x, y) = \frac{E^2(x^T y)}{E(x^T x)E(y^T y)}, \tag{3.22}$$

with E being the expected value. According to the *"tsclust"* package documentation in Montero et al. (2014), two different measures of dissimilarity between two time series based on the estimated Pearson's correlation can be computed. These can be $d_1 = \sqrt{2(1-\rho)}$ or $d_2 = \sqrt{\left(\frac{1-\rho}{1+\rho}\right)^\beta}$, where β specifies the regulation of the convergence.

Dynamic Time Warping

The DTW's idea is that the sequences can be extended by repeating elements and the distance is calculated between extended sequences. DTW can handle input sequences of different lengths (Gan et al., 2007).

Let $x = [x_1 \, x_2 \, ... \, x_r]^T$ and $y = [y_1 \, y_2 \, ... \, y_s]^T$ be two time series, where lengths r and s are not necessarily equal. Let M be an $r \times s$ matrix with the (i, j) element containing the squared Euclidean distance between two points x_i and y_j. Each element (i, j) in M corresponds to the alignment between two points x_i and y_j. Let $W = w_1, w_2, ..., w_k$ be a warping path, where the kth element $w_k = (i_k, j_k)$. Then $\max(r, s) \leq K < r + s - 1$ with the warping paths having the following restrictions: monotonicity, continuity and boundary conditions. There are exponentially many paths that satisfy these conditions, the optimal path being the one which minimizes the warping cost (Keogh et al., 2000):

$$DTW(x, y) = \frac{\sqrt{\sum_{l=1}^{K} w_l}}{K} = \frac{\sqrt{\sum_{l=1}^{K} d(x_{i_l}, y_{j_l})}}{K}, \tag{3.23}$$

where $(i_l, j_l) = w_l$ for $l = 1, 2, ..., K$. Then the optimal path can be found through dynamic programming as

$$\gamma(i,j) = d(x_i, y_i) + \min\left[\gamma(i-1,j), \gamma(i-1,j-1), \gamma(i,j-1)\right], \tag{3.24}$$

where $\gamma(i,j)$ is the dynamic time warping distance between the sub sequences $x_1, x_2, ..., x_i$ and $y_1, y_2, ..., y_j$.

Integrated Periodogram Distance

This distance based on the normalised periodogram was introduced in Caiado et al. (2006). Let $P_x(w_j) = (1/n)|\sum_{t=1}^{n} x_t e^{-itw_j}|^2$ and $P_y(w_j) = (1/n)|\sum_{t=1}^{n} y_t e^{-itw_j}|^2$ be periodograms of time series x and y, respectively, at frequencies $w_j = 2\pi j/n, j = 1, ..., [n/2]$ in the range $[0, \pi]$, $[n/2]$ being the largest integer less or equal to $n/2$. The interest lies only in its correlation structure, so normalized periodogram is used and it is defined by $NP(w_j) = P(w_j)/\hat{\gamma}$, where $\hat{\gamma}$ is the sample variance of the time series. Also, since the variance of the periodogram ordinates is proportional to the spectrum value at the corresponding frequencies, logarithms can be taken and therefore, the distance between x and y can be defined by Caiado et al. (ibid.) as

$$d_{LNP}(x,y) = \sqrt{\sum_{j=1}^{[n/2]} \left[\log(NP_x(w_j)) - \log(NP_y(w_j))\right]^2}. \tag{3.25}$$

Knowing that the periodogram has the equivalent representation

$$P(w_j) = 2\left[\hat{\gamma}_0 + \sum_{k=1}^{n-1} \hat{\gamma}_k cos(w_j k)\right], \tag{3.26}$$

where $\hat{\gamma}_k$ is the sample autocovariance function (defined in more detail in Brockwell et al. (1991)) which leads to

$$d_{NP}(x,y) = 2\sqrt{n}\sqrt{\sum_{k=1}^{n-1} (\hat{\rho}_{k,x} - \hat{\rho}_{k,y})^2}. \tag{3.27}$$

3.4 Early warning signals and Dimensionality reduction

The current state-of-the-art of failures analysis to HVAC applications, either applies mathematical methods to specific assets, or install specific hardware. For this work there is not enough data to build models about specific HVAC assets nor hardware can be installed, therefore data availability restricts this problem. Due to the lack of applications of early warning signals to HVAC systems under these specific conditions, other areas of application with a similar dynamic are used as a reference for choosing the methodologies to apply.

For this problem, several methods used for applications of EWS to climate-related problems and vibration analysis in manufacturing are considered. Climate and manufacturing-related fields use methodologies such as variation, power spectrum, auto-correlation, and VAE for early pattern detection. They have the key characteristics, specially in climate analysis, that can be compared to BMS sensor data analysis as both are dynamical, interdependent systems.

3.4.1 Early warning signals

There are a number of techniques to detect Early Warning Signals (EWS) as mentioned in Section 2.2.2. For the purpose of early failure detection applied to BMS four indicators are used based on variance, power spectrum, ACF1 and VAE. These early warning indicators are used with a chosen sliding window along the time series preceding the onset of a transition. In this case, the choice of the length of the sliding window is a trade-off between time-resolution (data availability) and the clarity of the change of the signal prior to transition. In order to identify trends before the transition, Kendall τ correlation coefficient is used (Kendall, 1948) in one of the indicators, PS, as this indicator presents a more chaotic behavior prior to transition. A positive Kendall τ coefficient indicates increasing trends in the indicators prior to transition, as applied in Lenton et al. (2012).

Variance
The first indicator used is the variance of time series, which is defined by:

$$s^2 = \frac{\sum(x_i - \overline{x})^2}{n-1}, \tag{3.28}$$

where s^2 is the sample variance, x_i is the value of one observation, $\overline{x}$ is the mean value of all observations and n is the number of observations. The reason to use variance is to compare a more simple approach with other EWS indicators.

Power Spectrum
Another EWS indicator is the power spectrum scaling exponent β , which is calculated by estimating the slope of the power spectrum $S(f)$ of the data, plotted on logarithmic axes (Bak et al., 1987) in medium term range (Prettyman et al., 2018). Exponent β can be estimated according to Livina et al. (2007) as:

$$S(f) \sim f^{-\beta}, \tag{3.29}$$

where the power spectrum is approximated by the periodogram, obtained from the absolute value of the fast Fourier transform. Then β is obtained by measuring the slope inside of the frequency range $10^{-2} \leq f \leq 10^{-1}$.

Lag-1 Auto-Correlation Function
Autocorrelation function is also used for the purpose of this work. This function measures correlation of the time series within itself at different time lags. According to Box et al. (2015), the definition of the lag-k autocorrelation function is:

$$r_k = \frac{\sum_{i=1}^{n-k}(x_i - \overline{x})(x_{i+k} - \overline{x})}{\sum_{i=1}^{n}(x_i - \overline{x})^2} \tag{3.30}$$

Lag-1 ($k = 1$) autocorrelation function (ACF1) is used, estimated in sliding windows to obtain ACF-indicator.

Variational Auto-Encoder (VAE)
This technique is a novel methodology for EWS. According to Kingma et al. (2019), the VAE is a network which attempts to represent the input with a PDF instead of several hidden nodes. This is illustrated in Kingma et al. (2013), with a training set $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_A]^T$ being $\boldsymbol{x}$ vectors from time $t = 1$ to time $t = A$.

VAE uses a neural network for the probabilistic encoder $q_\theta(\boldsymbol{z}|\boldsymbol{x})$ to approximate the posterior of the generative model $p_\theta(\boldsymbol{x}, \boldsymbol{z})$. Let the prior over the latent variables be the

centered isotropic multivariate Gaussian $p_\theta(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{0}, \boldsymbol{I})$. The parameters of the distribution are computed with a fully-connected neural network with a single hidden layer that attempts to reduce the reconstruction error. According to Kingma et al. (2013), the posterior is approximated with a multivariate Gaussian with a diagonal covariance structure:

$$\log q_\theta(\boldsymbol{z}|\boldsymbol{x}^{(i)}) = \log \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{2(i)}\boldsymbol{I}) \tag{3.31}$$

where mean and standard deviation, $\boldsymbol{\mu}^{(i)}$ and $\boldsymbol{\sigma}^{(i)}$, are the outputs of the encoding half of the network. Concerning the decoder half of the network, samples are computed from the posterior $\boldsymbol{z}^{(i,l)} \sim q_\theta(\boldsymbol{z}|\boldsymbol{x}^{(i)})$ using $\boldsymbol{z}^{(i,l)} = g_\phi(\boldsymbol{x}^{(i)}, \boldsymbol{\epsilon}^{(l)}) = \boldsymbol{\mu}^{(i)} + \boldsymbol{\sigma}^{(i)} \odot \boldsymbol{\epsilon}^{(l)}$ where $\boldsymbol{\epsilon}^{(l)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, with $\odot$ being an element-wise multiplication. As both prior ($p_\theta(\boldsymbol{z})$) and posterior ($q_\theta(\boldsymbol{z}|\boldsymbol{x})$) are assumed to be Gaussian, the resulting estimator for the model and datapoint $\boldsymbol{x}^{(i)}$ is, according to Kingma et al. (ibid.):

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}^{(i)}) \simeq \frac{1}{2}\sum_{j=1}^{J}(1 + \log((\sigma_j^{(i)})^2) - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2) + \frac{1}{L}\sum_{l=1}^{L}\log p_\theta(\boldsymbol{x}^{(i)}|\boldsymbol{z}^{(i,z)}),$$
$$\text{where } \boldsymbol{z}^{(i,l)} = \boldsymbol{\mu}^{(i)} + \boldsymbol{\sigma}^{(i)} \odot \boldsymbol{\epsilon}^{(l)} \text{ and } \boldsymbol{\epsilon}^{(l)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}). \tag{3.32}$$

It is first assumed a system working under normal conditions and training the model parameters ($\boldsymbol{\mu}^{(i)}$ and $\boldsymbol{\sigma}^{(i)}$) with such data. As EWS indicator, the reconstruction error is tracked, which is defined with the root mean square error (RMSE), described by:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}, \tag{3.33}$$

where n denotes the number of sampled points in a sequence, and $\hat{y}$ and y are the reconstructed output and the actual state. The VAE indicator is constructed, therefore, by measuring the reconstruction error according to the RMSE, defined as the Variational Autoencoder Reconstruction Error (VAERE). An application in manufacturing of this methodology for EWS can be seen in Figure 3.8. The Health Indicator (HI) is constructed based on VAE for this work for a specific manufacturing application (Wen et al., 2018).

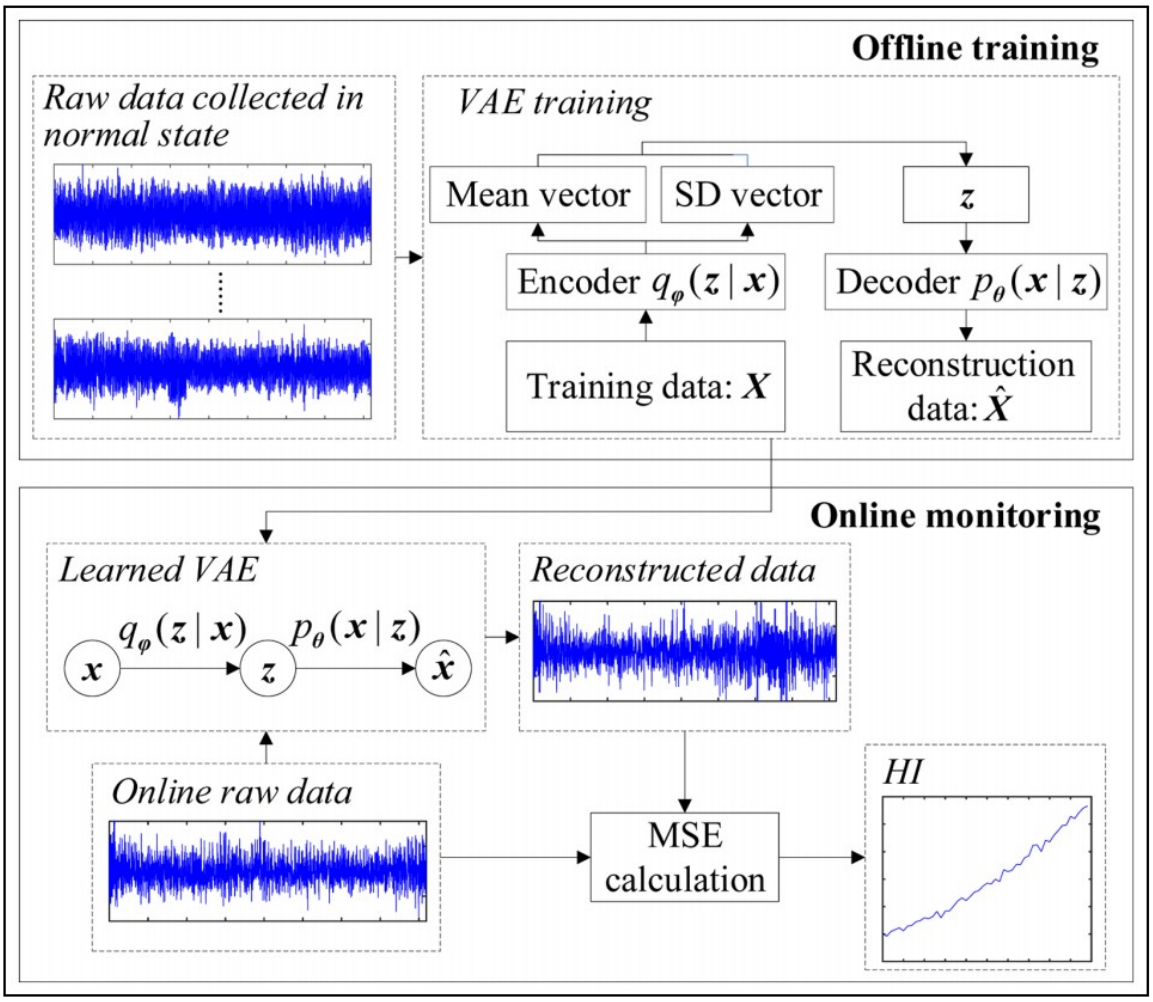


Figure 3.8: Health Indicator (HI) construction method based on VAE (Wen et al., 2018)

3.4.2 Principal component analysis

Principal Component Analysis (PCA) was first introduced in Pearson (1901). It is a methodology that obtains an r-dimensional basis that best captures the variance in the data. Given input data $\boldsymbol{D} \in \mathbb{R}^{n \times d}$ and the desired threshold α, it selects the smallest set of dimension r that captures at least an α fraction of the total variance. The steps are shown in Algorithm 2.

Algorithm 2: Principal Component Analysis (D,α), reproduced from Zaki et al. (2014):

1 $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$; // Compute mean
2 $\boldsymbol{Z} = \boldsymbol{D} - \mathbf{1} \cdot \boldsymbol{\mu}^T$; // Center data
3 $\boldsymbol{\Sigma} = \frac{1}{n}(\boldsymbol{Z}^T\boldsymbol{Z})$; // Compute covariance matrix
4 $(\lambda_1, \lambda_2, ..., \lambda_d) = eigenvalues(\boldsymbol{\Sigma})$; // Compute eigenvalues
5 $\boldsymbol{U} = (\boldsymbol{u}_1\ \boldsymbol{u}_2\ ...\ \boldsymbol{u}_d) = eigenvectors(\boldsymbol{\Sigma})$; // Compute eigenvectors
6 $f(r) = \frac{\sum_{i=1}^{n}\lambda_i}{\sum_{i=1}^{d}\lambda_i}, \forall r = 1, 2, ..., d$; // Fraction of total variance
7 Choose smallest r so that $f(r) \geq \alpha$; // Choose dimensionality capturing α fraction of total variance
8 $\boldsymbol{U}_r = (\boldsymbol{u}_1\ \boldsymbol{u}_2\ ...\ \boldsymbol{u}_r)$; // Reduced basis
9 $\boldsymbol{A} = \{\boldsymbol{a}_i | \boldsymbol{a}_i = \boldsymbol{U}_r^T\boldsymbol{x}_i, i = 1, ..., n\}$; // Reduced dimensionality data

The trajectory with the largest projected variance is called the first principal component, the orthogonal trajectory to the first one which captures the second largest projected variance is the second principal component, and so on.

3.5 Text classification methods

This section describes the text classification algorithms used for the purpose of BMS tagging. For classification, the study deals with multi-class and multi-label problems. There are several methods compared for the purpose of this work: logistic regression, decision trees, random forests, multinomial Naive Bayes and SVM. Considering that the text to classify has the peculiarity of being very short (i.e. maximum 11 words per input), the choice of the method works better is investigated in this case.

From the different methodologies explored in the literature review for the text classification part, the ones explored in this Thesis are the ones whose applications are similar to the problem defined here. The text explored in this work for BMS application has a maximum length of 11 words, therefore past applications to short text classification with a high accuracy have been explored.

Other applications for short-text classification, such as the diverse tweets classifiers, utilised random forests, xgboost, logistic regression, Naive Bayes and SVM among others. These classifiers proved to deliver good accuracy for these particular problems. Therefore the methodologies explained in this section are the ones that are used to obtain the results, due to the length of the text used here.

All methods have been implemented in Python v3.7, and the library used to reproduce the algorithms is *sci-kit learn* (Pedregosa et al., 2011).

Logistic regression classification

Logistic regression is also called logit regression, maximum-entropy classification (MaxEnt) or the log-linear classifier. Despite its name, this linear model works as a classifier more than as a regressor. The logistic function is a monotonic function defined between 0 and 1 (Wright, 1995)

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}}, f(x) \in [0,1], \tag{3.34}$$

where x_0 is the value of the sigmoid midpoint, L is the saturation point of the curve and k the logistic growth rate or steepness of the curve. The objective function of the logistic regression maximizes the likelihood function. The Maximum Likelihood Estimation (MLE) can be written as follows (White, 1982):

$$argmax_\beta : \log \left\{ \prod_{i=1}^{n} P(y_i|x_i)^{y_i} (1 - P(y_i|x_i))^{(1-y_i)} \right\}, \tag{3.35}$$

where y_i is the output between 0 and 1, $P(y_i|x_i)$ the posterior probability which is equal to $1/(1 + e^{-f})$, and β is the vector of weights/coefficients.

Decision trees

Decision trees form classification (or regression) models in the form of a tree structure. They break the dataset down into increasingly smaller subsets while the decision tree is incrementally developed. The resulting decision tree has decision nodes and leaf nodes.

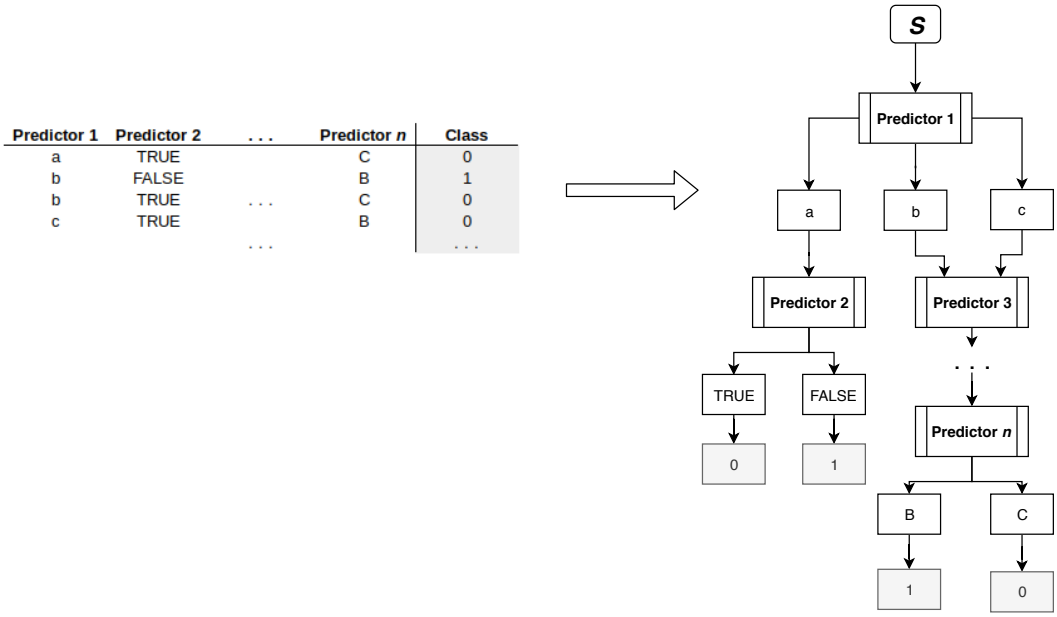


Figure 3.9: Schematics of decision table and decision tree.

The core algorithm for building decision trees is the Iterative Dichotomiser 3 (ID3), developed by Quinlan (1986). The algorithm begins with the original set, iterates on every unused attribute of the set S and calculates the entropy $H(S)$, defined as $H(S) = \sum_{x \in X} -p(x) \log_2 p(x)$, where S is the current dataset, X the set of classes in S and $p(x)$ the proportion of the number of elements in class x with respect to the number of elements in set S.

Random Forests

Random forests or random decision forests (Barandiaran, 1998) are an ensemble learning method for classification, that constructs a finite number of decision trees at training time, increasing the number of results for a better output. This ensemble method should,

by definition, provide better, although sometimes very similar, results.

After training, predictions for unobserved samples x' can be made by averaging the predictions from all the individual regression trees on x', $\hat{f} = \sum_{b=1}^{B} f_b(x')$. In the case of classification trees, the alternative option is performed by taking the majority vote, also known as *voting algorithm*.

XGBoost

xgboost was developed by Chen et al. (2016). The methodology creates a scalable end-to-end tree boosting system and introduce a sparsity-aware algorithm for parallel tree learning. It uses a gradient boosting framework. Generally, xgboost is fast when compared to other implementations of gradient boosting. The summary of the main features are listed below:

- **Regularisation:** It penalises more complex models through lasso and Ridge regularisation to prevent overfitting.
- **Sparsity awareness:** It automatically captures missing values depending on training loss and handles different types of sparsity patterns more efficiently.
- **Cross-validation:** The algorithm has a built-in cross-validation method at each iteration, thus excluding the need to hard-code this search and to specify the number of iterations required.
- **Parallelisation:** It uses parallelised implementation. This is possible due to the interchangeable nature of the different loops, building many different trees in parallel. This feature allows many users to run state-of-the-art algorithm without requiring a very powerful computer.

Multinomial Naive Bayes

Multinomial Naive Bayes (Maron, 1961) is a specialized version of Naive Bayes (NB) that is widely used in text analysis. Whereas simple NB would model presence and absence of particular words, multinomial naive Bayes explicitly models the word counts and adjusts the underlying calculations, as explained in McCallum et al. (1998), combining probability distribution of P_r with fraction of documents belonging to each class for each class j and word i, at a word frequency of f_i, according to McCallum et al. (ibid.):

$$Pr(j) \propto \pi_j \prod_{i=1}^{|V|} Pr(i|j)^{f_i}, \tag{3.36}$$

where $\pi_j = \frac{class_j}{\sum_{n=1}^{N} class_n}$ is the fraction of documents or labels on each class, and $|V|$ the feature space. The sum of logs is used to smooth the probability going increasingly up when a word re-appears several times, for which the log frequency is taken:

$$Pr(j) = \log(\pi_j) + \sum_{i=1}^{|V|} \log(1 + f_i) \log(Pr(i|j)). \tag{3.37}$$

Support vector machines classification

SVMs are discriminative classifiers originally developed by Vapnik et al. (1963), based on a separating hyperplane. Given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In a two-dimensional space, this hyperplane is a line dividing a plane into two parts, and each class lays on each side.

Let us consider the case of two classes. A training dataset has n points of the form $(\vec{x_1}, y_1)...(\vec{x_n}, y_n)$, where y_i has the value of either 1 or -1, indicating the class to which $\vec{x_i}$ belongs. The goal is to find the "maximum-margin hyperplane" that divides the group of points of both classes. Any hyperplane can be written as the set of points $\vec{x}$ that satisfies:

$$\vec{w} \cdot \vec{x} - b = 0, \tag{3.38}$$

where $\vec{w}$ is the normal vector to the hyperplane, and $\frac{b}{||\vec{w}||}$ determines the offset of the hyperplane from the origin along the normal vector. For the *hard-margin* case, the minimisation problem to solve is:

$$\text{minimise} \quad ||\vec{w}|| \quad \text{subject to:} \quad y_i(\vec{w} \cdot \vec{x_i} - b) \geq 1 \quad \text{for} \quad i = 1, ..., n \tag{3.39}$$

In the *soft-margin* case, the minimise function according to McCallum et al. (1998), is:

$$\left[\frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i(\vec{w} \cdot \vec{x_i} - b))\right] + \lambda||\vec{w}||^2, \tag{3.40}$$

where the parameter λ determines the trade-off between increasing the margin size and ensuring that each point belongs to the correct side of the margin.

The way to transform SVM to create a non-linear classifier is by means of the kernel trick, a method of using a linear classifier to solve a non-linear problem. Boser et al. (1992) suggested a way to create non-linear classifiers by applying the kernel trick to maximum margin hyperplanes. The resulting algorithm is similar, except that every dot-product is replaced by a non-linear kernel function. This allows the algorithm to fit the maximum-margin hyperplane in a transformed feature space.

As explained in the literature in Section 2.2.3, Rane et al. (2018) compares some of these methodologies for sentiment analysis. The comparison of the different methodologies for this particular work can be seen in Figure 3.10.

Classifier	Precision	Recall	F- Measure
Decision Tree	63%	64.6%	64.5%
Random Forest	85.6%	86.5%	86.5%
SVM	81.2%	84.4%	84.8%
Gaussian Naïve Bayes	64.2%	64.7%	64.6%
AdaBoost	84.5%	83.5%	83.5%
Logistic Regression	81%	81.6%	81.9%
KNN	59%	59.2%	59.3%

Figure 3.10: Accuracy of classifier for tweets sentiment analysis by Rane et al. (2018)

Chapter 4

BMS AI Design implementation and results

4.1 Triad forecast

A period of 4 years of national electricity demand data is used for the analysis, as there are policies that change in a relatively short period of time. The Triad forecasting model will be validated for four different periods, all of them from November to February, when the Triad season occurs. LSTM is applied and trained for this period.

4.1.1 Triad data analysis

In this section, the historic data is analysed and the relationship between the settled demand data and the energy generation are found with some of the renewable sources. Next, a plot with the historical Triad is discussed, and last, correlations in the data are sought that are going to be used as inputs for the LSTM model, as well as to identify seasonality in the training data.

Settled data and renewable generation

The data used for selecting Triad days is not the Initial Demand Out-turn (INDO), but the Settlement Final/ 1st Reconciliation/ 2nd Reconciliation (SF/R1/R2), which is the actual load on the grid once the BM units have submitted all the sub-meter data. This data is settled at around 9, 20 and 90 days post the event and is the data that Triad is calculated against. The main difference between INDO and settled is the removal of the station load (the load the power station uses to power itself). This is why the output forecast first is the SF/R1/R2.

There are limitations in the models depending on the amount, type and quality of data available. There are forecasts provided for the INDO; however, as there are no forecasts for the Settled Demand (SD) available, a model needs to be defined based on the available data. The idea is to find parameters that keep a relationship with the difference between the INDO and the SD. Generation renewables (wind and solar generation), and the mentioned difference between actual INDO and the SD keep such relationship. As shown in Figure 4.1, plotting $INDO - SD$ against the sum of solar and wind generation, the points obtained can be approximated by a linear regression, which makes solar and wind generation possible predictors for the SD forecast system.

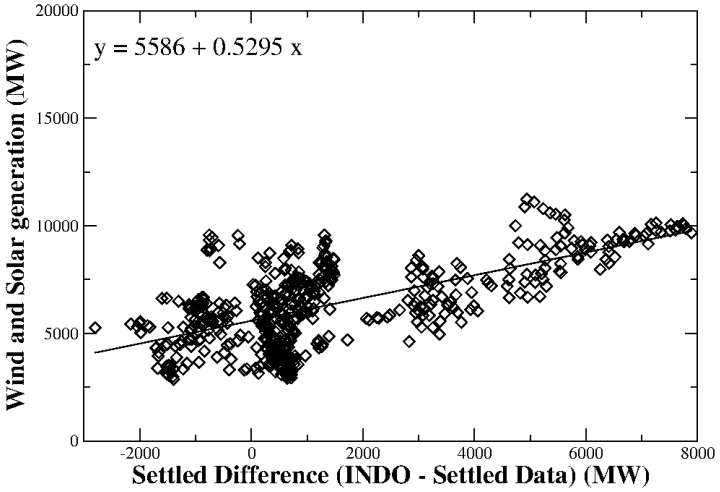


Figure 4.1: Correlation between renewables generation and settled data

The fact that this data is correlated means that it can be used for a predictive model that forecasts the SD as a first step for Triad forecasting.

Historical Triads

In order to analyse Triad, it is useful to know when Triad historically happened, so statistical insights can be gained for future decisions.

From the 2008/09 to the 2015/16 winter seasons, 45.8% of Triad occurred on Mondays and 29% on Thursday, with other weekdays only accounting for one in four Triads. Out of the total 24 cases, 22 occurred between 17:00 and 17:30 and 2 occurred between 17:30 and 18:00. A reason why a Triad may happen in a latter hour around February may be explained by the number of hours of sunlight, which grows longer after the January period and thus, moving forward the second peak of electricity demand, meaning that users switch on lighting a bit later than before, generating possible peaks later than in the beginning of the Triad season.

4.1.2 LSTM inputs analysis

First of all, it is necessary to study the influence factors of the SD. From the model point of view, it is also interesting to identify correlations between the temperature and INDO.

- National Demand Forecast (NDF)
- Wind Generation Forecasting (WIND)
- Transmission Demand Forecast (TSDF)
- Solar Generation Forecasting (SOLAR)

The four input data variables are forecasts for the next 48 half-hour periods predicted by ELEXON, obtained 24 hours before the event of interest. Solar and wind data is based on historical data and local wind and solar forecasts, used by National Grid to forecast likely levels of solar and wind generation. NDF is based on historically metered generation output for Great Britain. The values shown here take into account transmission losses and include station transformer load, pump storage demand and interconnector demand (ELEXON, 2018). In the plots of National Demand Forecast and Transmission

Demand Forecast data of several years as shown in Figure 4.2, it can be seen that both of them are decreasing over the years. This means that the actual demand does decrease and that this fact needs to be taken into consideration.

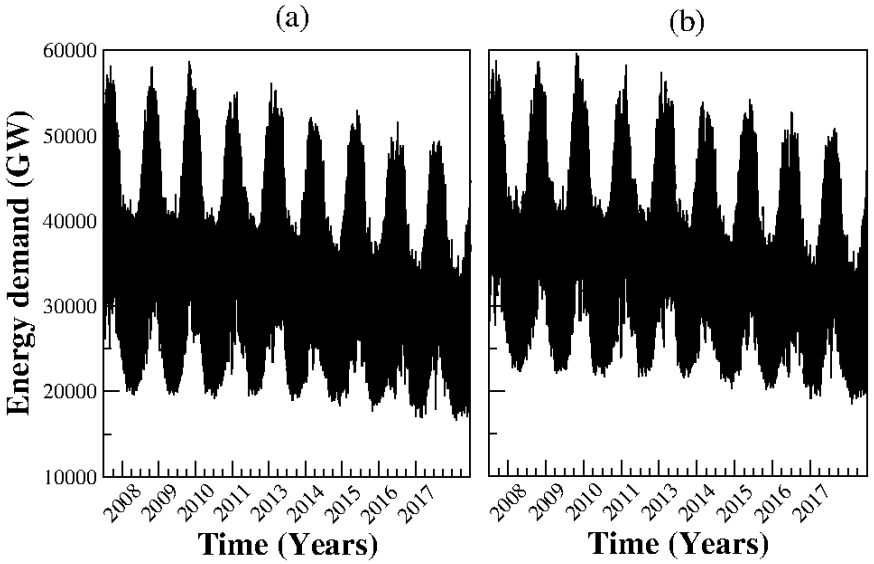


Figure 4.2: National Demand Forecast (a) and Transmission Demand Forecast (b) from January 2008 to October 2017

Also, it is useful to display every quarter of the year for the actual INDO to see the differences in patterns between seasons. So, taking the year 2017 as an example, each quarter of the year is plotted in Figure 4.3.

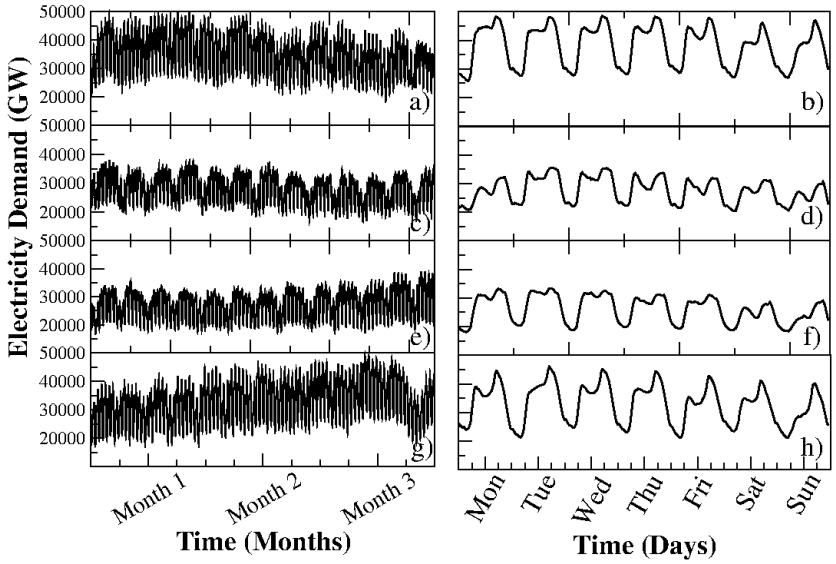


Figure 4.3: UK Electricity Demand for the 4 quarters of the year 2017 on the left: Panel (a) January - March, panel (c) April - June, panel (e) July - September and panel (g) October - December. Random week taken from each corresponding quarter, Monday to Sunday, on the right: 16/01/2017 to 22/01/2017 (b), 01/05/2017 to 07/05/2017 (d), 07/08/2017 to 13/08/2017 (f) and 06/11/2017 - 12/11/2017 (h).

As can be seen in Figure 4.3, the patterns of energy use are different depending on the season and/or day of the week. As for wind and solar energy, they depend on weather

conditions. For the Triad season 2016/17, these values are displayed in Figures 4.4 and 4.5.

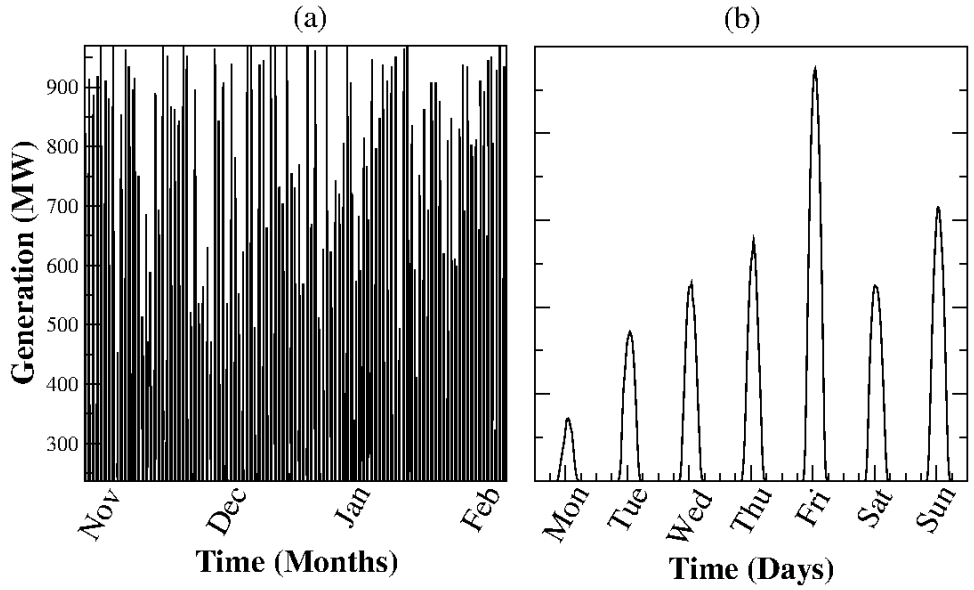


Figure 4.4: Observed data of the national total solar generation from November 2016 to February 2017 (a) and from 16th January 2017 to 22nd January 2017 (b). Data source: ELEXON (2018).

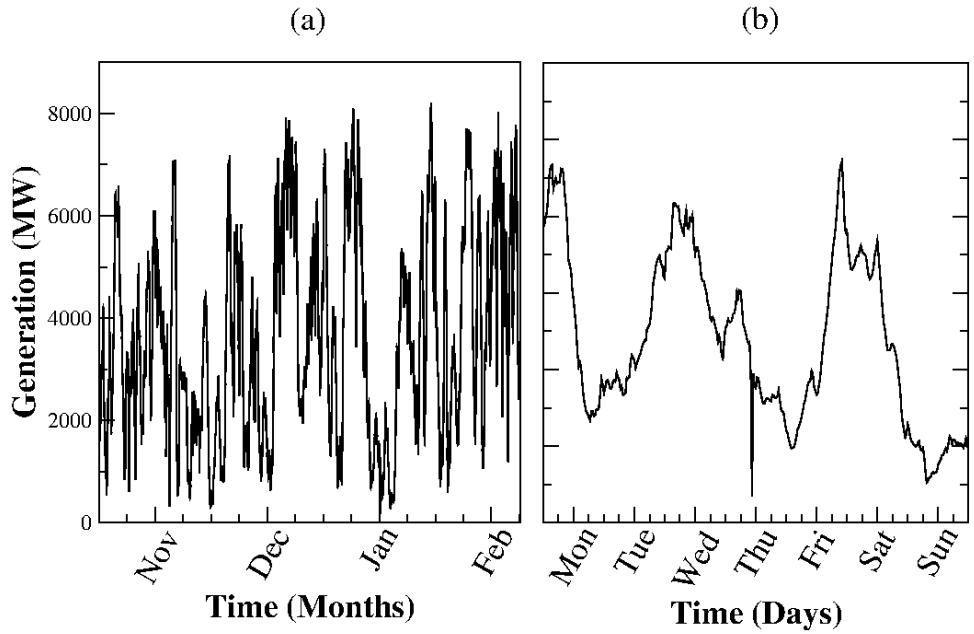


Figure 4.5: Observed data of the national total wind generation from November 2016 to February 2017 (a) and from 16th January 2017 to 22nd January 2017 (b). Data source: ELEXON (2018).

To determine data correlations, the standard correlation coefficient (*Pearson's* r) can be computed. The result of the four inputs that are being considered as inputs can be seen in Table 4.1.

Table 4.1: Pearson's r coefficients between Settled Data and the other variables.

Variables	Pearson's r with SD
NDF	0.99
TSDF	0.99
SOLAR	0.18
WIND	-0.15

As expected, a strong positive correlation exists between SD and NDF and TSDF values, and these are going to define the shape of the forecast. Also there are correlation between the wind and solar generation, and SD data. Now that the relationships between variables and the data has been discussed, the model will be built and tested.

4.1.3 System configuration

The goal of our modeling is to produce Triad signals (as few as possible) to determine a DSR intervention.

First of all the data is rescaled between 0 and 1 as this is a beneficial machine learning practice: when feeding the algorithm, the weights are assigned during the training stage of the system, and having different scales of values may lead to bad fitting and not reaching a global maximum solution. The rest of the system, which is also described in Figure 4.6, is as follows:

- **LSTM forecasting**: Provides with the SD forecasted values for the next 48 half-hour settlement periods. This architecture built consists of 40 concatenated cells, with 250 epochs, and a batch size worth of 6 days of data.
- **Peak extraction**: The maximum demand peak of the day is extracted and added to a vector with the previous forecast peaks. For filtering purposes, weekends and Christmas period (23^{rd} of December to 2^{nd} of February) are excluded from the dataset.
- **Filters**: After the demand of the next 24 hours has been forecast, two different filters have been used based on a simple approach. The idea is to use two Exponential Moving Averages (EMAs) multiplied by a factor. As an example of what the filter values may be, for this work the percentages to 3.5% and 4% are set for the soft and hard filters, lower and higher risk respectively.

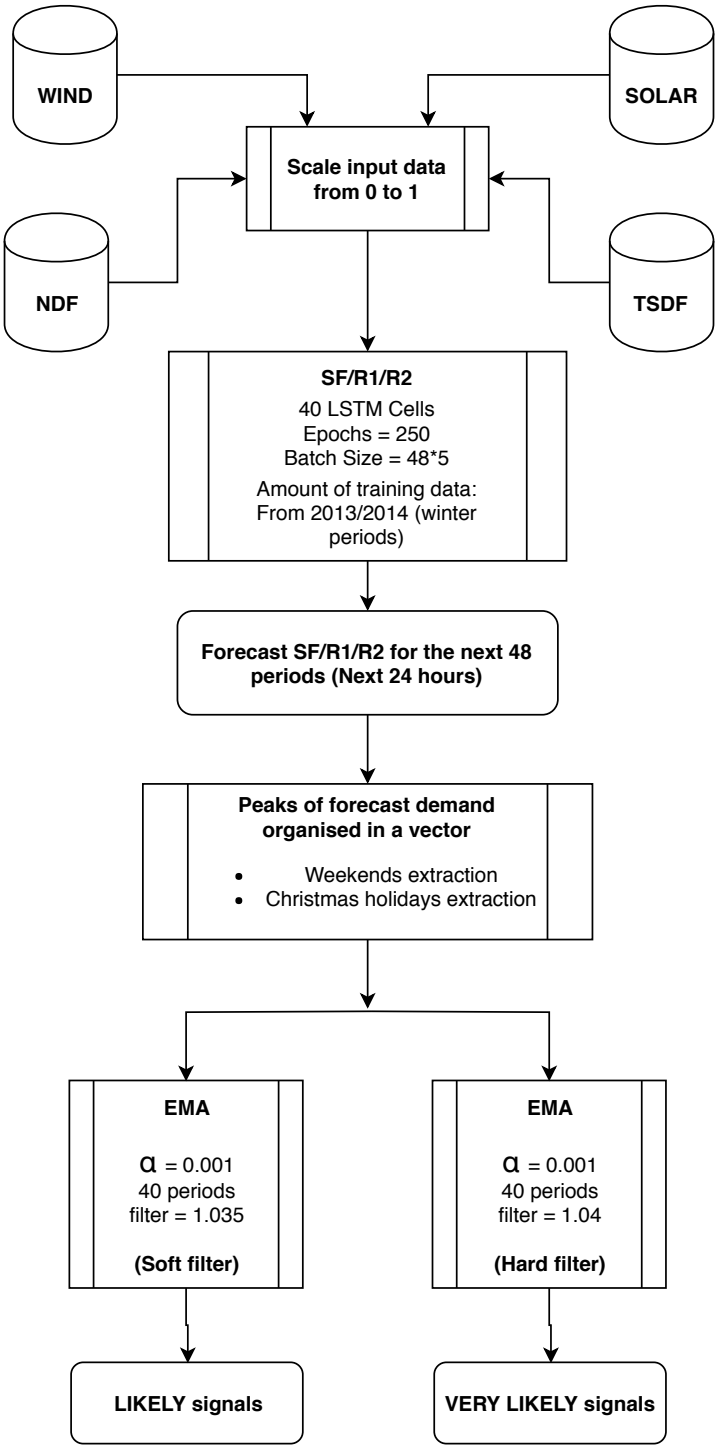


Figure 4.6: Triad detection procedure. After Settlement Final, First Reconciliation and Second Reconciliation (SF, R1, R2) have been forecasted, irrelevant forecast data such as weekends and Christmas holidays are extracted for their exclusion. Then, Exponential Moving Averages (EMAs) are calculated over forecasted values to obtain areas of high Triad risk.

The implementation of the detection procedure was carried using Python, with the Keras library. The dataset was divided into the following subsets for model deployment. The training set includes the data from the four different inputs (WIND, SOLAR, NDF and TSDF). Both training and test sets refer to final settled data (SF/R1/R2):

- Training Set: 1^{st} November 2015 to 14^{th} of February 2018. November to February for each year is the period for which the SF/R1/R1 data is available. Training and test set are consecutive, meaning that the period used for the test is consecutive to the training, so time sequence is taken into account.
- Test set: next 14 days after the last day of the training set (15^{th} to 28^{th} of February 2018). The purpose of this testing set is to estimate the performance when changing the hyperparameters in order to choose the optimum combination of these. This test data must not have been used for training.

A summary of the datasets can be found in Table 4.2.

Table 4.2: Training and testing data for Triad forecast.

	Training	**Testing**
Number of points	11472	672
(%)	94.46%	5.54%
Date range	01.11.2015 to 28.02.2018	15.02.2018 to 28.02.2018

These train-test split percentages are used due to the need of using as much data from previous and actual periods as possible to capture long-term trends, and two weeks of test data would suffice to show the periodicity captured in the forecast, both intraday and weekly. Long-term forecast is performed, thus the model is trained on the date from the previous years. The horizon of the forecasting is four months. As our model has four inputs, based on the description in 4.1.1, the LSTM architecture will contain a fixed number of the concatenated cells. This number is to be determined during the experiment.

The schematic of the model can be seen in Figure 4.7.

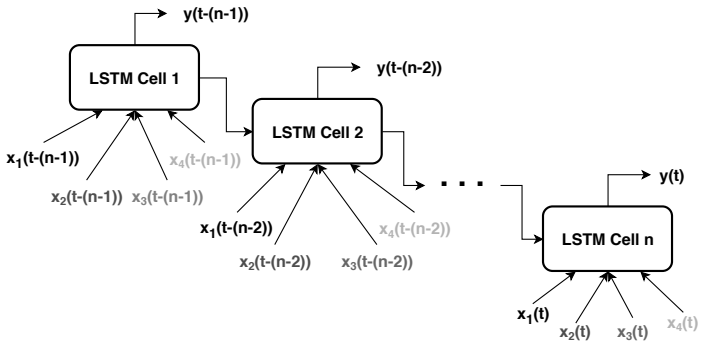


Figure 4.7: LSTM schematic with 4 inputs

4.1.4 1^{st} calibration stage: training data size

As mentioned in section 3.1.1, the LSTM parameters need to be tuned in order to achieve better performance. The hyperparameters of this network are calibrated according to the number of epochs, the number of years of data for the training stage and the number of cells constituting the network.

The experiment has been carried out by increasing the number of epochs, as well as the number of years of the testing period, varying the batch size used for training the LSTM. One epoch makes reference to the number of times the model has seen a number

of instances equal to the number of instances in the training data. The batch size is the number of values taken by the algorithm at every step, in this case 6 days worth of data (48 half-hours each day).

Table 4.3 shows the metrics with the best result obtained that correspond to 250 epochs with 288 batch size (6 days with 48 periods each) with the whole dataset. This means that the earlier the year of testing the shorter the amount of training data. The length of the test period has an influence on the number of epochs. Therefore for further experiments, the whole dataset before the test period is to be taken because less data is available.

Table 4.3: Training and testing accuracy metrics according to the numbers of epochs.

Epochs	**50**	**150**	**250**	**300**	**350**
Training accuracy					
RMSE (MW)	740.85	694.44	**693.11**	699.76	698.65
RRMSE (%)	2.09	1.94	**1.94**	1.96	1.96
MAE (MW)	574.24	522.95	**521.28**	527.60	526.56
MAPE (%)	1.66	1.52	**1.51**	1.53	1.53
Testing accuracy					
RMSE (MW)	833.44	799.75	**793.53**	796.59	798.27
RRMSE (%)	2.35	2.24	**2.22**	2.23	2.24
MAE (MW)	625.15	578.4	**578.15**	577.67	578.43
MAPE (%)	1.75	1.60	**1.60**	1.60	1.60

4.1.5 2^{nd} calibration stage: number of neurons

Next, the number of neurons of the LSTM needs to be determined by using the results obtained in the prior calibration stage. For the experiment in Table 4.3, 30 concatenated cells were used and the average of the values for ten runs have been taken for each metric. In Table 4.5 the results of this experiments can be seen together with a boxplot with the Forecasted Error (FE) (Al-Musaylh et al., 2018), which is the difference between observed and predicted values, for each model presented in Figure **??**.

Table 4.4: Boxplot for each model of forecast values realisation with different numbers of LSTM cells.

Number of neurons	**10**	**20**	**30**	**40**	**50**	**60**
RMSE (MW)	801.82	798.85	795.54	795.66	794.33	797.97
RRMSE (%)	2.25	2.24	2.23	2.23	2.22	2.23
MAE (MW)	585.26	578.86	576.48	575.99	578.95	581.17
MAPE (%)	1.61	1.60	1.60	1.60	1.61	1.61

Table 4.5: Accuracy metrics for the number of neurons in the LSTM.

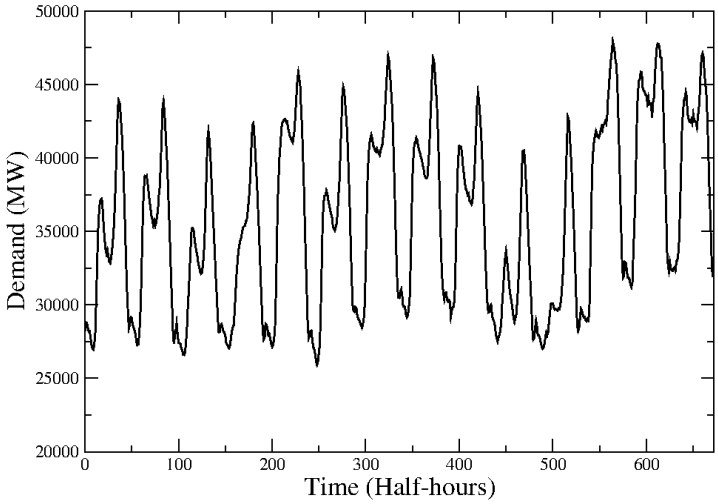
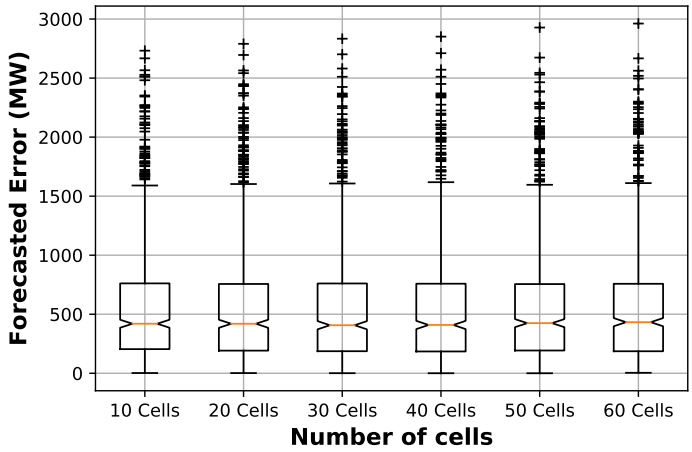


Figure 4.8: LSTM modeled energy demand time series.



In this experiment, the values for 10 runs are averaged. It can be concluded that for this forecasting horizon, the number of concatenated cells to be chosen is 40, as the metrics in Table 4.5 show the best results for this number of cells.

The training algorithm used is the Adam optimiser, with a learning rate $= 0.001$ (Kingma et al., 2014). The total number of parameters, weights and biases is 7241 and the number of training instances is 11472.

The modeled demand time series can be seen in Figure 4.8. In this experiment, 14 days (672 points for 48 half-hours per day) were modeled in February 2018, from Wednesday to Tuesday. It also can be seen that the weekly patterns are lower than for weekdays. The next section will consider the filters and signals for Triad as the last stage of the modeling system.

4.1.6 LSTM comparison with other models

In this section, the performance of LSTM is compared with several other popular methods. First, the LSTM is compared with the mean-only model and then with a simpler version of ANN with the same characteristics in terms of the number of cells and learning rate used for the LSTM. Also the LSTM model is compared with SVM regression, ran-

Table 4.6: Comparison of models based on accuracy metrics.

Model	LSTM	Mean-only	ANN	SVM	Random forests	Bayesian regression
RMSE (MW)	795.66	6013.81	850.82	2321.85	1868.25	806.05
RRMSE (%)	2.23	16.91	2.37	6.50	5.27	2.27
MAE (MW)	575.99	5295.46	619.42	2009.95	1588.54	587.68
MAPE (%)	1.60	15.24	1.74	5.78	4.55	1.62

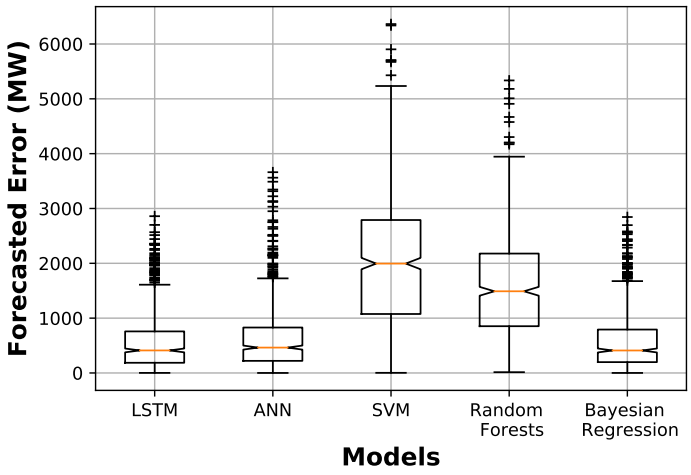


Figure 4.9: Forecast error of the five models.

dom forests and Bayesian regression models. The results of the comparison are shown in Table 4.6 and illustrated in Figure 4.9.

This shows that LSTM model outperforms other models. It is important to mention that Bayesian regression results are following closely the performance of the LSTM in the second place, which makes this methodology a good alternative worth testing in similar DSR scenarios.

4.1.7 Cross-validation

Generalised performance of a learning method and its prediction capability rely on independent test data (Friedman et al., 2001). Therefore, cross-validation is necessary to ensure that the results are reliable when new data is introduced. Time-series data is the object of the forecast, therefore a cross-validation that considers the serial correlation inherent to the problem is needed (Arlot et al., 2010). Therefore, one step ahead cross validation is performed (Hyndman et al., 2018) consisting of use $1, \ldots, k$ samples, to predict $k+1$ value (or alternatively $k+1, \ldots, k+m$ values). This is performed for the whole period of validation and then 24 hours of data are added to the model to obtain the following day's output.

4.1.8 Filters configuration

Applying the data filters is the last stage of data processing. They play the role of data points classification between Triad and no Triad.

For the filters, as mentioned in section 3.1.1, EMAs have been used. The idea is to call Triads as any value above the 3.5% and 4% for hard and soft filters, respectively, of the 40 days EMAs. The values have been chosen to capture all three Triads causing the least

disruption possible. For filters validity in terms of parameters selection they must satisfy the following:

- To successfully predict the three Triads at least with one of the two filters;
- To be able to successfully call at least two Triads using the signals of both filters.
- To call the minimum number of Triad possible so the energy supply disruptions in buildings are kept to minimum;

The level of financial/energy risk that is to be taken into account depends on the user. This is why two filters are used to manage the risk. In Figures 4.10 - 4.13 the forecast peaks for the Triad seasons 2014/15, 2015/16, 2016/17 and 2017/18 respectively are shown, together with both filters:

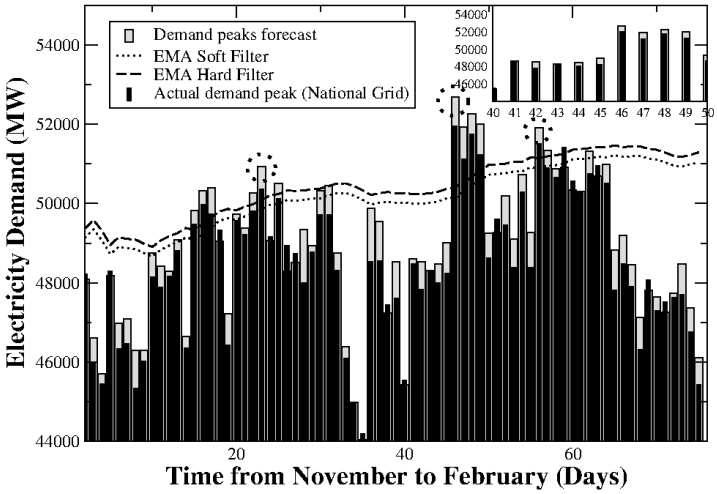


Figure 4.10: 2014/15 forecast peaks for each day and EMA filters.

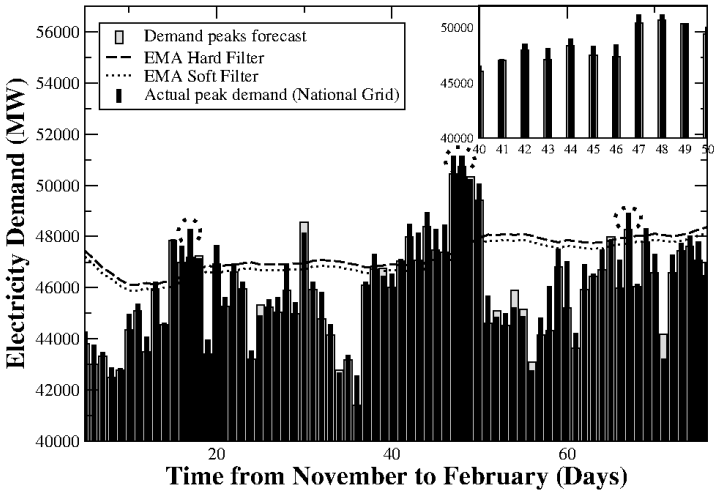


Figure 4.11: 2015/16 forecast peaks for each day and EMA filters.

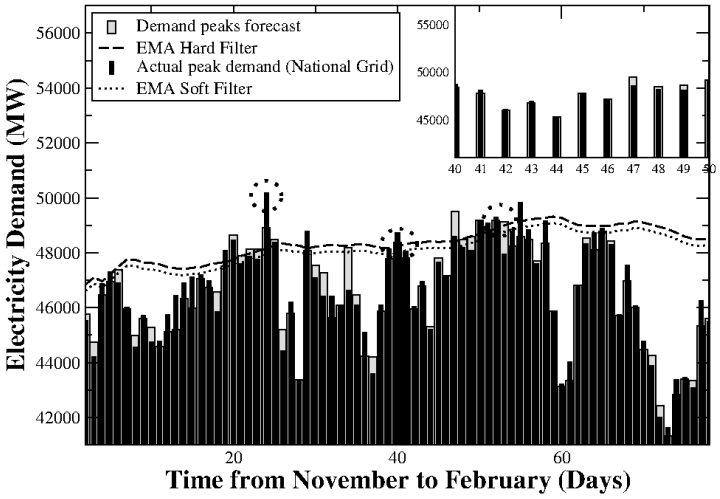


Figure 4.12: 2016/17 forecast peaks for each day and EMA filters.

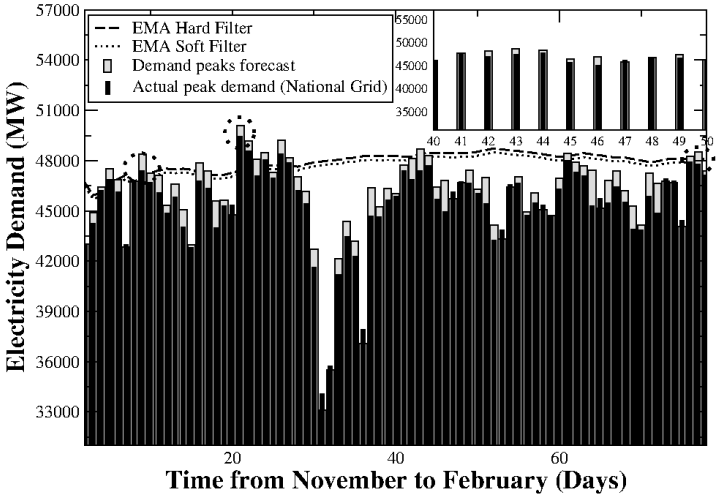


Figure 4.13: 2017/18 forecast peaks for each day and EMA filters.

The performance of the filters can be measured by counting the number of signals that our model generates versus the number of Triad predicted in the hindcast. The summary of these results can be seen in Table 4.7, in which, for each year, the number of signals given by each filter, as well as the number of actual Triad days predicted. It shows the number of positive signals generated and the last two are the number of these signals that predicted the actual Triads.

Table 4.7: Number of signals calling Triad for each filter and number of Triads predicted.

Seasons	**Soft Filter**	**Hard Filter**	**Triad Predicted (Soft)**	**Triad Predicted (Hard)**
2014/15	19	14	3	3
2015/16	21	19	3	3
2016/17	19	14	3	3
2017/18	21	16	3	2

The condition respect to the soft filter has been met by properly forecasting the three Triads in the first three testing years (2014 - 2017) and successfully predicting 2 out of 3

over the 2017/2018 period. The aim is to call the minimum number possible of signals, so EMA parameters are valid for all three years of testing, meaning that the three conditions for filters calibration mentioned in Section 4.1.8 have been met.

This system failed to predict only one of the Triads with one of the filters but, in this case certain parameters have been left fixed for all the testing periods. The flexibility of this system allows to adjust the filters in future scenarios.

This system was implemented in the Mitie company in December 2018 and it runs automatically every day in winters since then. A schematic of the software can be seen in AppendixA.

4.1.9 Findings summary

Development of a deep learning system for detecting and forecasting events of high energy demand at a national level for general DSR interventions, with $RRMSE = 2.23\%$ achieved in LSTM and being able to predict the Triads with a fixed level of risk.

4.2 Risk balancing of renewables PPA

The objective of this study is two-fold: estimation of solar and wind generation for the following year, and optimisation of price and shape of the business consumption profile. The estimation of both wind and solar generation relies only on observed data, therefore no other predictors are used. For the electricity prices are estimated by averaging a certain range of prices given for year 2018, which is the training period used. Information about total volume generated and prices for the 2018 can be seen in Table 4.8:

Table 4.8: Annual volume (KWh) and price (£/MWh) for different assets and consumption profile.

	Wind Farm 1	**Wind Farm 2**	**Wind Farm 3**	**Wind Farm 4**	**Solar Farm 1**	**Consumption Profile**
Annual volume (MWh)	77728.30	19579.12	55530.87	41753.34	58776.61	195511.72
Price (£/MWh)	51.50	53.98	55.60	61.03	48.40	-

Temporal resolution of data is half-hourly, therefore the total length of the annual data is 17520 points for all wind and solar farms, and for the consumption profiles. Specific characteristics such as turbine model and farm locations are not provided, due to data confidentiality.

4.2.1 Wind generation

After applying the switchpoint model described in Section 3.1.2 to wind asset 1, these are different dates obtained for wind pattern change, and the respective histograms for each. Note that in this study 5 switchpoints are used.

Our goal is to generate similar patterns by using Bayesian modeling and MCMC. Figure 4.14 shows the weight, α and β values for the three distributions comprising the

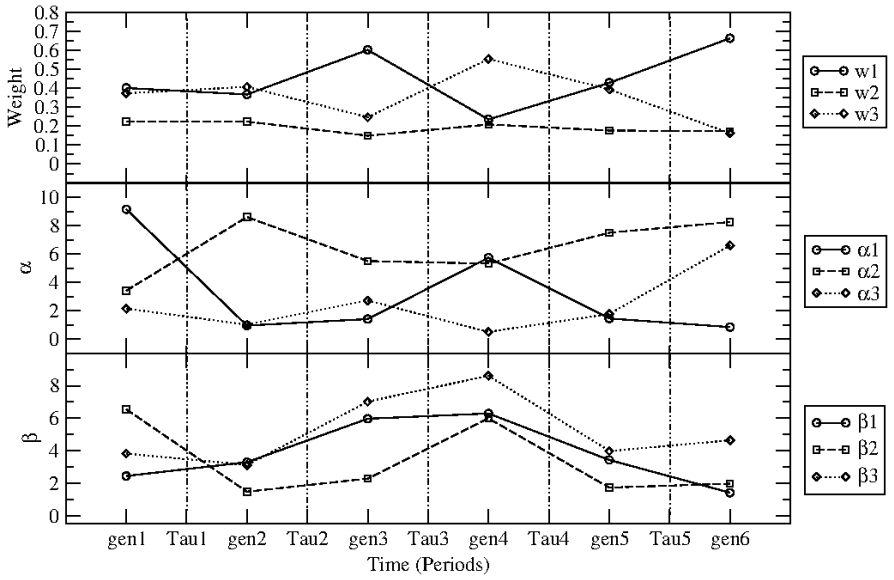


Figure 4.14: Parameters for the Bayesian mixture model in each interval between switch-points.

mixture model. The weights show the importance of each of the distributions in each period of generation. The results are used for wind simulation for the following year, as our main interest is to capture a range where the wind generation values are more likely to be homogeneous. The results of this model for every period can be seen in Figure 4.15:

It can be seen that 20 simulations of the mixture model of the three beta distributions approximate the histogram of the data very well for every period. This figure displays the differences between summer and winter period in both observed and simulated data. Due to the smoothness of the mixture data, it can be seen that the simulated data could miss some spikes in generation data close to 0.9, but due to the nature of the beta distribution, it captures the small values well in all cases. This shows the flexibility of beta distribution in different scenarios.

The generated simulations are compared with their respective observations. Results can be seen in Table 4.9.

Table 4.9: Comparison of actual and simulated total annual volume for 2019 for the four wind farms (hindcast exercise).

	Wind farm 1	**Wind farm 2**	**Wind farm 3**	**Wind farm 4**
Actual volume (MWh)	77728.30	19579.12	55530.87	41753.34
Simulated volume (MWh)	80448.22	23164.32	60067.23	48774.21

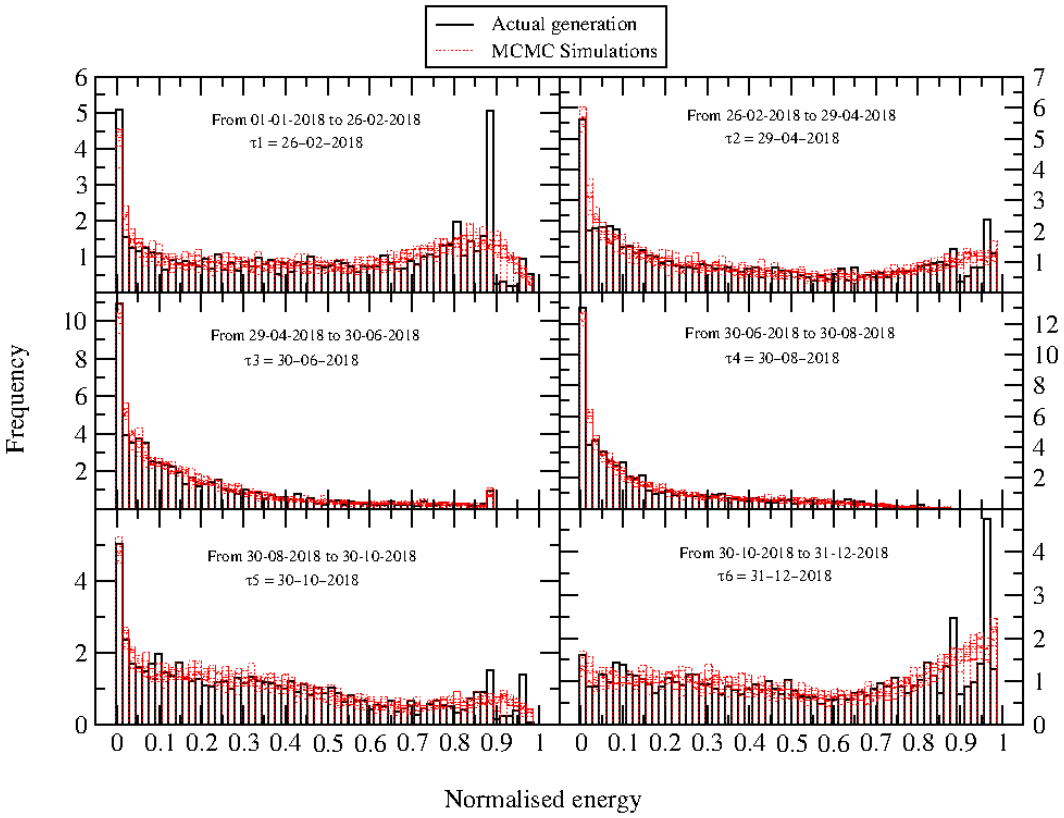


Figure 4.15: Histograms of wind generation and MCMC estimations for a wind asset (for six separated sub-periods). A total of 20 MCMC traces have been used to create the histograms in every sub-period. The data has been scaled between 0 and 1.

4.2.2 Solar generation

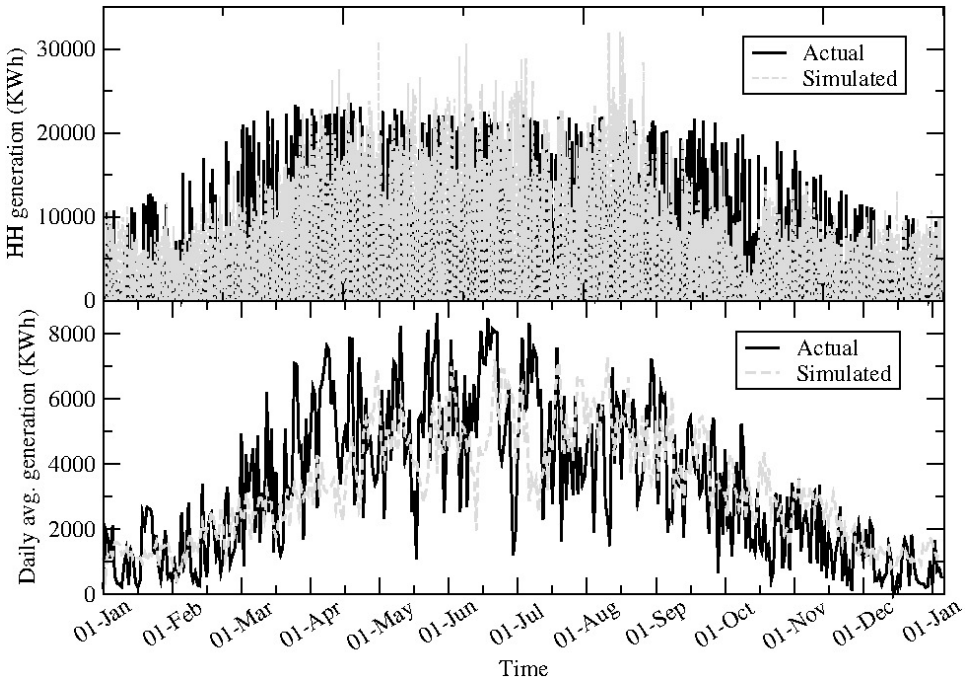


Figure 4.16: Actual and simulated solar generation for year 2019. The simulations have been carried out with the stochastic model described in 3.1.3. The top panel represents data of half hour temporal resolution. The bottom panel represents daily average generation. Year 2018 has been used as a reference period for training in order to simulate values to the following year.

Figure 4.16 shows solar generation for 2019 and its corresponding simulations for the same year. The reference year used for training the stochastic model is 2018, therefore rate of decay is applied over the following year. As the model considers minimum and maximum peaks on the range of its 30 period moving average, the pattern followed by the seasonal trend is similar. Oscillations within the same day are modeled by the red noise.

Daily modeled data is shown with the seasonal trend and the noise. The noise amplitude increases from winter to summer, and then decreases back to winter levels. These irregularities are reproduced by the simulated data, which shows, not only the same seasonality pattern, but also the same noise pattern.

Also, the forecast for the three years with available data set are compared, in terms of the total volume, in MWh. These results can be seen in Table 4.10.

Table 4.10: Comparison of actual and forecast total annual volume of solar generation for three consecutive years, from January 2017 to December 2019.

	2017	**2018**	**2019**
Actual volume (MWh)	58776.61	62362.27	58424.66
Simulated volume (MWh)	59845.69	59789.51	59955.11

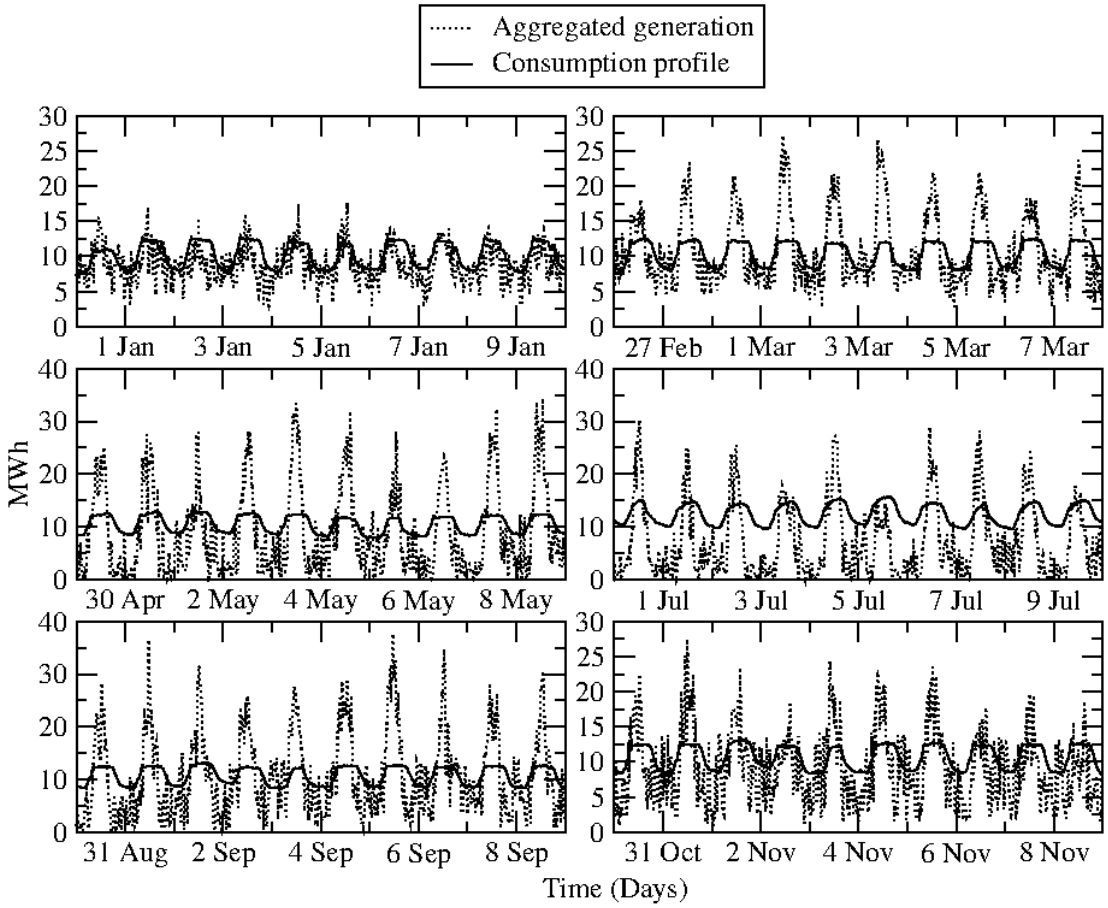


Figure 4.17: Optimal mixed generation of wind and solar, and energy consumption. Every panel corresponds to ten days in six different periods of year 2019. Time resolution is half-hourly.

Now that the estimations for both wind and solar generation have been calculated, the same model is applied for the rest of wind assets and find the best modeled output that minimises total cost and difference of shape between total aggregated and consumption.

4.2.3 Optimal cost shape match

After estimating future wind and solar generation, the same methods have been used for estimating generation in the rest of the farms to find the optimal combination of wind and solar generation. For the consumption profile, it is assumed that a very similar pattern repeats year after year, as it has been observed in the historical data.

As shown in Figure 4.17, the total aggregated generation sometimes exceeds the consumption curve. The total aggregated generation struggles to reach consumption for baseload during the summer period, due to the lack of wind generation available. However, there is normally a compensation of solar generation within the day during peakload.

Figure 4.18 shows daily aggregated data for both optimal generation and consumption, where the lack of wind generation in summer can be seen. Also, the plot shows that the consumption increases in summer, therefore the total difference is even higher. Wind generation tends to decrease in summer, yet solar generation usually increases.

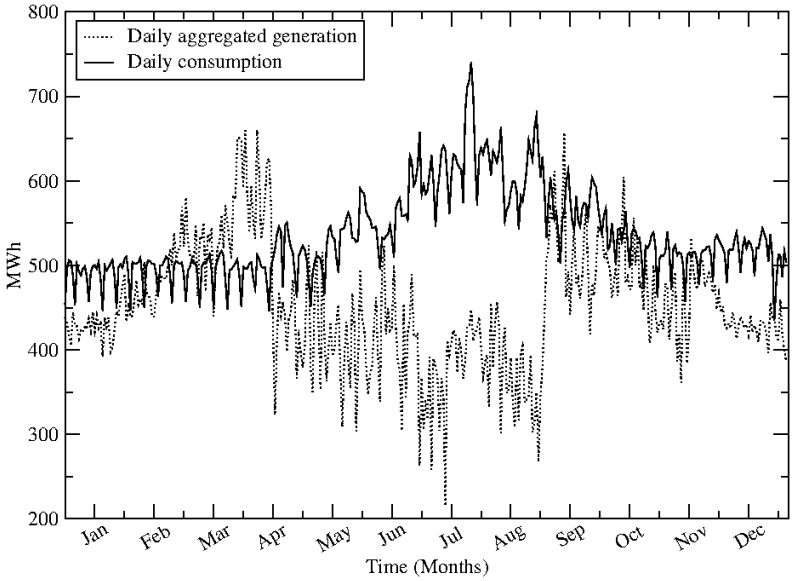


Figure 4.18: Optimal daily mixed generation of wind and solar and energy consumption in 2019. Temporal resolution is daily.

For solving this optimisation problem several solvers were applied (TNC, COBYLA, Powell, etc.), but they did not converge to a suitable solution. Therefore, the problem has been solved heuristically, by prioritising cost over shape.

Optimisation has been performed over the half-hourly data (Figure 4.17). For optimisation, half-hourly resolution data is chosen to assess more accurately the amount of electricity to buy or sell within the day. This helps one to assess the risk of purchasing energy blocks in the intraday or mid-term energy market.

4.3 Electricity generation and consumption profile optimisation with no cost

This section details a different minimisation problem that takes into account the minimisation of the difference between consumption and the sum of the generation profiles in order to get an optimum combination of generators to meet demand.

This minimisation problem has not been built from a forecast as the previous problem description does. Instead, actual consumption profiles are used. The reason for this is that landfill gas generation sources are used (together with wind and solar generation) and this type of generation source for forecast is not considered, as the most common scenario for a PPA based on renewable sources are wind and solar generation due to their cheaper price per MWh.

This experiment uses a total of 19 landfill gas assets, 7 wind assets and 1 solar asset. This is limited by the amount of asset data of each type that could be obtained. For the electricity consumption, profiles from four different companies have been obtained. Their names could not be disclosed due to data confidentiality, but their characteristics can be seen in Table 4.11:

Table 4.11: Annual energy volume (KWh) of four different industry samples.

	Consumption Profile 1	**Consumption Profile 2**	**Consumption Profile 3**	**Consumption Profile 4**
Industry	Restaurant	Supermarket	Banking	Property
Annual volume 2019 (MWh)	495030.79	52301.81	231836.36	62347.60

Due to the many gaps in the generation data, only a month of data has been used for both consumption and generation profiles: 01-06-2019 to 01-07-2019.

For the purpose of the optimal combination of renewables assets, actual consumption and generation are first matched. For this, the solver that minimised the the difference between aggregated generation and target consumption is used, as defined in Eq. 3.20.

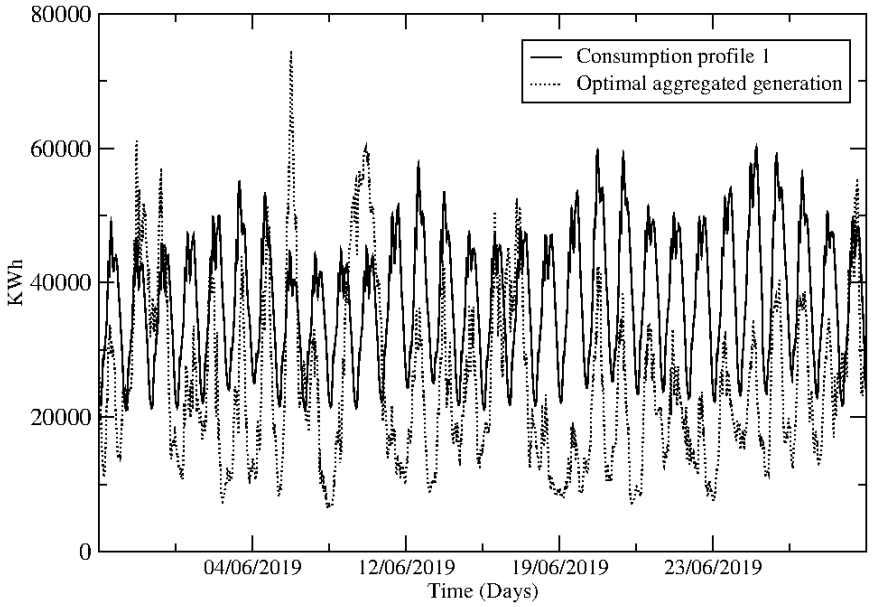


Figure 4.19: 30 min resolution data, from 1^{st} of June to 1^{st} of July 2019, of consumption profile 1 and its corresponding optimal combination of generation assets from the given asset list.

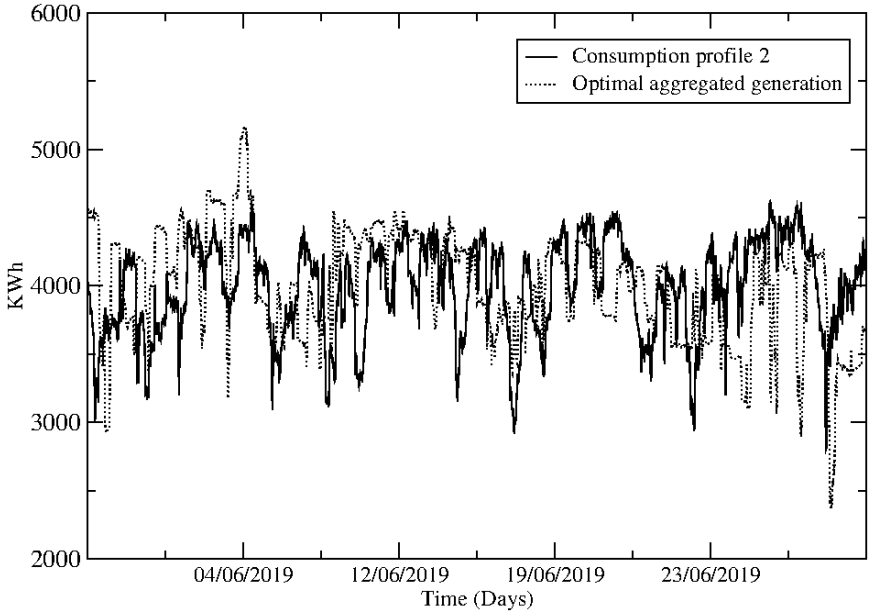


Figure 4.20: Half-hourly data of consumption profile 2 and its corresponding optimal combination of generation assets from the given asset list.

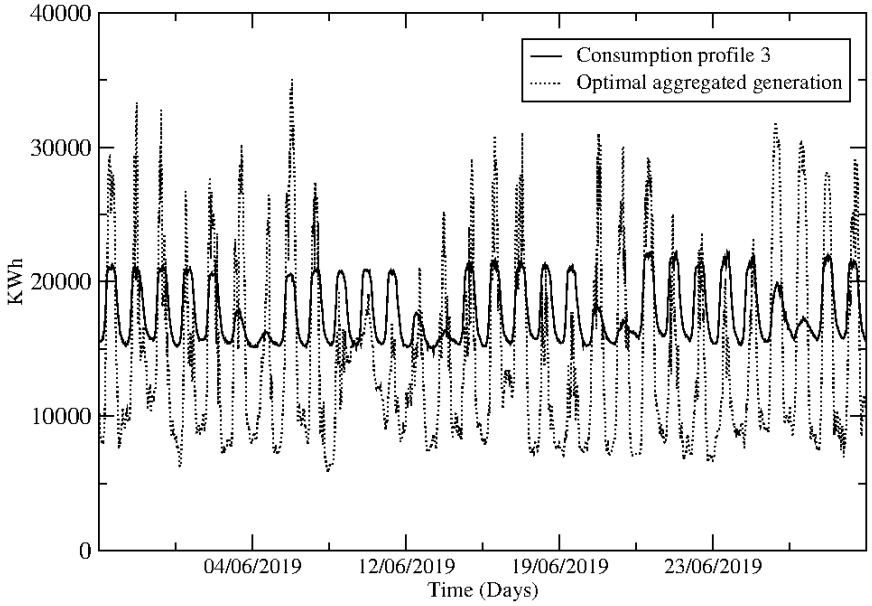


Figure 4.21: Half-hourly data of consumption profile 3 and its corresponding optimal combination of generation assets from the given asset list.

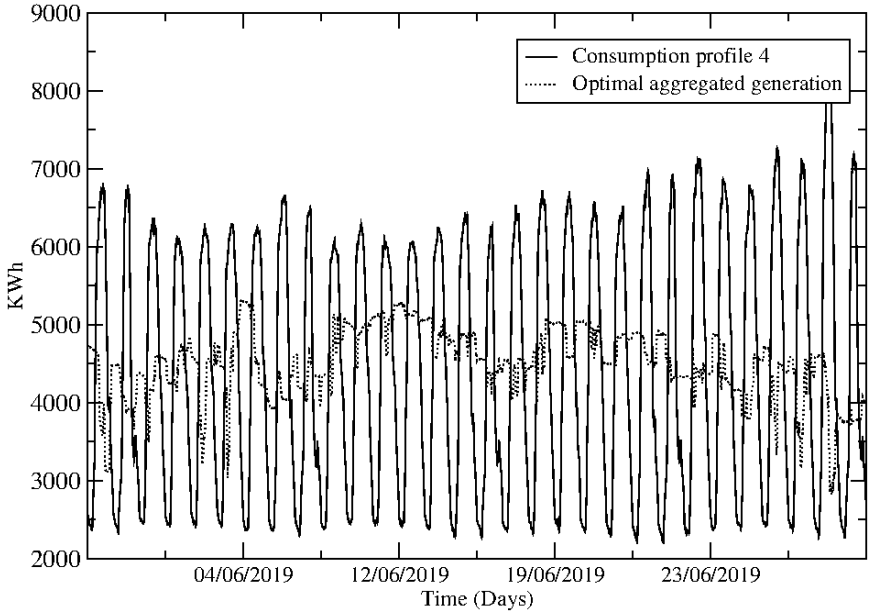


Figure 4.22: Half-hourly data of consumption profile 4 and its corresponding optimal combination of generation assets from the given asset list.

Figures 4.19 to 4.22 show the results of the consumption profile and aggregated generation match. It can be seen here the role of each type of asset: wind and landfill gas compensate most part of the baseload, whereas solar covers generation peaks. Landfill gas generation is generally more steady and less seasonal, and the character of solar generation compensates for intraday consumption. The role of solar for peakload is very important, as solar energy is generally cheaper and it compensates consumption within a number of hours when electricity is more expensive.

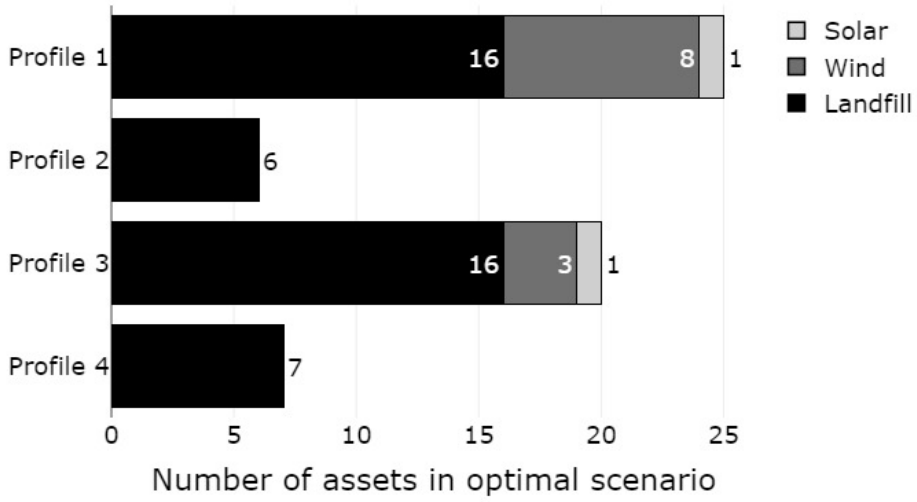


Figure 4.23: Optimal combination of assets for each profile.

In Figure 4.23, it can be seen the asset distribution for optimal scenarios. In the two

Table 4.12: Root mean square error of different solvers for the optimisation problem (KWh).

Solvers / Profile	**L-BFGS-B**	**TNC**	**COBYLA**	**SLSQP**	**Nelder -Mead**	**Powell**	**BFGS**
Profile 1	6591.66	**6298.82**	6421.64	10926.69	7485.87	6698.55	6750.58
Profile 2	1033.18	525.69	508.07	546.19	**490.56**	1168.20	1168.21
Profile 3	**1772.56**	**1772.56**	1781.71	4712.48	2416.77	1946.07	1946.07
Profile 4	**18306.65**	**18306.65**	**18306.65**	18458.90	18469.47	18345.50	18345.50

profiles with the highest consumption (*1* and *3*), there are renewable assets other than landfill. This shows that when only shape is considered, landfill gas is the most versatile renewable asset. The reason for this is that the random component inherent to generation is much lower than in wind. Also, profiles 1 and 3 are the only ones that include solar assets. The reason for this can be that only one solar asset data is available, and adding this profile would add a generation which is too high even for peakload consumption. With a wider range of solar generation assets or the possibility of buying just a part of generation (not considered in this case study) the results would change and peakload would be compensated better.

Table 4.12 shows the RMSE of different solvers used for optimisation. For the purpose of this problem, the result is picked with the smallest RMSE.

4.3.1 Findings summary

Development of Bayesian model for long-term wind generation forecast and stochastic model for long-term solar generation forecast and a linear programming-based method for optimal combination of renewable assets with and without consideration of costs per Kwh, with no weather data available.

4.4 AHU linkage using BMS correlations

4.4.1 Experiments 1 and 2: Manufacturing facility (Ground floor)

The building used in the case study is a large facility that consists of office spaces on the one hand, and manufacturing halls on the other hand. The office spaces contain meeting rooms and common spaces such as canteen, kitchen, changing room and open spaces. The manufacturing part comprises several dedicated spaces as this is an automotive manufacturing plant. In terms of cooling, heating and ventilation systems, the building has 3 chillers, 6 boilers, 6 fan coil units and 16 multi-speed fan AHUs. As the focus of this work are AHU systems, the actual linkage between AHU-space, control strategy and number of sensors can be seen in Table 4.13. Some of the AHUs have been excluded due to the lack of sensors in their respective areas of influence, which makes them irrelevant to this study.

Table 4.13: AHUs of the manufacturing facility, their associated areas of influence and the number of temperature sensors in such areas. AHUs' temperatures are controlled by the average value of the temperature sensors located in their respective areas. Data has temporal resolution of 10 minutes. Excluded AHUs, AHU06 and AHU15 have also shown.

AHU	**Associated zone**	**Control strategy**	**N. of temp. sensors**
AHU01	lab rooms	Avg. room temp - fixed set point	4
AHU02	support room	Avg. room temp - fixed set point	3
AHU03	wax room	Avg. room temp - fixed set point	4
AHU04	shell room	Avg. room temp - fixed set point	4
AHU05	NPI room	Avg. room temp - fixed set point	3
AHU06	clean room	AHU Return temp - fixed set point	0
AHU07A/7B	foundry area	Avg. room temp - fixed set point	8
AHU09/10	finish room	Avg. room temp - fixed set point	4
AHU11	inspection/X-Ray rooms	PI control loop - variable set point	5
AHU12	canteen	Avg. room temp - fixed set point	1
AHU14	pre-fire room	Avg. room temp - fixed set point	2
AHU15	fresh air make-up unit	Fixed supply temp	0
AHU16	shell rooms	Avg. room temp - fixed set point	2

Our experiments are based on Table 4.13 that contains "ground truth" for the case study. The data has been obtained from the manufacturer for every AHU individually (*AHU Specification (Barkell)*). Technical specifications include dimensions of boxes and specifications of motors, fans, thermal wheel (if applicable), coils, etc. Every sensor is tagged with a specific name describing the company and followed by building location and subset (Ventilation, Metering, Cooling, Heating, Globals, Terminals and Lighting).

The building used for this case study is a Rolls Royce plant located in Rotherham, with 1639 BMS points in total. The considered period is June to July 2018 (30 days). The facility comprises two main areas: the production plant on the ground floor and the offices on the upper floor.

Time series data provided by sensor points is not often reliable, as there are data gaps, unnecessary units, and anomalies, (data subsets misplaced between points because of extraction). The selected data points are:

- **AHU Supply Air Temperature (SAT) points:** These points measure the temperature of the air supplied to the area. There is one of these points per AHU.

- **AHU Return Air Temperature (RAT) points:** These points measure the temperature of the air extracted from the space prior to re-circulation or disposal.
- **Room temperature points:** Sensors are located in each room to measure temperature. There are between one and four sensors located in each room, and measurements is the average of the sensor values.
- **Fan Coil Unit (FCU) room temperature:** Output temperature of FCUs. This piece of equipment is connected (or close) an AHU.

4.4.2 Experiments 3: Office spaces (First floor)

The office spaces contain the meeting rooms and common spaces such as restaurant, canteen, kitchen, changing room and open spaces. A schematic is shown in Figure 4.24. In this case study, FCUs feed air to specific meeting rooms in the office above the manufacturing facilities.

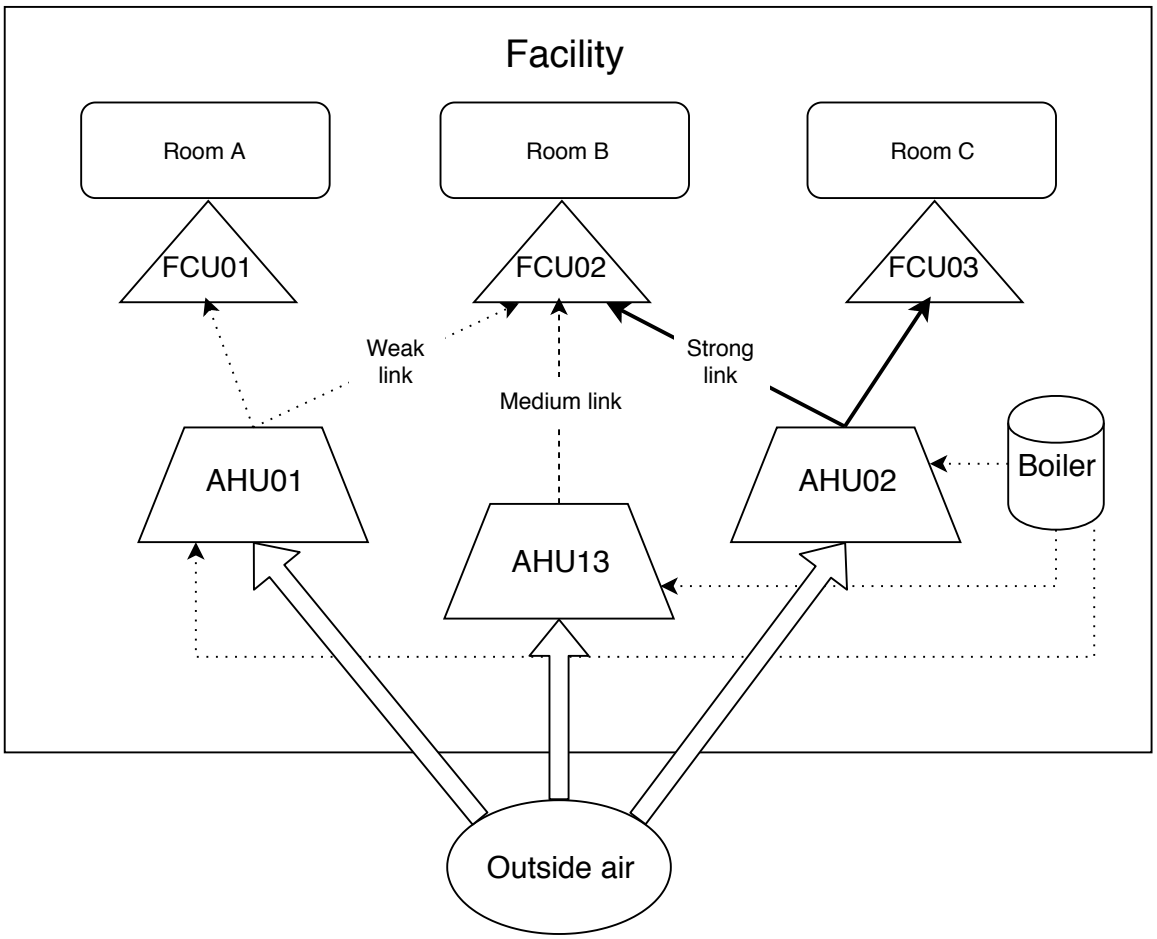


Figure 4.24: Schematic of AHUs and FCUs linkage

Meeting rooms have the following generic names: *Rhenium*, *Tugsten*, *Nickel*, *Cobalt*, *Tantalum*, and the gym. The main goal of this experiment is to link the respective rooms, each being fed by a different FCU, with their closest AHU unit. The strength of these links can be described by the Person's correlation coefficient, its high values showing a strong, very likely correlation, medium showing a moderate strength and low a weak

link.

Various experiments are performed to relate AHUs with their respective work spaces, as well as FCUs with their respective AHUs. In experiment 1, a comparison of the performance for different clustering techniques is performed in a controlled experiment consisting of only a small part of the facility. Then the best performing methodologies from experiment 1 are used in experiment 2 for the whole facility. Experiment 3 shows a different clustering focus, linking a different piece of equipment, fan coil units, with their respective areas of influence.

The concept of difference in temperature is introduced and a comparison of different clustering techniques is performed in a controlled experiment consisting of only a small part of the facility. Then the best performing methodologies from the prior experiment are used for the whole facility. After this, the next experiment has a different clustering focus, linking a different piece of equipment and fan coil units, with their respective areas of influence.

4.4.3 Linkage significance and difference in temperature

In order to test if the correlations are significant, the following is done. If the link between room sensors and SAT is established, the internal difference in temperature is to take into consideration. All spaces in the facility may generate heat internally (kitchen, manufacturing process, people, computer equipment, etc.) which could make it difficult to identify the respective AHUs. This is why this internal heat disruption is measured in the first place, and then gradually add complexity to the clustering.

For this purpose, the difference between supply and return air temperature has been taken into account,

$$\Delta T = |SAT - RAT| \ (^{\circ}C). \tag{4.1}$$

Table 4.14 represents the mean, variance, minimum and maximum value of the element-wise difference between supply and return air temperatures for each AHU.

Table 4.14: Mean, variance, minimum and maximum values of the difference bettween supply and return air temperature (C^o)

AHU number	Mean	Variance	Min Value	Max Value
AHU01	2.39	7.25	0.00	11.30
AHU02	6.20	14.37	0.01	13.45
AHU03	8.01	0.90	0.48	14.62
AHU04	2.87	0.97	0.02	8.47
AHU05	3.74	0.63	0.04	10.89
AHU06	3.97	0.75	0.06	6.20
AHU07A	17.16	1.06	12.92	20.18
AHU07B	15.49	1.13	12.28	18.34
AHU09	10.37	1.40	8.23	14.86
AHU10	10.12	1.59	6.69	13.80
AHU11	2.28	0.98	0.00	4.99
AHU12	3.50	5.50	0.00	10.69
AHU14	8.09	17.38	0.00	15.61
AHU16	1.04	0.56	0.00	7.90

Table 4.14 shows that some AHUs, such as 7A, 7B, 9 and 10, present a much higher mean value of their difference in supply and return temperatures. However it can be seen that the variance in some of them is not very high, meaning that the high mean difference in temperature is stable. This probably happens in an area which has been constantly influenced by external factors. On the other hand, areas with a relatively low mean and with a high variance can be found, meaning that the physical space presents temperature disruptions very often.

These values of difference in temperature are to be used as a measure as to how much the defined AHU linkage can be trusted. Therefore a limit for the mean difference in temperature as $7.5C^o$ is set, above which the cluster is considered as not reliable as the internal heat disruption may be too high. The reason for choosing this value is that the clustering algorithms fail in linking AHUs with sensors above an approximate value of $8C^o$ in experiment 2.

4.4.4 Linking three AHUs to five different rooms

For the first experiment, three AHUs have been selected. According to Table 4.14, the ones with a relatively low mean and variance are chosen i.e. the ones with a lower internal difference in temperature. Also it is necessary to validate how good the solution is, so the site engineers are consulted on the room names to which the AHUs are supplying air to. The chosen AHUs and corresponding rooms are:

- **AHU05** supplying air to the **"NPI room"**
- **AHU11** supplying air to the **"Visual inspection"**, **"Manual inspection"** and **"X Ray"** rooms
- **AHU16** supplying air to the **"Shell drying"** room

The techniques discussed are applied in the first experiment and later, these results are compared with the real connections. In a properly clustered group, the AHU's SAT

Table 4.15: Lasso clustering results

	AHU	Room temperature sensors
Cluster 1:	**AHU05 Supply Air Temp**	Room NPI Room Temp No3 Room NPI Room Temp No2 Room NPI Room Temp No.1
Cluster 2:	**AHU16 Supply Air Temp**	Drying Shell Drying Cell Temp No2 Drying Shell Drying Cell Temp No1
Cluster 3:	**AHU11 Supply Air Temp**	Room Manual Inspection Room Temp Room Manual Inspection Room Temp.1 Room Visual Inspection Room Temp Room Visual Inspection Room Temp.1 Ray X Ray Room Temp

Table 4.16: Distance between clusters considered for creating a boundary for eac2h AHC metric.

Distance metrics	**Correlations**	**DTW**	**IPD**
Clusters distance limit	1.4	10000	150

should be in the same group together with their corresponding space temperature sensors. The clustering dendrograms correspond to the AHC clustering technique with three distance metrics. AHC can be represented as a dendrogram because the algorithm progressively separates the clusters based on the distance metrics until all the points form a separate cluster. For this reason, the lasso clustering methodology is represented separately in Table 4.15, because in this algorithm the search for the optimal penalisation parameter is done in an iteratively refined grid.

For the experiments, it is assumed that it is known that the BMS points belong to different AHU systems and room temperature sensors. What it is assumed not to know beforehand is which of the 13 AHUs are associated with which of the 11 physical spaces. The parent-child relationship is known once the AHU has been associated with its corresponding physical space.

In the dendrograms, the distance chosen determines which branches below that distance form a cluster. The same distances are used later in the experiment with the whole building evaluation. An ideal cluster should contain the AHU sensors together with their respective temperature sensors within the same space. Figures 4.25, 4.26 and 4.27 show the clustering results for different distance metrics.

The fact that the AHUs' SAT belongs to the same cluster means in the case of AHC that, in the lowest levels of the dendrogram, the time series are closer (in terms of the chosen distance) to each other, and form groups that are more distant from each other as the branches go up. In the case of graphical lasso, it means that the elements corresponding to the estimated inverse of the covariance matrix are zero between clusters, thus forming groups of time series that present the most similar number of common features.

Figure 4.25 shows that correlations-based AHC clusters AHU05 and AHU11 correctly with their corresponding temperature sensors. AHU16 is in a cluster together with one of

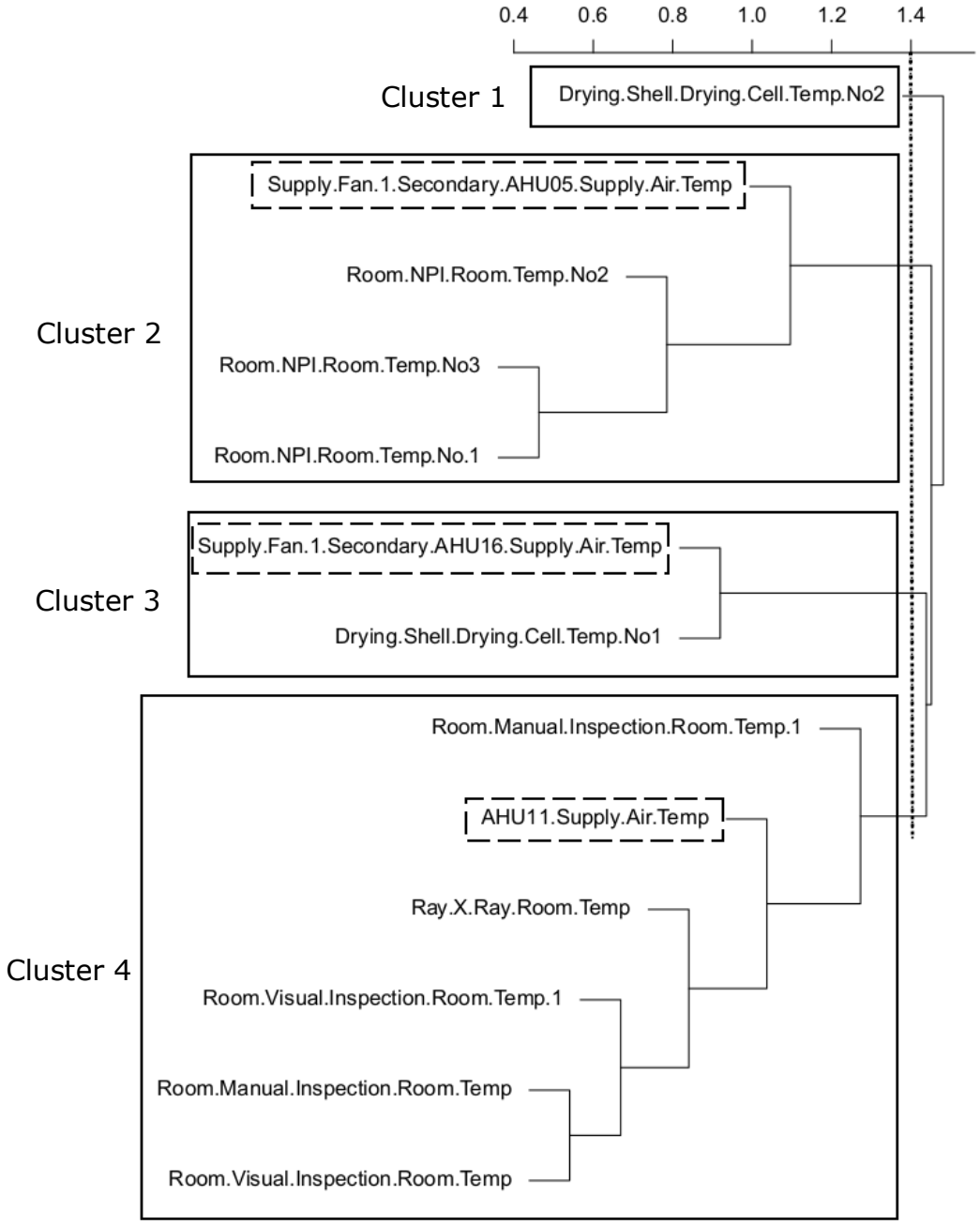


Figure 4.25: Experiment 1: Correlations-based AHC with 3 AHUs. The numbers above the dendrogram indicate the incremental distance between clusters according to this metric. The dashed box indicates the supply air temperature of the respective AHU. The other points correspond to space temperature sensors.

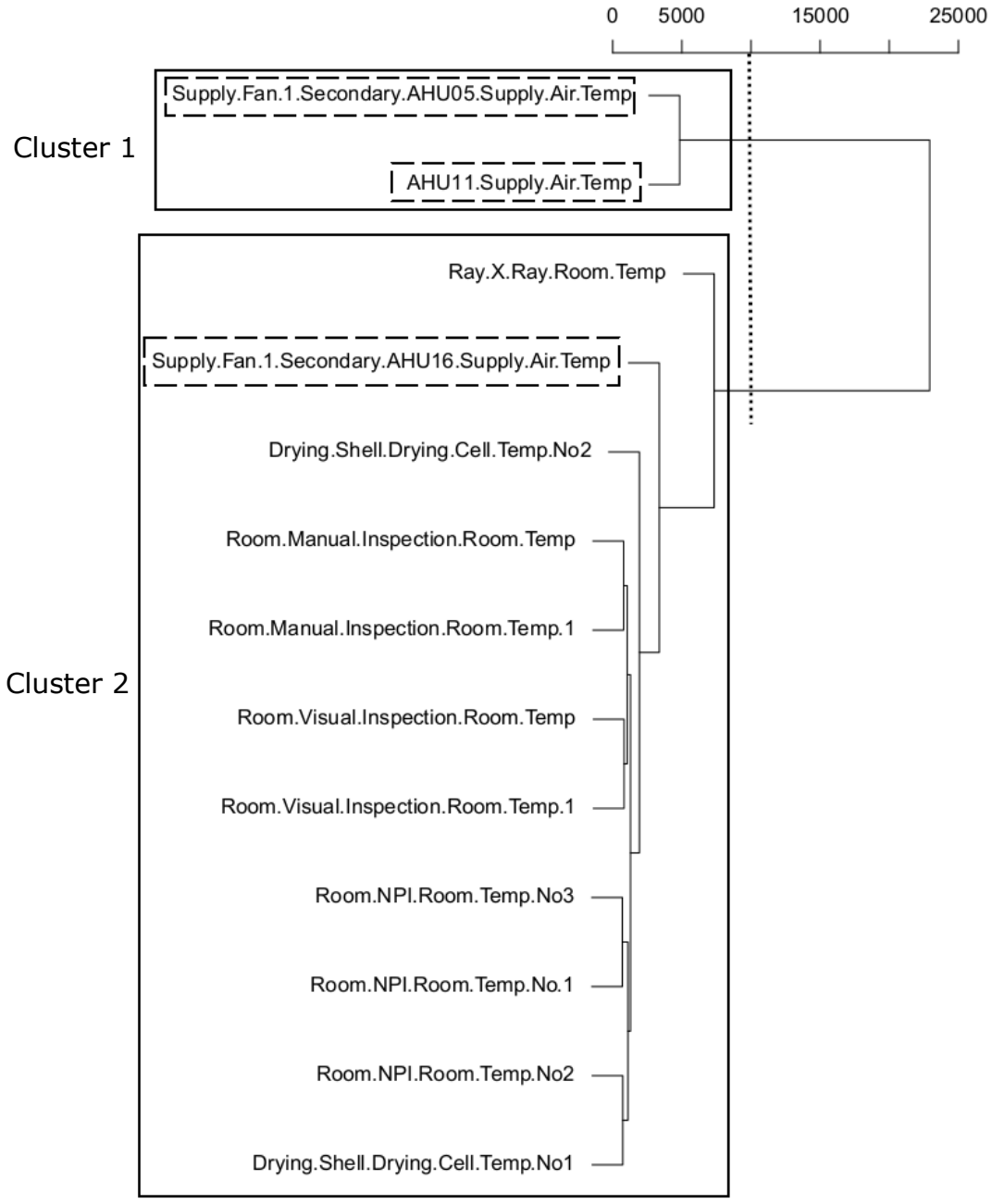


Figure 4.26: Experiment 1: DTW-based AHC with 3 AHUs. The numbers above the dendrogram indicate the incremental distance between clusters according to this metric. The dashed box indicates the supply air temperature of the respective AHU. The other points correspond to space temperature sensors.

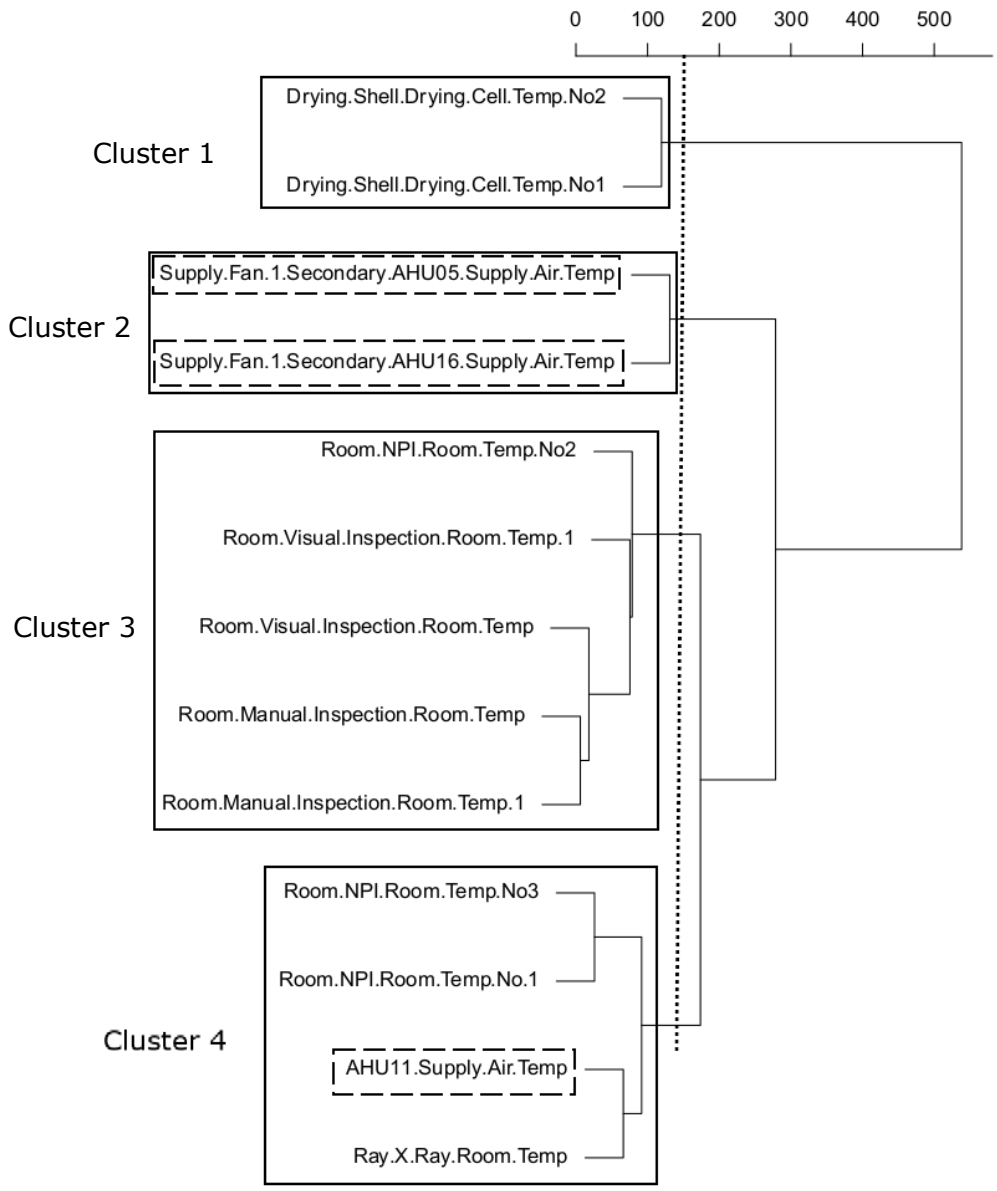


Figure 4.27: Experiment 1: IPD-based AHC with of 3 AHUs. The numbers above the dendrogram indicate the incremental distance between clusters according to this metric. The dashed box indicates the supply air temperature of the respective AHU. The other points correspond to space temperature sensors.

Table 4.17: Success rate for every clustering methodology. Total % of success rate has been chosen in relation to the total number of aggregated sensors correctly clustered for all three AHUs.

	Correlations AHC	**DTW AHC**	**IPD AHC**	**Lasso**
AHU 05	3/3 (100%)	0/3 (0%)	0/3 (0%)	3/3 (100%)
AHU 11	5/5 (100%)	0/5 (0%)	1/5 (20%)	5/5 (100%)
AHU 16	1/2 (50%)	2/2 (100%)	0/2 (0%)	2/2 (100%)
% success rate	**90%**	**20%**	**10%**	**100%**

its sensors, however the other sensor is excluded, thus forming a separate cluster. Figure 4.26 shows the same clustering methodology but based on DTW. It can be seen that the clusters do not respond to the logic of the physical connections. Similarly, in Figure 4.27 the clusters do not separate correctly either. Table 4.15 shows the results of lasso clustering. In this case, all the clusters are correct. The three AHUs are properly linked to their respective temperature sensors within the same cluster.

Results have been summarised in Table 4.17. The percentages are obtained based on the rate of room temperature sensors contained within the cluster with their corresponding AHU each. For instance, correlations AHC clusters elements properly within AHUs 05 and 11 but missed one of the two elements in cluster containing AHU16. The one showing the best results is the graphical lasso clustering, which correctly groups AHUs with their respective areas of influence. This performance is followed by correlations clustering.

Now that the techniques that perform best have been identified to be correlations-based AHC and Lasso clustering, they are used for the next experiment.

4.4.5 Linking all AHUs to rooms (whole building evaluation)

In experiment 2, the two best performing techniques discussed in section 4.4.4, Pearson's distance-based AHC and lasso clustering are used, with all AHUs and all temperature sensors within the building. In the results of the Pearson's correlation distance-based AHC shown in Figure B.1, correct clusters are considered by using the same distance to analyse the clusters as in experiment 1. As an example for correct cluster, Figure B.1 shows that AHU01 is contained within the same branch levels as the four lab rooms, as expected. Similarly, AHU07A and AHU07B are clustered with the *Foundry* area. Other areas such as *Finish* rooms, *Wax* rooms and the canteen, they are in different clusters with respect to their AHUs. Interestingly, it is observed that the related rooms are usually found together in the lowest level of the branches. Another observation is that AHUs that share common spaces are also clustered together, such as AHU07A and AHU07B. Results for lasso clustering table have been included in Appendix B. In the results of lasso clustering in Table B.1, same as in experiment 1, AHUs and room temperature sensors are found within the same cluster. Some of the clusters only group room temperature sensors, but no AHU SAT is present, as is the case of clusters 2, 5 and 9. On the other hand, clusters with only AHUs and no temperature sensors are observed in cluster 11. AHU06 is discarded for the lack of sensors in the room.

As defined in section 4.4.3, a limit value of $7.5C^o$ is set for the difference between SAT and RAT. In Table 4.18 the obtained results are summarised. The table presents

the average difference in temperature, confirmation/denial of linkage in the appropriate category and the issue associated with the AHU if any.

- **Issue (a)**: Open plan space. Heat exchange occurring between adjacent rooms.
- **Issue (b)**: The AHU is enabled and provides 100% cooling nearly constantly but is having no effect on setpoint (the temperature value set) or is not enough to cool to the set-point. This implies the AHU is not mechanically sound or capable to meet requirements (onsite investigation may confirm this).
- **Issue (c)**: Heat gain is too high. Issues reported with heat coming from internal sources: kitchen, servers, crowded areas, etc.

Table 4.18: Summarised results of experiment 2. Total % of success rate has been chosen in relation to the total number of aggregated sensors correctly clustered for all three AHUs. Rows with bold font numbers represent AHUs whose mean ΔT is above the limit.

AHU	**Mean ΔT**	**Associated room**	**Success rate AHC (Correlations)**	**Success rate Graphical lasso**	**Reported issue**
AHU01	2.39	Lab rooms	4/4 (100%)	4/4 (100%)	N/A
AHU02	6.20	Support room	3/3 (100%)	3/3 (100%)	N/A
AHU03	**8.01**	Wax room	**0/4 (0%)**	**4/4 (100%)**	N/A
AHU04	2.87	Shell room	4/4 (100%)	0/4 (0%)	N/A
AHU05	3.74	NPI room	3/3 (100%)	3/3 (100%)	N/A
AHU07A/AHU07B	**17.16/15.49**	Foundry area	**8/8 (100%)**	**1/8 (12.5%)**	(a)
AHU09/AHU10	**10.37/10.12**	Finish room	**0/4 (0%)**	**0/4 (0%)**	(b)
AHU11	2.28	Inspection/X-ray rooms	5/5 (100%)	5/5 (100%)	N/A
AHU12	3.50	Canteen	0/1 (0%)	0/1 (0%)	(c)
AHU14	**8.09**	Pre-fire room	**0/2 (0%)**	**0/2 (0%)**	N/A
AHU16	1.04	Drying rooms	1/2 (50%)	1/2 (50%)	N/A
% Sensors clustered correctly			**70%**	**52.5%**	

Table 4.18 shows the results of this experiment. The performance of both methodologies is very similar, except that lasso fails to cluster wax rooms with their respective temperature sensors. In general, it can be said that correlations-based AHC performs better than graphical lasso in terms of the number of sensors belonging to the correct cluster. 70% reveals that this methodology identifies underlying connections between AHUs and their respective spaces. Also, it can be observed that the AHUs with a mean ΔT above the set-up limit fail to predict the matches between AHUs and physical spaces in general, although exceptions can be seen in graphical lasso and AHU03. AHUs 07 A and B are in the same branch as their respective sensors in the AHC technique. AHUs 2 and 12 have a very high variance (as shown in Table 4.14), which does not seem to affect the performance of the algorithms in the case of AHU 2. For AHU 12, both methods fail to predict its only sensor with the canteen. When looking at its variance, it seems to be higher than other AHUs, as the canteen is crowded mainly during lunch time. This, together with the fact that its only sensor may be misplaced, could explain this issue.

The intention was to test the fact that the difference in temperature has determinant effect on obtaining these relationships. In general this relationship proves correct but there are sometimes exemptions. In the case of AHU07 A & B and despite being an open space, correlations-based AHC has been able to properly identify all 8 sensors in this more challenging case. Therefore it can be concluded that this fact cannot be confirmed or the experiment is insufficient.

4.4.6 Linking AHUs with FCUs

The main goal of this experiment is to link the respective rooms, each being fed by a different FCU, with their closest AHU unit. These relationships are shown per FCU, and the links of these with all (if any) of the correlated AHUs. This is summarised in Table 4.19.

Table 4.19: AHU and FCUs linkage. Numerical association with correlation and the degree of strength defined by flag category. Bold numbers represent the highest correlation value encountered per space, which is linked to a specific AHU. The linkage has been defined according to the value of the coefficient as *low* (0-40),*medium* (40-70) and *high* (70-100).

AHU	FCU Linked	Pearson's correlation coefficient	Linkage
AHU01	Nickel	0.68	Medium
	Tugsten	0.34	Low
	Gym	**0.29**	Low
AHU02	Nickel	**0.89**	High
	Cobalt	0.51	Medium
	Tugsten	**0.51**	Medium
	Rhenium	**0.48**	Medium
	Tantalum	0.05	Low
AHU04	Cobalt	0.14	Low
	Nickel		
AHU07A	Tugsten	0.25	Low
	Nickel	0.24	
	Gym	0.06	
AHU07B	Nickel	0.20	Low
	Tugsten	0.14	
	Gym	0.06	
AHU09	Nickel	0.02	Low
	Gym	0.01	
AHU10	Nickel	0.07	Low
	Tugsten	0.04	
	Gym	0.02	
AHU11	Cobalt	0.06	Low
	Rhenium	0.02	
AHU13	Cobalt	**0.55**	Medium
	Rhenium	0.45	Medium
	Tantalum	**0.16**	Low
AHU16	Cobalt	0.08	Low
	Tantalum	0.03	
	Rhenium	0.02	

It is concluded from correlations that the most influential AHUs to the FCUs are AHU 01, 02 and 13. After validation with the engineers, it has been found out that AHU02 is the one that has the highest influence over these areas. Therefore, some other potential dependencies are uncovered from the supply air temperatures of AHUs 01 and 13 to some of the rooms (*Nickel*, *Tugsten* and *Gym* for AHU01 and *Cobalt*, *Rhenium* and *Tantalum* for AHU13).

4.4.7 Findings summary

Methods for obtaining relationships between HVAC assets and their respective areas of influence in large facilities through clustering time series analysis from temperature sensor data. Results show a 70% of correct association of AHUs with their respective areas of operation in a very large facility (16 AHUs) and a correct FCU-room association by using correlations.

4.5 Early warning signals

The goal of this part of the thesis is to apply EWS techniques for early failure detection in Building Management Systems.

Our variable of interest is the supply air temperature (SAT) of a particular zone of the building in the manufacturing plant. SAT is controlled by the average room temperature measured by four sensors installed in the zone using the control signal of pre-established fixed setpoint. This means that target temperature is set to a determined value. Average room temperature is then used by the system to regulate the percentage of openness of heating or cooling valves, according to the setpoint. Another factor that influences the temperature of the room is the Outside Air Temperature (OAT). These four variables are represented in Figure 4.28, where the time 0, denotes the empirical onset of the failure. In the x-axis, time zero indicates the moment of the failure. Negative and positive values are therefore, the moments before and after the failure. It has been designed like this so the anticipation of the signal with respect to the failure can be easily visualised.

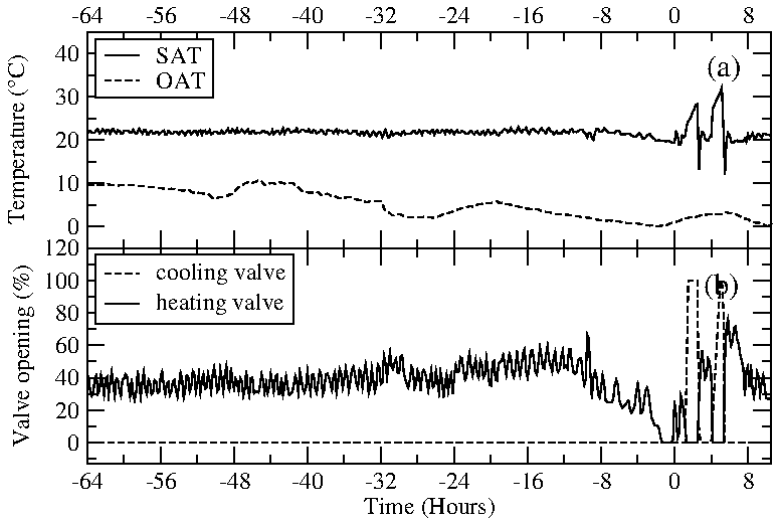


Figure 4.28: Sensor variables affecting room temperature: (a) Supply Air Temperature and Outside Air Temperature in degrees Celsius. (b) Percentage of opening for each of the heating and cooling valves. Figure showns data data from 5:30AM of the 25^{th} of January 2019 starting 64 hours prior to the failure.

The system operates as follows:

- Outside Air Temperature controls frost coil valve. Frost coil valve protects the Air Handling Unit (AHU) from very low temperatures: the parts contain liquid elements, and they can contract and damage the AHU.
- Supply Air Temperature controls heating and cooling valves. As this is a winter period, the cooling valve will be closed most of the time, but during anomalous overheating it activates and provides cooling to the environment.

First, univariate analysis of the SAT is conducted by using different indicators introduced in Section 3.4.1, and then other variables are taken into account, reducing the system dimensionality and applying the same techniques to compare both approaches and identify which one provides an EWS at the earliest time.

4.5.1 Univariate analysis

For the purpose of this study, univariate analysis is performed only with SAT, then multivariate analysis is performed with all variables controlling SAT by reducing the dimensionality before applying these techniques.

Results for univariate analysis are presented in Figure 4.29, where SAT is presented together with the EWS indicators. The plot shows data from 64 hours prior to the failure at moment 0, which is when the temperature increases anomalously for the first time. Different windows are used to obtain every EWS: 22 hours for variance and ACF1, 8 hours for PS and 4 hours for VAERE. The window size has been selected according to the clarity of the signal they provided on each indicator. Y-axis of the variance has been represented in logarithmic scale, for convenience.

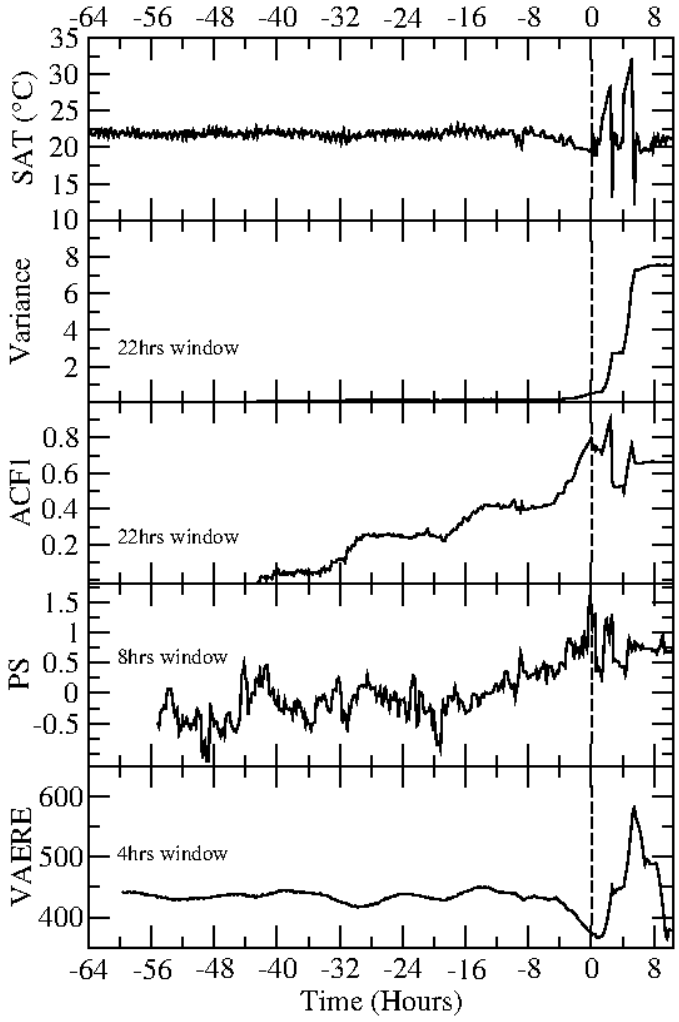


Figure 4.29: EWS of supply air temperature with four indicators: variance, autocorrelation function, power spectrum and variational autoencoder. The moment of the failure (when the supply air temperature increases anomalously for the first time) is 9:30AM of the 27^{th} of January 2019. Plot starts at 5:30AM of the 25^{th} of January, 64 hours prior to the failure. Window sizes used: 22 hours for variance and ACF1, 8 hours for PS and 4 hours for VAERE, as these sizes were found to work best when applying these methodologies. Data resolution is 10 minutes.

As shown in Figure 4.29, ACF1 and variance are the indicators presenting the most

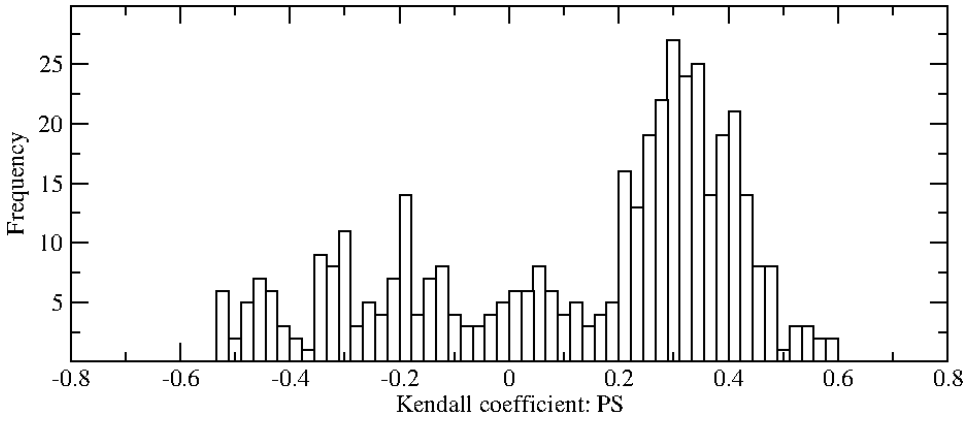


Figure 4.30: Histogram of the frequency distribution of the Kendall trend statistic for the PS indicator for the univariate case, for the different windows across the indicator.

clear signals. It can be observed that PS also increases before the failure occurs, although it gives a signal with a shorter period. VAE has been first trained with the system functioning under normal conditions. The output displayed shows the VAERE. This seems to show a drop right before the tipping point, but less than 4 hours prior to event.

As can be seen in Figure 4.30, Kendalls show a more robust positive trend than negative, as the majority of values are greater than zero. However, some negative values can be seen as oscillations occur. PS has been the only methodology chosen for this purpose, as it is the indicator presenting the most chaotic behaviour.

4.5.2 Multivariate analysis

Multivariate analysis for EWS is shown in Figure 4.31. The plot on the top corresponds to OAT, cooling valve and heating valve reduced with PCA to their first principal component or direction to the minimum projection variance. The windows used for each indicator for multivariate analysis are: 22 hours for variance and ACF1, 16 hours for PS and 20 hours for VAERE.

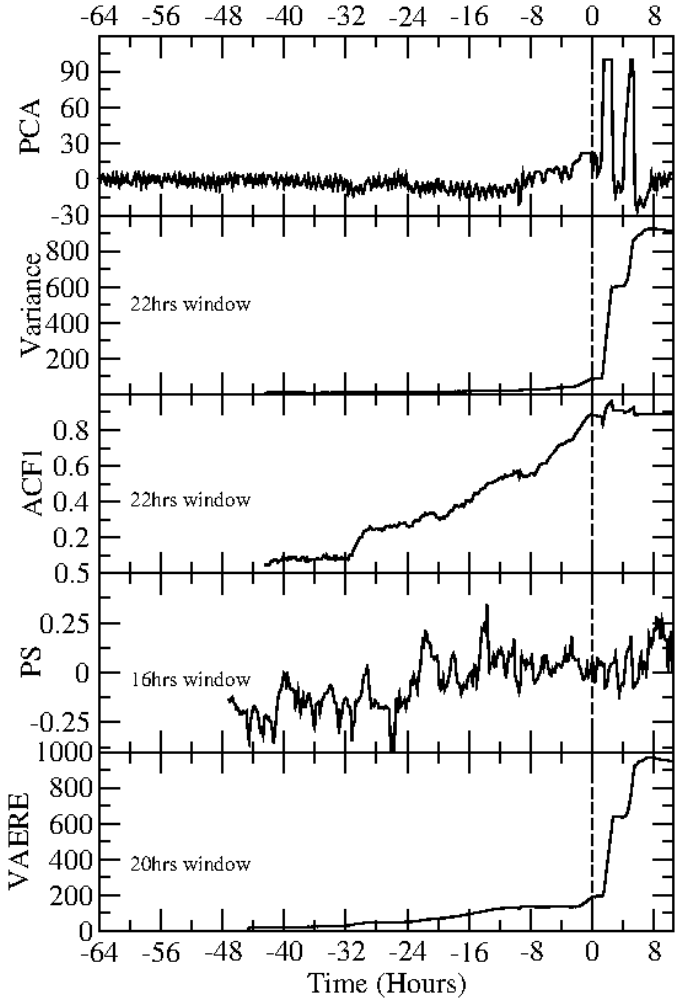


Figure 4.31: Multivariate EWS analysis of the temperature failure with PCA and four indicators: variance, auto-correlation function, power spectrum and variational autoencoder. The moment of the failure (when the supply air temperature increases anomalously for the first time) is 9:30AM of the 27^{th} of January 2019. Plot starts at 5:30AM of the 25^{th} of January, 64 hours prior to the failure. Window sizes used: 22 hours for variance and ACF1, 16 hours for PS and 20 hours for VAERE, as these sizes were found to work best when applying these methodologies. Data resolution is 10 minutes.

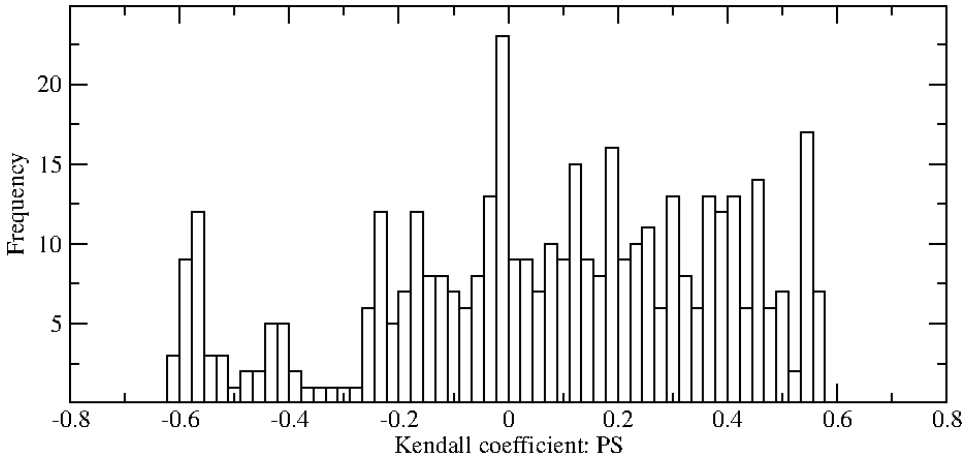


Figure 4.32: Histogram of the frequency distribution of the Kendall trend statistic for the PS indicator for the multivariate case, for the different windows across the indicator.

Comparing this plot with the univariate one, it is observed that ACF1 and variance present clear EWS signals, whereas the variance represents a more steady slope in the multivariate case. PS shows a more prompt signal in the multivariate case, 24 hours prior to failure in comparison to 8 hours prior to the failure in the univariate analysis. However, values of the Kendalls coefficient incline a bit more towards negative values than in the univariate case, although still positive values are predominant. This means that PS indicator is less robust in the multivariate case. An earlier signal is also given by the VAE, whose indicator for the multivariate signal gives a clear EWS after 32 hours before the event, in comparison to the 4 hours prior to failure of the univariate case.

4.5.3 Findings summary

Development of early warning signals for temperature sensor data that detects anomalies up to 32 hours prior to failure, considering both the SAT for the univariate case and all variables affecting SAT for the multivariate case, and without considering the control loop for this temperature sensor.

4.6 Machine learning for BMS tag classification

The existing infrastructure uses various naming conventions for sensors and equipment of buildings using labels that are given at the moment of the installation for brief description of the component's type, location, parent-relationship, etc. For example, *Boiler 1 Temp Sensor* describes a sensor that measures the temperature of the water in boiler number 1. There are several problems related to this descriptive naming system:

- **There exists no naming standard.** This complicates tagging when mobilising several buildings to the same analytics platform. Most of these analytics use naming conventions to create general rules that apply to all buildings.
- **Some tags are incomplete.** The labels are created to give a short description. This means that some words would appear much shortened. Such as *Temp* instead of *Temperature* or *Grnd flr* instead of *Ground floor*.
- **Duplicates.** If the building is large enough, with a lot of equipment, there can be duplicates in labels. This is due, for example, to upgrades of the equipment that may occur when renewing older systems, or if new areas are built.

The goal of this part of the thesis is to automatically create tags, defined by the Haystack specification (Quan et al., 2003), based on the information provided from BMS. A tag is a name/value pair applied to an entity (sites, equipment, sensor points, etc.) following the Haystack standard, as shown in Haystack (2019), which is an open source initiative that standardises semantic data models to make easier data handling. Their applications include automation, control, energy, lighting, HVAC and other BMS applications. A tag defines a property or attribute of a BMS unit. Some elements are already tagged by default, whereas others need to be tagged according to some data criteria.

There are three main groups on which to enclose every tag considered for this case study: *Point tags*, *Service type* and *Equip tags*. These are the descriptions of every main category according to Haystack (ibid.), from the lowest to the highest level of concision:

- *Service type*: Used to classify the labels into eight main categories, namely *cooling*, *heating*, *lighting*, *ventilation*, *metering*, *monitoring*, *terminals* and *globals*.
- *Equip tags*: Refers to equipment type. Equipment is often a physical asset such as an AHU, boiler or chiller. These tags can also refer to logical grouping such as a chiller plant. There are a total of 26 *equip tags* including category $'0'$ (when a point does not belong to any of them). Each label must belong to one category at least. There are never more than two categories assigned to it.
- *Point tags*: It refers to a lower level abstraction of the labels. There are 36 main types of *Point tags* in the dataset and they represent the most complex classification part, as there can be many tags refering to one label and even its manual assignation is difficult.

Tagging stage 1: *Point Tags* classification

First the focus lies on the multi-label classification problem concerning *point tags* classification. The focus on this tag types is due to the complexity of the problem. Then, a second experiment is carried out using the rest of the main categories with the most successful technique, from the lowest to the highest level of complexity, to provide with a complete solution of the tagging problem, as well as an overall accuracy result.

The data is extracted from different versions of the same Trend BMS controller type (TREND, 2018). The raw data is used for the purpose of training/testing the algorithms. An example of the extracted data is shown in Table 4.20:

Table 4.20: Excerpt of BMS raw data. Type, units and interval refer to characteristics of the data, whereas outstation and module refer to the characteristics of the BMS.

Label	Type	Outstation	Module	Units	Interval
IL4-6 Damper	Boolean	15	D11(Sv)	None	300
AHU2 Low Temp Hold Off SP	Numeric	12	K1(V)	°C	3600
Extact Fan28 Override	Boolean	15	W3(S)	None	3600
AHU Heating Coil	Numeric	17	D2(Sv)	None	300
		...			

Table 4.21: Example of tagged data (other categories have been removed from the table for simplicity)

Label	Point tags
AHU1 Dampers	sensor, damper, recirc
AHU1 Frost Stat	sensor, valve, frost
AHU1 Max Supply Temp	sp, temp, air, discharge, oneA
UPS Rm Fire Sys Fault	sensor, alarm
...	...

The tags to assign to each row of data are defined by the Haystack specification (Haystack, 2019), and they are separated into several categories for different purposes. Some of the examples of these tagging categories are shown in Table 4.21.

All points are classified as *sensors*, *commands* and *setpoints (sp)* using one of the following three tags:

- **sensor**: input, analogue/digital input, sensor
- **cmd**: output, analogue/digital output, actuator, command
- **sp**: setpoint, internal control variable, schedule

This work uses BMS controller data, according to the real system deployed at the Mitie company (U.K.). The dataset contains data from 37 buildings, of which 36 have been used for training and 1 for testing.

Tagging part 2: All categories

For this part, all three categories are used to define the problem: *Service type*, *equip tags* and *point tags*. These three separated problems, each connected to the next. Because of this, the problem is solved with an increased level of complexity. With a label as an input, the algorithm has to perform three stages of classification to identify correct types and tags. The first problem classifies the labels according to *service type*, the output of this problem is used to separate the data in these groups, which are then treated as different problems for *equip tags* classification. The outputs of each problem are then used as predictors for *point tags* classification, after which the final result is obtained.

An example of data classification can be seen below in Table 4.22:

Table 4.22: Example of tagging problem with all categories

Label	Service type	Equip tags	Point tags
CHW Pump 1 Enable	Cooling	cooling pump	sensor, run
East VT Valve	Heating	vtHeating	cmd, heat
HWS TEMP SETPOINT	Heating	boiler	sp, leaving, temp, water
Space cooling setpoint	Terminals	fcu	sp, air, cool, temp, zone
...	...	...	...

4.6.1 *Point Tags* classification

Process overview

First, the data is pre-processed and sparse matrices are concatenated before the two-step model classification. The process is shown in Figure 4.33.

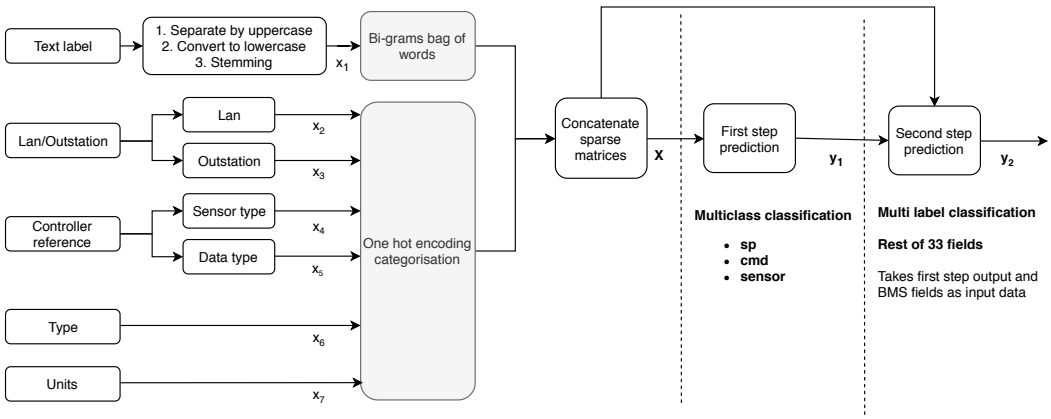


Figure 4.33: BMS label classification for pre-processing data and predicting categories.

One of the biggest challenges in BMS data pre-processing is that the default names of the sensors (text labels) are introduced manually. Therefore many typos, acronyms, groups of words written together separated by upper case, etc can be found. The text label is first separated by upper case letters, taking into consideration those which contain acronyms. Then these are converted to lower case. Next, stemming is applied, which is the process of reducing words to their stem. This refers to the roots of the words known as lemma. With this, the classification algorithm is more likely to capture similarities. Then, the result is categorised with bi-grams (groups of two words) bag of words (Zhang et al., 2010).

Lan and *outstation* fields can be found together, so they can be separated before categorisation. Same with controller reference, such fields contain sensor type information (I/O such as switches, temperature sensors, etc.) and data type (static, variable, etc.).

After that, all the fields with their respective vector representations are concatenated in a matrix (in which most elements are zeros), whose rows serve as inputs for the first step. The first step aims to predict only *sp*, *cmd* or *sensor*, as all labels always belong to one of these three categories. Second step aims to predict the rest of the labels by using the first step prediction, so extra information (plus the input data) is added for better accuracy.

Text classification results

The results of the experiments have been presented in two tables: Table 4.23 presents the train and test accuracy for the first and second classification steps. The algorithm counts an element as correctly tagged when 100% true positives and 100% true negatives per label are obtained. Table 4.24 presents the individual accuracy per tag type for a sample of the eight first tags and per method used.

Table 4.23: Text classification results of applying five techniques and their % of absolute and relative accuracy. Bold font highlights the best performing methodology in the test set.

	LogReg	RndForests	XGBoost	MultinomialNB	LinearSVC
			First Step		
Train	99.80	99.99	99.78	98.75	99.78
Test	99.73	99.07	**99.84**	87.69	99.50
			Second Step		
Train	83.75	96.34	81.43	55.84	85.21
Test	**89.01**	87.14	88.68	58.24	87.64

Table 4.24: Results of applying five techniques to the individual tags, % of accuracy per tag type. Bold font denotes the highest score for each tag type.

Tag	LogReg	RndForests	XGBoost	MultinomialNB	LinearSVC
air	98.68	98.24	97.52	**98.85**	97.91
alarm	**99.18**	98.90	98.68	96.04	99.07
chilled	99.89	99.94	**100.0**	88.52	99.84
co2	99.56	99.50	**99.89**	90.66	99.84
cool	99.89	**100.0**	99.89	98.74	**100.0**
damper	**99.57**	99.56	99.50	90.44	99.56
discharge	98.52	**98.90**	98.79	94.40	98.57
enable	**100.0**	**100.0**	**100.0**	95.99	**100.0**

Results in Table 4.23 show that for the experiments performed with our data, xgboost algorithm provides the best result for test accuracy, closely followed by logistic regression and linear SVC. The best accuracy achieved at the second step, however, is by logistic regression, followed by xgboost and linear SVC.

Results per tag type in Table 4.24 show that the accuracy per tag type varies with every different method. In fact, it can be observed that every method outperforms on at least one predicted tag.

Assessment of errors

The class probability for each prediction is considered, which is the probability for each label of belonging to a certain class, to calculate the confidence of the prediction and to discard all the elements below a certain boundary. Gneiting et al. (2007) provided summary measures for the evaluation of probabilistic forecasts, by assigning a numerical score based on the predictive distribution. The probabilities of our system do not seem to follow a clearly defined distribution, as shown in Figure 4.34.

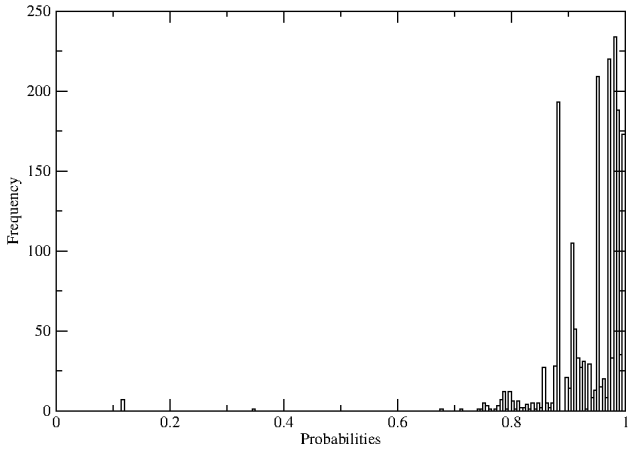


Figure 4.34: Prediction probabilities histogram of test set.

Therefore, it has been decided to create a different metric to assess and discard values based on the prediction confidence. Prediction confidence consists of re-formulating each probability, so that a probability around 0.5 results in a confidence around 0. For each prediction probability, obtained as the algorithm's output, P_i, the confidence score is defined, C_i, calculated as shown in Eq. 4.2:

$$C_i = 2 \cdot |P_i - 0.5| \,. \tag{4.2}$$

For simplicity, the all-tags averaged confidence per label is calculated. The aim of this is to filter the values by how strong the choice of the algorithm is, therefore the value of the confidence around a prediction probability of 0.5, will have a confidence value close to zero, but a probability close to either 0 or 1 will result in a confidence value close to 1. The resulting chart with all the confidence values of the test set can be seen in Figure 4.35, in which a boundary of 0.85 has been set.

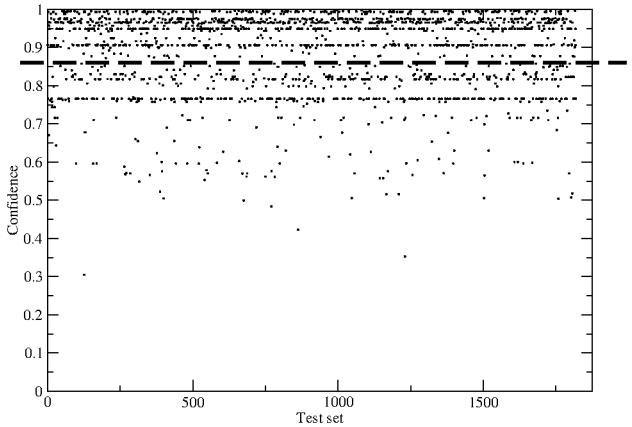


Figure 4.35: Prediction confidence of test set calculated according to Eq 4.2. The dashed line represents the boundary, currently set to 0.85.

With the chosen boundary of 0.85, the results can be found in Table 4.25. Each dot of the dataset represents a particular confidence score for that particular label. The choice of the boundary is related to the strength of the decisions. A lower boundary value implies a lower value of true and false negatives, but a higher value of true and false positives.

Table 4.25: Percentages of true/false positives/negatives with respect to the total length of the test set, 1875 labels.

	Positive	**Negative**
True	83.74%	3.95%
False	8.52%	3.79%

True positives are the labels that pass the boundary and whose classification is correct, true negatives are the ones that did not pass the filter because they are predicted with low confidence, but the classification is incorrect, false positives are the elements with an incorrect classification but that are not detected because they are predicted with high confidence and false negatives are the elements whose prediction is correct but with low confidence overall. As shown in Table 4.25, 83.74% of the test set is correctly put in the category of good predictions, whereas a 3.95% of the test set is correctly identified as misclassified elements.

4.6.2 All categories evaluation

The complete text classification problem scheme has been illustrated in Figure 4.36. XGBoost methodology has been used in all prediction stages based on the previous experiment.

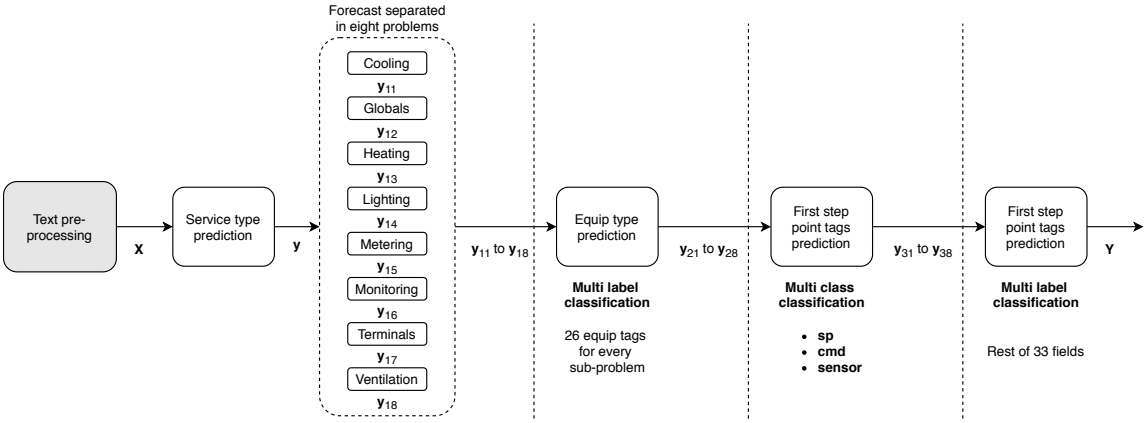


Figure 4.36: BMS label classification of multi-stage process.

As can be seen, the predictions start from the same pre-processing as the previous experiment. Then the first prediction is performed for *service type*. As the results obtained for both train and test sets are above 95%, the output is sub-divided into the eight service type categories that serve individually as training inputs for the following step. The reason for doing this is to improve the chances of success for further category predictions. For example, a predicted *equip tag* of boiler would come from a heating *service type* for sure. Therefore there is no need of training the *equip tag* problem according to other service categories. Then, every individual result is moved to *point tag* predictions as it was done in the previous experiment.

The results for every prediction stage can be seen in Table 4.26.

Table 4.26: Percentages of true/false positives/negatives in text classification with respect to the total length of the test set for all predictions, 1875 labels.

	Service type		Equip tags		Point tags	
	Positive	**Negative**	**Positive**	**Negative**	**Positive**	**Negative**
True	93.85%	0.99%	92.37%	1.59%	90.29%	2.08%
False	1.09%	4.06%	4.96%	1.09%	5.45%	2.18%

The percentage of true positives decreases as the complexity of the forecast increases. Also, the number of false positives increases, as every stage inherits the errors from the previous one.

The final rate of true positives of the output is 90.29%. In comparison with the previous experiment, it can be seen that the accuracy in this one is higher. This means that sub-dividing the problem into the eight predicted *service type* categories is favourable to the accuracy of the solution.

4.6.3 Findings summary

The findings of this part include an automatic method for translating building sensor points into a haystack naming standard that uses multi-stage text classification and that achieves a 90% of true positives. Also, the addition of a confidence formula to create a boundary for detecting false positives has been developed.

Chapter 5

Conclusions and further work

The goal of this thesis is to develop a series of solutions for the built environment: from mechanisms to regulate electricity prices to automatic solutions with BMS data involving sensors clustering, early warning signals analysis of sensor failure and automatic BMS text classification.

5.1 Conclusions

Key Findings:

- Development of a deep learning system for detecting and forecasting events of high energy demand at a national level for general DSR interventions. With $RRMSE = 2.23\%$ achieved in LSTM and being able to predict Triads with a fixed level of risk.
- Development of methods for long-term wind and solar generation forecast and a linear programming-based method for optimal combination of renewable assets with no weather data available.
- Methods for obtaining relationships between HVAC assets and their respective areas of influence in large facilities through clustering time series analysis from temperature sensor data. Results show a 70% of correct association of Air Handling Units (AHUs) with their respective areas of operation in a large facility with 16 AHUs and a correct Fan Coil Unit (FCU)-room association.
- Development of early warning signals for temperature sensor data that detects anomalies up to 32 hours prior to failure.
- An automatic method for translating building sensor points into a haystack naming standard that uses multi-stage text classification and that achieves a 90% of true positives. Also, the addition of a confidence formula to create a boundary for detecting false positives has been developed.

5.1.1 DSR events forecasting

A system for load forecasting of DSR events is designed, either long or short term, depending on the DSR intervention performed. The model is composed of two steps: Load forecasting and highest peaks extraction with respect to the latest n days. The goal is to forecast all the three Triad peaks with the least possible number of calls in order to reduce the number of DSR interventions. The LSTM model is calibrated and its performance is compared with ANN, SVM, random forests, Bayesian regression and the mean-only model. This demonstrates that LSTM outperforms other models, and that its

performance is closely followed by Bayesian regression. It is shown that, over the 4 years of testing, 11 peaks are forecast in total, showing that the number of signals for the soft and hard filters are, respectively, 19 and 14 for the winter of 2014, 21 and 19 for the winter of 2015, 19 and 14 for winter of 2016 period and 21 and 16 for the winter of 2017.

Once a Triad signal is positive, then as much equipment as possible is switched off and generators are run, consuming fuel for every DSR signal. The factors chosen for the filters are to be defined by the user, in this case, 3.5% and 4% (soft and hard filter respectively). This defines the level of risk that the company, building manager or DSR manager want to take. The risk assessment would determine the number of signals that the organisation can afford in terms of fuel/disruption, and the risk of missing the Triad, which is subject to a cost.

The energy system layout may change in the future, as can be seen in the re-distribution of transmission losses per region according to the P350 amendment approved on the 24th of March 2017 (Grid, 2018). This could affect the way the forecast performs, and may lead to a correction factor for a better forecasting, as well as factors re-calibration. The most limiting factor in the current system design is data availability.

Due to recent changes in energy systems, it is necessary to focus on more generalised methodologies that offer a certain degree of flexibility in order to be adapted to DSR interventions. This may lead to further developments in the area of more flexible forecast for different long/short term DSR scenarios.

The future energy systems will require either nonlinear growth of infrastructures, which is not sustainable, or wider-scale, smart interventions which are agile, low-cost and reduce carbon emissions. The UK energy market presents a set of DSR interventions which are economically grounded and of high potential of implementation in other countries with similar demand for energy, without large investments into infrastructure. This makes modeling and forecasting of DSR programmes of high relevance to international energy markets. The modeling approach introduced is concise, accurate, computationally light and flexible for further tuning, according to market and risk management requirements.

5.1.2 PPA and renewable energy generation forecasting

In this part of the thesis, two techniques are applied, Bayesian and MCMC, to estimate wind energy generation: one for detecting change points in the data using beta distribution, and another for seasonal period separation. A stochastic model is developed to forecast solar generation. In order to find the optimal combination of renewable assets with respect to a particular consumption profile, linear programming is used to minimise costs and fulfill the required electricity demand. The optimal combination of intermittent output of renewable projects could support energy buyers, balancing-related costs of the PPA and last, but not least, create a basis which can reduce some balancing-related tautology costs of the buyer's import utility contract.

This analysis shows that, despite the intermittent behavior that makes wind forecast a very challenging task, a good forecast is achieved for one year horizon. By breaking the problem down into several sub-periods, with different parameters of beta distribution, a forecast could be achieved, whose histograms were good based on the comparison of their posterior with respect to their actual distribution. A different approach was followed for solar generation forecasting. The optimal combination of several probability

densities provides a decision on which farms to choose for a particular consumption profile.

This work is a simplified version of a real case scenario. In normal PPA negotiations, many projects would be offered from several generators with different price structures. The length of these contracts would lie between five and ten years instead of one. However, this model still serves as a basis for real case scenarios, as they would use the same forecasting principles that can be extrapolated to a longer forecasting horizon and a higher number of assets.

5.1.3 BMS sensors clustering

A solution is proposed to obtain equipment relationships based on data correlations and clustering that relate AHUs to their respective areas of operation. This is an important industrial challenge that requires a smart solution.

Graphical lasso and correlations-based AHC are the two best performing techniques in the BMS experiments. In the first experiment, lasso clustering proved slightly better than correlations distance-based AHC. In the second experiment, however, their performance is quite similar, with few differences on *Wax* and *Shell* rooms but with the same number of successfully clustered elements. Experiment 3 shows satisfactory results as the physical spaces linked to the FCUs correspond to the real case. This approach allows to quickly scan the system, as one AHU can be connected to more than one FCU, being in the same area of influence in the case they are not directly attached.

This methodology can be generalised to large buildings with a similar problem. The system could potentially input labeled time series data corresponding to sensors and output the clusters. Although the accuracy is not 100%, this data-driven approach removes weeks of engineering visits to the facility to establish these relationships manually and looking at documentation. By applying this methodology, it becomes quite straightforward to establish a first step towards more advanced analytics or to determine a standard naming system that encompasses parent-child relationships.

This top-down perspective aims not to use internal parameters of AHUs and other BMS information, but instead their direct output. Systems internal information is less likely to be available in comparison to the room temperature sensors which are commonly installed, but the hardware connections are not documented. The most recent evolution in smart buildings implies installation of a significant amount of sensors in large facilities, so this creates the necessity of generating real value from the data through data mining. Time series clustering can be used on a daily basis by the site engineers who need to trace back faults without the need of an extensive installation information on that particular building.

5.1.4 EWS for BMS failures

Several early warning signal techniques are applied to analyse BMS temperature sensor data and percentage of openness of heating and cooling valves to activate heating and/or cooling systems, respectively, in a certain area of a large industrial facility. The analysis included univariate and multivariate EWS, using the following indicators: variance, ACF, PS and VAE. PCA has been used for dimensionality reduction in the multivariate case.

The analysis shows that, in general, the indicators provide an earlier and more reliable signal in the multivariate case. This improvement can be seen especially in PS and VAE. In the case of PS, the difference is that the indicator changes its pattern 24 hours prior to failure. With the VAE, the reconstruction error does not reproduce the gradual increase before the failure in the univariate case despite of the abrupt temperature change. In the multivariate case, the VAE does show the expected behaviour, producing an early warning signal almost 32 hours prior to failure. The reason that the VAE does not provide a good signal in the univariate case may lay in what is considered a normal functioning system when training the model using only SAT. Such distribution makes small changes due to external factors, which actually control its non-stationary behaviour. In the multivariate case, the behavior of the system is defined mainly by the heating valve, which steadily opens and closes to provided heating to the environment within an established range or set point.

The early detection of such failures gives time to on-site engineers to make adjustments when necessary before these failures actually happen. This not only reduces maintenance and operational costs, but also produces energy savings by advising when parts of the system should not be activated in some given period, thus compensating the ”blind spots” from the BMS control system, and extracting real value from the data generated.

5.1.5 BMS label classification

A two-stage text classification of BMS labels is performed. The results show that xgboost performs better than the other four methods, but the others make good candidates for this stage too, except maybe for multinomial Naive Bayes, which shows slightly worse results. The outperformer in the second stage classification, the multi-label problem, is logistic regression. The top performers that follows are xgboost algorithm. Again, the Naive Bayes method performs the worst of the five. The accuracy per tag type shows that certain algorithms may be better in predicting certain tags than others. In this part of the work, xgboost and logistic regression have been considered to design the system, but the aim for further work will be a combination of methods for the second stage, using each method for doing only the classifications they are best at, to improve the general accuracy of the whole implementation. Sub-dividing the problem into several problems improves its accuracy for the whole system, as expected.

In terms of the model's deployment, the assessment of errors is very important. The main problem are the false positive results. The false positives are the incorrectly tagged elements that passed to the building analytics software. These elements may be difficult to detect, especially for buildings with a big number of points. Increasing the confidence boundary may help solve this problem and reduce false positives to a minimum. This also may reduce the number of true positives, increasing the amount of manual work.

Our research provides a novel machine learning solution for the real-world BMS, which can be applied in several systems, or even re-trained with new requirements that could appear in the future.

5.2 Further work

In load forecasting for DSR events, the developed model produces a satisfactory load forecast at the national level. Although this model captures the demand trend, it does not consider indoor physical factors, such as occupancy, internal system's efficiency, which

may require more power from the grid in the case of older infrastructures. The future research may include charging electric vehicles in the consumption patterns, as well as varying electricity prices, as those have an impact on the electricity generation patterns. Also, due to the satisfactory results produced by Bayesian regression, further work may include this methodology for comparison with LSTM if similar data is used.

The main limitation of the PPA optimisation problem is related to data availability and quality. Two years of data could be obtained for every wind farm, and three years for the solar farm. To test this model by using a longer forecasting horizon, especially for wind, would have been more ideal. Also, regarding data availability, only data from one solar farm could be obtained, so repeating the experiment with a higher number of solar farms would give a wider perspective of solar energy market. As can be seen in the results for wind simulations, satisfactory results are achieved by breaking the problem down to five switchpoints. However for further research in this area, it would be interesting to create even more granularity to consider individual estimations and finding an optimal number of switchpoints, preventing overfitting. Also, as the novelty of this work is to perform an estimation of wind and solar power in the very long term, the comparison of these results with other models leaves room for further development.

Text classification also leaves room for further development. In this problem, DNN architectures have not been tested due to the need of large datasets with a large corpus to extract features with a high accuracy. This does not mean that a specific DNN architecture modification cannot be arranged to produce similar or better results, therefore there could be potential for improvement if more data is provided.

It can be said that data availability and quality have been the two major issues encountered. This would make somehow difficult the implementation of the methods, as their data extraction was not trivial. For the BMS clustering, the lack of official documentation made the verification of the results time consuming (with on-site engineers). Similar for the BMS tagging problem, as the verification of the results needed constant feedback from control engineers in order to know if the results obtained from new imported buildings were correct (as tagging data is obviously not available for new buildings). Also, as the goal is to compare some of the most popular text classification algorithms in the literature for very short text classification, there is room for comparison with deep learning models for this specific type of text in the future. In the case for EWS applications to BMS system, lack of data and difficulty of automation acquired a different perspective. Here, the problem lies, not only in the automation of the data acquisition part, but also in the location of specific failures in order to test the algorithms. Further work to improve these results would imply an improvement of software solutions for data extraction, as well as the location of new failures and case studies to further improve and generalise these results.

Publications/conferences list

Publications

- J. J. Mesa-Jimenez, L. Stokes, C. Moss, Q. Yang, and V. N. Livina *"Modelling Energy Demand Response Using Long-Short Term Memory Neural Networks"*, Springer, Energy efficiency (2020) (DOI: 10.1007/s12053-020-09879-z)
- J. J. Mesa-Jimenez, L. Stokes, Q. Yang, and V. N. Livina *"Machine learning for BMS analysis and optimisation"*, IOP, Engineering Research Express (2020) (DOI: 10.1088/2631-8695/abbb85)
- J. J. Mesa-Jimenez, L. Stokes, Q. Yang, and V. N. Livina *"Machine Learning for Text Classification in Building Management Systems"*, Elsevier, Journal of Building Engineering (Under review)
- J. J. Mesa-Jimenez, L. Stokes, Q. Yang, and V. N. Livina *"Early Warning Signals of Failures in Building Management Systems"*, Springer, Applied Intelligence (Under review)
- J. J. Mesa-Jimenez, A. Tzianoumis, L. Stokes, Q. Yang, and V. N. Livina *"Long-Term Wind and Solar Energy Generation and Optimisation of Power Purchase Agreements"*, Springer, Energy Systems (Under review)

Conferences

- Postgraduate Institute for measurement science (PGI) conference. Sustaining our environment. Title: *Technologies in the built environment to reduce electricity consumption.* Teddington, London, UK - November 06, 2019.
- Institute for Research Development, Training and Advice (IRDTA). DeepLearn 2019. Title *3rd International Summer School on Deep Learning.* Warsaw, Poland - July 22-26, 2019.

Appendix A

Appendix. Triad software

The main purpose of this application is to generate signals for Triad, DSR in other words, to output the half hour periods when a Triad is more likely to happen. Internally, the system is composed of a forecasting model and two boundaries. The forecasting model determines the value of next day's national electricity demand, and the boundaries provide a simple method, that call Triads or no-Triads (1 or 0, respectively) if the value of the forecast demand is above/below the respective boundary.

A.1 Initial set-up

Settings.ini, as well as dbo.LSTM_ModelParameters contain all the variables that can be modified to run the algorithm. This is something that will have to be set up at the beginning, with none or few changes in the future.

Then, *EMALoader.py* is run. It is the file that loads the first sets of data to perform the initialization of the exponential moving averages. This is done just once, after adjusting the parameters that are to be obtained from the DataBase. This file will need to be run initially once, but there may be readjustments after changing parameters from settings.ini, then it may be run later again.

TrainingLoader.py has been built with the purpose of extracting data for the daily training but some of this data, for example first to the final reconciliation demand data, is not available in the API of ELEXON currently, so it has been built for the hypothetical case that this data could be available in the future.

Training data has been extracted manually and added to dbo.UK_INDO_DemandData. which contains the training data for the next whole season. The file contains the Training.csv data, in case the training data needs to be loaded again.

Figure A.1 summarises the procedure to follow in case of data base initialisation.

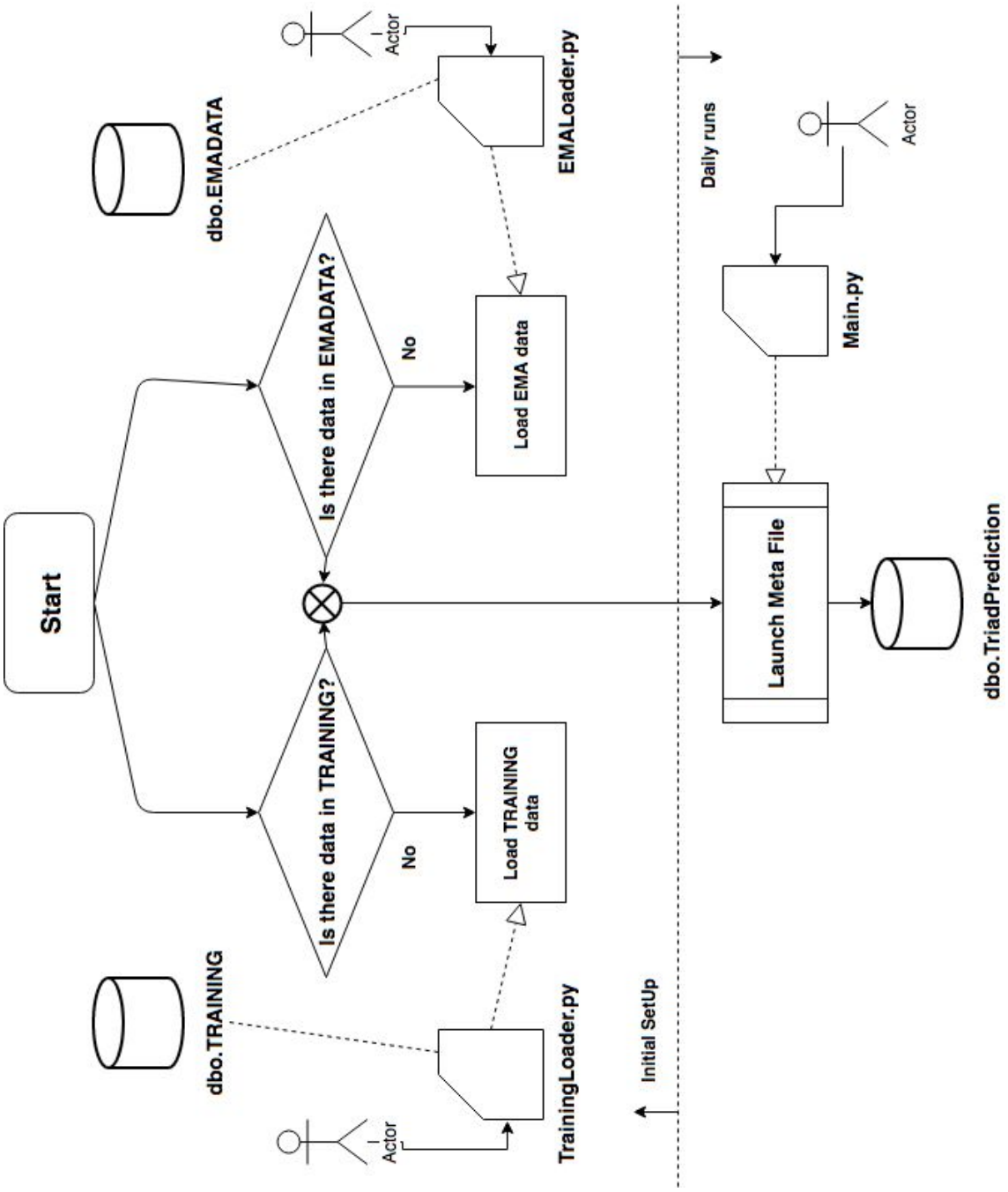


Figure A.1: Schematic of the Triad software.

A.2 List of files and daily production of results

The TRIAD software is expected to run every day at 8:00am by executing the *main.py* script, obtaining automatically the data from the API. *Main.py* is the only file that needs to be run every day, it either grabs the data to make predictions for today, or it takes data for doing the prediction on a specified day (in *settings.ini*). Software running period is from 1st November to 28th February (Triad season). From 23rd December to 2nd January the software will not run, due to Christmas period and the subsequent low demand.

- API_ELEXON.py: contains functions to extract data from the API.

- DB.py: contains the class that calls all the DataBase connection function.
- EMALoader.py: loads initial EMA data when launched .
- Main.py: file that performs the daily runs. It contains the main blocks of the daily working system.
- Parameters.py: obtains all the necessary parameters from both DBs and Settings.py.
- TrainingLoader.py: loads the data for the purpose of training the algorithm. (Not in use at the moment as this data is not available from the API).
- TRIADfunctions.py: contains adjacent functions used by other scripts.
- config.ini: configuration file. Contains the parameters that may be changed (DataBase Credentials, API Key and links, date of daily running).
- ReadMe file.

Appendix B

Appendix. Experiment 2: Whole building evaluation

Table B.1: Experiment 2. Lasso clustering results

	AHU in cluster	**Room temperature sensors in cluster**
Cluster 1:	**AHU02 Supply Air Temp**	Support Rm3 Temp Support Rm2 Temp Support Rm1 Temp Room Inspection Svcs Rm Temp1
Cluster 2:	(No AHU in cluster)	Foundry Room Temp4 Foundry Room Temp3 Foundry Room Temp2 Foundry Room Temp1 Foundry Room Temp 3 Foundry Room Temp 2 Foundry Room Temp 1
Cluster 3:	**AHU07A Supply Air Temp** **AHU07B Supply Air Temp**	Radiant Foundry Verification Rm Temp Canteen Room Temp Room Finish Temp 4 Room Finish Temp 3 Room Finish Temp 2 Room Finish Temp 1 Room Finish Room Temp 3 Room Finish Room Temp 2 Room Finish Room Temp 1 Room Finish Room Temp
Cluster 4:	**AHU11 Supply Air Temp**	Room Manual Inspection Room Temp Room Manual Inspection Room Temp.1 Room Visual Inspection Room Temp Room Visual Inspection Room Temp.1 Ray X Ray Room Temp
Cluster 5:	(No AHU in cluster)	Room Inspection Svcs Rm Temp4 Room Inspection Svcs Rm Temp3 Room Inspection Svcs Rm Temp2
Cluster 6:	**AHU10 Supply Air Temp** **AHU01 Supply Air Temp** **AHU14 Supply Air Temp**	Lab Scope Rm Temp Lab Polish Rm Temp Lab Mount Cutoff Rm Temp Room Lab Shell Rm Temp,
Cluster 7:	**AHU05 Supply Air Temp**	Room NPI Room Temp No3 Room NPI Room Temp No2 Room NPI Room Temp No.1
	AHU in cluster	**Room temperature sensors in cluster**
Cluster 8:	**AHU16 Supply Air Temp**	Drying Shell Drying Cell Temp No1
Cluster 9:	(No AHU in cluster)	Drying Shell Drying Cell Temp No2 Room Shell Room Temp No4 Room Shell Room Temp No3 Room Shell Room Temp No2 Room Shell Room Temp No1
Cluster 10:	**AHU03 Supply Air Temp** **AHU09 Supply Air Temp**	Room Inspection Svcs Rm Temp1 Room Wax Room Temp No4 Room Wax Room Temp No3 Room Wax Room Temp No2 Room Wax Room Temp No1
Cluster 11:	**AHU12 Supply Air Temp** **AHU04 Supply Air Temp**	

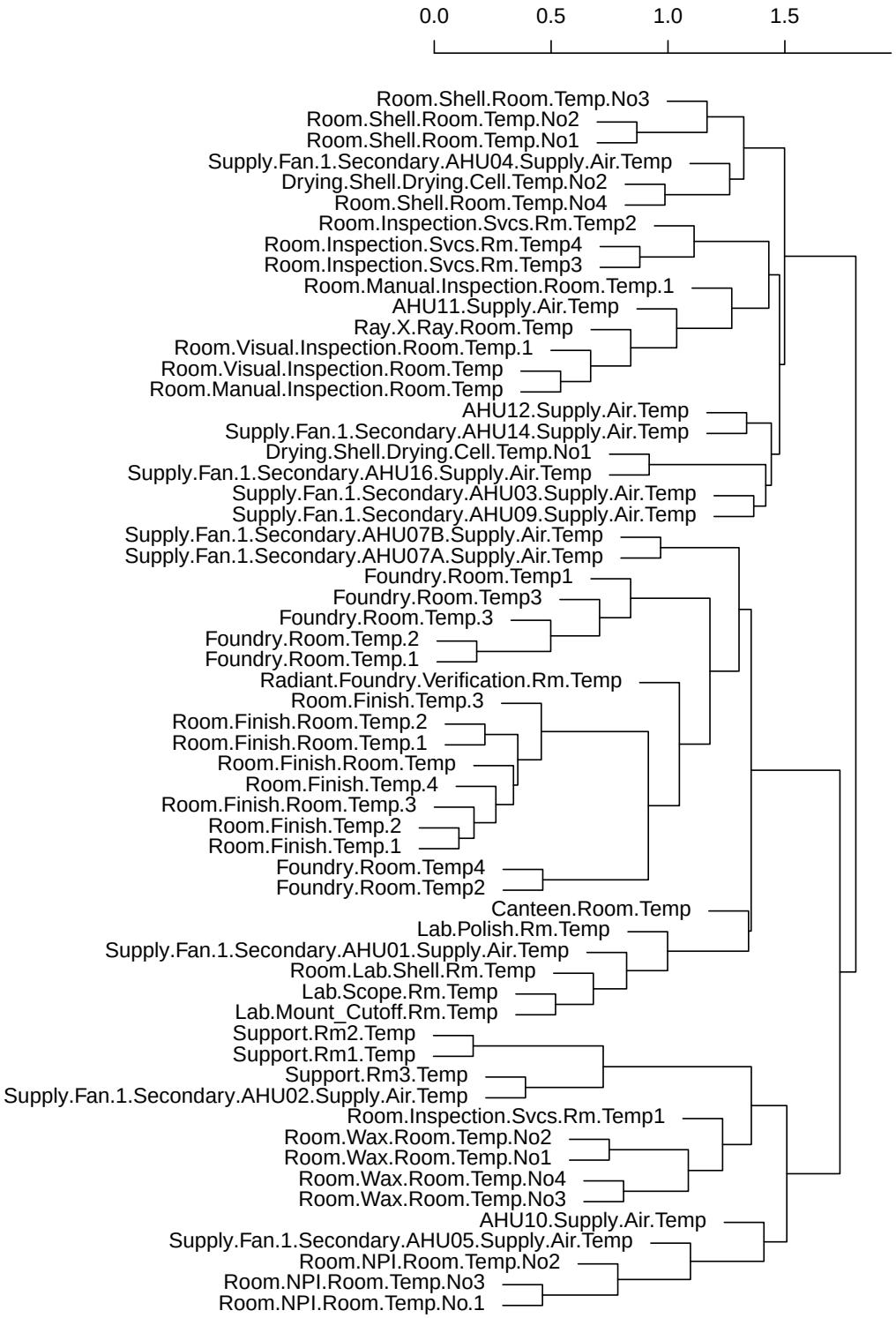


Figure B.1: Experiment 2: Correlations based AHC clustering with the complete set of AHUs in the facility. The number above indicates the incremental distance between clusters according to this metric.

Bibliography

Abad, L. et al. (2020). "A Short-Term Load Forecasting Algorithm Using Support Vector Regression & Artificial Neural Network Method (SVR-ANN)". In: *2020 11th IEEE Control and System Graduate Research Colloquium (ICSGRC)*. IEEE, pp. 138–143.

Abdel-Nasser, M. and K. Mahmoud (2019). "Accurate photovoltaic power forecasting models using deep LSTM-RNN". *Neural Computing and Applications* 31.7, pp. 2727–2740.

Abuella, M. and B. Chowdhury (2015). "Solar power forecasting using artificial neural networks". In: *2015 North American Power Symposium (NAPS)*. IEEE, pp. 1–5.

Aghabozorgi, S., A. S. Shirkhorshidi, and T. Y. Wah (2015). "Time-series clustering–A decade review". *Information Systems* 53, pp. 16–38.

Ahmad, W. et al. (2020). "Towards Short Term Electricity Load Forecasting Using Improved Support Vector Machine and Extreme Learning Machine". *Energies* 13.11, p. 2907.

AHU Specification (Barkell). (last accessed on 10/10/2018). URL: `https://www.barkell.co.uk/web/air-handling-units/ahu-specification.htm`.

Akçay, H. and T. Filik (2017). "Short-term wind speed forecasting by spectral analysis from long-term observations with missing values". *Applied Energy* 191, pp. 653–662.

Akinyelu, A. A. and A. O. Adewumi (2014). "Classification of phishing email using random forest machine learning technique". *Journal of Applied Mathematics* 2124.17.

Al-Musaylh, M. S. et al. (2018). "Short-term electricity demand forecasting with MARS, SVR and ARIMA models using aggregated demand data in Queensland, Australia". *Advanced Engineering Informatics* 35, pp. 1–16.

Ali, J. et al. (2012). "Random forests and decision trees". *International Journal of Computer Science Issues* 9.5, p. 272.

Alonso, A. M. and D. Peña (2019). "Clustering time series by linear dependency". *Statistics and Computing* 29.4, pp. 655–676.

Alsaleem, S. (2011). "Automated Arabic Text Categorization Using SVM and NB." *Int. Arab J. e-Technol.* 2.2, pp. 124–128.

Aly, H. H. (2020). "A proposed intelligent short-term load forecasting hybrid models of ANN, WNN and KF based on clustering techniques for smart grid". *Electric Power Systems Research* 182, p. 106191.

Andrews, J. T., E. J. Morton, and L. D. Griffin (2016). "Detecting anomalous data using auto-encoders". *International Journal of Machine Learning and Computing* 6.1, p. 21.

Arlot, S., A. Celisse, et al. (2010). "A survey of cross-validation procedures for model selection". *Statistics surveys* 4, pp. 40–79.

Azad, H. B., S. Mekhilef, and V. G. Ganapathy (2014). "Long-term wind speed forecasting and general pattern recognition using neural networks". *IEEE Transactions on Sustainable Energy* 5.2, pp. 546–553.

Bacher, P., H. Madsen, and H. A. Nielsen (2009). "Online short-term solar power forecasting". *Solar energy* 83.10, pp. 1772–1783.

Bak, P., C. Tang, and K. Wiesenfeld (1987). "Self-organized criticality: An explanation of the 1/f noise". *Physical review letters* 59.4, p. 381.

Balaji, B. et al. (2016). "Brick: Towards a unified metadata schema for buildings". In: *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*, pp. 41–50.

Bandara, K., C. Bergmeir, and S. Smyl (2020). "Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach". *Expert Systems with Applications* 140.

Barandiaran, I. (1998). "The random subspace method for constructing decision forests". *IEEE transactions on pattern analysis and machine intelligence* 20.8, pp. 1–22.

Barbounis, T. and J. B. Theocharis (2006). "Locally recurrent neural networks for long-term wind speed and power prediction". *Neurocomputing* 69.4-6, pp. 466–496.

Bayes, T. (1763). "XLIII. A letter from the late Reverend Mr. Thomas Bayes, FRS to John Canton, MA and FR S". *Philosophical Transactions of the Royal Society of London* 53, pp. 269–271.

Benesty, J. et al. (2009). "Pearson correlation coefficient". In: *Noise reduction in speech processing*. Springer, pp. 1–4.

Berndt, D. J. and J. Clifford (1994). "Using dynamic time warping to find patterns in time series." In: *KDD workshop*. Vol. 10. 16. Seattle, WA, pp. 359–370.

Bianchi, F. M. et al. (2017). "An overview and comparative analysis of recurrent neural networks for short term load forecasting". *arXiv preprint arXiv:1705.04378*.

Bode, G. et al. (2019). "A time series clustering approach for Building Automation and Control Systems". *Applied Energy* 238, pp. 1337–1345.

Boser, B. E., I. M. Guyon, and V. N. Vapnik (1992). "A training algorithm for optimal margin classifiers". In: *Proceedings of the fifth annual workshop on Computational learning theory*. ACM, pp. 144–152.

Bossanyi, E. (1985). "Short-term wind prediction using Kalman filters". *Wind Engineering* 9.1, pp. 1–8.

Bouktif, S. et al. (2020). "Multi-sequence LSTM-RNN deep learning and metaheuristics for electric load forecasting". *Energies* 13.2, p. 391.

Box, G. E. et al. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.

Bracale, A. and P. De Falco (2015). "An advanced bayesian method for short-term probabilistic forecasting of the generation of wind power". *Energies* 8.9, pp. 10293–10314.

Brindley, G. (2019). *How can corporates minimise risks and maximise opportunities when entering renewable PPAs?: RE-Source Platform*. `http://resource-platform.eu/news/risk-mitigation-for-corporate-renewable-ppas/`, (last accessed on 20/06/2020).

Brockwell, P. J., R. A. Davis, and S. E. Fienberg (1991). *Time Series: Theory and Methods: Theory and Methods*. Springer Science & Business Media.

Brown, P. F. et al. (1992). "Class-based n-gram models of natural language". *Computational linguistics* 18.4, pp. 467–479.

Caiado, J., N. Crato, and D. Peña (2006). "A periodogram-based metric for time series classification". *Computational Statistics & Data Analysis* 50.10, pp. 2668–2684.

Cao, X. et al. (2015). "A Data-Driven Hybrid Optimization Model for Short-Term Residential Load Forecasting". In: *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on*. IEEE, pp. 283–287.

Carli, R. et al. (2020). "Iot based architecture for model predictive control of hvac systems in smart buildings". *Sensors* 20.3, p. 781.

Carlo, C. M. (2004). "Markov chain Monte Carlo and Gibbs sampling". *Lecture notes for Biostatistical Analysis* 581.

Casado-Vara, R. et al. (2020). "IoT network slicing on virtual layers of homogeneous data for improved algorithm operation in smart buildings". *Future Generation Computer Systems* 102, pp. 965–977.

Chai, K. M. A., H. L. Chieu, H. T. Ng, et al. (2002). "Bayesian online classifiers for text classification and filtering". In: *Special Interest Group on Information Retrieval*. Vol. 2, pp. 97–104.

Chan, N. H., C. Y. Yau, and R.-M. Zhang (2014). "Group LASSO for structural break time series". *Journal of the American Statistical Association* 109.506, pp. 590–599.

Chatterjee, S., P. G. Jose, and D. Datta (2019). "Text Classification Using SVM Enhanced by Multithreading and CUDA." *International Journal of Modern Education & Computer Science* 11.1.

Chen, B.-J., M.-W. Chang, et al. (2004). "Load forecasting using support vector machines: A study on EUNITE competition 2001". *IEEE transactions on power systems* 19.4, pp. 1821–1830.

Chen, C. et al. (2011). "Online 24-h solar power forecasting based on weather type classification using artificial neural network". *Solar Energy* 85.11, pp. 2856–2870.

Chen, C. and H. Liu (2020). "Medium-term wind power forecasting based on multi-resolution multi-learner ensemble and adaptive model selection". *Energy Conversion and Management* 206, p. 112492.

Chen, S.-T., D. C. Yu, and A. R. Moghaddamjo (1992). "Weather sensitive short-term load forecasting using nonfully connected artificial neural network". *IEEE Transactions on Power Systems* 7.3, pp. 1098–1105.

Chen, T. and C. Guestrin (2016). "Xgboost: A scalable tree boosting system". In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, pp. 785–794.

Cheng, W. Y. et al. (2017). "Short-term wind forecast of a data assimilation/weather forecasting system with wind turbine anemometer measurement assimilation". *Renewable Energy* 107, pp. 340–351.

Conn, A. R., K. Scheinberg, and P. L. Toint (1997). "On the convergence of derivative-free methods for unconstrained optimization". *Approximation theory and optimization: tributes to MJD Powell*, pp. 83–108.

Constantinescu, E. et al. (2009). *Unit commitment with wind power generation: integrating wind forecast uncertainty and stochastic programming*. Tech. rep. Argonne National Lab.(ANL), Argonne, IL (United States).

Dalal, M. K. and M. A. Zaveri (2011). "Automatic text classification: a technical review". *International Journal of Computer Applications* 28.2, pp. 37–40.

Daut, M. A. M. et al. (2017). "An Improved Building Load Forecasting Method using a combined Least Square Support Vector Machine and modified Artificial Bee Colony". *ELEKTRIKA-Journal of Electrical Engineering* 16.1, pp. 1–5.

Davis, E. P. and D. Karim (2008). "Comparing early warning systems for banking crises". *Journal of Financial stability* 4.2, pp. 89–120.

De Paola, A. et al. (2020). "A fog-based hybrid intelligent system for energy saving in smart buildings". *Journal of Ambient Intelligence and Humanized Computing* 11.7, pp. 2793–2807.

Defays, D. (1977). "An efficient algorithm for a complete link method". *The Computer Journal* 20.4, pp. 364–366.

Devlin, J. et al. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv, v2, May 24*. Tech. rep. Accessed 2019-11-30.

Dey, M., S. P. Rana, and S. Dudley (2020a). "A case study based approach for remote fault detection using multi-level machine learning in a smart building". *Smart Cities* 3.2, pp. 401–419.

Dey, M., S. P. Rana, and S. Dudley (2020b). "Smart building creation in large scale HVAC environments through automated fault detection and diagnosis". *Future Generation Computer Systems* 108, pp. 950–966.

ELEXON (2018). *ELEXON web site, URL: https://www.elexon.co.uk/*. (last accessed on 20/12/2017).

Elnagar, A., R. Al-Debsi, and O. Einea (2020). "Arabic text classification using deep learning models". *Information Processing & Management* 57.1, p. 102121.

Faruqui, A. and S. Sergici (2010). "Household response to dynamic pricing of electricity: a survey of 15 experiments". *Journal of regulatory Economics* 38.2, pp. 193–225.

Feng, Y. et al. (2019). "Short term load forecasting of offshore oil field microgrids based on da-svm". *Energy Procedia* 158, pp. 2448–2455.

Fernandez, A. A. and J. L. Deneubourg (2011). "On following behaviour as a mechanism for collective movement". *Journal of theoretical biology* 284.1, pp. 7–15.

Filik, Ü. B., Ö. N. Gerek, and M. Kurban (2011). "A novel modeling approach for hourly forecasting of long-term electric energy demand". *Energy Conversion and Management* 52.1, pp. 199–211.

Friedman, J., T. Hastie, and R. Tibshirani (2001). *The elements of statistical learning*. Vol. 1. 10. Springer series in statistics New York.

— (2008). "Sparse inverse covariance estimation with the graphical lasso". *Biostatistics* 9.3, pp. 432–441.

Froufe, M. M. et al. (2020). "Smart buildings: Systems and drivers". *Buildings* 10.9, p. 153.

Fulbright, N. R. et al. (2016). *Corporate Renewable Power Purchase Agreements: Scaling up globally*. `https://www.wbcsd.org/Programs/Climate-and-Energy/Climate/Resources`, (last accessed on 20/06/2020).

Gan, G., C. Ma, and J. Wu (2007). *Data clustering: theory, algorithms, and applications*. Vol. 20. Siam.

Gao, J. and M. Bergés (2018). "A large-scale evaluation of automated metadata inference approaches on sensors from air handling units". *Advanced Engineering Informatics* 37, pp. 14–30.

Gargiulo, F. et al. (2019). "Deep neural network for hierarchical extreme multi-label text classification". *Applied Soft Computing* 79, pp. 125–138.

Genkin, A., D. D. Lewis, and D. Madigan (2007). "Large-scale Bayesian logistic regression for text categorization". *Technometrics* 49.3, pp. 291–304.

Gensler, A. et al. (2016). "Deep Learning for solar power forecasting—An approach using AutoEncoder and LSTM Neural Networks". In: *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 002858–002865.

Geron, A. (2017). *Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems*. Sebastopol, CA: O'Reilly Media. ISBN: 978-1491962299.

Gholamzadehmir, M. et al. (2020). "Adaptive-predictive control strategy for HVAC systems in smart buildings–A review". *Sustainable Cities and Society*, p. 102480.

Gil, A et al. (2010). "Influence of wind energy forecast in deterministic and probabilistic sizing of reserves". In: *9th International Workshop on Large-Scale Integration of Wind Power into Power Systems as well as on Transmission Networks for Offshore Wind Power Plants*. ISBN 978-3-9813870-2-5; TRN: DE11G8846, pp. 83–88.

Gilman, D. L., F. J. Fuglister, and J. M. Mitchell Jr (1963). "On the power spectrum of "red noise"". *Journal of the Atmospheric Sciences* 20.2, pp. 182–184.

Gneiting, T. and A. E. Raftery (2007). "Strictly proper scoring rules, prediction, and estimation". *Journal of the American Statistical Association* 102.477, pp. 359–378.

Golestaneh, F., P. Pinson, and H. B. Gooi (2016). "Very short-term nonparametric probabilistic forecasting of renewable energy generation—With application to solar energy". *IEEE Transactions on Power Systems* 31.5, pp. 3850–3863.

González-Romera, E et al. (2008). "Monthly electric energy demand forecasting with neural networks and Fourier series". *Energy Conversion and Management* 49.11, pp. 3135–3142.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep learning*. MIT press.

Goodman, J. T. (2001). "A bit of progress in language modeling". *Computer Speech & Language* 15.4, pp. 403–434.

Gopi, A. P. et al. (2020). "Classification of tweets data based on polarity using improved RBF kernel of SVM". *International Journal of Information Technology*, pp. 1–16.

Graves, A., A.-r. Mohamed, and G. Hinton (2013). "Speech recognition with deep recurrent neural networks". In: *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, pp. 6645–6649.

Grid, N. (2018). *National Grid UK (https://www.nationalgrid.com/uk)*. URL: `https://www.nationalgrid.com/uk`.

Grippo, L, F Lampariello, and S Lucidi (1989). "A truncated Newton method with nonmonotone line search for unconstrained optimization". *Journal of Optimization Theory and Applications* 60.3, pp. 401–419.

Hamedmoghadam, H., N. Joorabloo, and M. Jalili (2018). "Australia's long-term electricity demand forecasting using deep neural networks". *arXiv preprint arXiv:1801.02148*.

Harrag, F., E. El-Qawasmeh, and P. Pichappan (2009). "Improving Arabic text categorization using decision trees". In: *2009 First International Conference on Networked Digital Technologies*. IEEE, pp. 110–115.

Hasanli, H. and S. Rustamov (2019). "Sentiment Analysis of Azerbaijani twits Using Logistic Regression, Naive Bayes and SVM". In: *2019 IEEE 13th International Conference on Application of Information and Communication Technologies (AICT)*. IEEE, pp. 1–7.

Haystack (2019). *Project Haystack web site, URL: https://project-haystack.org/ (last accessed on 25/02/2019)*.

He, W. (2017). "Load forecasting via deep neural networks". *Procedia Computer Science* 122, pp. 308–314.

Hedges, A., N. R. Fulbright, and M. Duvoort (2019). *How multi-technology PPAs could help companies reduce risk*. `https://www.wbcsd.org/Programs/Climate-and-Energy/Energy/REscale/Resources`, (last accessed on 20/06/2020).

Heinemann, D., E. Lorenz, and M. Girodo (2006). "Forecasting of solar radiation". *Solar energy resource management for electricity generation from local level to global scale. Nova Science Publishers, New York*, pp. 83–94.

Hölzle, U. (2016). *Achieving Our 100 Renewable Energy Purchasing Goal*. `https://www.google.com/green/pdf/achieving-100-renewable-energy-purchasing-goal.pdf`, (last accessed on 20/06/2020).

Hochreiter, S. and J. Schmidhuber (1997). "Long short-term memory". *Neural computation* 9.8, pp. 1735–1780.

Hong, W.-C. (2009). "Electric load forecasting by support vector model". *Applied Mathematical Modelling* 33.5, pp. 2444–2454.

Huang, C.-Y. et al. (2011). "Short term wind speed predictions by using the grey prediction model based forecast method". In: *2011 IEEE Green Technologies Conference (IEEE-Green)*. IEEE, pp. 1–5.

Hyndman, R. J. and G. Athanasopoulos (2018). *Forecasting: principles and practice*. Publisher: OTexts.

Ifrim, G., G. Bakir, and G. Weikum (2008). "Fast logistic regression for text categorization with variable-length n-grams". In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 354–362.

Jain, A. K. and R. C. Dubes (1988). "Algorithms for clustering data". *Englewood Cliffs: Prentice Hall, 1988*.

Jaskie, K., C. Elkan, and A. Spanias (2019). "A modified logistic regression for positive and unlabeled learning". In: *2019 53rd Asilomar Conference on Signals, Systems, and Computers*. IEEE, pp. 2007–2011.

Jenkins, G., H. Lim, et al. (1999). *An Integrated Analysis of a Power Purchase Agreement*. Tech. rep. JDI Executive Programs.

Jeong, Y.-S., M. K. Jeong, and O. A. Omitaomu (2011). "Weighted dynamic time warping for time series classification". *Pattern Recognition* 44.9, pp. 2231–2240.

Jiang, Y., Z. Song, and A. Kusiak (2013). "Very short-term wind speed forecasting with Bayesian structural break model". *Renewable Energy* 50, pp. 637–647.

Joachims, T. (1998). "Text categorization with support vector machines: Learning with many relevant features". In: *European conference on machine learning*. Springer, pp. 137–142.

— (2001). "A statistical learning learning model of text classification for support vector machines". In: *Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval*. ACM, pp. 128–136.

Jones, E., T. Oliphant, P. Peterson, et al. (2001–). *SciPy: Open source scientific tools for Python*. `http://www.scipy.org/`, (last accessed on 10/05/2020).

Juban, J., N. Siebert, and G. N. Kariniotakis (2007). "Probabilistic short-term wind power forecasting for the optimal management of wind generation". In: *2007 IEEE Lausanne Power Tech*, pp. 683–688.

Kandananond, K. (2011). "Forecasting electricity demand in Thailand with an artificial neural network approach". *Energies* 4.8, pp. 1246–1257.

Kendall, M. G. (1948). *Rank correlation methods.* Publisher: Griffin.

Keogh, E. J. and M. J. Pazzani (2000). "Scaling up dynamic time warping for datamining applications". In: *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 285–289.

Khiali, L. et al. (2019). "Detection of spatio-temporal evolutions on multi-annual satellite image time series: A clustering based approach". *International Journal of Applied Earth Observation and Geoinformation* 74, pp. 103–119.

Khwaja, A. et al. (2017). "Boosted neural networks for improved short-term electric load forecasting". *Electric Power Systems Research* 143, pp. 431–437.

Kingma, D. P. and J. Ba (2014). "Adam: A method for stochastic optimization". *arXiv preprint arXiv:1412.6980*.

Kingma, D. P. and M. Welling (2013). "Auto-encoding variational bayes". *arXiv preprint arXiv:1312.6114*.

Kingma, D. P., M. Welling, et al. (2019). "An introduction to variational autoencoders". *Foundations and Trends® in Machine Learning* 12.4, pp. 307–392.

Koh, J. et al. (2016). "Quiver: Using control perturbations to increase the observability of sensor data in smart buildings". *arXiv preprint arXiv:1601.07260*.

Koh, J. et al. (2018). "Plaster: An integration, benchmark, and development framework for metadata normalization methods". In: *Proceedings of the 5th Conference on Systems for Built Environments*, pp. 1–10.

Kong, W. et al. (2019). "Short-term residential load forecasting based on LSTM recurrent neural network". *IEEE Transactions on Smart Grid* 10.1, pp. 841–851.

Kouroupetroglou, P.-N. and G. Tsoumakas (2017). "Machine learning techniques for short-term electric load forecasting". *Thesis*.

Kraft, D. (1988). *A software package for sequential quadratic programming*. Publisher: Wiss. Berichtswesen d. DFVLR.

Kurnia, R. I., Y. D. Tangkuman, and A. S. Girsang (2020). "Classification of user comment using Word2vec and SVM classifier".

Lai, C. S. et al. (2020). "Load forecasting based on deep neural network and historical data augmentation". *IET Generation, Transmission & Distribution* 14.24, pp. 5927–5934.
Lai, S. et al. (2015). "Recurrent convolutional neural networks for text classification". In: *Twenty-ninth AAAI conference on artificial intelligence*, pp. 2267–2273.
Lance, G. N. and W. T. Williams (1967). "A general theory of classificatory sorting strategies: 1. Hierarchical systems". *The computer journal* 9.4, pp. 373–380.
Lange, B. et al. (2006). "Wind power prediction in Germany–Recent advances and future challenges". In: *European Wind Energy Conference, Athens*. Vol. 1. 5, pp. 73–81.
Le, C.-C. et al. (2019). "Text Classification: Naïve Bayes Classifier with Sentiment Lexicon". *IAENG International Journal of Computer Science* 46.2, pp. 141–148.
Lee, W.-Y., J. M. House, and N.-H. Kyong (2004). "Subsystem level fault diagnosis of a building's air-handling unit using general regression neural networks". *Applied Energy* 77.2, pp. 153–170.
Lee, W.-Y. et al. (1996). "Fault diagnosis of an air-handling unit using artificial neural networks". *Transactions-American society of heating refrigerating and air conditioning engineers* 102, pp. 540–549.
Lenton, T. et al. (2012). "Early warning of climate tipping points from critical slowing down: comparing methods to improve robustness". *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 370, pp. 1185–1204.
Li, G. et al. (2019). "Recurrent neural networks based photovoltaic power forecasting approach". *Energies* 12.13, p. 2538.
Li, L.-L. et al. (2020a). "Short-term wind power forecasting based on support vector machine with improved dragonfly algorithm". *Journal of Cleaner Production* 242, p. 118447.
Li, P.-H. et al. (2020b). "Investigating UK consumers' heterogeneous engagement in demand-side response". *Energy Efficiency*, pp. 1–28.
Li, S., D. Hong, and H. Wang (2017). "Relation Inference among Sensor Time Series in Smart Buildings with Metric Learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 4, pp. 4683–4690.
Li, S., L. Goel, and P. Wang (2016). "An ensemble approach for short-term load forecasting by extreme learning machine". *Applied Energy*. URL: `https://www.sciencedirect.com/science/article/pii/S0306261916302707`.
Liu, B. et al. (2002). "Partially supervised classification of text documents". In: *International Conference on Machine Learning*. Vol. 2. Citeseer, pp. 387–394.
Liu, J.-p. and C.-l. Li (2017a). "The short-term power load forecasting based on sperm whale algorithm and wavelet least square support vector machine with DWT-IR for feature selection". *Sustainability* 9.7, p. 1188.
Liu, J. et al. (2017b). "Deep learning for extreme multi-label text classification". In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 115–124.
Liu, P. et al. (2019). "Parallel naive Bayes algorithm for large-scale Chinese text classification based on spark". *Journal of Central South University* 26.1, pp. 1–12.
Livina, V., A. Lewis, and M. Wickham (2019). "Tipping point analysis of electrical resistance data with early warning signals of failure for predictive maintenance". *arXiv preprint arXiv:1904.04636*.
Livina, V. N., F Kwasniok, and T. M. Lenton (2010). "Potential analysis reveals changing number of climate states during the last 60 kyr." *Climate of the Past* 6.1.
Livina, V. N. and T. M. Lenton (2007). "A modified method for detecting incipient bifurcations in a dynamical system". *Geophysical research letters* 34.3.
Livina, V. et al. (2013). "Forecasting the underlying potential governing the time series of a dynamical system". *Physica A: Statistical Mechanics and its Applications* 392.18, pp. 3891–3902.

Lowery, C. and M. O'Malley (2012). "Impact of wind forecast error statistics upon unit commitment". *IEEE Transactions on Sustainable Energy* 3.4, pp. 760–768.
Lu, Y.-C., C.-H. Shen, and Y.-C. Wei (2013). "Revisiting early warning signals of corporate credit default using linguistic analysis". *Pacific-Basin Finance Journal* 24, pp. 1–21.
Marmaras, C. et al. (2017). "Predicting the energy demand of buildings during triad peaks in GB". *Energy and Buildings* 141, pp. 262–273.
Maron, M. E. (1961). "Automatic indexing: an experimental inquiry". *Journal of the ACM* 8.3, pp. 404–417.
Matos, M. A. and R. J. Bessa (2010). "Setting the operating reserve using probabilistic wind power forecasts". *IEEE transactions on Power Systems* 26.2, pp. 594–603.
Mauch, B. et al. (2013). "An effective method for modeling wind power forecast uncertainty". *Energy Systems* 4.4, pp. 393–417.
McCallum, A., K. Nigam, et al. (1998). "A comparison of event models for naive bayes text classification". In: *AAAI-98 workshop on learning for text categorization*. Vol. 752. 1. Publisher: Citeseer, pp. 41–48.
McLean Sloughter, J, T. Gneiting, and A. E. Raftery (2013). "Probabilistic wind vector forecasting using ensembles and Bayesian model averaging". *Monthly Weather Review* 141.6, pp. 2107–2119.
Miaschi, A. and F. Dell'Orletta (2020). "Contextual and Non-Contextual Word Embeddings: an in-depth Linguistic Investigation". In: *Proceedings of the 5th Workshop on Representation Learning for NLP*, pp. 110–119.
Mikolov, T., Q. V. Le, and I. Sutskever (2013a). "Exploiting similarities among languages for machine translation". *arXiv preprint arXiv:1309.4168*.
Mikolov, T., W.-t. Yih, and G. Zweig (2013b). "Linguistic regularities in continuous space word representations". In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 746–751.
Mikolov, T. et al. (2013c). "Efficient estimation of word representations in vector space". *arXiv preprint arXiv:1301.3781*.
Mohanad, S. A.-M., C. D. Ravinesh, and L. Yan (2018). "Particle swarm optimized–support vector regression hybrid model for daily horizon electricity demand forecasting using climate dataset". In: *E3S Web of Conferences*. Vol. 64. EDP Sciences, p. 08001.
Mokhtari, A. and A. Ribeiro (2014). "RES: Regularized stochastic BFGS algorithm". *IEEE Transactions on Signal Processing* 62.23, pp. 6089–6104.
Montero, P., J. Vilar, et al. (2014). "TSclust: An R package for time series clustering". *Journal of Statistical Software* 62.1, pp. 1–43.
Mtibaa, F. et al. (2020). "LSTM-based indoor air temperature prediction framework for HVAC systems in smart buildings". *Neural Computing and Applications*, pp. 1–17.
Negnevitsky, M. and C. W. Potter (2006). "Innovative short-term wind generation prediction techniques". In: *2006 IEEE PES Power Systems Conference and Exposition*, 7–pp.
Newbery, D. (2011). "High level principles for guiding GB transmission charging and some of the practical problems of transition to an enduring regime". *Report Prepared for and Commissioned by Project TransmiT, Great Britain Office of Gas & Electricity Markets*.
Park, J. Y., B. Lasternas, and A. Aziz (2018). "Data-driven framework to find the physical association between AHU and VAV terminal unit–pilot study". Session 11.
Pearson, K. (1901). "Principal components analysis". *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 6.2, p. 559.
Pedregosa, F. et al. (2011). "Scikit-learn: Machine Learning in Python". *Journal of Machine Learning Research* 12, pp. 2825–2830.
Perez, R. et al. (2010). "Validation of short and medium term operational solar radiation forecasts in the US". *Solar Energy* 84.12, pp. 2161–2172.

Pinson, P., C. Chevallier, and G. N. Kariniotakis (2007). "Trading wind generation from short-term probabilistic forecasts of wind power". *IEEE Transactions on Power Systems* 22.3, pp. 1148–1156.

Pinson, P. et al. (2009). "From probabilistic forecasts to statistical scenarios of short-term wind power production". *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology* 12.1, pp. 51–62.

Plageras, A. P. et al. (2018). "Efficient IoT-based sensor BIG Data collection–processing and analysis in smart buildings". *Future Generation Computer Systems* 82, pp. 349–357.

Powell, M. J. (1964). "An efficient method for finding the minimum of a function of several variables without calculating derivatives". *The computer journal* 7.2, pp. 155–162.

— (1973). "On search directions for minimization algorithms". *Mathematical programming* 4.1, pp. 193–201.

Prabhat, A. and V. Khullar (2017). "Sentiment classification on big data using Naïve Bayes and logistic regression". In: *2017 International Conference on Computer Communication and Informatics (ICCCI)*. IEEE, pp. 1–5.

Prettyman, J, T. Kuna, and V Livina (2019). "Generalized early warning signals in multivariate and gridded data with an application to tropical cyclones". *Chaos: An Interdisciplinary Journal of Nonlinear Science* 29.7, p. 073105.

Prettyman, J., T. Kuna, and V. Livina (2018). "A novel scaling indicator of early warning signals helps anticipate tropical cyclones". *EPL (Europhysics Letters)* 121.1, p. 10002.

Pritoni, M. et al. (2015). "Short paper: A method for discovering functional relationships between air handling units and variable-air-volume boxes from sensor data". In: *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, pp. 133–136.

Qingle, P. and Z. Min (2010). "Very short-term load forecasting based on neural network and rough set". In: *Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on*. Vol. 3. IEEE, pp. 1132–1135.

Qiu, J. et al. (2015). "The early-warning model of equipment chain in gas pipeline based on DNN-HMM". *Journal of Natural Gas Science and Engineering* 27, pp. 1710–1722.

Quan, D., D. Huynh, and D. R. Karger (2003). "Haystack: A platform for authoring end user semantic web applications". In: *International Semantic Web Conference*. Springer, pp. 738–753.

Quinlan, J. R. (1986). "Induction of decision trees". *Machine learning* 1.1, pp. 81–106.

Rakthanmanon, T. et al. (2012). "Searching and mining trillions of time series subsequences under dynamic time warping". In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 262–270.

Ramadhan, W., S. A. Novianty, and S. C. Setianingsih (2017). "Sentiment analysis using multinomial logistic regression". In: *2017 International Conference on Control, Electronics, Renewable Energy and Communications (ICCREC)*. IEEE, pp. 46–49.

Rane, A. and A. Kumar (2018). "Sentiment classification system of Twitter data for US airline service analysis". In: *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*. Vol. 1. IEEE, pp. 769–773.

Ranjitha, K., B. V. Prasad, et al. (2020). "Optimization scheme for text classification using machine learning naïve Bayes classifier". In: *ICDSMLA 2019*. Springer, pp. 576–586.

Reikard, G. (2009). "Predicting solar radiation at high resolutions: A comparison of time series forecasts". *Solar Energy* 83.3, pp. 342–349.

Reynolds, D. A. (2009). "Gaussian Mixture Models." *Encyclopedia of biometrics* 741.

Rogers, B. M. et al. (2018). "Detecting early warning signals of tree mortality in boreal North America using multiscale satellite data". *Global change biology* 24.6, pp. 2284–2304.

Saini, L. and M. Soni (2002). "Artificial neural network based peak load forecasting using Levenberg–Marquardt and quasi-Newton methods". *IEEE Proceedings-Generation, Transmission and Distribution* 149.5, pp. 578–584.

Salvatier, J., T. V. Wiecki, and C. Fonnesbeck (2016). "Probabilistic programming in Python using PyMC3". *PeerJ Computer Science* 2, e55.

Schein, J. et al. (2006). "A rule-based fault detection method for air handling units". *Energy and Buildings* 38.12, pp. 1485–1492.

Sehovac, L. and K. Grolinger (2020). "Deep Learning for Load Forecasting: Sequence to Sequence Recurrent Neural Networks With Attention". *IEEE Access* 8, pp. 36411–36426.

Sfetsos, A and A. Coonick (2000). "Univariate and multivariate forecasting of hourly solar radiation with artificial intelligence techniques". *Solar Energy* 68.2, pp. 169–178.

Shi, J. et al. (2013). "Hybrid forecasting model for very-short term wind power forecasting based on grey relational analysis and wind speed distribution features". *IEEE Transactions on Smart Grid* 5.1, pp. 521–526.

Shukla, R. M. and S. Sengupta (2020). "Scalable and Robust Outlier Detector using Hierarchical Clustering and Long Short-Term Memory (LSTM) Neural Network for the Internet of Things". *Internet of Things*, p. 100167.

Sibson, R. (1973). "SLINK: an optimally efficient algorithm for the single-link cluster method". *The computer journal* 16.1, pp. 30–34.

Sideratos, G., A. Ikonomopoulos, and N. D. Hatziargyriou (2020). "A novel fuzzy-based ensemble model for load forecasting using hybrid deep neural networks". *Electric Power Systems Research* 178, p. 106025.

Singh, G. et al. (2019). "Comparison between multinomial and bernoulli Naïve Bayes for text classification". In: *2019 International Conference on Automation, Computational and Technology Management (ICACTM)*. IEEE, pp. 593–596.

Singh, P. and P. Dwivedi (2018). "Integration of new evolutionary approach with artificial neural network for solving short term load forecast problem". *Applied energy* 217, pp. 537–549.

Snoonian, D. (2003). "Smart buildings". *IEEE spectrum* 40.8, pp. 18–23.

Solari, G. et al. (2012). "The wind forecast for safety management of port areas". *Journal of Wind Engineering and Industrial Aerodynamics* 104, pp. 266–277.

Squartini, T., I. Van Lelyveld, and D. Garlaschelli (2013). "Early-warning signals of topological collapse in interbank networks". *Scientific reports* 3, p. 3357.

Stahley, B. (2019). *Commercial solar panel degradation: What you should know and keep in mind*. `https://businessfeed.sunpower.com/articles/what-to-know-about-commercial-solar-panel-degradation`, (last accessed on 21/05/2020).

Stopps, H and M. Touchie (2019). "Reduction of HVAC system runtime due to occupancy-controlled smart thermostats in contemporary multi-unit residential building suites". In: *IOP Conference Series: Materials Science and Engineering*. Vol. 609. 6. IOP Publishing, p. 062013.

Sun, A., E.-P. Lim, and Y. Liu (2009). "On strategies for imbalanced text classification using SVM: A comparative study". *Decision Support Systems* 48.1, pp. 191–201.

Sun, Z. and M. Zhao (2020). "Short-Term Wind Power Forecasting Based on VMD Decomposition, ConvLSTM Networks and Error Analysis". *IEEE Access* 8, pp. 134422–134434.

Tian, C. et al. (2018). "A deep neural network model for short-term load forecast based on long short-term memory network and convolutional neural network". *Energies* 11.12, p. 3493.

Tong, S. and D. Koller (2001). "Support vector machine active learning with applications to text classification". *Journal of machine learning research* 2.Nov, pp. 45–66.

TREND (2018). *TREND web site, URL: https://www.trendcontrols.com/en-GB/Pages/default.aspx*. (last accessed on 12/11/2018).

Tucker, A., S. Swift, and X. Liu (2001). "Variable grouping in multivariate time series via correlation". *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 31.2, pp. 235–245.

Urquhart, B et al. (2015). "Development of a sky imaging system for short-term solar power forecasting". *Atmospheric Measurement Techniques* 8.2, p. 875.

Van Cutsem, O. et al. (2020). "Cooperative energy management of a community of smart-buildings: A Blockchain approach". *International Journal of Electrical Power & Energy Systems* 117, p. 105643.

Vapnik, V. and A. Y. Lerner (1963). "Recognition of patterns with help of generalized portraits". *Avtomat. i Telemekh* 24.6, pp. 774–780.

Venter, G. (2010). "Review of optimization techniques". *Encyclopedia of aerospace engineering*.

Wang, J. et al. (2016). "Dimensional sentiment analysis using a regional CNN-LSTM model". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vol. 2, pp. 225–230.

Wang, X. et al. (2019). "CharCNN-SVM for Chinese text datasets sentiment classification with data augmentation". *International Journal of Innovative Computing, Information and Control* 15.1, pp. 227–246.

Ward Jr, J. H. (1963). "Hierarchical grouping to optimize an objective function". *Journal of the American statistical association* 58.301, pp. 236–244.

Wen, J. and H. Gao (2018). "Degradation assessment for the ball screw with variational autoencoder and kernel density estimation". *Advances in Mechanical Engineering* 10.9, p. 1687814018797261.

Wendorff, A., E. Botero, and J. J. Alonso (2016). "Comparing different off-the-shelf optimizers' performance in conceptual aircraft design". In: *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, p. 3362.

Weng, T. and Y. Agarwal (2012). "From buildings to smart buildings—sensing and actuation to improve energy efficiency". *IEEE Design Test of Computers* 29.4, pp. 36–44.

White, H. (1982). "Maximum likelihood estimation of misspecified models". *Econometrica: Journal of the Econometric Society*, pp. 1–25.

Wilczak, J. et al. (2015). "The Wind Forecast Improvement Project: A public–private partnership addressing wind energy forecast needs". *Bulletin of the American Meteorological Society* 96.10, pp. 1699–1718.

Wright, R. E. (1995). "Logistic regression."

Xu, B. et al. (2012). "An improved random forest classifier for text categorization." *JCP* 7.12, pp. 2913–2920.

Xu, H. et al. (2016). "End-to-end learning of driving models from large-scale video datasets". *arXiv preprint*.

Yang, S. et al. (2020). "Model predictive control with adaptive machine-learning-based model for building energy efficiency and comfort optimization". *Applied Energy* 271, p. 115147.

Yao, L., C. Mao, and Y. Luo (2019). "Graph convolutional networks for text classification". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33, pp. 7370–7377.

Yu, L. et al. (2020). "Multi-agent deep reinforcement learning for HVAC control in commercial buildings". *IEEE Transactions on Smart Grid*.

Yu, Y., D. Woradechjumroen, and D. Yu (2014). "A review of fault detection and diagnosis methodologies on air-handling units". *Energy and Buildings* 82, pp. 550–562.

Yun, Z. et al. (2008). "RBF neural network and ANFIS-based short-term load forecasting approach in real-time price environment". *IEEE Transactions on power systems* 23.3, pp. 853–858.

Zaki, M. J., W. Meira Jr, and W. Meira (2014). *Data mining and analysis: fundamental concepts and algorithms*. Cambridge University Press.

Zhang, M., Z. Yu, and Z. Xu (2020). "Short-Term Load Forecasting Using Recurrent Neural Networks With Input Attention Mechanism and Hidden Connection Mechanism". *IEEE Access* 8, pp. 186514–186529.

Zhang, M., X. Ai, and Y. Hu (2019). "Chinese text classification system on regulatory information based on SVM". In: *IOP Conference Series: Earth and Environmental Science*. Vol. 252. 2. IOP Publishing, p. 022133.

Zhang, X., J. Zhao, and Y. LeCun (2015). "Character-level convolutional networks for text classification". In: *Advances in neural information processing systems*, pp. 649–657.

Zhang, Y., R. Jin, and Z.-H. Zhou (2010). "Understanding bag-of-words model: a statistical framework". *International Journal of Machine Learning and Cybernetics* 1.1-4, pp. 43–52.

Zheng, J. et al. (2017). "Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network". In: *Information Sciences and Systems (CISS), 2017 51st Annual Conference on Information Sciences and Systems (CISS)*. IEEE, pp. 1–6.

Zhu, C. et al. (1997). "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization". *ACM Transactions on Mathematical Software (TOMS)* 23.4, pp. 550–560.