COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO

GENERATION OF ELECTRICITY LOAD PROFILES FOR NON-RESIDENTIAL BUILDINGS USING STATISTICAL METHODS OF MACHINE LEARNING

Autor: Daniel Elechiguerra Batlle

Director: Jan Richarz

Madrid

Agosto de 2019

intereses a causa de la cesión.

d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusive del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 26 de Agosto de 2019

ACEPTA

Fdo. Daniel Elechiguerra Batlle

Elechiguerra

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título "GENERATION OF ELECTRICITY LOAD PROFILES FOR NON-RESIDENTIAL BUILDINGS USING STATISTICAL METHODS OF MACHINE LEARNING" en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2018-2019 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: Daniel Elechiguerra Batlle Fecha: 26/ 08/ 2019

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Jan Richarz Fecha: 26/ 08/ 2019

COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO

GENERATION OF ELECTRICITY LOAD PROFILES FOR NON-RESIDENTIAL BUILDINGS USING STATISTICAL METHODS OF MACHINE LEARNING

Autor: Daniel Elechiguerra Batlle

Director: Jan Richarz

Madrid

Agosto de 2019

GENERACIÓN DE PERFILES DE DEMANDA ELÉCTRICA PARA EDIFICIOS NO RESIDENCIALES UTILIZANDO MÉTODOS ESTADÍSTICOS DE APRENDIZAJE AUTOMÁTICO

Autor: Elechiguerra Batlle, Daniel.

Director: Richarz, Jan.

Entidad colaboradora: RWTH Aachen University.

RESUMEN DEL PROYECTO

Introducción

El sector energético ha sido una de las principales preocupaciones en los últimos años debido a sus implicaciones en el cambio climático. Los gobiernos están haciendo un esfuerzo cada vez mayor para lograr un desarrollo sostenible dentro de este sector, por un lado, a través de la expansión de las energías renovables y, por otro, a través de un uso más eficiente de la energía. El sector inmobiliario representa el 40% del consumo final de energía en la Unión Europea y se espera que esta demanda aumente [1]. En Alemania, los edificios no residenciales representan el 37% del consumo energético del parque inmobiliario [2]. Adicionalmente, los edificios de nueva construcción representan una pequeña fracción de todo el parque inmobiliario. Por lo tanto, los edificios no residenciales existentes son un objetivo relevante para lograr el desarrollo sostenible.

Para alcanzar este ambicioso propósito, la principal solución pasa por la integración de las energías renovables en los sistemas energéticos, sustituyendo la actual generación centralizada no renovable por una generación distribuida más eficiente y fiable, basada principalmente en energías renovables [3]. Esta generación distribuida depende directamente de la adaptación de las energías renovables al consumo energético de los edificios distribuidos. La planificación energética de los edificios consiste en seleccionar la combinación óptima de dispositivos tales como paneles fotovoltaicos, bombas de calor o tecnologías de cogeneración que sean capaces de satisfacer las necesidades energéticas del edificio, minimizando al mismo tiempo las emisiones de gases de efecto invernadero y maximizando la eficiencia energética [4].

La modernización y rehabilitación de edificios existentes para la mencionada planificación sólo es posible si se conoce el comportamiento de consumo energético de un edificio, es decir, el perfil de demanda eléctrica. Sin embargo, estos perfiles raramente son conocidos para la mayoría de los edificios. En consecuencia, se necesitan modelos precisos para poder estimar los perfiles eléctricos característicos de los edificios existentes, sin necesidad de medir su consumo de energía [1]. Esta tarea, en la cual se centra este trabajo, se denomina «modelado» para distinguirla de «pronóstico», que implica la predicción del consumo futuro de energía. En este último, se emplea un algoritmo de aprendizaje que utiliza información del pasado para predecir resultados futuros, y típicamente consiste en un enfoque longitudinal, puesto que genera predicciones sobre los mismos edificios utilizados para entrenar el modelo [5].

Se han propuesto numerosos modelos en el área del modelado de perfiles de demanda eléctrica, que se pueden dividir principalmente en dos enfoques diferentes. Por un lado, los enfoques *bottom-up* se basan en el análisis del comportamiento individual de los aparatos eléctricos mediante la determinación de la distribución de probabilidad de su consumo de energía. Estas distribuciones se combinan con variables específicas del edificio para crear múltiples subsistemas. En última instancia, los diferentes subsistemas se agregan en un sistema complejo que puede modelar el perfil eléctrico agregado del edificio. Por otro lado, los enfoques *top-down* parten del consumo agregado de los edificios y son capaces de extraer la relación entre el consumo de energía y las variables de entrada, obteniendo información sobre ellas. La idea que subyace a los modelos *top-down* es la descomposición del perfil inicial en sus subsistemas, de donde se pueden derivar las relaciones. En estos casos, se emplea normalmente un algoritmo predictivo para llevar a cabo la tarea.

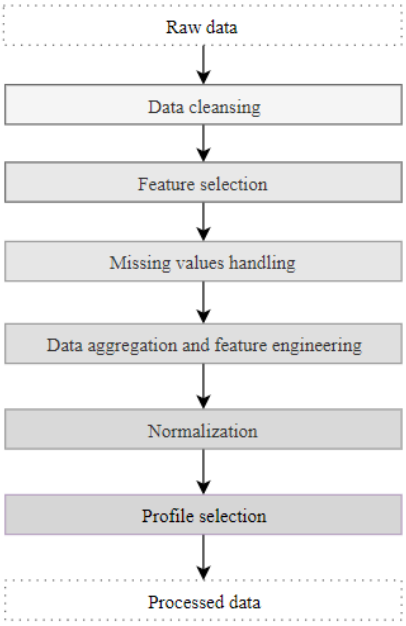
En cuanto al modelado de perfiles eléctricos, se han desarrollado modelos *top-down* relativamente sencillos, centrados principalmente en edificios residenciales. Por ejemplo, Ge et al. [6] propusieron en su trabajo un modelo de regresión basado en la caracterización de los perfiles eléctricos mediante una superposición de cinco distribuciones normales. A continuación, se realiza un análisis paramétrico para extraer las dependencias entre los parámetros de las distribuciones y el número de dormitorios y ocupantes. McLoughlin et al. [7] propusieron un enfoque alternativo basado en técnicas de agrupamiento tales como *k-means* y *k-medoids*. En este trabajo, los perfiles eléctricos de hogares individuales se agrupan en clases de perfiles. Posteriormente, estas clases de perfiles se vinculan a las características del hogar mediante un modelo de regresión logística que, en una última etapa, se utiliza para clasificar los nuevos hogares en las clases de perfiles obtenidas, en función de sus características.

Por el contrario, se han desarrollado métodos más complejos para el pronóstico del consumo de energía. A pesar de que el pronóstico del consumo de energía no es el objeto de este trabajo, los métodos y enfoques utilizados para esta tarea generalmente se pueden aplicar al modelado de perfiles de demanda eléctrica. En cuanto a estos métodos, se han propuesto diversos algoritmos, siendo la regresión lineal, máquinas de soporte vectorial, redes neuronales, bosques aleatorios y árboles de decisión con potenciación del gradiente, los más utilizados. Además, estudios recientes han propuesto el uso de métodos de aprendizaje conjunto, como *stacking*, para mejorar la precisión del modelo predictivo.

Dado que el desarrollo de modelos precisos para el modelado de perfiles de demanda eléctrica en edificios no residenciales sigue siendo una cuestión clave para la generación distribuida, será el tema central de este trabajo. El objetivo de este trabajo es desarrollar un modelo predictivo basado en algoritmos de aprendizaje automático que, mediante el uso de información transversal como los datos de aparatos eléctricos y los datos de habitaciones, pueda utilizarse para modelar con precisión los perfiles eléctricos de los edificios no residenciales, sin necesidad de medir su consumo de energía. Un propósito complementario es utilizar el modelo obtenido para analizar las dependencias entre los perfiles eléctricos y los parámetros seleccionados.

Metodología

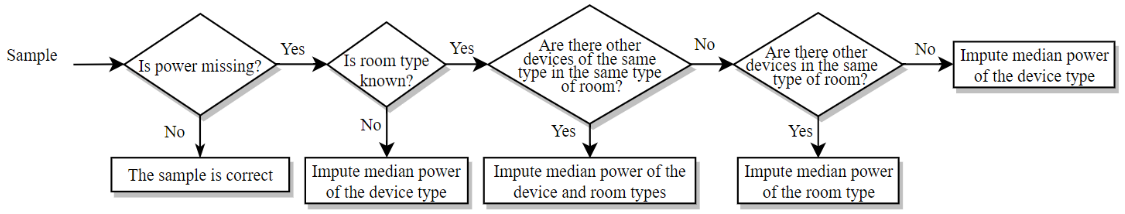
En este trabajo, se proponen dos modelos *top-down* diferentes para llevar a cabo la tarea prevista, con el fin de obtener los mejores resultados posibles. Ambos han sido implementados en Python 3.7.3, utilizando principalmente las librerías *Scikit-learn* [8], *Pandas* [9] y *NumPy* [10]. Para ambos modelos, los datos recopilados deben ser preprocesados para obtener un conjunto de datos adecuado. El procedimiento se ilustra en la siguiente figura:



Los datos han sido recogidos de 70 edificios del Centro de Investigación de Jülich. Contienen información sobre su consumo de energía, que se ha utilizado como salida para los modelos, e información sobre las habitaciones y los aparatos eléctricos del edificio, como calderas, ventilación, refrigeración, calefacción y aparatos de suministro de agua fría, que han servido como entrada para el modelo.

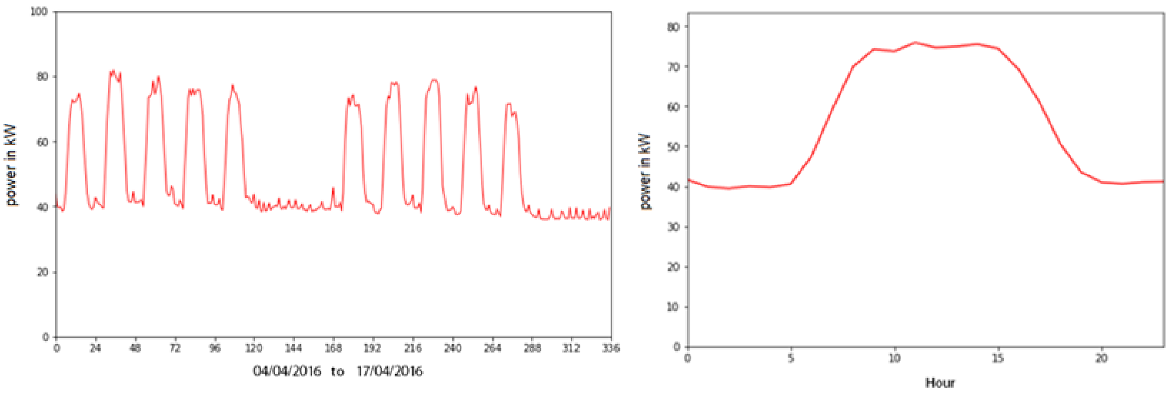
El primer paso, *data cleansing*, implica la detección y eliminación de valores atípicos del consumo de energía de los edificios en cuestión. A continuación, se seleccionan las variables relevantes que se utilizarán en los modelos implementados.

Los datos dados no han sido recogidos expresamente para este proyecto, lo cual ha resultado en una importante falta de información. En cuanto a las variables numéricas, faltaba el 58% de la información. Por lo tanto, se han realizado diferentes métodos de imputación múltiple para cada variable numérica, con el fin de compensar parcialmente la falta de datos. En la siguiente figura se muestra un ejemplo de imputación múltiple para la potencia de los aparatos de refrigeración:


Sample
Is power missing?
Yes
No
The sample is correct
Is room type known?
Yes
No
Impute median power of the device type
Are there other devices of the same type in the same type of room?
No
Yes
Impute median power of the device and room types
Are there other devices in the same type of room?
No
Yes
Impute median power of the room type
Impute median power of the device type

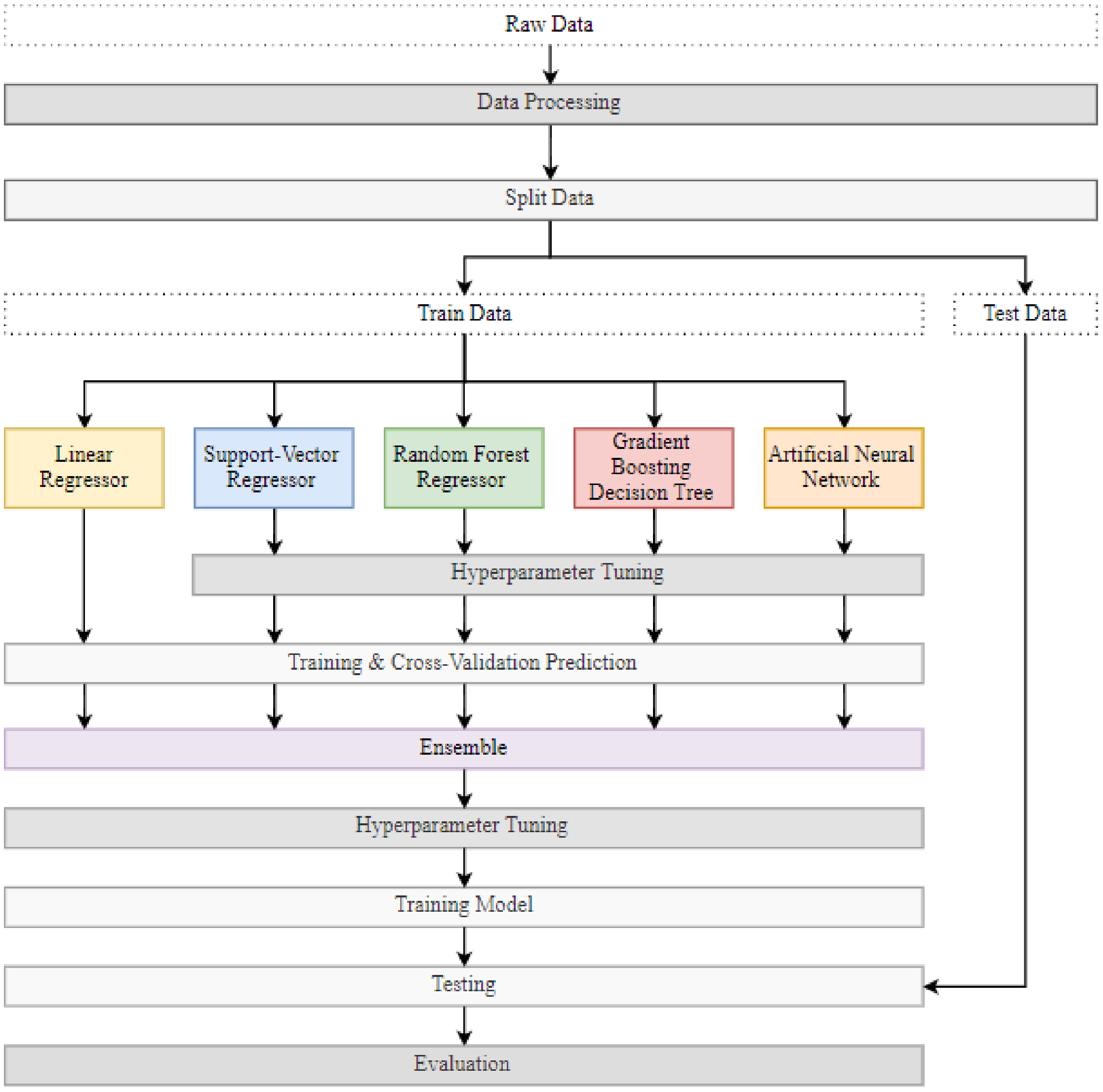
Los datos vienen estructurados por aparato y habitación, respectivamente. Sin embargo, como el resultado de los modelos consiste en los perfiles eléctricos de los edificios, es necesario agregar los datos de entrada para cada edificio. En este paso, ha sido necesario agregar las diferentes variables de distintas maneras, prestando especial atención a las variables categóricas, con el fin de evitar una pérdida significativa de información. Posteriormente, el conjunto de entrada resultante del paso anterior es normalizado, puesto que es un requisito para la mayoría de los algoritmos predictivos.

El último paso del preprocesamiento consiste en la selección de los perfiles de demanda eléctrica utilizados como salida para el modelo. Para obtener perfiles eléctricos característicos de los edificios, los perfiles de salida se obtienen promediando el consumo energético de los días laborables de dos semanas consecutivas, como se representa en la siguiente figura. Con el fin de considerar las variaciones estacionales en los perfiles eléctricos, se ha repetido el procedimiento para perfiles de abril, junio y diciembre.



En cuanto a los modelos propuestos, se han seguido dos enfoques diferentes. El primer modelo se basa únicamente en la regresión, que se realiza para cada hora del día, mientras que el segundo modelo se basa en una combinación de agrupación y clasificación de perfiles eléctricos normalizados y una regresión para el consumo máximo de potencia. En ambos casos, se ha aplicado el método de *stacking* para mejorar el rendimiento. En el *stacking*, se utilizan varios algoritmos para predecir la salida deseada y, en lugar de seleccionar el algoritmo con el mejor rendimiento, se entrena un algoritmo adicional (*ensemble*) con las salidas de los algoritmos anteriores para hacer la predicción final.

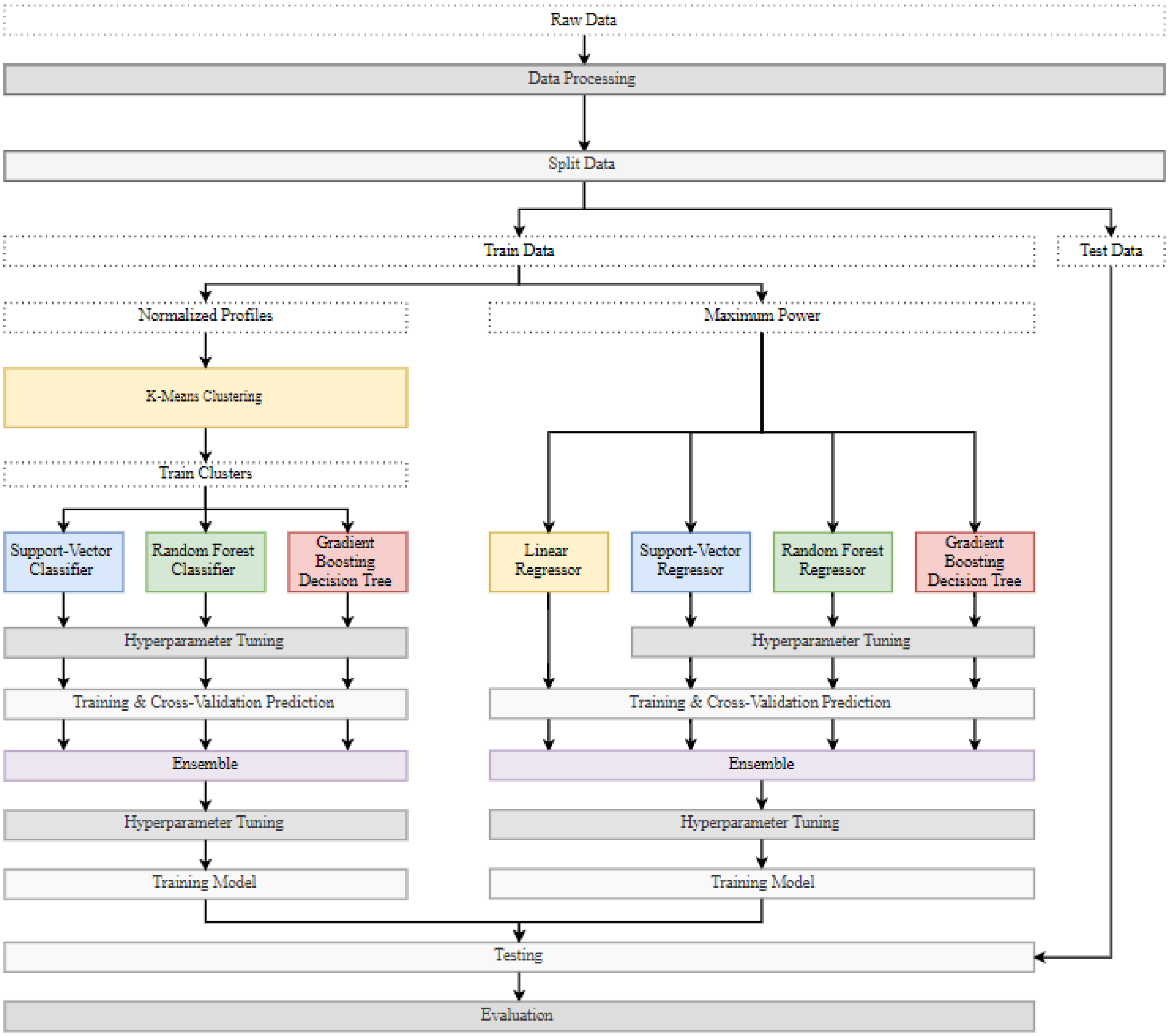
Para el modelo de regresión, los datos preprocesados se dividen en los conjuntos de entrenamiento y de pruebas. A continuación, los datos de entrenamiento se introducen en los algoritmos base, que pueden verse en la siguiente figura. Los mejores hiperparámetros, que controlan el proceso de aprendizaje, se seleccionan automáticamente para cada algoritmo. Estos algoritmos se entrenan con los datos dados y se obtienen las predicciones de validación cruzada para el conjunto de entrenamiento. Las predicciones mencionadas se utilizan como entrada al algoritmo adicional, que es un bosque aleatorio. Los hiperparámetros de este algoritmo se seleccionan y el algoritmo se entrena de la misma manera que antes. Finalmente, el modelo implementado se utiliza para estimar los perfiles de los edificios de prueba, y los resultados se comparan con los perfiles reales para evaluar el rendimiento del modelo.



La implementación del segundo modelo se basa en la agrupación y clasificación de los perfiles eléctricos y es ligeramente más complicado que el anterior. Dado que el tamaño del conjunto de datos disponible es reducido y que el rango de potencia de los edificios existentes es relativamente grande, estos perfiles eléctricos deben normalizarse. A continuación, se realiza un análisis de regresión para predecir la potencia máxima del edificio.

Una vez más, los datos preprocesados se dividen en los conjuntos de entrenamiento y de pruebas. Posteriormente, los perfiles de entrenamiento se normalizan y se utiliza un algoritmo *k-means* para agruparlos en cinco grupos. Los grupos de perfiles resultantes servirán como clases de salida para entrenar los algoritmos de clasificación. Para la tarea siguiente se utilizan diferentes clasificadores, mostrados en la siguiente figura. La selección de los hiperparámetros y el entrenamiento de estos algoritmos, así como del bosque aleatorio adicional, se realizan de la misma manera que en el modelo anterior.

En este punto, se extrae la potencia máxima de los perfiles de entrenamiento originales y se realiza un análisis de regresión de la potencia máxima para predecir la potencia máxima de los edificios de pruebas. Este paso es esencialmente igual al enfoque anterior. La etapa final consiste en multiplicar los perfiles normalizados por la potencia máxima prevista para obtener los perfiles eléctricos modelados para el conjunto de pruebas, que pueden ser comparados con los perfiles reales y con los obtenidos en la sección anterior para evaluar el rendimiento de este modelo alternativo.



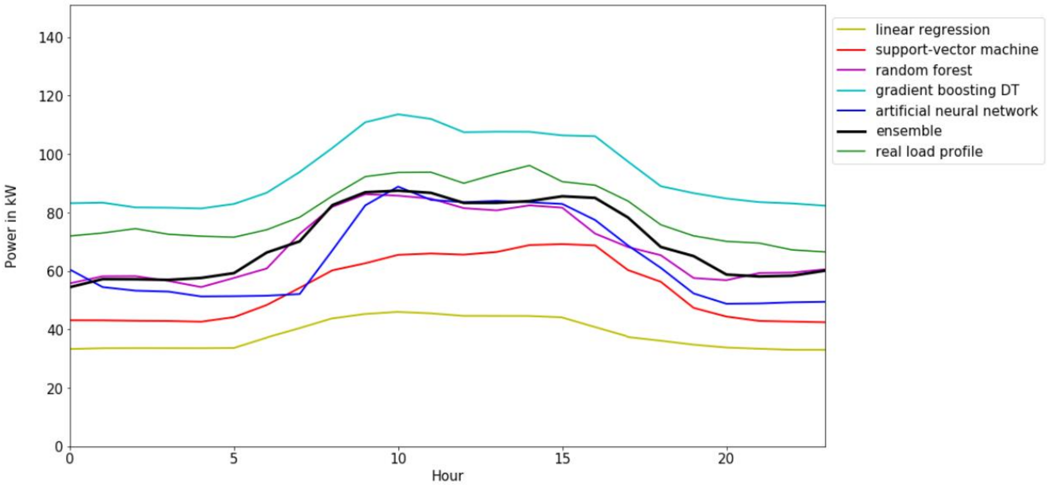
Finalmente, para evaluar el rendimiento de los modelos, se utiliza como métrica el error porcentual absoluto medio (MAPE), puesto que es la métrica más utilizado para el modelado de perfiles de demanda eléctrica. Sin embargo, el MAPE tiene ciertas limitaciones que pueden dar lugar a resultados erróneos. Por lo tanto, en el presente trabajo se utiliza, adicionalmente, una métrica alternativa derivada de la anterior, conocida como error porcentual absoluto medio simétrico (sMAPE).

Resultados

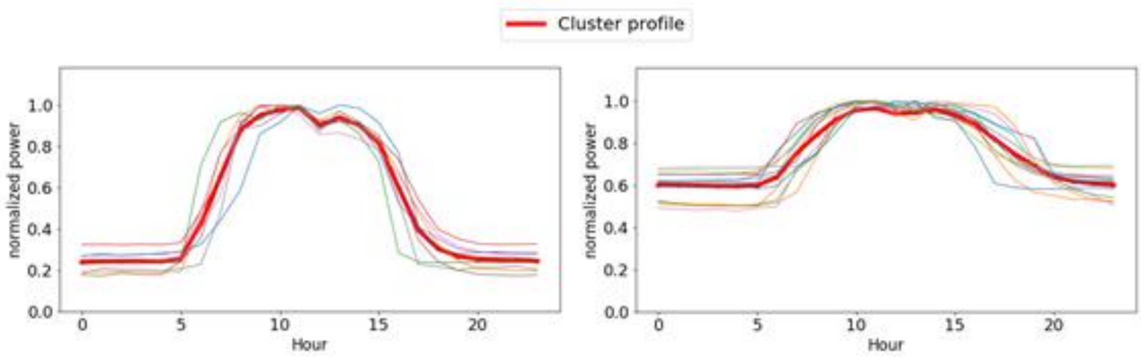
El modelo de regresión permite la predicción de perfiles eléctricos para cada uno de los algoritmos. Por lo tanto, sus rendimientos pueden compararse tal y como se ilustra en la siguiente tabla, donde se puede apreciar que el método de *stacking* (ensemble) deriva en una mejora de la precisión:

	MAPE medio [%]	SD MAPE [%]	sMAPE medio [%]	SD sMAPE [%]
Regresión lineal	90.64	61.36	74.38	29.84
Máquina de soporte vectorial	56.47	40.76	46.72	20.60
Bosque aleatorio	44.24	33.51	36.04	20.55
Árbol de decisión con p. de g.	115.03	122.53	57.13	32.90
Red neuronal	63.73	51.14	58.98	43.25
Ensemble	39.36	28.98	32.79	17.90

A continuación, se muestra un ejemplo de la predicción de un perfil eléctrico.



Para el segundo enfoque, dos de los cinco grupos de perfiles eléctricos normalizados obtenidos en el agrupamiento se ilustran en la siguiente figura. Debido al reducido tamaño del conjunto de datos, se generan errores relativamente grandes en este paso.



Seguidamente, los perfiles normalizados se clasifican en los cinco grupos. La siguiente tabla muestra la exactitud de esta clasificación que, una vez más, se ve afectada por el reducido tamaño del conjunto de datos.

	Árbol de decisión con potenciación de grad.	**Bosque aleatorio**	**Máquina de soporte vectorial**	**Ensemble**
Exactitud [%]	35.57	42.86	14.29	50.00

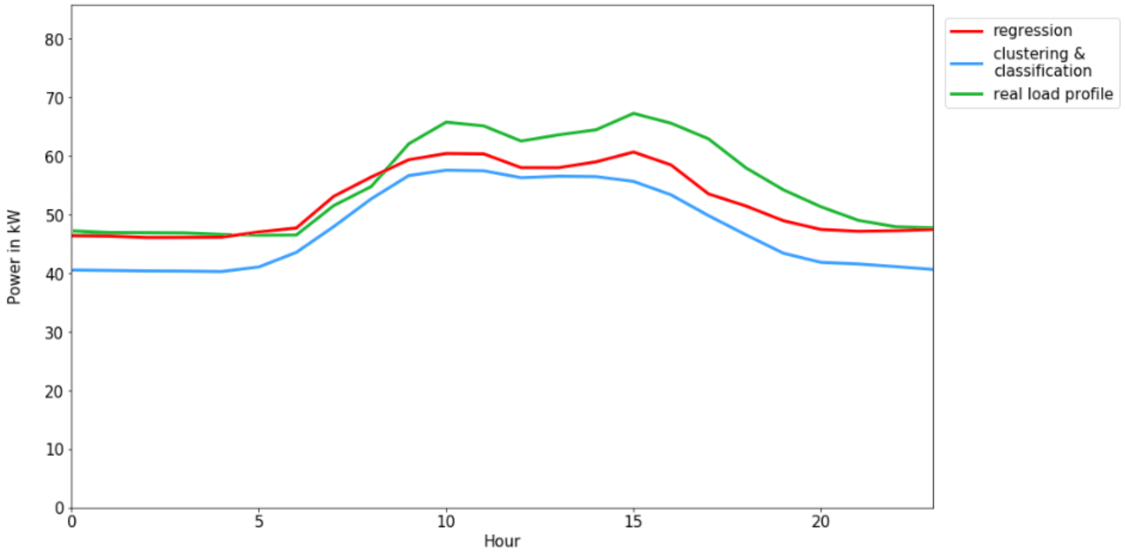
El análisis de regresión para la potencia máxima resulta una etapa crítica del modelo debido al error asociado, como puede apreciarse en la siguiente tabla.

	MAPE [%]	**sMAPE [%]**
Regresión lineal	80.59	69.37
Máquina de soporte vectorial	58.20	48.62
Bosque aleatorio	34.96	33.09
Árbol de decisión con potenciación de gradiente	41.34	34.93
Ensemble	30.63	29.36

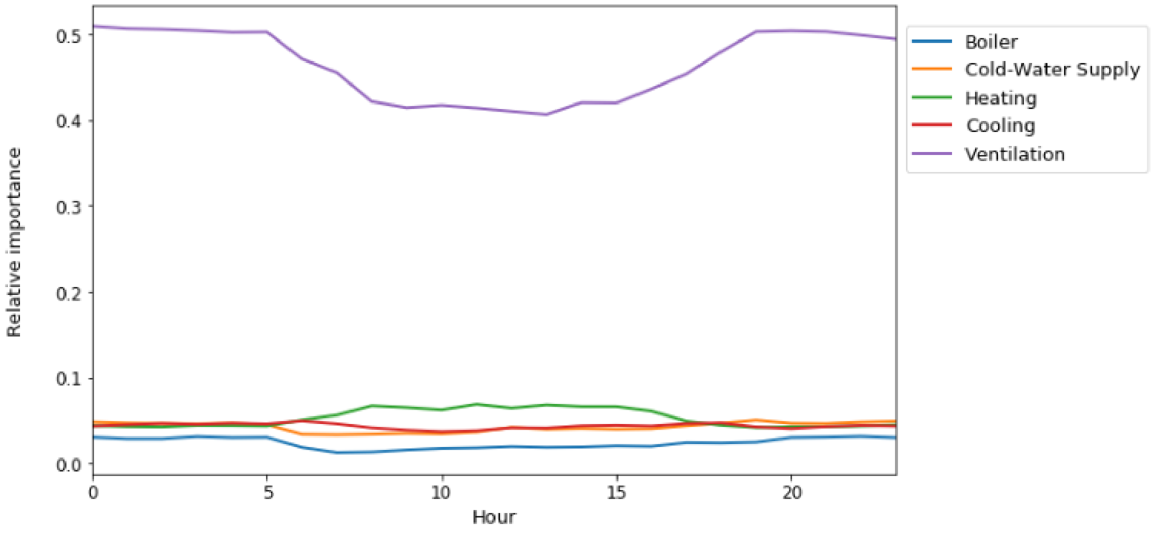
Finalmente, se combinan los perfiles normalizados y las potencias máximas previstas para obtener la predicción final del segundo modelo. En la siguiente tabla se comparan los resultados de ambos enfoques, donde se aprecia que el primer enfoque es más preciso y genera mejores resultados para la tarea y el conjunto de datos dados.

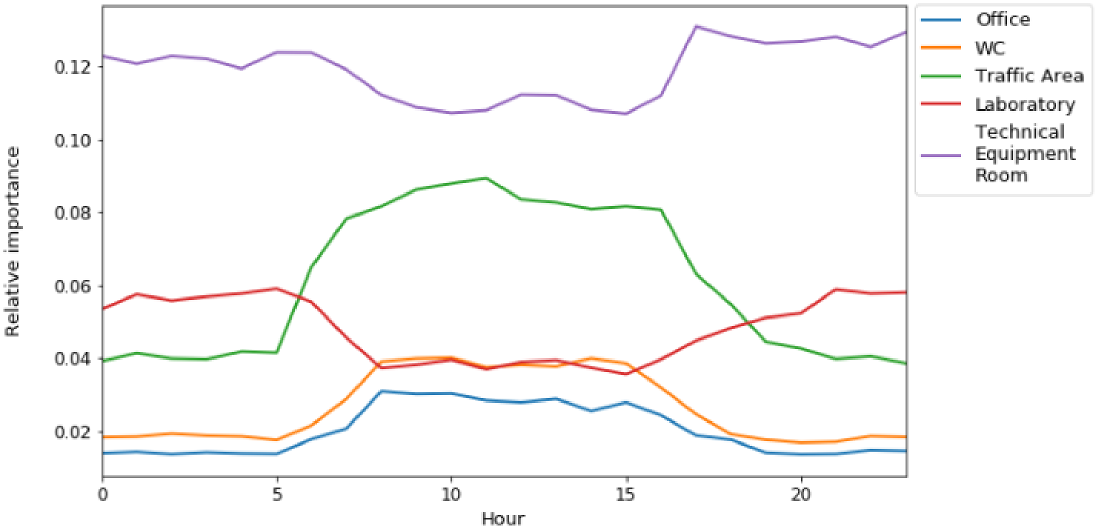
	MAPE medio [%]	**SD MAPE [%]**	**sMAPE medio [%]**	**SD sMAPE [%]**
Regresión	39.36	28.98	32.79	17.90
Agrupamiento y clasificación	45.17	33.27	38.45	19.53

En la figura representada a continuación se muestra un ejemplo de los perfiles eléctricos generados con ambos modelos:



Finalmente, el primer modelo se utiliza para extraer las dependencias entre los perfiles eléctricos y las variables de entrada seleccionadas. En las siguientes figuras, se ilustran la descomposición temporal de la importancia relativa para el modelo de cada aparato eléctrico, así como de los principales tipos de habitaciones. En ellas se puede apreciar, por un lado, la relevancia de los aparatos de ventilación y, por otro, las variaciones de la importancia relativa de las diferentes variables durante las horas laborables.





Conclusiones

Los resultados muestran que los modelos implementados pueden predecir con precisión los perfiles de demanda eléctrica, demostrando ser herramientas útiles para el propósito de modelar perfiles eléctricos de edificios no residenciales. Sin embargo, los resultados se ven afectados por la falta de calidad de los datos de entrada y, por lo tanto, se debe realizar una evaluación adecuada con datos de mayor calidad. Además, a pesar de que los modelos presentados fueron desarrollados para ser entrenados con un conjunto de datos de edificios no residenciales con el fin de predecir los perfiles de otros edificios, estos modelos pueden ser utilizados para modelar cualquier tipo de edificio, puesto que el número de variables de entrada es flexible.

En todos los casos estudiados a partir de los edificios dados, el modelo de regresión obtuvo el mejor rendimiento. Esto se debe a que ambos enfoques realizan un análisis de regresión que deriva en errores similares, pero el segundo enfoque incluye adicionalmente una agrupación y una clasificación, lo que resulta en una disminución del rendimiento. En cuanto a los algoritmos utilizados, los bosques aleatorios demostraron tener un mejor rendimiento que cualquier otro para este conjunto de datos. Además, la utilización del método de *stacking*, también implementado con un bosque aleatorio, resulta en un incremento del rendimiento con respecto a los otros algoritmos, lo cual es consistente con los resultados encontrados en otros estudios [11].

Las dependencias obtenidas muestran que los aparatos de ventilación son más importantes para el modelo que cualquier otro aparato o habitación, representando aproximadamente la mitad de la importancia de todas las variables. La descomposición temporal de las dependencias a lo largo del día muestra que, en general, pueden dividirse en tres intervalos de tiempo diferentes que corresponden a las horas laborables y no laborables. Durante las horas laborables, la dependencia entre los perfiles eléctricos por un lado, y los aparatos de calefacción y habitaciones como oficinas, zonas de tránsito y aseos por otro, aumenta significativamente, lo que significa que tienen una gran influencia en las variaciones del consumo de energía durante el día. Esta situación se debe probablemente a la relación de estos últimos tipos de habitaciones con la ocupación de los edificios, lo cual tiene un efecto directo en el consumo de energía.

A pesar de que los modelos implementados han resultado ser adecuados para realizar la tarea prevista, la metodología propuesta podría mejorarse en diferentes aspectos. Por una parte, deberían explorarse otras técnicas de selección de variables con el fin de reducir la dimensión del conjunto de datos y simplificar el proceso de predicción.

Por otro lado, la inclusión de nuevas variables en el conjunto de entrada, lo cual no requeriría ninguna modificación de los modelos implementados, podría derivar en múltiples mejoras. Por ejemplo, la inclusión de variables de ocupación de los edificios resultaría en una mejora significativa del rendimiento de los modelos, puesto que los tipos de habitaciones relacionadas con las personas han mostrado tener una gran influencia en los perfiles eléctricos. La inclusión de variables estacionales o de temperatura media podría permitir el modelado conjunto de perfiles eléctricos de diferentes estaciones, en lugar de entrenar los modelos por separado. Con el objetivo de aumentar el número de edificios que pueden ser modelados conjuntamente, la inclusión de variables geográficas implicaría la capacidad de predecir perfiles para edificios en áreas más amplias, incluso a nivel nacional, aumentando drásticamente el potencial de los modelos implementados.

Finalmente, la mejora de la calidad de los datos es, en cualquier caso, de gran importancia para realizar una evaluación adecuada de los modelos y obtener resultados significativamente mejores.

Referencias

[1] A. Mickaitytė, E. Zavadskas, A. Kaklauskas and L. Tupėnaitė, "The concept moel of suistainable buildings refurbishment", International Journal of Strategic Property Management, vol. 12, no. 1, pp. 53-68, 2008.

[2] Deutsche Energie-Agentur, "Der dena-Gebäudereport: Statistiken und Analysen zur Energieeffizienz im Gebäudebestand", 2016.

[3] European Union Energy Initiative Partnership Dialogue Facility (EUEI PDF), "Energy and Climate Change Adaptation in Developing Countries", 2017.

[4] A. Facci, V. Krastev, G. Falcucci and S. Ubertini, "Smart integration of photovoltaic production, heat pump and thermal energy storage in residential applications", Solar Energy, 2018.

[5] J. Massana, C. Pous, L. Burgas, J. Melendez and J. Colomer, "Short-term load forecasting in a non-residential building contrasting models and attributes", *Energy and Buildings*, vol. 92, pp. 322-330, 2015.

[6] Y. Ge, C. Zhou and D. Hepburn, "Domestic electricity load modelling by multiple Gaussian functions", *Energy and Buildings*, vol. 126, pp. 455-462, 2016.

[7] F. McLoughlin, A. Duffy and M. Conlon, "A clustering approach to domestic electricity load profile characterisation using smart metering data", Applied Energy, vol. 141, pp. 190-199, 2015.

[8] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[9] W. McKinney, “Data structures for statistical computing in python,” in Proceedings of the 9th Python in Science Conference, S. van der Walt and J. Millman, Eds., 2010, pp. 51–56.

[10] S. van der Walt, S. C. Colbert, and G. Varoquaux, “The numpy array: A structure for efficient numerical computation,” Computing in Science & Engineering, vol. 13, no. 2, pp. 22–30, 2011.

[11] F. Divina, A. Gilson, F. Goméz-Vela, M. García Torres and J. Torres, "Stacking Ensemble Learning for Short-Term Electricity Consumption Forecasting", Energies, vol. 11, no. 4, p. 949, 2018.

GENERATION OF ELECTRICITY LOAD PROFILES FOR NON-RESIDENTIAL BUILDINGS USING STATISTICAL METHODS OF MACHINE LEARNING

Author: Elechiguerra Batlle, Daniel.

Director: Richarz, Jan.

Collaborating entity: RWTH Aachen University.

PROJECT SUMMARY

Introduction

The energy sector has been a major concern in recent years due to its implications on climate change. Governments are making an increasing effort to achieve sustainable development within this sector, on the one hand, through the expansion of renewable energies and, on the other hand, a more efficient use of energy. The building sector accounts for 40% of the final energy consumption in the European Union, and this demand is expected to increase [1]. In Germany, non-residential buildings account for 37% of the energy consumption of the building stock [2]. Moreover, new constructed buildings represent a small fraction of the entire building stock. Therefore, existing non-residential buildings are a relevant target to achieve the sustainable development.

In order to achieve this ambitious goal, the main solution relies on the integration of renewable energies into energy systems by substituting the current centralized non-renewable generation by a distributed more efficient and reliable generation, mainly based on renewable energies [3]. Such a distributed generation directly depends on the adaptation of renewables energies to the energy consumption of distributed buildings. This building energy planning involves selecting the optimal combination of devices such as photovoltaic panels, heat pumps or combined heat and power devices that are able to supply the energy needs of the building while minimizing greenhouse gases emissions and maximizing the energy efficiency of its operation [4].

The modernization and refurbishment of existing buildings for the mentioned planning is only possible when the energy consumption behavior of a building, the load profile, is known. However, these buildings do not usually have a measured load profile, so it can only be estimated. In consequence, precise models are needed in order to be able to estimate the characteristic load profiles of existing buildings, without the need of measuring its power consumption [1]. This task, which is the focus of this work, is referred as *modeling* in order to distinguish it from *forecasting*, which involves the prediction of future energy consumption. In the latter, a learning algorithm uses information from the past is used in order to predict future outcomes and it typically consists on a longitudinal approach, since it generates predictions on the same buildings used to train the model [5].

Several models have been proposed in the area of load profiles modeling, that can mainly be divided into two different approaches. On the one hand, bottom-up approaches are based on the analysis of the individual end-use appliances behavior by determining the probability distribution of their energy consumption. These distributions are combined with building-specific variables to create multiple subsystems. Ultimately, the different subsystems are aggregated into a complex system that can model the aggregated load profile of the building. On the other hand, top-down approaches start from the aggregated consumption of buildings and are able to extract the relationship between the energy consumption and the input variables, gaining information about them. The idea behind top-down models is to break down the initial profile into its compositional sub-systems from where dependencies can be derived. For these cases, a predictive algorithm is normally implemented to conduct the task.

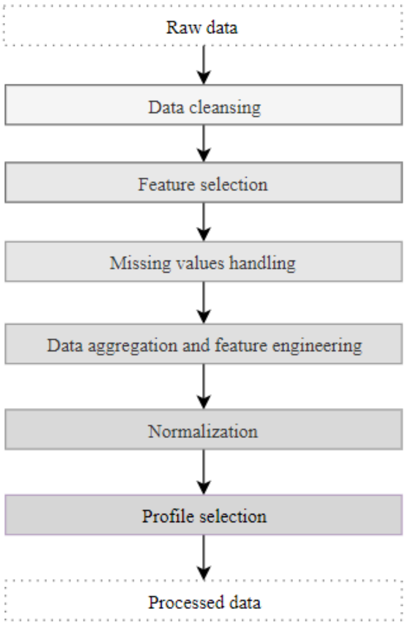
Regarding the modeling of load profiles, relatively simple top-down models have been developed and mainly focused on residential buildings. For instance, Ge et al. [6] proposed in their work a regression model based on the characterization of load profiles by a superposition of five Gaussian distributions. Then, a parameter analysis is performed to extract the dependencies between the parameters of the distributions and the number of bedrooms and occupants. McLoughlin et al. [7] proposed an alternative approach based on clustering techniques such as k-means and k-medoids. In this work, load profiles from individual households are clustered into profile classes. Then, these classes are linked to household characteristics by means of a logistic regression model which, in a final stage, is used to classify new households into the obtained profile classes according to their characteristics.

In contrast, more complex methods have been developed for the forecasting of energy consumption. Despite forecasting is not the object of this work, methods and approaches used for forecasting generally apply to the modeling of load profiles. Within these methods, several algorithms have been proposed, being linear regression, support-vector machines, artificial neural networks, random forests and gradient-boosting decision trees the most widely used. In addition, recent studies have proposed the use of ensemble learning methods such as stacking in order to improve the accuracy of the predictive model.

Since the development of accurate models for the modeling of load profiles of non-residential buildings is still a key issue for distributed generation, it will be the focus of the present work. This work aims to develop a predictive model based on machine learning algorithms that, by using cross-sectional information such as appliances and room data, can accurately be used to model the load profiles of non-residential buildings without the necessity of measuring its energy consumption. A complementary purpose is to use the obtained model to analyze the dependencies between the load profiles and the given parameters.

Methodology

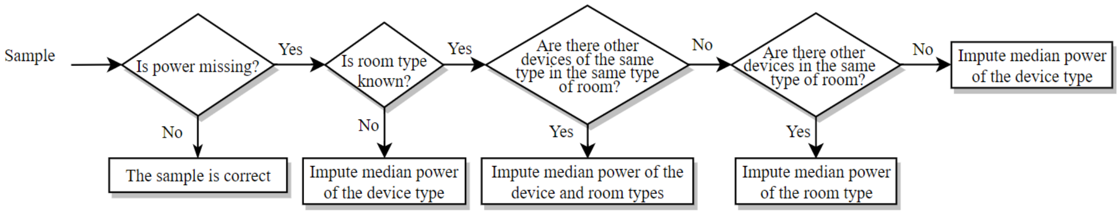
Two different top-down approaches have been developed to accomplish the intended task, in order to obtain the best possible results. Both have been implemented in Python 3.7.3, by mainly using the *Scikit-learn* [8], *Pandas* [9] and *NumPy* [10] libraries. For both approaches, the collected data needs to be preprocessed to obtain a suitable data set for the models, and the procedure is illustrated in the following figure:



This data has been collected from 70 buildings of the Jülich Research Centre, containing information about their energy consumption, which has been used as output for the models, and information about the rooms and technical equipment of the building such as boilers, ventilation, cooling, heating and cold-water supply appliances, which have served as input for the model.

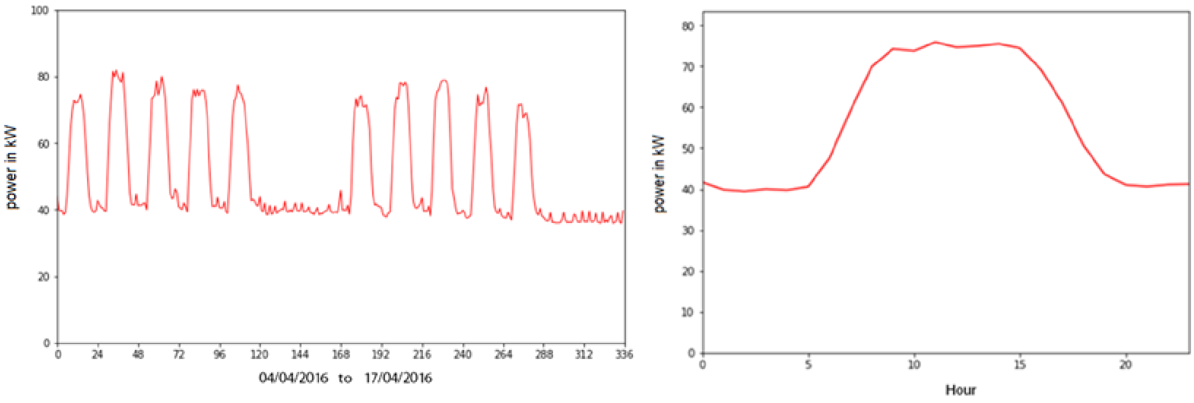
The first step, data cleansing, involves the detection and removal of outliers from the energy consumption of the given buildings. Then, relevant features are selected to be used in the implemented models.

The given data was not collected expressly for this project and thus, an important lack of information had to be faced. Regarding numeric features, 58% of the information was missing. Therefore, different multiple imputation approaches have been performed depending for each of the numeric features so as to partially compensate the lack of data. An example of this multiple imputation for the power of cooling appliances process is shown in the figure below:


Sample
Is power missing?
Yes
No
The sample is correct
Is room type known?
Yes
No
Impute median power of the device type
Are there other devices of the same type in the same type of room?
No
Yes
Impute median power of the device and room types
Are there other devices in the same type of room?
No
Yes
Impute median power of the room type
Impute median power of the device type

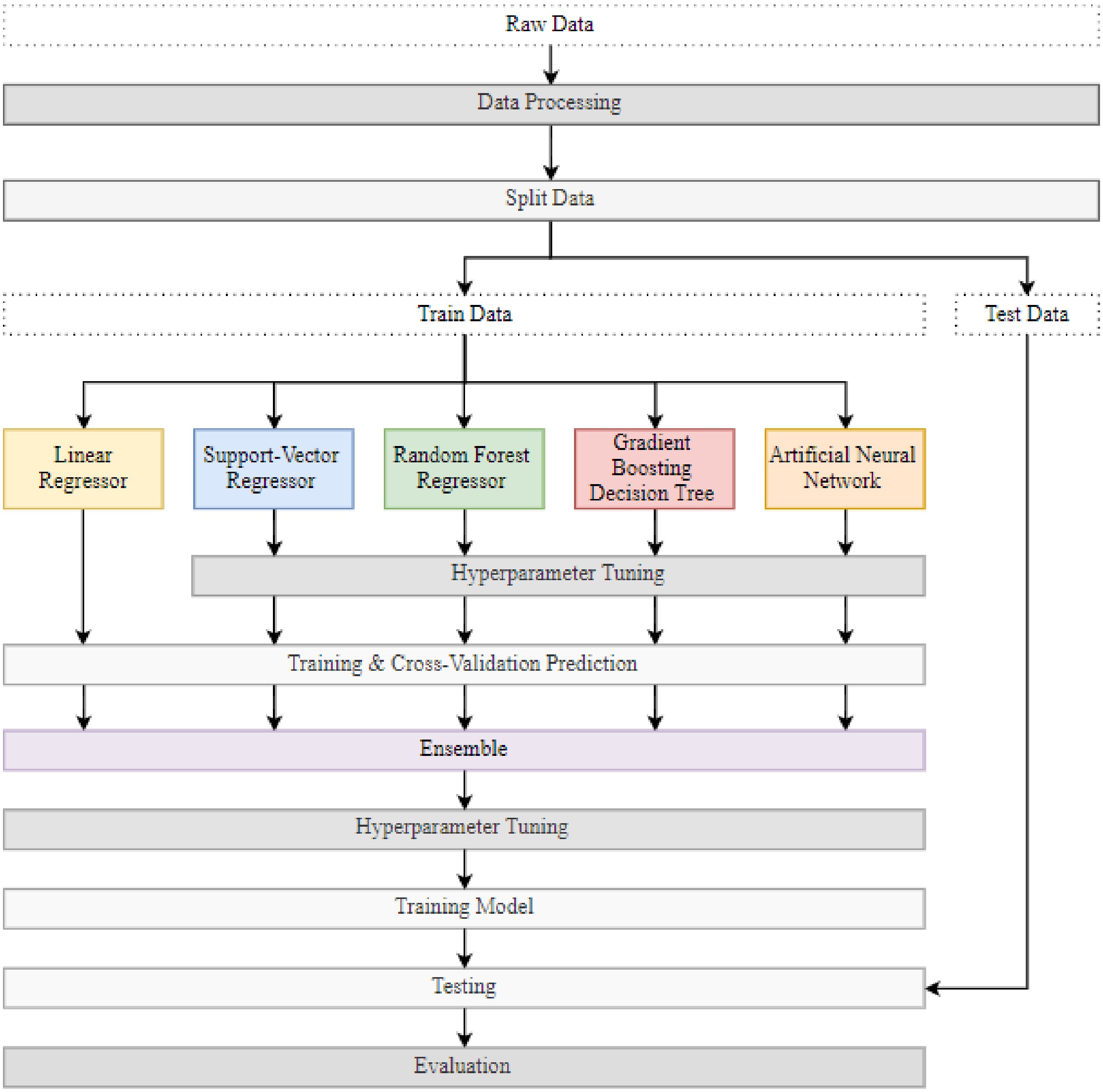
The data came sampled by appliance and room, respectively. However, since the output consists on the load profiles of the buildings, the input data needed to be aggregated into buildings. In this step, it was necessary to aggregate different features in different ways, paying especial attention to categorical variables in order to avoid a significant loss of information. Then, the resulting input set from the previous step is normalized, as it is a requirement for most of the predictive algorithms.

The last step of preprocessing consists on the selection of the load profiles used as output for the model. To obtain characteristic load profiles of the buildings, the output profiles are obtained by averaging the load profiles of workdays from two consecutive weeks, as it is depicted in the upcoming figure. To account for seasonal changes in the load profiles, the procedure has been repeated for load profiles from April, June and December.



Regarding the proposed models, two different approaches have been followed. The first model is entirely based on regression, which is performed for every hour, while the second model relies on a combination of clustering and classification of normalized load profiles and a regression for the maximum power consumption. In both cases, the stacking method has been applied in order to improve the performance. In stacking, several algorithms are used to predict the desired output and, instead of selecting the algorithm with the best performance, a stacked algorithm is trained with the outputs from the previous algorithms to make the final prediction.

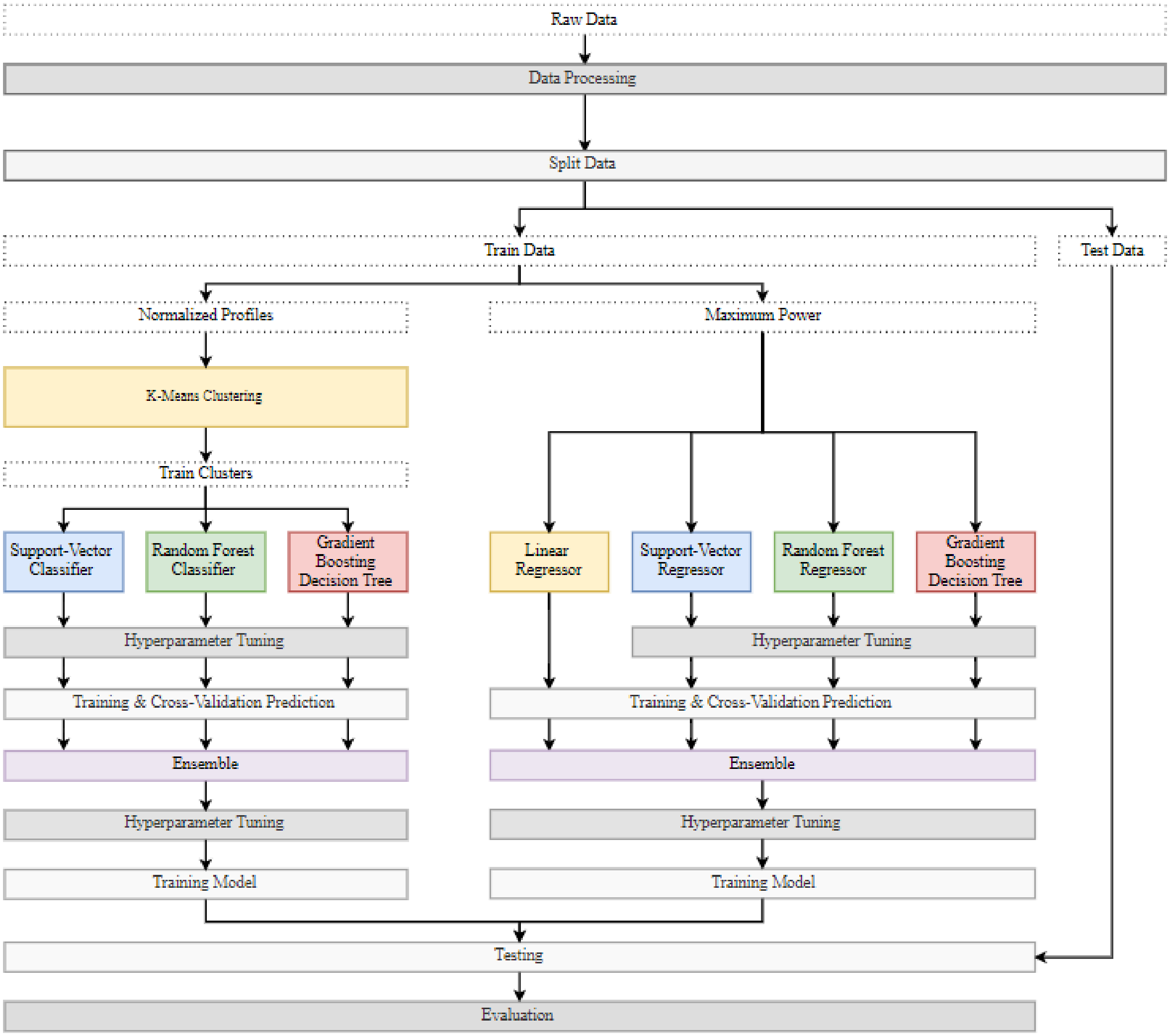
For the regression model, preprocessed data is divided into the training and test sets. Then, training data is fed into the base learner algorithms, which can be seen in the following figure. The best hyperparameters, which control the learning process, are automatically selected for each algorithm. These algorithms are trained with the given data and the cross-validation predictions for the training set are obtained. The mentioned predictions are used as input to the stacked algorithm, which is a random forest. The hyperparameters of this algorithm are selected and the algorithm is trained in the same way as before. Finally, the implemented model is used to estimate the load profiles of the test set and the results are compared to the real profiles in order to evaluate the performance of the model.



The implementation of the second approach is based on the clustering and classification of load profiles, and it is slightly more complicated than the previous one. Since the size of the available data set is small and the power range of existing buildings is relatively large, these load profiles have to be normalized and then, a regression analysis is performed to predict the maximum power of the building.

Again, preprocessed data is divided into the training test sets. Afterwards, training profiles are normalized, and a k-means algorithm is used to cluster them into five clusters. The resulting clusters of the load profile will serve as output labels to train the classification algorithms. Different classifiers are used for the subsequent task, which are shown in the following figure. The selection of hyperparameters and training of these algorithms and the stacked random forest are accomplished in the same way as explained for the previous approach.

At this point, the maximum power of the original training profiles is extracted, and a regression analysis of the maximum power is performed to predict the maximum power of the test buildings. This step is essentially equal to the previous approach. The final stage is to multiply the predicted normalized profiles by the predicted maximum power to obtain the modeled load profiles for the test set, which can be compared to the real profiles and the obtained in the previous section to evaluate the performance of this alternative approach.



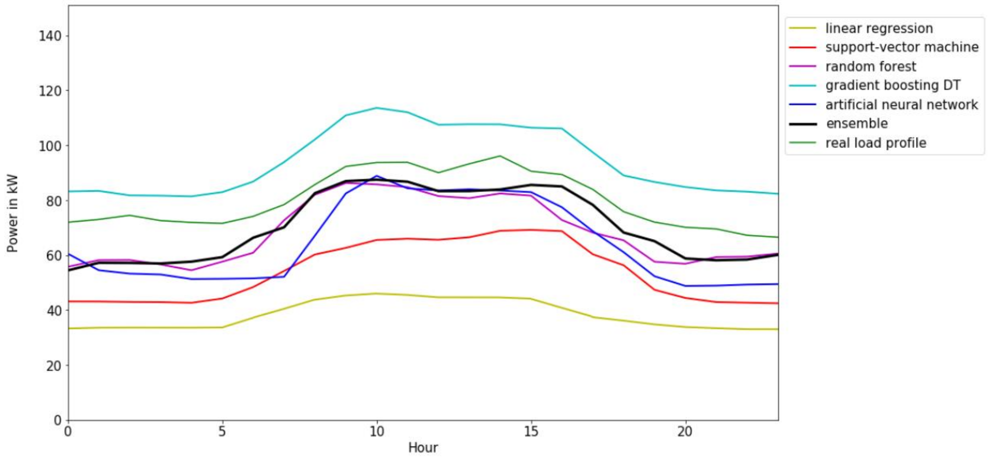
Finally, to evaluate the performance of the models, the mean absolute percentage error (MAPE) is used as accuracy metric, since it is the most popular metric for the modeling of load profiles. However, the MAPE has certain limitations that can lead to erroneous results. Therefore, an alternative metric, derived from the previous one is used in the present work, known as symmetric mean absolute percentage error (sMAPE).

Results

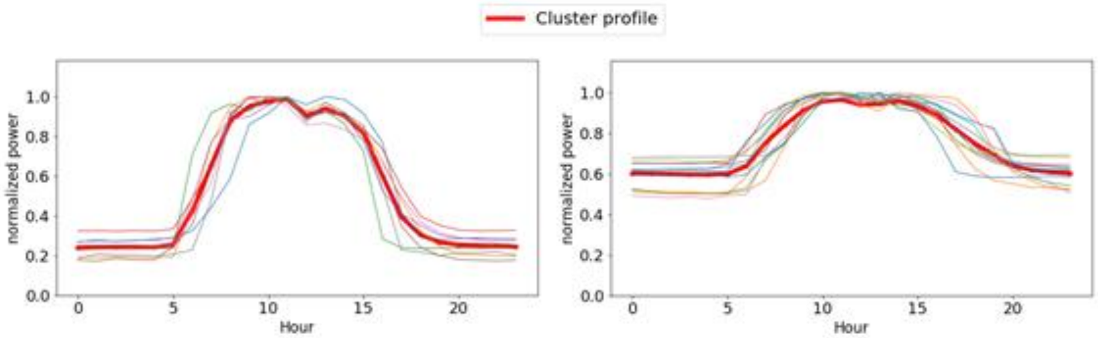
The regression approach allows the prediction of load profiles for each of the algorithms. Therefore, their performance can be compared as it is illustrated in the following table, where it can be appreciated that the ensemble method derives in an improvement of the accuracy:

	mean MAPE [%]	SD MAPE [%]	mean sMAPE [%]	SD sMAPE [%]
Linear regression	90.64	61.36	74.38	29.84
Support-vector machine	56.47	40.76	46.72	20.60
Random forest	44.24	33.51	36.04	20.55
Gradient boosting decision tree	115.03	122.53	57.13	32.90
Artificial neural network	63.73	51.14	58.98	43.25
Ensemble	39.36	28.98	32.79	17.90

The representation of an exemplary load profile is shown hereunder.



For the second approach, two of the five obtained clusters of normalized load profiles are illustrated in the following figure. Due to the small size of the data set, relatively large errors are generated in this step.



Then, normalized profiles are classified into the five clusters. The following table shows the accuracy of this classification which, again, suffers from the small size of the data set.

	Gradient boosting decision tree	**Random forest**	**Support-vector machine**	**Ensemble**
Accuracy [%]	35.57	42.86	14.29	50.00

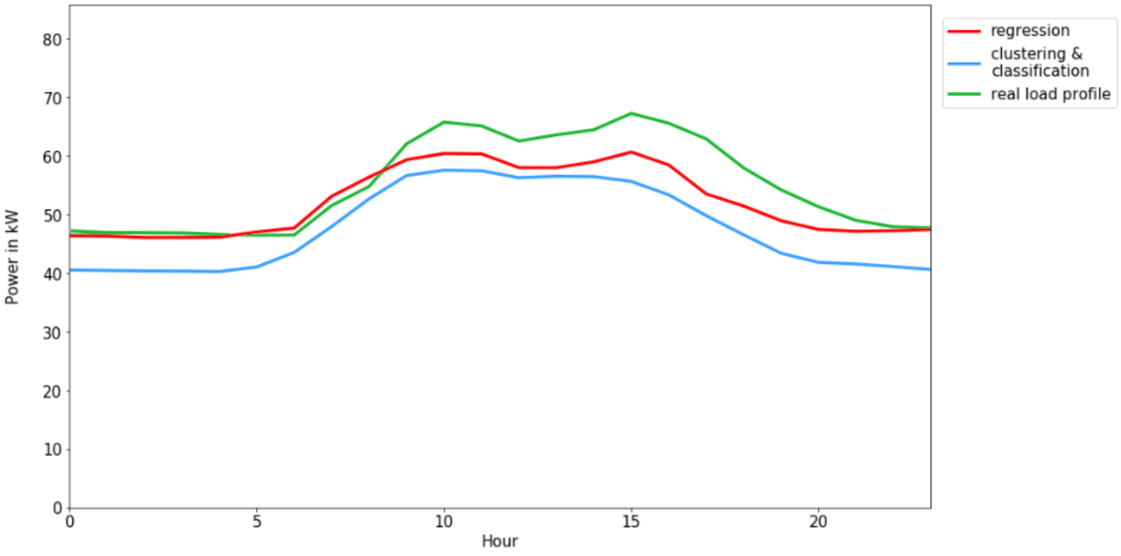
The regression analysis for the maximum power shows to be the critical stage of the model, as it can be appreciated in the following table.

	MAPE [%]	**sMAPE [%]**
Linear regression	80.59	69.37
Support-vector machine	58.20	48.62
Random forest	34.96	33.09
Gradient boosting decision tree	41.34	34.93
Ensemble	30.63	29.36

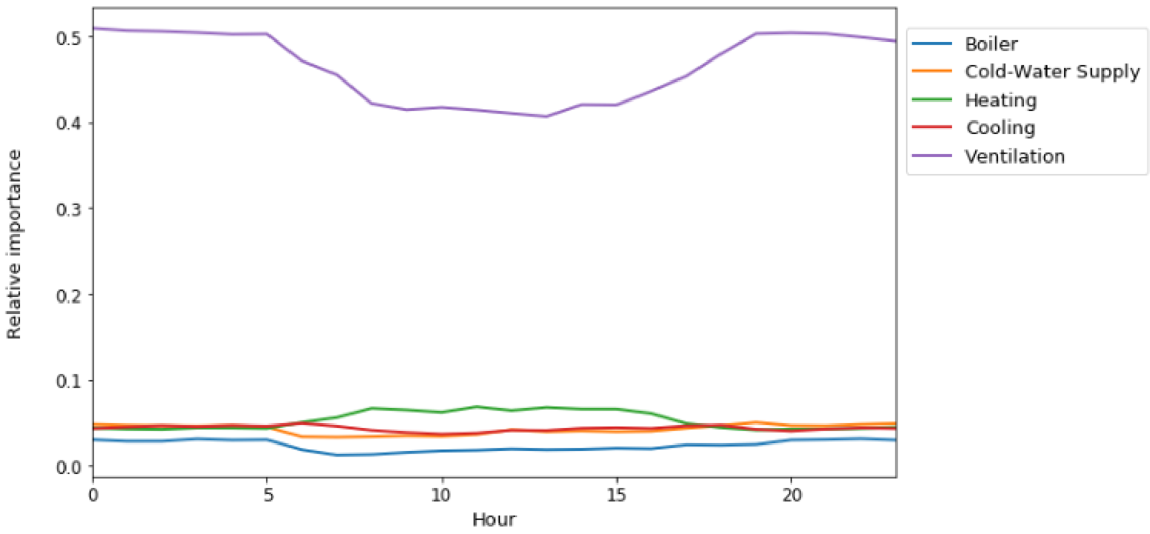
Finally, predicted normalized profiles and the predicted maximum powers are combined to obtain the final prediction of the second approach. In the upcoming table, performances from both approaches are compared, showing that the first approach is more accurate and performs better for the given task and data set.

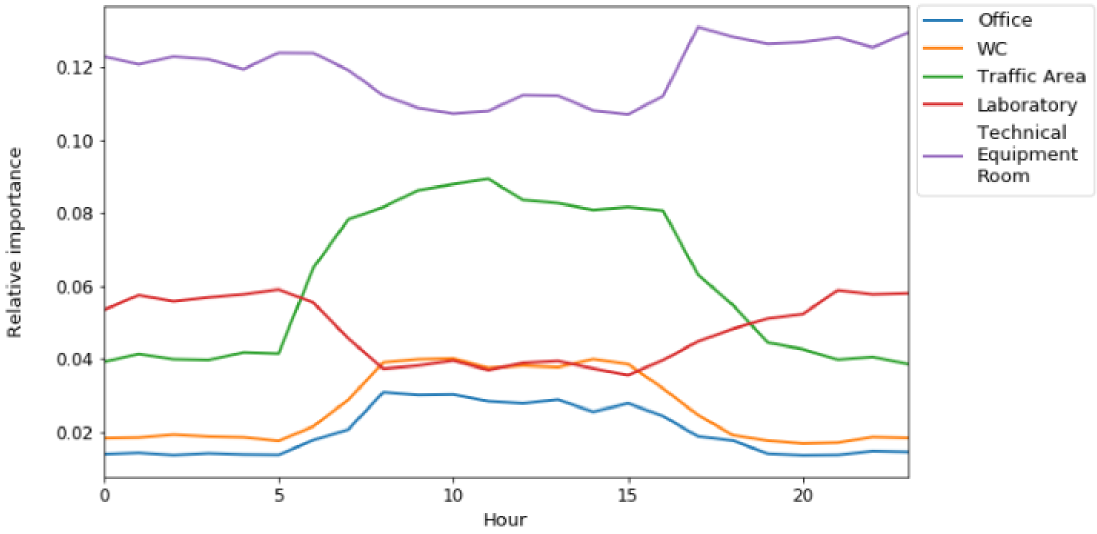
	mean MAPE [%]	**SD MAPE [%]**	**mean sMAPE [%]**	**SD sMAPE [%]**
Regression	39.36	28.98	32.79	17.90
Clustering and classification	45.17	33.27	38.45	19.53

An example of the load profiles predicted with both approaches is depicted in the represented figure underneath:



Finally, the first model is used to extract the dependencies of load profiles and the selected input features. The time decomposition of the relative importance of each appliance, as well as of the main room types is illustrated in the following figures, showing the relevance of ventilation appliances and the variations of the relative importance during working hours.





Conclusions

Results demonstrate that the implemented models can accurately predict load profiles, proving to be useful tools for the intended purpose of modeling load profiles. However, performances suffer from the lack of quality of the input data and thus, a proper evaluation should be made with higher quality data. In addition, despite the presented models were developed to be trained with a set of data from non-residential buildings in order to predict the load profiles of other buildings, they can be used to model any type of buildings, since the number of input parameters is flexible.

For all studied cases from the given buildings, the regression model obtained the best performance. This is due to the fact that both approaches perform a regression analysis which derives in similar errors, but the second approach includes the classification and clustering steps, resulting in a decrease of the performance. Regarding the algorithms, random forests proved to perform better than any other for this specific data set. Moreover, the use of an ensemble method, also implemented with a random forest, results in an increase of the performance with respect to the other algorithms, which is consistent with other results found in the literature [11].

The obtained dependencies exhibit that ventilation devices are more important for the model than any other room or appliance, accounting for about half of the total importance of the features. The time decomposition of the dependencies throughout the day show that they can generally be divided into three different time intervals corresponding to the working and non-working hours. During the working hours, the dependency between load profiles and heating devices and rooms such as offices, traffic areas and WCs increases significantly, meaning that they have a high influence on the variations of the energy consumption during the day. This situation is probably due to the relation of the latter room types to the occupancy of buildings, which has a direct effect on energy consumption.

Even though the implemented models proved to be suitable to accomplish the intended task, the proposed methodology could be improved in different aspects. On the one hand, further feature selection techniques should be explored with the aim of reducing the dimensionality of the data set and simplifying the prediction process.

On the other hand, the inclusion of new variables in the input set, which does not need any modification of the implemented model, could derive in multiple improvements of the model. For instance, the inclusion of occupancy features would result in a significant improvement of the performance of the models, since people-related room types had a high influence on the load profiles. Including seasonal or mean temperature variables could allow the joint modeling of load profiles from different seasons instead of training models separately. With the aim of increasing the number of buildings that can be modeled together, the inclusion of geographic variables would result in the ability of predicting profiles for buildings in wider areas, even at a country level, drastically increasing the potential of the implemented models.

To conclude, the improvement of the quality of the data is, in any case, of major importance to perform a proper evaluation of the model and obtain significantly better results.

References

[1] A. Mickaitytė, E. Zavadskas, A. Kaklauskas and L. Tupėnaitė, "The concept moel of suistainable buildings refurbishment", International Journal of Strategic Property Management, vol. 12, no. 1, pp. 53-68, 2008.

[2] Deutsche Energie-Agentur, "Der dena-Gebäudereport: Statistiken und Analysen zur Energieeffizienz im Gebäudebestand", 2016.

[3] European Union Energy Initiative Partnership Dialogue Facility (EUEI PDF), "Energy and Climate Change Adaptation in Developing Countries", 2017.

[4] A. Facci, V. Krastev, G. Falcucci and S. Ubertini, "Smart integration of photovoltaic production, heat pump and thermal energy storage in residential applications", Solar Energy, 2018.

[5] J. Massana, C. Pous, L. Burgas, J. Melendez and J. Colomer, "Short-term load forecasting in a non-residential building contrasting models and attributes", *Energy and Buildings*, vol. 92, pp. 322-330, 2015.

[6] Y. Ge, C. Zhou and D. Hepburn, "Domestic electricity load modelling by multiple Gaussian functions", *Energy and Buildings*, vol. 126, pp. 455-462, 2016.

[7] F. McLoughlin, A. Duffy and M. Conlon, "A clustering approach to domestic electricity load profile characterisation using smart metering data", Applied Energy, vol. 141, pp. 190-199, 2015.

[8] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[9] W. McKinney, “Data structures for statistical computing in python,” in Proceedings of the 9th Python in Science Conference, S. van der Walt and J. Millman, Eds., 2010, pp. 51–56.

[10] S. van der Walt, S. C. Colbert, and G. Varoquaux, “The numpy array: A structure for efficient numerical computation,” Computing in Science & Engineering, vol. 13, no. 2, pp. 22–30, 2011.

[11] F. Divina, A. Gilson, F. Goméz-Vela, M. García Torres and J. Torres, "Stacking Ensemble Learning for Short-Term Electricity Consumption Forecasting", Energies, vol. 11, no. 4, p. 949, 2018.

Table of contents

Nomenclature

Formula symbols and units

Symbol	Meaning	Unit
k	Number of folds	-
β	Parameter of an algorithm	-
$f(X)$, y, A	Target variable	kW
X	Input variable	-
M	Margin	-
ϵ	Slack-variable	-
C	Capacity	-
$k(x,y)$	Kernel function	-
a	Output	-
b	Bias	-
w	Weight matrix	-
$\sigma(\cdot)$	Activation function	-
B	Number of boosting rounds	-
λ	Learning rate	-
F	Predicted value	kW
K	Number of clusters	-
m	Cluster centroid	-
Q_1	First quartile	-
Q_3	Third quartile	-

Indexes and abbreviations

Symbol	Meaning
ANN	Artificial neural network
DT	Decision tree
GBDT	Gradient boosting decision tree
JRC	Jülich Research Centre
LR	Linear regression
MAE	Mean absolute error
MAAPE	Mean arctangent absolute percentage error
MAPE	Mean absolute percentage error
MASE	Mean absolute scaled error
MSE	Mean squared error
PCA	Principal component analysis
R^2	Coefficient of determination
RF	Random forest
RMSE	Root-mean-square error
RSS	Residual sum of squares
sMAPE	Symmetric mean absolute percentage error
SVM	Support-vector machine

List of figures

List of tables

1 Introduction

The energy sector has been a major concern in recent years due to its implications on climate change, and governments are making an increasing effort to achieve sustainable development within this sector. For instance, the German government aims to reduce the non-renewable primary energy demand in 2050 by 80% with respect to 2008 through the expansion of renewable energies and a more efficient use of energy [1].

The building sector accounts for 40% of the final energy consumption in the European Union, and this demand is expected to increase [2]. In Germany, non-residential buildings account for 37% of the energy consumption of the building stock [3]. Moreover, new constructed buildings represent a small fraction of the entire building stock. Therefore, existing non-residential buildings are a relevant target to achieve a *nearly climate neutral* building stock by 2050 [1].

In order to achieve this ambitious goal, the main solution relies on the integration of renewable energies into energy systems by substituting the current centralized non-renewable generation by a distributed more efficient and reliable generation, mainly based on renewable energies [4]. The limitation is that, despite most of these renewable energies, such as wind and solar energy, are fluctuating and cannot be controlled, the energy supply must be guaranteed [5].

As a result, efficiency and reliability of the distributed generation directly depends on the adaptation of renewables energies to the energy consumption of distributed buildings. This building energy planning involves selecting the optimal combination of devices such as photovoltaic panels, heat pumps or combined heat and power devices that are able to supply the energy needs of the building while minimizing greenhouse gases emissions and maximizing the energy efficiency of its operation [6].

The modernization and refurbishment of existing buildings for the mentioned planning is only possible when the energy consumption behavior of a building, the load profile, is known. However, these buildings do not usually have a measured load profile and it can only be estimated. In consequence, precise models are needed in order to be able of estimating the load profiles of existing buildings, without the need of measuring its power consumption [2]. For this reason, several approaches have been proposed in the area of load profiles estimation, which can be classified in different ways such as modeling and forecasting tasks or bottom-up and top-down approaches.

Since load profile estimation is still a key issue for distributed generation, it will be the focus of the present work. This work aims to develop a predictive model based on machine learning algorithms that, by using cross-sectional information such as appliances and room data, can accurately be used to model the load profiles of non-residential buildings without the necessity of measuring its energy consumption. A complementary purpose is to use the obtained model to analyze the dependencies between the load profiles and the given parameters.

First, relevant studies regarding the modeling of load profiles, as well as the forecasting of energy consumption will be presented to explore the different approaches and methodologies that have been developed for these tasks. In the following chapter, a general introduction to machine learning will offer an overview of several algorithms and concepts that are necessary for the development of this work. Then, two approaches are proposed to model load profiles, both of them implemented by a combination of predictive algorithms, intending to obtain the best possible performances. The first model consists on a regression analysis for each hour of the day that results in the generation of the entire load profile. The second one is based on the clustering and classification of normalized profiles which, combined with a regression analysis of the maximum load, is able to generate the desired load profiles. The proposed models will then be applied to buildings from the Jülich Research Centre in Germany and the obtained results will be presented and discussed. The conclusions of the methodology and results as well as possible improvements and a critical view on the implemented models will close this work.

2 State of the art

2.1 Load profiles prediction: modeling and forecasting

The concept of predicting load profiles can be used to refer to different tasks related to the estimation of power consumption, each of them being addressed with different approaches. Since the term prediction can be understood as the estimation of outcomes for unseen data, generally based on the experience of given observations [7], the breadth of this definition can lead to a confusion when facing the different tasks. In consequence, a clear distinction between the two most relevant classes of tasks needs to be made.

On the one hand, when a prediction is applied to time series, where a learning algorithm uses the information of the past to predict future outcomes, the corresponding term is *forecasting*. The temporal dimension is an essential aspect of forecasting. In the context of load profiles prediction, forecasting techniques generally consist on using historical time-dependent measurements (commonly smart meter data) of a building and predicting the future power consumption of the same building within a specified period of time [8]. Forecasting typically addresses the profile prediction from a longitudinal approach, since it generates predictions on the same buildings used to train the model, as it is shown in Figure 2.1.

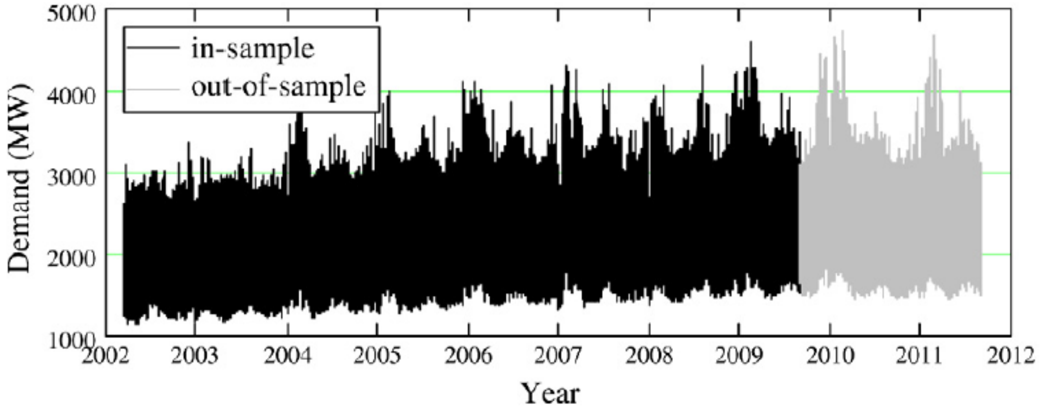


Figure 2.1: Load profile forecasting. training data (in sample) and testing data (out of sample) where predictions are to be made [9].

On the other hand, considering the temporal dimension is not always possible or desired, as it is the case of this study. Hence, for the cross-sectional approach of learning from available non-residential building load profiles in order to predict load profiles of other existing buildings by using general information (not time-dependent), the term forecasting does not seem suitable to describe the task. Instead, this kind of task is mostly referred as *modeling* [10] and is the central topic of this project.

Throughout this work, the terms *prediction* and *modeling* of load profiles will be used equally, while forecasting will be reserved to the estimation of future power consumption of the same building.

2.2 Modeling of load profiles

In the last years, the increasing interest in the modeling of load profiles has derived in the development of numerous models to deal with this task. Depending on the followed approach, the modeling of load profiles (and also forecasting), can be divided into bottom-up and top-down [11].

Bottom-up models rely on the analysis of the individual end-use appliances behavior by determining the probability distribution of their energy consumption, requiring a high level of expertise. These different distributions are combined with building-specific variables such as the number of appliances, number of occupants or the size of the building and other variables such as weather data to create multiple subsystems. Then, the different subsystems are aggregated into a complex system that can model the aggregated load profile of the building. Bottom-up models do not need historical load profiles in order to be implemented, but these are necessary to validate the results. Figure 2.2 illustrates the modeled consumption behavior of household appliances and the final bottom-up aggregated model.

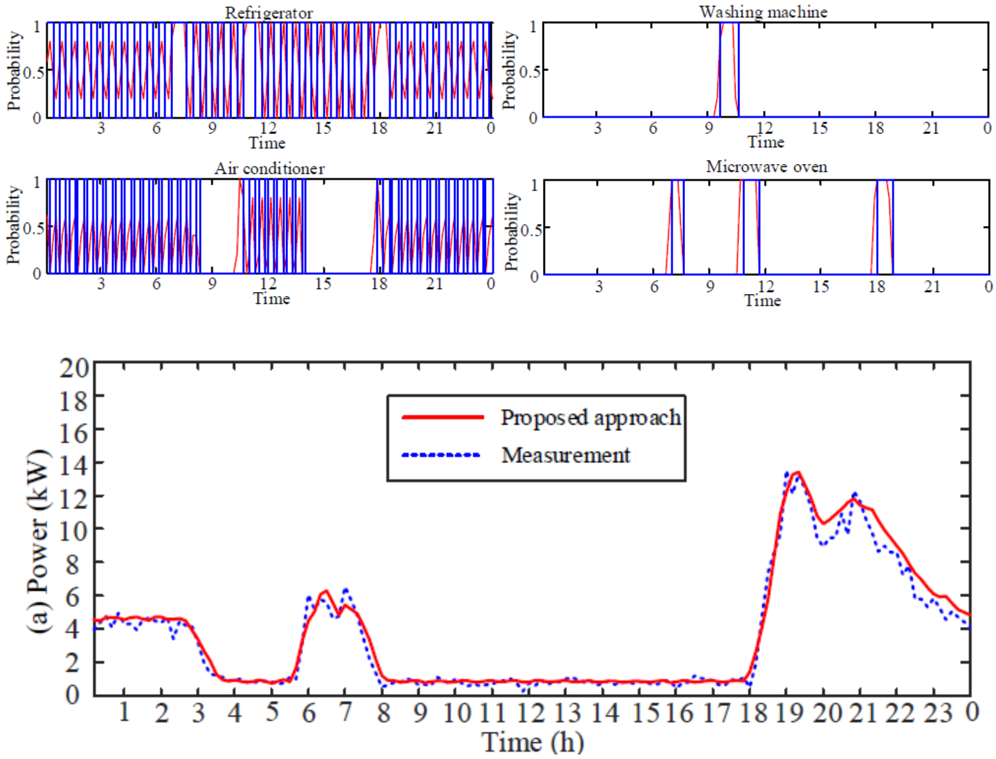


Figure 2.2: Illustration of four modeled appliances (up) and the aggregated model for a household using a bottom-up approach (down) [12].

The distinction between residential and non-residential buildings is especially significant for this kind of approach because, as the modeling of all appliances, occupancy and usage is required, very different aspects need to be considered for each case. Bottom-up models have been proposed for the modeling of both residential [13][14] and non-residential buildings [15]. However, since bottom-up models are implemented with probability distributions, this approach is not relevant for the proposed model and thus, will no longer be considered.

Top-down models start from the aggregated consumption of buildings and are able to extract the relationship between the energy consumption and the input variables, gaining information about them. The idea behind top-down models is to break down the initial profile into its compositional sub-systems from where dependencies can be derived. For these cases, a predictive algorithm is normally implemented to conduct the task, and unlike bottom-up approaches, top-down models do need historical data to implement the model. In the following, some top-down approaches will be presented.

The model proposed by Ge et al. [10] characterizes load profiles of residential buildings by a superposition of five Gaussian distributions, which is illustrated in Figure 2.3. Hence, load profiles are defined by 15 parameters, three for each Gaussian distribution: height, position and width. This results in a simplification of load profiles which, instead of dealing with 24 hours, only have to deal with 15 parameters. Within this model, a decomposition of households is made depending on the number of bedrooms and the number of occupants as variables. Then, a parameter analysis is performed to extract the dependencies between the parameters and both the number of bedrooms and occupants, which allows the modeling of individual household profiles in a simple way. The study concludes with the aggregation of individual profiles to obtain the regional load consumption of England and Wales.

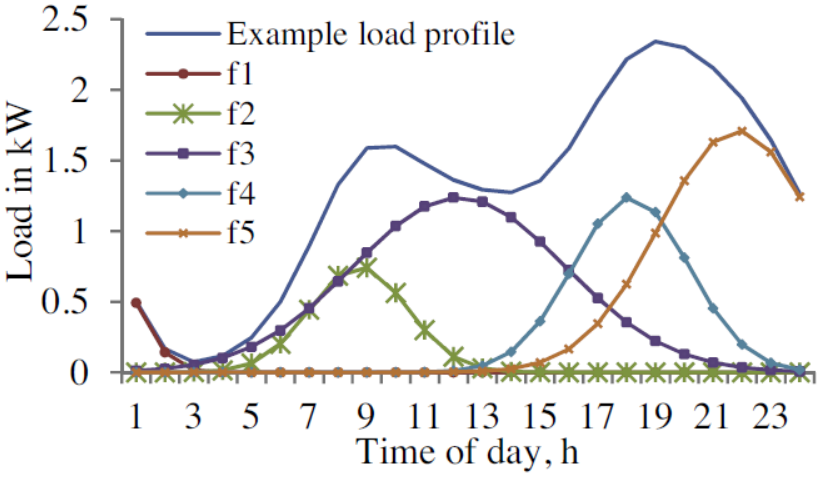
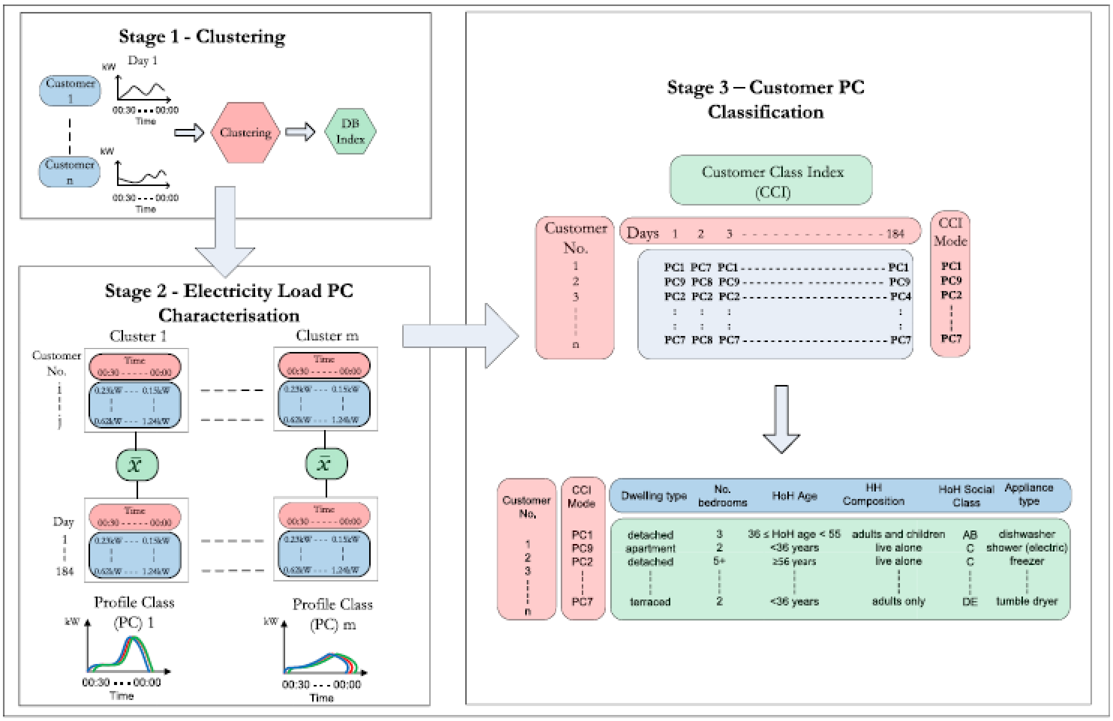


Figure 2.3: Decomposition of a load profile into five Gaussian distributions proposed by Ge et al. [10].

McLoughlin et al. [16] examined the influence of dwelling and occupant characteristics on domestic energy consumption in several Irish dwellings. However, the purpose of their work is not to model complete profiles but only aggregated parameters of households' consumption such as the total electricity consumption, maximum demand, load factor and time of use. With a multiple linear regression, household characteristics, information about appliances and other socio-economic variables are used to estimate the four parameters for other buildings. Although the followed approach is relatively simple, it provides a significant insight on the direct influence of the considered variables and consumption parameters.

In a later work, McLoughlin et al. [17] presented an alternative top-down modeling approach for the load profiles of Irish households based on clustering techniques such as k-means and k-medoids. They observe that traditional methods often result in an expensive loss of information when averaging or aggregating load profiles. In contrast, they state that data mining techniques permit a segmentation of the load profiles before entering the statistical model, allowing a dimension reduction with a minimal information loss. In the proposed model, load profiles of individual households are clustered for each day of the week into profile classes. Then, each profile class is linked to household characteristics by means of a multi-nominal logistic regression model that, in a final stage, is used to classify new households into the obtained profile classes according to their characteristics. Figure 2.4 shows these three steps. Although the aforesaid approach turns to be robust, it needs large amounts of data.



Customer No.	CCI Mode	Dwelling type	No. bedrooms	HoH Age	HH Composition	HoH Social Class	Appliance type
1	PC1	detached	3	36 ≤ HoH age < 55	adults and children	AB	dishwasher
2	PC9	apartment	2	<36 years	live alone	C	shower (electric)
3	PC2	detached	5+	≥56 years	live alone	C	freezer
n	PC7	terraced	2	<36 years	adults only	DE	tumble dryer

Figure 2.4: Top-down approach for load profile modeling proposed by McLoughlin et al. [17].

In addition to these studies, forecasting of load profiles has been the focus of numerous papers related to energy consumption. Despite forecasting is not the object of this work, methods and approaches used for forecasting commonly apply to the modeling of load profiles. Therefore, reviewing some of these works will provide a useful background for the implementation of the predictive model.

Massana et al. [8] discuss the use of different predictive models to forecast the electric consumption in a non-residential (university) building. The aim of the study is to analyze the dependencies between the electric consumption and temperature, calendar and building occupancy data. Three predictive algorithms are compared within this work:

- multiple linear regression
- multilayer perceptron (a variant of neural networks)
- support-vector regression

For this specific task, support vector regression exhibited the best performance. The resulting model cannot be used to forecast other buildings electric consumption, yet the proposed approach is of relevance since an identical approach can be followed for the modeling of load profiles. A diagram of the applied process is illustrated in Figure 2.5. First, in the data preprocessing stage, relevant attributes are selected, missing values and outliers are filtered, and the resulting data is finally normalized. Then, the data set is divided into training and test sets. The training set is used to train the three models and the one with the best performance is selected. Finally, the model is evaluated with the test set.

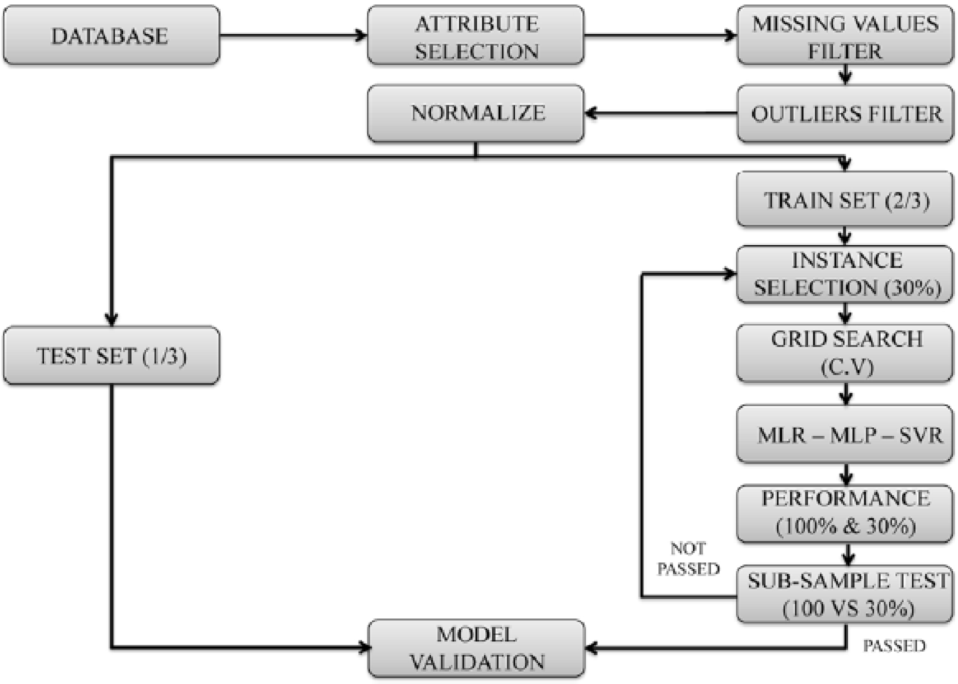


Figure 2.5: Diagram of the process followed by Massana et al. [8].

Seasonal variations in load profiles of residential buildings are studied by Wang et al. [9]. In their work, the year trend of average load profiles is approximated by a linear regression analysis. Then, load profiles of a complete year are decomposed into different seasons, as well as into workdays, Saturdays and Sundays. An average load for each segmentation is calculated and, at a final step, a quadratic regression algorithm is used to model intra-seasonal trends of average profiles. The developed model can subsequently be used to forecast the long-term energy consumption of buildings belonging to the studied population.

Additional studies stress the use of some of the aforementioned predictive algorithms in the forecasting of energy consumption such as linear regression [18] and neural networks [19], as well as alternative algorithms, for instance, random forests [20] and gradient boosting decision trees [21].

Furthermore, new methods have gained importance in the area of energy forecasting due to the significant performance improvements that can be achieved. Divina et al. [22] proposed in their work the use of an ensemble learning approach called stacking. In a similar way to the model implemented by Massana et al. [8], they use three predictive algorithms to forecast the energy consumption:

- evolutionary algorithm for decision trees
- random forest
- artificial neural network

The novelty of this work is the substitution of the selection stage, where the best algorithm is chosen to perform the predictions, by an additional predictive algorithm, a gradient boosting decision tree in this case (Figure 2.6). The latter algorithm will use the estimations made by the algorithms of the first layer and will perform the definitive prediction. Such a simple addition proves to result in a significant improvement of the accuracy compared to the three separated algorithms.

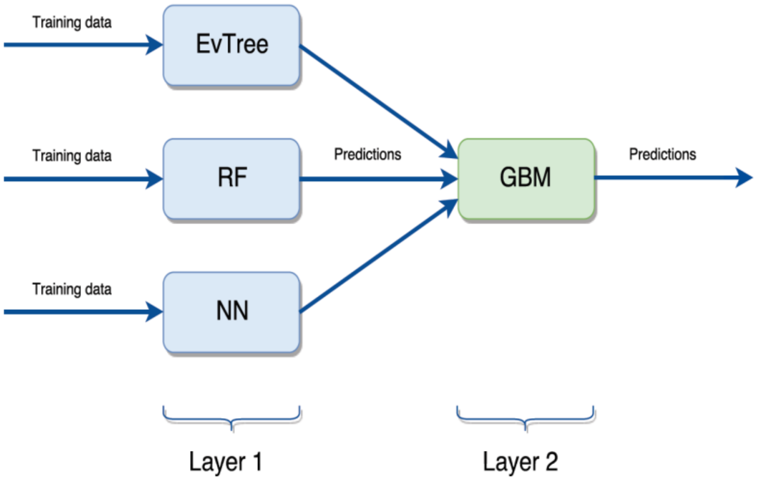


Figure 2.6: Stacking method implemented by Divina et al. [22].

Finally, Table 2.1 summarizes the works regarding the modeling and forecasting of load profiles that have been discussed in this section. It can be appreciated that, in general, modeling approaches have applied simpler methods than forecasting. However, to achieve the intended purpose of implementing accurate models that are able to predict load profiles of existing buildings, more complex methods should be used. Therefore, this work aims to offer an appropriate method to model load profiles. In the following chapter, the necessary concepts to accomplish this task will be introduced.

Work	**Approach**	**Prediction**	**Method**
Capasso et al.	bottom-up	modeling	probability distributions
Richardson et al.	bottom-up	modeling	probability distributions
Sandels et al.	bottom-up	modeling	probability distributions
Ge et al.	top-down	modeling	parameter analysis
McLoughlin et al. (1)	top-down	modeling	multiple linear regression
McLoughlin et al. (2)	top-down	modeling	clustering: k-means, k-medoids classification: multi-nominal log. regression
Massana et al.	top-down	forecasting	model selection: multiple linear regression, multilayer perceptron, support-vector regression
Wang et al.	top-down	forecasting	weekday decomposition and quadratic regression
Pedersen et al.	top-down	forecasting	linear regression
Bennett et al.	top-down	forecasting	neural network
Lahouar et al.	top-down	forecasting	random forest
Touzani et al.	top-down	forecasting	gradient boosting decision tree
Divina et al.	top-down	forecasting	decision tree, random forest, neural network, gradient boosting decision tree (stacking)

Table 2.1: Summary of presented modeling and forecasting works.

3 Theoretical foundations

Both the prediction of load profiles and their modeling are commonly based on machine learning techniques. Therefore, it is convenient to introduce in a general way what machine learning is and which are the most important branches, as well as to briefly describe the different algorithms most frequently used in the literature.

Machine learning is commonly defined, as: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E." [23]

The typical objective of a machine learning model is to learn a training set in order to be able to make accurate predictions on new data.

Machine learning algorithms can generally be divided depending on the type of data they input and output, and the task they intend to solve. Some of the most common types are [24]:

- Supervised learning: The data set contains both input and output and the algorithm must learn to predict the outputs from the inputs.
- Unsupervised learning: Only the input data is available and seeks to identify relations between samples.
- Semi-supervised learning: Only part of the output is known or labelled.
- Reinforcement learning: The algorithm learns how to act in an environment given an observation to maximize the reward.

The modeling of load profiles is a supervised learning task. Since for the training set, the output is known, and the purpose is to predict the load profile on new samples. In consequence, supervised learning will be developed in the following. Furthermore, different accuracy metrics will be presented in section 3.1.7 and discussed with the aim of selecting the most adequate metrics to evaluate the performance in relation to the objective of this project.

In addition, unsupervised learning will be briefly introduced at the end of this chapter, as it will turn to be a useful tool to address the task from a different perspective.

3.1 Supervised learning

It has been introduced that, for a supervised learning task, the algorithm infers a function that maps an input to an output, which can be used to map new samples in a reasonable way.

Within supervised learning, two kinds of task can be distinguished depending on whether the output is quantitative (section 3.1.3) or qualitative (section 3.1.4).

The previous use of the word reasonable comes from the fact that any supervised learning algorithm must face two main sources of error, the bias and the variance, giving name to the common issue known as the *bias-variance tradeoff* [25].

3.1.1 Bias-variance tradeoff and complexity

Every algorithm needs to be trained with a set of data, which should be representative for the studied population, called training set. Training an algorithm means finding the learning parameters that minimize a given loss function (error) for the mentioned training set. Then, a different set of data, the test set, is used to evaluate the expected error for unseen data. The expected generalization error, or prediction error, of an algorithm can be expressed as the sum of three terms, the bias, the variance and the irreducible error. The irreducible error is related to the noise of the target, and it is beyond the control of the algorithm, since under no circumstances will it be suppressed. Consequently, it sets a lower bound on the prediction error. In contrast, bias and variance do depend on the model and can be adjusted [25].

On the one hand, bias error is defined as a systematic error made by an algorithm due to erroneous assumptions. An algorithm has high bias when it misses significant patterns or relations between inputs and outputs. A biased estimator assumes the training error as irreducible when it is not [25].

On the other hand, variance is an error that comes from the sensitivity to noise in the training set. If an algorithm learns to model the noise in the training data as if it was a real relation between input and output, it is said to have high variance [25].

Generally, predictive models with a lower bias tend to have a higher variance and vice versa. The tradeoff of trying to minimize both errors is represented in Figure 3.1, while Figure 3.2 illustrates the issue with three examples of trained models. The complexity of a model involves the ability to model complex relations of the data. A low-complexity model is only able to fit simple relations such as linear ones while a high-complexity model is able to fit complex nonlinear relations. In Figure 3.1, a lower complexity of the model results in an unavoidable high error in both the training and test sets (high bias). In this case, the model fails on fitting the output data, which can be appreciated in the left example of Figure 3.2. Consequently, this situation is called *underfitting*. Opposed to this, Figure 3.1 shows that a higher complexity derives in a minimization of the training error, but an increase of the prediction error (high variance). In this case, the model fits the noise of the output in the training set, preventing itself from generalizing unseen data, which is illustrated in the right example of Figure 3.2. This situation is defined as *overfitting* [25].

None of these situations are desired, since both increase the prediction error. The ideal situation would be to choose a balanced model, corresponding to the example in the middle of Figure 3.2. This makes the selection of the model complexity a decisive step. A close observation of Figure 3.1 exhibits that the complexity selection cannot be done by minimizing the training error, since this common mistake would lead to overfitting. In contrast, the prediction error is related to the test error. So, by minimizing this second one, the expected prediction error would also be minimized. This is because the test samples are unseen observations for the model. However, using the test error to select the complexity of the model must always be avoided, because it is a way of *overfitting* the test set, and the obtained error would be lower than the expected for completely new data [26].

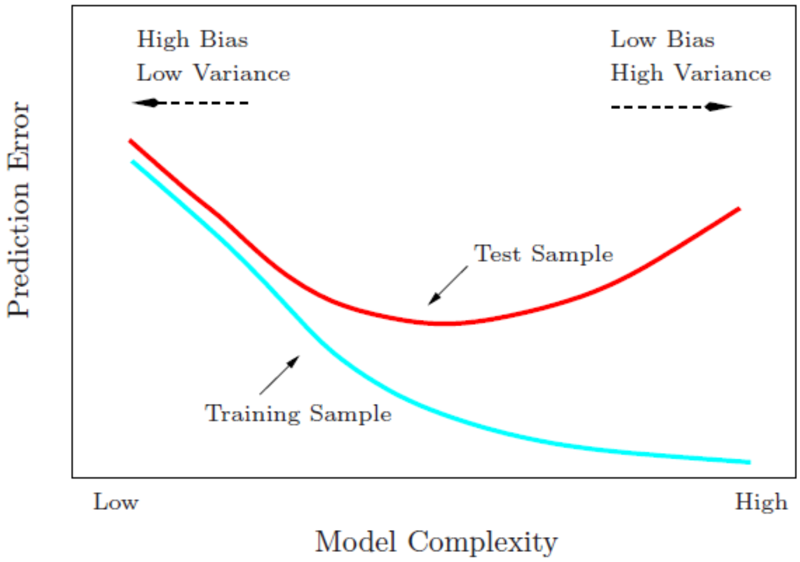


Figure 3.1: Illustration of the bias-variance tradeoff [25].

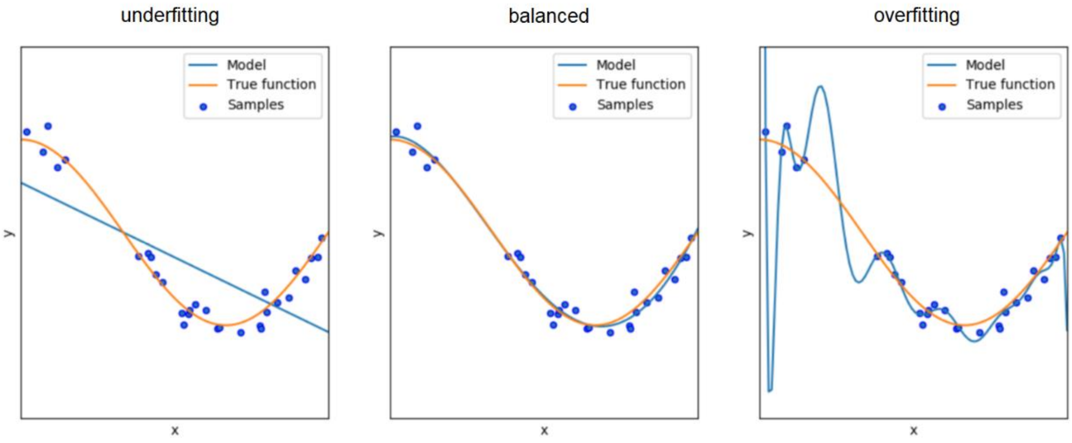


Figure 3.2: Comparison of the effect of model complexity in regression analysis [27].

For these reasons, complexity should never be analyzed neither with the training set nor the test set, but another one created expressly for this task, the validation set. The validation set is used to estimate the prediction error after training a model. Ultimately, the prediction error is evaluated with the test set.

3.1.2 Cross-validation

It has been mentioned that a validation set should be used to estimate the prediction error. However, in many situations where the size of the data set is relatively limited, as it is the case for this project (chapter 4), extracting two representative subsets for validating and testing the model can result excessively expensive in terms of the loss of training samples.

The solution is to wisely use the training set as the validation set using a technique called *cross-validation*. There are different types of cross-validation, but only k-fold cross-validation, the most common type, will be discussed [28].

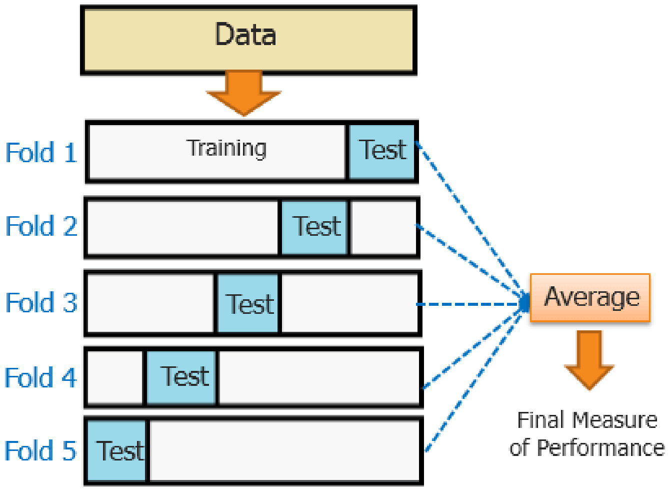


Figure 3.3: Example of a 5-fold cross-validation.

In k-fold cross validation, the training set is shuffled and divided into k folds, as shown in Figure 3.3. Then, the model is trained using all the folds except one, which is left out to use it as validation set, where the prediction error is calculated. This process is repeated k times, each of them leaving out a different fold. Finally, the validation errors of the k folds are averaged to obtain the cross-validation error.

The greater number of folds, the more accurate and reliable the validation error is, due to the reduction of the variance. However, selecting a high number of folds can result computationally expensive since more models need to be trained, so the election of the number of folds needs to be cautiously done.

3.1.3 Regression analysis

When the output of the predictive algorithm is quantitative (numerical), the term used to define the task is regression analysis, or just regression. For this task, the algorithm must be able to model the relations between input and output by inferring a function that maps inputs (independent variables) to outputs (dependent variables) as accurately as possible. Modeled relations can range from linear and quadratic to other nonlinear relations. Figure 3.2 depicts three examples of regression models. With the modeled relations, regression analysis permits to estimate not only the expected value of the dependent variable given some values of the dependent variables but also how does this variable change when one of the independent variables is varied [25].

Prediction of load profiles, either considering their forecasting or their modeling is, in most cases, a regression task and hence, it will be a key topic for this work.

3.1.4 Classification

Tasks where the output to be estimated is qualitative, also known as categorical, and the actual category membership is known, are referred as classification. It is relevant stress the fact that true classes are previously known, and therefore serve the model to look for patterns and relations between instances belonging to the same group, that will permit the model to classify correctly unseen samples [25].

While the general outputs of a classifier algorithm are the predicted classes of a set of new samples, in some occasions, the model outputs the predicted probabilities of new samples of belonging to each of the classes, and the selected classes will normally be the ones with the highest probability [25].

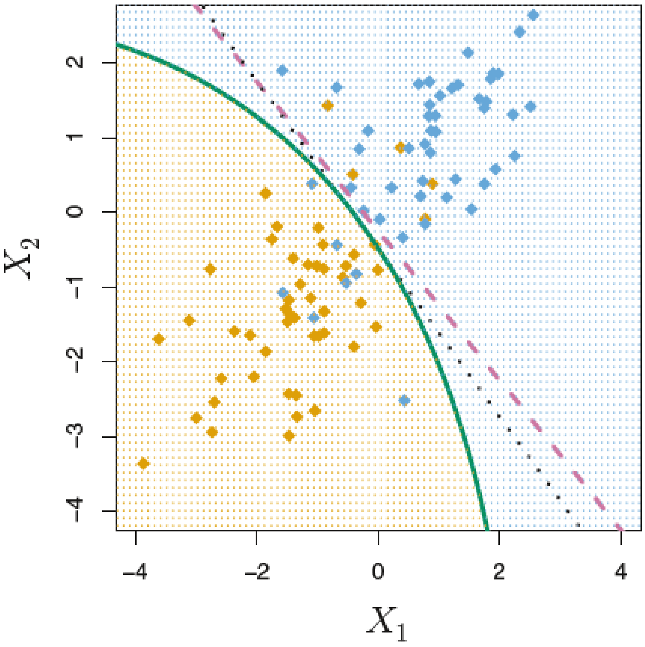


Figure 3.4: Example of classification task [25].

3.1.5 Algorithms of supervised learning

Since the appearance of machine learning, a wide variety of algorithms have been developed. One can expect any model to fit properly any kind of relation between input and output, when a suitable selection of the hyperparameters that control the learning process (see section 3.1.6), and thus of the complexity of the model, is made. Unfortunately, this is not the case, since algorithms have an inherent complexity and tackle the mapping of data in very different ways. There is not a better algorithm that performs better for every data set and every task [29]. For instance, Figure 3.5 illustrates that a simple linear model can accomplish certain tasks more accurately (top left) than a more complex decision tree (top right), while the latter may perform better in other situations (bottom right) than a linear model (bottom left). Consequently, the common procedure is to evaluate the performance of different models, and then select the one that generalizes better a concrete task.

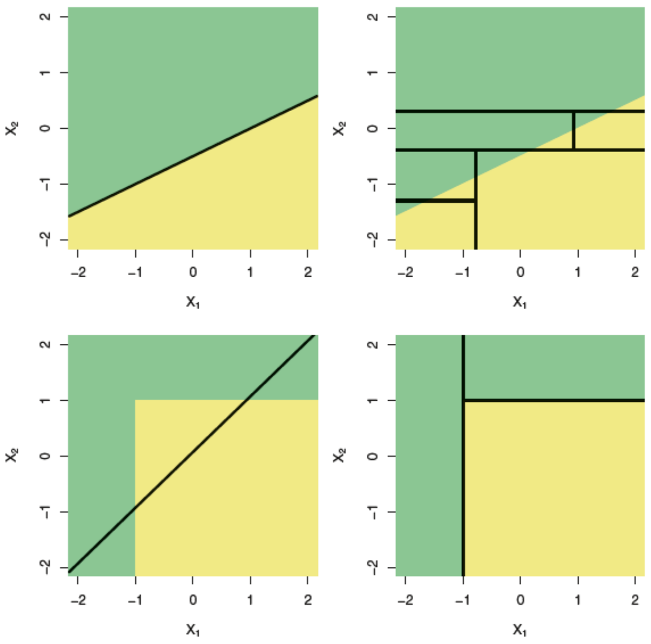


Figure 3.5: Comparison of the suitability of algorithms depending on the task [25].

Chapter 2 introduced that in the literature, the most common algorithms used to predict load profiles are linear regression, artificial neural networks, support-vector machines and random forests. In the following, these algorithms will be discussed, as well as another proposed algorithm, gradient boosting decision trees and an ensemble learning method known as *stacking*.

3.1.5.1 Linear Regression

According to Hastie [25], linear regression (LR) models assume a linear relation between input *X* and output *f(X)*, that can be expressed as follows:

$$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j \tag{3.1}$$

They were one of the first algorithms to be developed due to their simplicity, yet they are still widely used for many applications. The main advantages are that LR are simple and provide an interpretable relation between input and output. In addition, using a LR model to perform a given task, provides a reference for the performance that can be used to evaluate other algorithms since, for certain applications, a linear model can perform better than other complex models. Figure 3.6 illustrates two examples of linear regression.

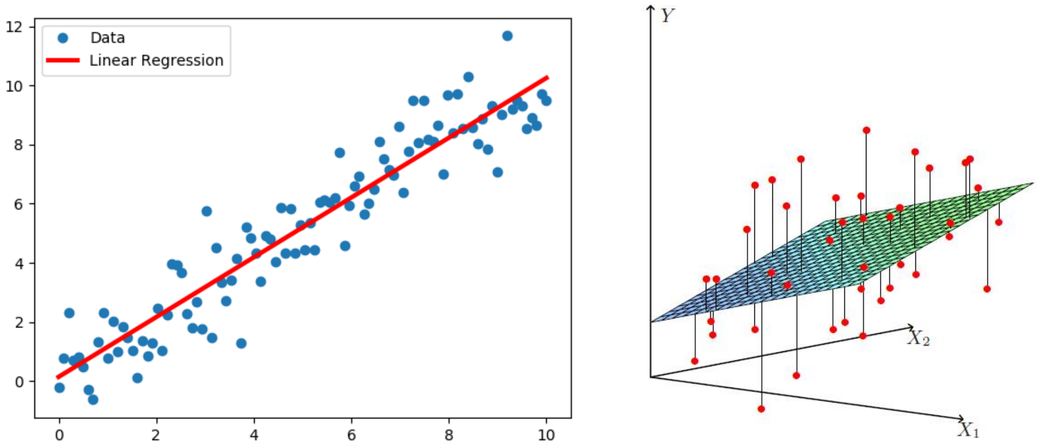


Figure 3.6: Illustration of a linear regression model with one input (left) and two inputs (right) [25].

To correctly model the relation, the algorithm must estimate the parameters β_j, $j \epsilon (0,p)$. This is usually done by using the least squares method, consisting on the minimization of the residual sum of squares (RSS):

$$RSS(\beta) = \sum_{i=1}^{N} \left(y_i - f(x_i)\right)^2$$

$$RSS(\beta) = \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 \qquad (3.2)$$

where $\beta = (\beta_0, \beta_1, . . ., \beta_p)^T$.

The RSS defined in Formula 3.2 can also be expressed as:

$$RSS(\beta) = (y - X\beta)^T (y - X\beta). \qquad (3.3)$$

Since the objective is to minimize the RSS, Formula 3.3 can be differentiated and equaled to zero, resulting in Formula 3.4, from where the estimation of the parameters can be obtained (Formula 3.5).

$$X^T(y - X\beta) = 0 \qquad (3.4)$$

$$\hat{\beta} = (X^T X)^{-1} X^T y \qquad (3.5)$$

This solution shows that a LR model can be directly implemented, and no hyperparameters need to be selected (section 3.1.6).

3.1.5.2 Support-Vector Machines

Support-vector machines (SVMs) were originally developed for classification tasks [30]. SVMs are based on the idea that, for any classification problem, only data points close to the boundary which separates the classes (boundary points) should be considered, since samples far from it will be correctly classified. The separation of classes is performed by the division of the input space with a hyperplane. However, an infinite number of hyperplanes can separate two non-overlapping classes. Consequently, SVMs seek to find the hyperplane that maximizes the distance between the boundary points of each class and the hyperplane. This distance is known as *margin*, and the resulting hyperplane is called the *maximum-margin hyperplane*. This is illustrated in Figure 3.7 (right), where the separating hyperplane, which is a line in this case, the boundary points and the margin can be appreciated.

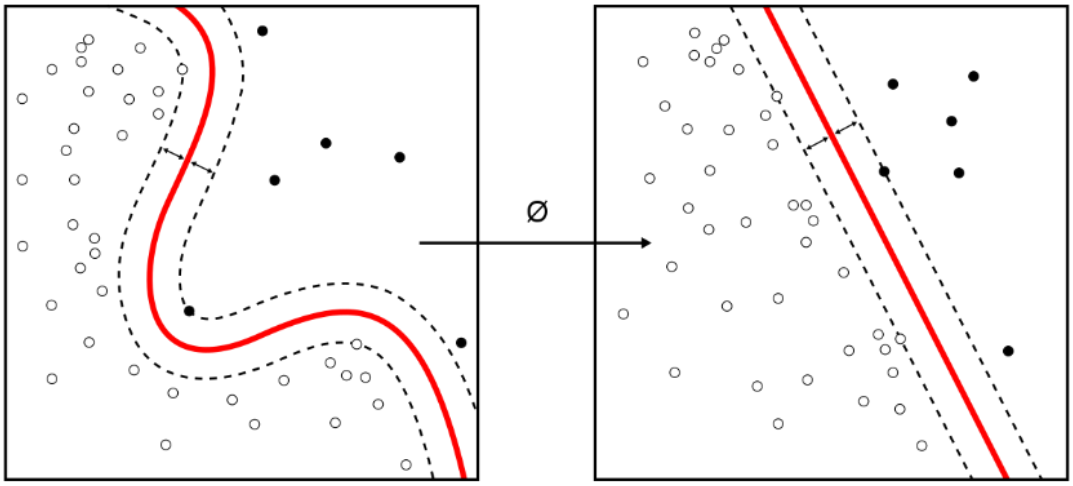


Figure 3.7: Nonlinear (left) and linear (right) of support vector machines [30].

The procedure to obtain the hyperplane is as follows [30]:

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{maximize}}\, M \tag{3.6}$$

$$subject\ to \sum_{j=1}^{p} \beta_j^2 = 1, \tag{3.7}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M\ \forall i = 1, \ldots, n. \tag{3.8}$$

where $\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} = 0$ is the separating hyperplane and M, the margin.

The interpretation of Formula 3.8 is that the distance from all points to the hyperplane must be greater than the margin. Unfortunately, in most cases, classes are not completely separable, since they overlap in the input space, so an extension of the introduced algorithm is necessary. While the previous algorithms are referred as *hard-margin* SVMs, the extended algorithms are called *soft-margin* SVMs, since they allow boundary points to be on the wrong side of the hyperplane, by introducing a slack-variable ϵ.

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{maximize}}\, M \tag{3.9}$$

$$subject\ to \sum_{j=1}^{p} \beta_j^2 = 1, \tag{3.10}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M(1 - \epsilon_i), \tag{3.11}$$

$$\epsilon_i \geq 0, \sum_{i=1}^{n} \epsilon_i \geq C \tag{3.12}$$

The capacity (C) is a tuning hyperparameter that bounds the sum of the ϵ_i's and therefore determines how severely can the margin or hyperplane be violated by boundary points.

For both soft-margin and hard-margin SVMs, the separation is constraint to be linear. However, many classification tasks cannot be performed by a linear separation in the original space. To overcome this problem, various kernel functions *k(x,y)* can be selected to transform the original space into a higher-dimensional space, where the different classes are linearly separable. This kernel transformation is illustrated in Figure 3.7, from left to right.

Even though SVMs were originally designed as classifiers, modified SVMs can be applied to regression tasks, essentially modifying the objective function, and will also be used within this project. SVMs are especially suitable algorithms when there are a large number of features or few data points [31].

3.1.5.3 Artificial Neural Networks

Artificial neural networks (ANNs) are computing systems composed by a succession of layers of nodes (neurons), where the first layer is the input and the last layer is the output. The remaining layers are called *hidden layers*. Neurons from different layers are connected by edges, each of them having a corresponding weight. Figure 3.8 illustrates a representative example of a neural network with one hidden layer. A brief explanation of the operation of ANNs will be given in this section.

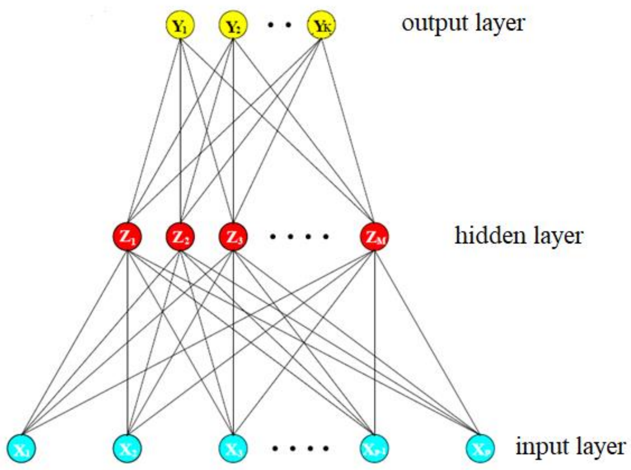


Figure 3.8: Representation of a one hidden layer neural network [25].

The output of an ANN, as well as the values of each layer, are obtained by performing operations to the values of the previous layer, starting from the input. First, the input values are multiplied by the edge weights and fed into the corresponding nodes of the next layer. The resulting inputs for each node are summed and a different constant called *bias* is added to each node. Then, the outputs of each layer are obtained by applying an activation function to each of the node values. This process is expressed in Formula 3.13 in a vectorized form, where a^l are the outputs of the lth layer, b^l the biases, w^l the weights of the edges connecting the lth layer and the previous one, and $\sigma(\cdot)$ is the activation function. The process is repeated for every layer until the output of the ANN is obtained. The described procedure is named *forward propagation*, as input values propagate through the network to produce the output [25].

$$a^l = \sigma(w^l a^{l-1} + b^l) \tag{3.13}$$

The most common method to calculate the appropriate edge weights is called *backpropagation*. In backpropagation, an initial set of weights is randomly assigned. Then, forward propagation is performed to make a prediction and the error of this prediction is calculated by using a given loss function. Then, starting from the prediction error, a backward propagation is done, analogous to the forward propagation, to calculate the error values for each node. The slope of the loss function with respect to each weight is calculated, multiplied by a learning rate and subtracted to the previous weights. This process will be applied iteratively to reduce the prediction error until the maximum number of iterations is reached, or no improvement is achieved [25].

The number of hidden layers in ANNs can range from zero (perceptron) to hundreds, or even thousands (multilayer perceptron). The latter has proven to be an extremely powerful tool for the modeling of nonlinear functions [25], and their use in the field of energy prediction is widely spread. However, ANNs generally need the use of large data sets in order to avoid overfitting and obtain good performances, which can be a drawback for certain applications.

3.1.5.4 Random Forests

A random forest (RF) is a tree-based algorithm that integrates multiple decision trees. Hence, a brief introduction of decision trees (DTs) is instructive to understand how RFs work.

DTs consist of a series of decision rules that successively split the input space into a determined number of regions. For each division of the space, a target value is assigned to all the samples belonging to that partition of the space. This value is obtained by taking the mean of the training samples belonging to the partition for regression tasks, or the mode of the samples for classification tasks. The successive splits are chosen in a way that minimize a given loss function e.g. RSS for DT regressors. The prediction of the output of new samples will be performed by applying the consecutive decision rules to each sample until a leaf node is reached, and the leaf value will be assigned to the sample [30].

Figure 3.9 shows de decision process of a DT, as well as the spatial representation of a fitted model. The similarity of this algorithm with a tree is what gives it its name. Analogously, the separation of the tree in each internal node are known as *branches*; and the terminal nodes, where no longer splitting is done, are called *leaves*. The number of levels of the tree, this is, the maximum number of splits that have to be made to reach a leaf is named *depth* [30].

DT have the advantage of being very easy to interpret and can easily handle categorical features. However, DTs do not generally perform better than other algorithms and tend to overfit the training set, they have high variance. Hyperparameters such as the maximum number of leaves and the maximum depth should be selected before training the algorithm, in order to limit its complexity and avoid overfitting [30].

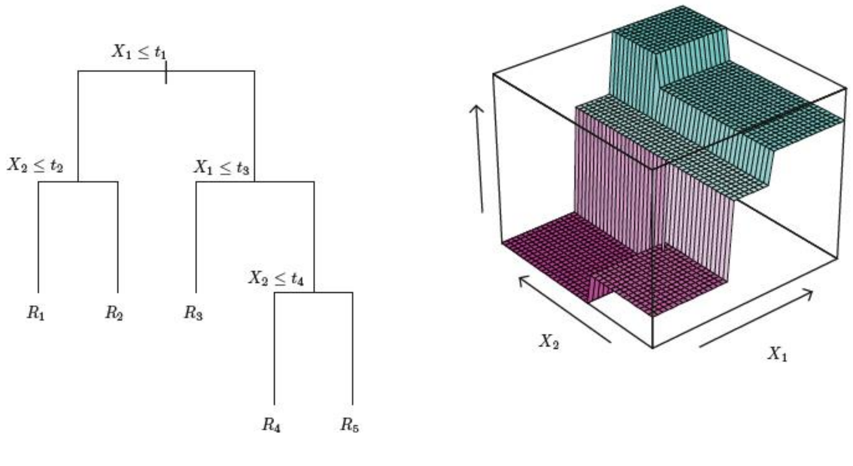


Figure 3.9: Illustration of the decision rules (left) and the partitioned space (right) of a decision tree [30].

Random forests, on their behalf, are ensemble learning algorithms. This means that they are built by the combination of multiple algorithms, which in this case are DTs. Ensemble learning algorithms can be implemented by using bagging (or bootstrap aggregating) and boosting. RTs apply the bagging technique to the DTs [25].

It has been mentioned that DTs with high depths tend to overfit. Training different models on different training sets, and then averaging the outputs would result in a decrease of the variance, however, only a training set is available. Instead, in bootstrap aggregating, different models are trained, but only using part of the training set, randomly selected with replacement. With this method, each model is trained independently with a different training set and will output different predictions. In addition to bagging, each DT is trained with only a random subset of the original features, which results in a decorrelation of the training estimators. The number of features to be used is usually equal to the square root of the total number of features. Ultimately, the output of the RF is obtained by averaging the outputs of all these DT estimators [25].

The combination of all the high-variance DTs into a RF will dramatically increment the performance of the model and overfitting is compensated. With RFs, accurate results can be expected even for small data sets. In addition to the DT hyperparameters, RFs include others such as the number of estimators and the number of features by tree [25].

The main disadvantage of RFs is loss of interpretability compared to DTs. Nonetheless, RFs provide useful tools that rank the importance of the features, which is especially practical to analyze the dependencies between the target and the features [25].

3.1.5.5 Gradient Boosting Decision Tree

Gradient Boosting Decision Trees (GBDTs) are also ensemble learning algorithms. GBDTs, unlike RFs, use boosting instead of bagging, which consists on the combination of several *weak* DTs into a single strong model iteratively [25].

In contrast to bagging, where the independent estimators are trained on different training sets, boosting involves growing trees sequentially, using the information from the previous trees, and using for each fit a modified version of the original data. The iterative algorithm, according to Hastie et al. [25], is the following:

$1. Set\ \hat{f}(x) = 0\ and\ r_i = y_i for\ all\ i\ in\ the\ training\ set$
$2. For\ b = 1,2, \dots, B, repeat:$

$(a)\ Fit\ a\ tree\ \hat{f}^b\ with\ d\ splits\ (d + 1\ terminal\ nodes) to\ the\ training\ data\ (X, r).$

$(b)\ Update\ \hat{f}\ by\ adding\ in\ a\ shrunken\ version\ of\ the\ new\ tree:$

$$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x) \tag{3.13}$$

$(b)\ Update\ the\ residuals,$

$$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i). \tag{3.14}$$

$3. Output\ the\ boosted\ model,$

$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x). \tag{3.15}$$

Here, rather than training overfitted deep trees, the boosting approach learns slowly. Initially, a DT is trained using the original output, and the resulting tree is multiplied by a learning rate (hyperparameter λ) that defines the speed of the learning process. Then, a new DT is trained using the residuals from the previous updated tree as output. The resulting tree is again multiplied by the learning rate, added to the previous tree and residuals are updated. This process is followed until the maximum number of boosting rounds is reached [25].

By using the residuals of the previous tree as output, trees tend to be rather small and will sequentially improve the performance where the previous model did not perform well. With this method, very robust algorithms can be implemented [25].

3.1.5.6 Ensemble learning: Stacking

The concept of ensemble learning algorithms has been introduced for RF and GBDT. Stacking is also an ensemble method based on the combination of different predictive models (base learners) that, in a similar way to the previous cases, are trained using the training set. However, instead of averaging or combining their outputs, a last stacked model (meta learner) is trained on the predictions of the previous models to make a final prediction. The general scheme is depicted in Figure 3.10.

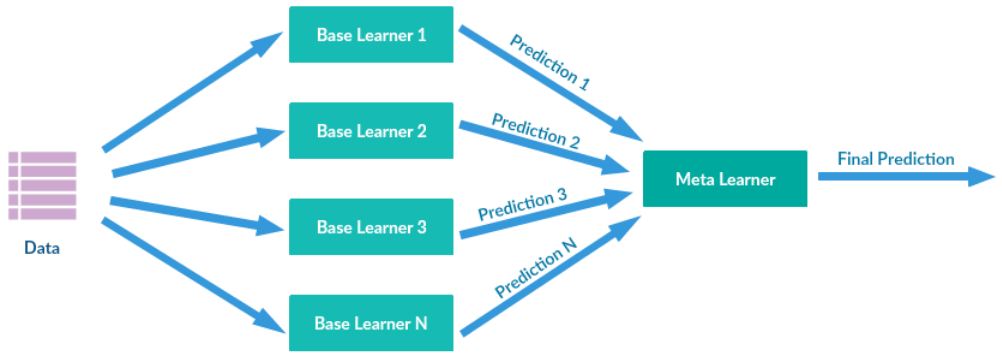


Figure 3.10: Representation of stacking.

It is a matter of importance that the input predictions to the meta learner are done on completely unseen data. Therefore, the predictions of the base learners are obtained by cross-validation using the exact same folds, and the meta learner is trained with the aforementioned predictions. This serves as a good approximation to real situations.

Whereas RFs and GBDTs use only a combination of DTs, in stacking a variety of very different algorithms is desired. This is because, as it was previously mentioned, every algorithm performs differently for the same task, and generally one will be more accurate. If stacking is not applied, the model with the best performance should be selected.

However, it might happen that other models perform better in certain situations. With a stacked meta learner, benefits from all the base learners can be combined to obtain even better results. In consequence, stacking is said to perform better than any of the base learners since, in the worst case, the performance will be equal to the one of the best learner [32].

3.1.6 Hyperparameter tuning

Together with the selection of the predictive models to be used for a given task, the selection of the corresponding hyperparameters of each model should also be performed. The difference between the parameters and the hyperparameters of a model is that, while parameters are estimated by the model itself during the training process according to an objective function (such as β_j in linear regression), hyperparameters control the learning process and need to be specified [33]. Some of them have already been mentioned such as capacity for SVMs, number of estimators for RFs and learning rate for GBDTs. This section will explore how should they be selected.

It is possible to implement any predictive model without defining the hyperparameters, since algorithms have a default value for them. Nevertheless, it is unlikely that this preselection of hyperparameters is the optimal combination because for any task and dataset, optimal hyperparameters will be different. Likewise, selecting the hyperparameters of other models found in the literature will rarely offer the best performance and therefore, a strategy to select the optimal hyperparameters must be followed. This is known as *hyperparameter tuning* [33].

The general procedure to tune the hyperparameters consists on the use of cross-validation or a held-out validation set, mentioned in sections 3.1.1 and 3.1.2. In first place, the model is trained with a combination of hyperparameters and then, the trained model is used to predict the outputs of the validation set (or training set if cross-validation is used), where the performance is evaluated. The process is subsequently repeated for different combinations of hyperparameters and finally, the combination resulting in the best validation performance will then be selected.

Although successively trying the different combinations can be done manually, it turns out to be an ineffective and inefficient task, as the number of possible combinations can grow to unmanageable quantities. In contrast, automated methods offer more efficient techniques that will lead to better results. In the most common method, called *grid search*, all the preselected possible values of the hyperparameters to be tuned must be defined. Then, all the combinations are generated (hyperparameter grid), and an exhaustive searching is made by trying each of these combinations. The preselection of possible values must be done carefully since grid search suffers the curse of dimensionality, but it can be easily parallelized [33].

3.1.7 Accuracy measurement

In order to evaluate the performance of the proposed models, it is necessary to establish an accuracy metric that fits the objectives of this project. In the literature, a large number of metrics have historically been used, which can be divided into two categories, scale-dependent and scale-independent accuracy metrics. In this section, some of the most used ones will be introduced.

Scale-dependent accuracy metrics

This type of metrics, sometimes referred as absolute or unscaled metrics, provide a measure of the difference between the unscaled real and predicted values. They are generally used for evaluating predictive models where the absolute value of the error is to be minimized. These metrics can also be useful for comparing methods on the same set of data, but they should never be used to compare the prediction accuracy of different models where the scale of the used data is significantly different. The main scale-dependent metrics are the mean squared error (MSE), root-mean-square error (RMSE) and the mean absolute error (MSE).

The MSE has been the most used metric historically [34], and it is calculated according to

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(F_i - A_i)^2, \tag{3.16}$$

by averaging the sum of squared errors, where A_i is the actual target value, F_i is the predicted value, and n is the number of samples considered. The MSE corresponds to the second moment (relative to the origin) of the error and, for an unbiased estimator (the mean of the predictions is equal to the mean of the measures), it is analogous to the variance. Because the scale of the MSE is the square of the scale of the original data, it is relatively difficult to interpret. Therefore, the RMSE is frequently preferred, since it uses the original scale.

$$RMSE = \sqrt{MSE} \tag{3.17}$$

Due to the fact that the MSE and RMSE are calculated by squaring the errors, they tend to penalize large errors more than small errors. This turns to be a problem in the case of outlying estimates, which would have very high impact on the metric, leading to a difficult interpretation of the accuracy in normal cases. To overcome this problem, the MAE can be calculated using the absolute difference of the error, making it a more robust measure against outliers, since it penalizes all errors equally. For this reason, MAE may be more appropriate than MSE and RMSE in case that the errors are not normally distributed or when larger penalizations are not required for large errors [35]. Consequently, MAE has been increasingly used in the last years for the modeling and forecasting of load profiles.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|F_i - A_i| \tag{3.18}$$

Scale-independent accuracy metrics

Measuring the difference between the target and predicted values could result inappropriate when different studies are to be compared, because the unscaled performance is calculated for specific data sets. The solution is to use scale-independent accuracy metrics, also known as relative or scaled metrics. The coefficient of determination (R^2) and the mean absolute percentage error (MAPE) are two scale-independent metrics widely used in the literature [34].

The coefficient of determination, given by

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(A_i - F_i)^2}{\sum_{i=1}^{n}(A_i - \bar{A})^2}, \tag{3.19}$$

offers a quantification of the linear relation between the target and predicted values. For general applications, R^2 normally ranges from 0 to 1. It equals to one if the predicted values match the targets, and zero in the case that the predicted values correspond to the mean of the target values. In this sense, it can be understood as the proportion of variance of the target variable that can be explained by the model. However, R^2 can become smaller than zero, if an average straight line estimates better the target. This implies that, while the R^2 metric is bounded above by one, there is not a lower bound. Very high negative values can be reached when the target has a very low variance, with very similar values, but the predictions have a significantly larger variance. This situation becomes a problem when it comes to load profile modeling, since buildings with a constant consumption will derive in very large errors, biasing the model, which will fail to model the load profiles. In consequence, R^2 is not suitable for the purpose of this work and will therefore not be considered as a metric.

A more robust metric is the mean absolute percentage error (MAPE). It has become, by far, the most popular accuracy metric for both forecasting and modeling purposes [8]. According to the following formula,

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{A_i - F_i}{A_i}\right| \cdot 100\%, \tag{3.20}$$

errors can be interpreted as a deviation of the actual target, and of the same order, relative to the target, are treated equally, as can be seen in Table 3.1. The reason for its extended use is that it provides a very easily interpretable metric while, at the same time, allows a direct comparison of different methods. Nevertheless, the MAPE also has some disadvantages [36]. On the one hand, negative errors ($A_i < F_i$) are weighted more heavily than positive ($A_i > F_i$), which can bias the model and lead to a misinterpretation of the results. This can also be seen in Table 3.1. Furthermore, negative errors are bounded by 100%, whereas positive errors are not bounded. On the other hand, very small values of the target can derive in unacceptably high errors or, in the case that the target is zero, undefined error values.

These disadvantages have led to multiple extensions of the MAPE, such as the symmetric mean absolute percentage error (sMAPE), mean absolute scaled error (MASE) or the mean arctangent absolute percentage error (MAAPE).

The sMAPE [37] aims to deal with the aforementioned disadvantages by averaging in the denominator the target and predicted values, as shows

$$sMAPE = \frac{1}{n} \sum_{i=1}^{n} \frac{|A_i - F_i|}{(|A_i| + |F_i|)/2} \cdot 100\%, \tag{3.21}$$

The main advantages of this extended metrics are that it is no longer asymmetrical, weighting equally positive and negative errors, and it us bounded between 0% (forecast equals target) and 200 % (target or prediction are zero), making it a more robust metric than MAPE. A comparison of MSE, MAE, MAPE and sMAPE, can be seen in Table 3.1. In Figure 3.11, the bounds and symmetry of both scale-independent metrics can be observed.

	A	F	MSE	MAE	MAPE	sMAPE
1 - Positive error	15	10	25	5	33.3%	40%
2 - Negative error	10	15	25	5	50%	40%
3 - Positive error	150	100	2500	50	33.3%	40%
4 - Negative error	100	150	2500	50	50%	40%

Table 3.1: Comparison of MSE, MAE, MAPE and sMAPE for different scenarios.

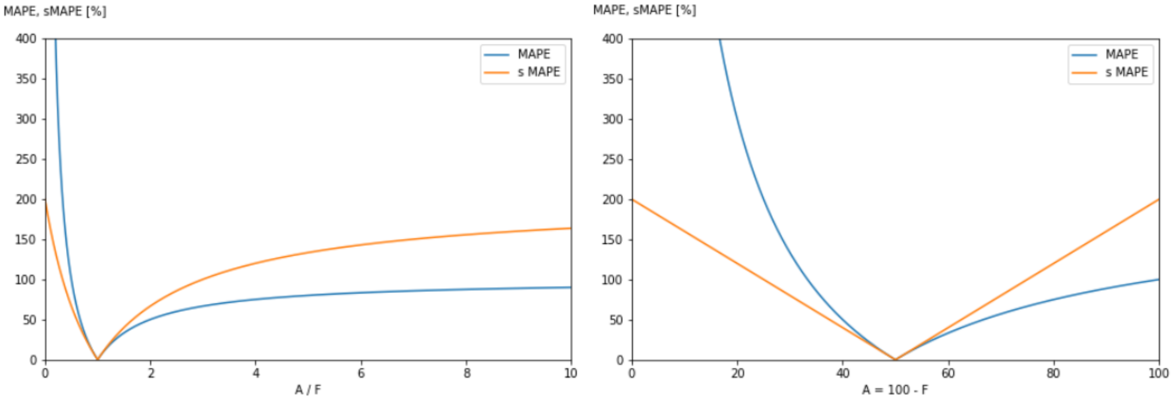


Figure 3.11: Graphical comparison between MAPE and sMAPE.

The use of MAPE as the main accuracy metric is a common practice in load profile prediction. Since, for the purpose of this work, a good comparability of different sized load profiles is needed, and a good interpretability is desired, the MAPE will be used in the following as accuracy metric. However, the sMAPE will also be included, due to its advantages relative to the common MAPE.

3.2 Unsupervised Learning

The purpose of unsupervised learning is to find relations and patterns which were previously unknown in a non-labeled data set. Depending on the type of patterns that are to be found, two main types of unsupervised learning can be defined, clustering or cluster analysis and principal component analysis. In clustering, relations between training samples are explored, creating groups of samples which are considered to be similar. Principal component analysis (PCA), another unsupervised learning algorithm, consists on an orthogonal transformation that convert correlated features into uncorrelated features. The new features resulting from a combination of the original ones, are now *aligned* with the directions of greater variance, allowing a dimensionality reduction by removing low-variance features [30].

Even though the dimensionality reduction of PCA can result a useful tool, it is not within the scope of this project and thus, only clustering will be discussed.

3.2.1 Clustering

Clusters are groups of samples which, in some sense, are more similar to each other than they are to samples belonging to other clusters. An example of clustering is presented in Figure 3.12, where data has been divided into two clusters. It is an analogous procedure to classification but where the true classes are unknown.

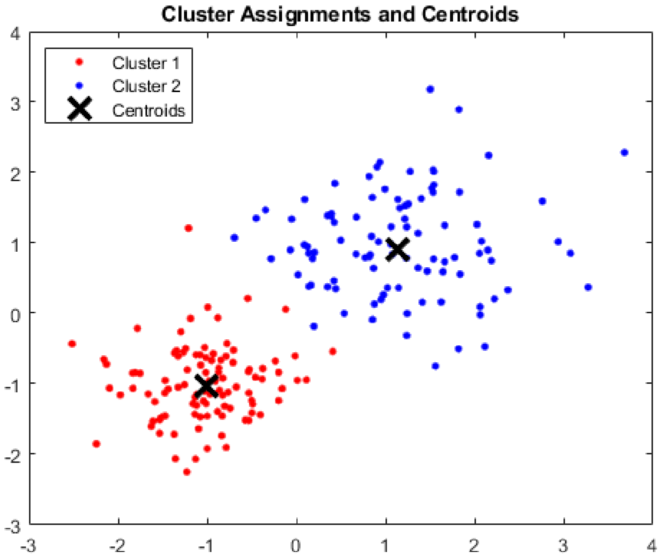


Figure 3.12: Example of a clustering task.

One of the most popular clustering algorithms is k-Means, a centroid-based algorithm that divides the space into k Voronoi cells that will define the clusters, and consists in an iterative process [25]:

An initial set of K centroids are generated: $m_1, \dots, m_K$

Repeat until no centroid variation is experienced:

1. Cluster *C(i)* is assigned for each sample based on

$$C(i) = \underset{1 \le k \le K}{\operatorname{argmin}} \|x_i - m_k\|^2 \tag{3.22}$$

2. Centroids are updated

$$m_k = \frac{1}{|S_k|} \sum_{x_i \in S_k} x_i, \tag{3.23}$$

where S_k is the set of all the samples assigned to cluster k.

One aspect to point out from this algorithm is that the number of clusters has to be specified in advance as input to the model. An appropriate value of k must be chosen since a too small number of clusters will not be able to model all the groups present in the data and a too large value will create more clusters than needed, separating samples into different clusters when they should belong to the same group [3.8]. Therefore, the number of clusters should be iteratively increased until no significant reduction of the clustering error is achieved. Figure 3.13 shows an example where three clusters should be selected.

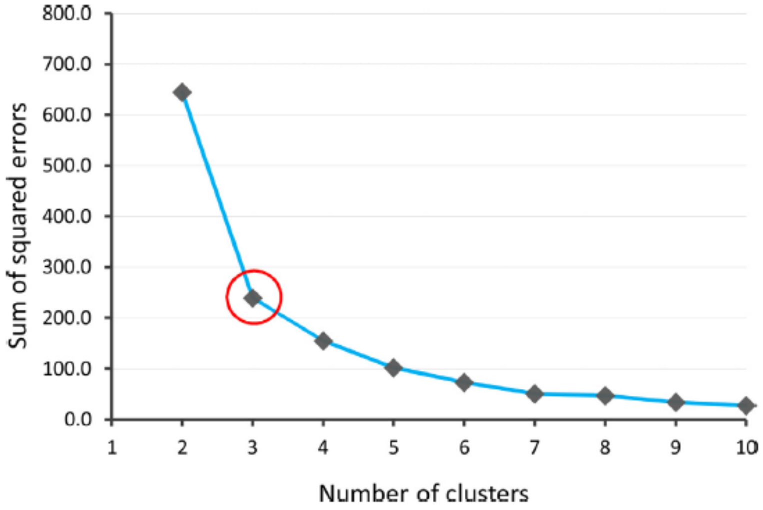


Figure 3.13: Clustering error depending on the number of clusters [39].

According to Formula 3.22, for the task of clustering load profiles, k-Means algorithm will cluster samples by minimizing the squared error between each profile and the centroid's profile. It is also a useful method to detect outlier load profiles that are not similar to the others.

4 Methodology

Once the fundamental concepts for the development of the work have been introduced, this chapter explains the preparation and implementation of the predictive model used for the modeling of load profiles, as well as the results obtained from it.

The proposed structure suits the necessary steps for the realization of this project, divided into the following sections. First, the data used in this work will be presented. Then, the preprocessing of the data will be developed and, subsequently, two approaches used to address the modeling of the building load profiles will be developed.

The purpose of the subsequently presented models is to achieve an accurate prediction of load profiles and therefore, both models rely on an elaborated implementation of various machine learning algorithms. Nevertheless, they have been entirely automated with the aim of providing an easy tool for future users. In addition, the provided models can also be used for other data sets, fulfilling the desired objective of building a transferable tool. Ultimately, both models have been implemented in Python 3.7.3 by using the *Scikit-learn* library [27], mainly complemented via the *Pandas* [40] and *NumPy* [41] libraries.

4.1 Data from the Jülich Research Centre

The data used for this work corresponds to data measured from the Jülich Research Centre (JRC), one of the largest interdisciplinary research centers in Europe. The JRC was founded on 1956 and, currently, counts on over 200 buildings, covering an area of 2.2 km^2.

The original information comes from different data sets, one of them containing the historical measurements of the power consumption of the buildings, which will be used as target. For most of the buildings, hourly energy consumption from the years 2015 to 2018 is available. Before filtering, the load data set contained historical profiles of 131 buildings. However, since a large number of these profiles do not have any input data, only 70 of all the buildings can be used to train and test the model. In any case, the size of the training set is relatively small, so the collection of suitable information for this task would result in significant improvements on the performance of the model.

Five input sets are used, containing information about the following technical equipment of the buildings:

- boilers
- heating appliances (all systems generating or distributing heat and warm water except boilers)
- cold-water supply appliances
- cooling appliances
- ventilation appliances

Furthermore, one data set containing information about the rooms and their usage is also used as input. The information summarized in these data sets was collected for previous studies and therefore, the collected information does not necessarily suit the necessities of this study. In consequence, the data needs to be prepared for the models.

The appliance data sets contain categorical attributes such as the code, description, building, room, manufacturer, type, as well as numeric attributes as the age, power and other appliance-specific attributes, which in general, have significant amounts of missing information. The rooms data set, on its behalf, includes the building, room type (usage) and area. Table 4.1 exposes the variables of all the six data sets that were available for this project.

4.2 Data Preprocessing

Data comes in a format that is rarely suitable for the predictive models. Therefore, to be able to use the data for the proposed application, it first needs to be preprocessed. Data preprocessing is, in most cases, one of the most time-consuming steps, and needs to be accomplished with extreme care, since the result of this step is the training set. Moreover, a proper preprocessing can lead to significant improvements in the accuracy of the model while, at the same time, allows a generalization of the method, making it a useful tool to obtain similar results from other data sets [42].

According to Kotsiantis et al. [42], data preprocessing can be divided into *instance selection*, *data cleansing* or *data cleaning*, *missing values handling*, *normalization*, *feature selection* and *feature construction*. Because of the complexity of the data set, a large number of features, and the large amount of missing information, these processes could only be partially automatized, and required a deep study of the data. The mentioned processes will be discussed subsequently, as they were a critical part of this work, apart from instance selection, which was not necessary due to the small number of samples. Additionally, the aggregation of the data into a building-wise format, and the selection of the load profiles will also be discussed. The resulting process is depicted in Figure 4.1.

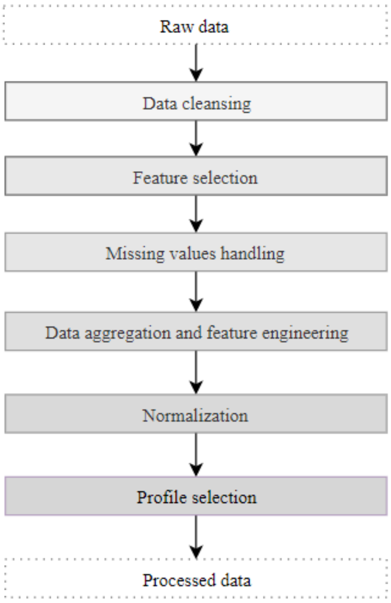


Figure 4.1: Data preprocessing flow diagram.

4.2.1 Data cleansing

Data cleansing is defined as the process of detecting and correcting or removing incorrect or inaccurate samples from the data set. The main task concerning data cleansing is detection. An outlier is an observation which is significantly deviated from the other observations. This type of observations may lead to a decrease of the performance of the predictive model and hence, they should be avoided [42]. Figure 4.2 shows the presence of an outlier in the load profile of a building.

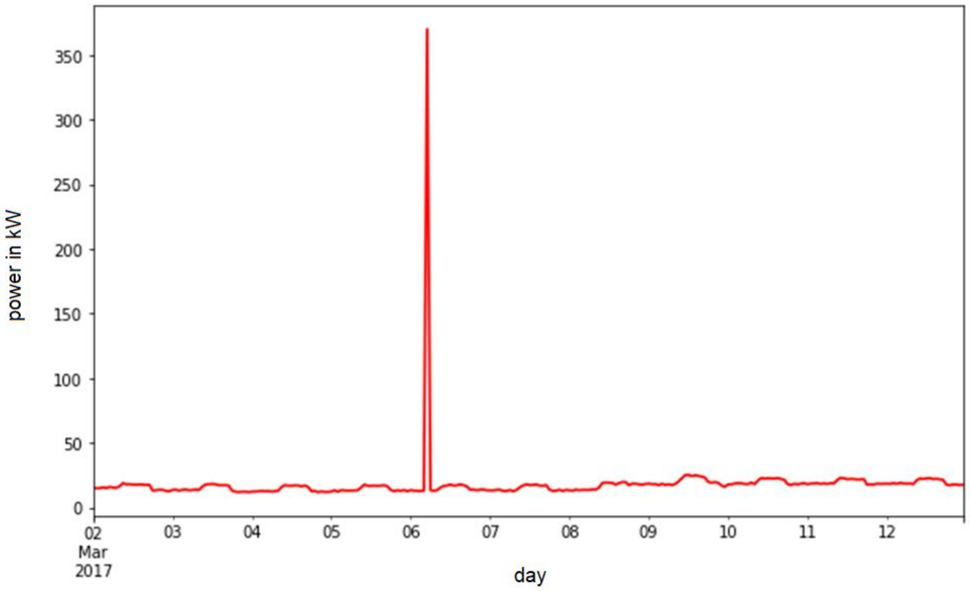


Figure 4.2: Load profile of a building which has an outlier.

Outliers are identified according to the Tukey's fences [43], based on the interquartile range, where values outside the range

$$[Q_1 - k(Q_3 - Q_1), Q_3 + k(Q_3 - Q_1)] \tag{4.1}$$

are considered as outliers, for some nonnegative k. Q_1 corresponds to the first quartile, value below which 25% of the data is located and Q_3 corresponds to the third quartile, value below which 75% of the data is located. The difference of both values is the mentioned interquartile range, containing 50% of the sample data. The definition of the range depends on the election of the value of k. The typical value for identifying data as an outlier is k = 1.5, while a greater value of k = 3 is typically used for identifying data as *far out*, or extreme outlier.

It is within the selection of the time periods to be used in the modeling of the load profile of a building that outlier detection is performed. If, according to Formula 4.1, an outlier is detected within the selected period, it is consequently removed from the data set.

Special attention needs to be given to observations with a value of 0kW. Since the goal of the study is to model load profiles of real buildings, values of 0kW are considered unrealistic for the normal operation of the buildings within the scope of the study and would derive in undesired noise. In many cases, these values will be detected as outliers and hence, they will automatically be filtered. However, for buildings with large load variations, and especially for small buildings, these values may not be filtered as outliers, and were filtered in a subsequent step.

Finally, buildings with an insufficient number of non-zero observations to model the load profile are deleted from the data set. Figure 4.3 illustrates an invalid profile of a building that needs to be removed from the data set.

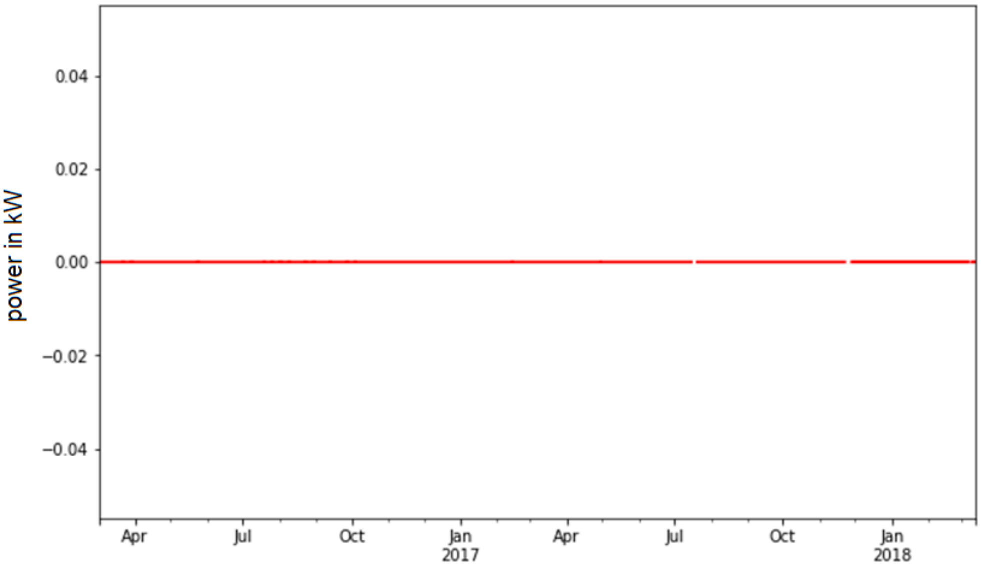


Figure 4.3: Illustration of an invalid load profile.

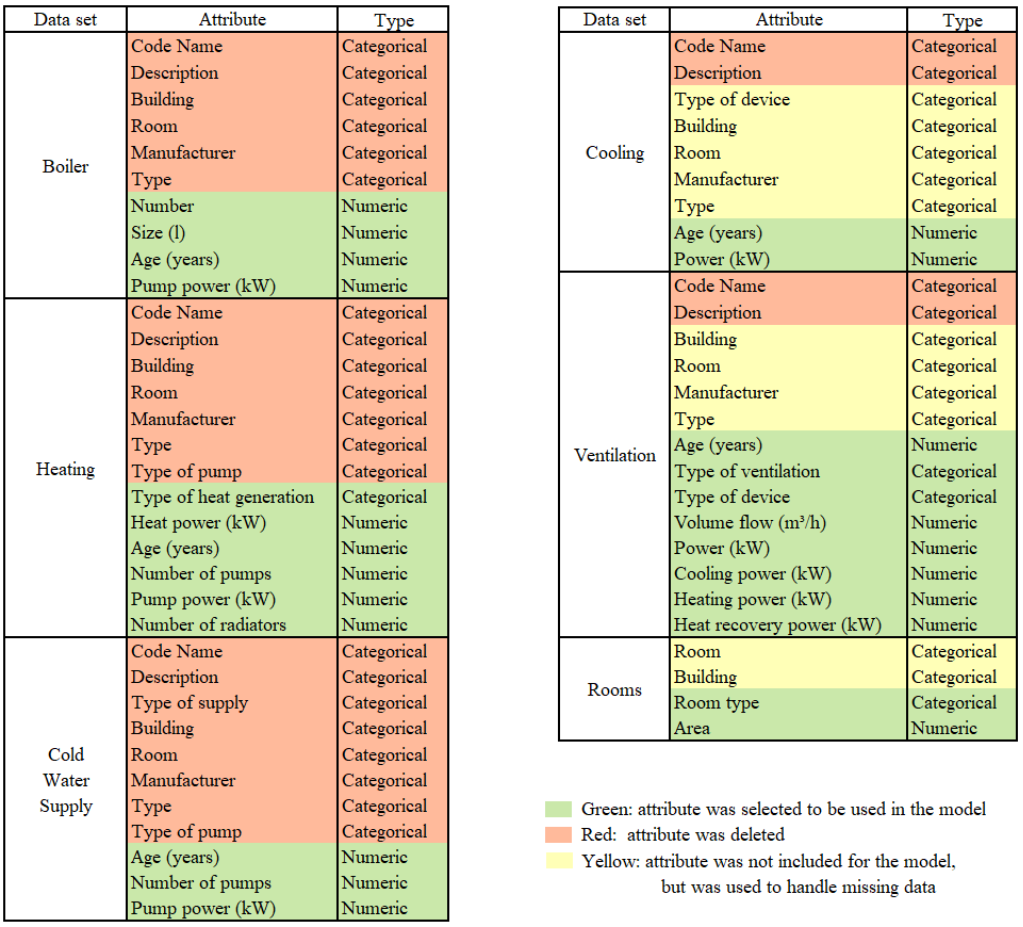
4.2.2 Feature selection

A predictive model does not necessarily need to use all the features included in the data set and therefore a feature selection is performed in order to reduce the dimension of the input data but avoiding a significant loss of information.

Table 4.1, containing the input attributes, is colored depending on whether they were selected or not to be used in the model.

In general, the influences on the load profile of the code, description, building, room, manufacturer, type, and type of pump attributes are inexistent or irrelevant and hence, can be left out of the input set. However, some of them will be useful to handle missing values, so they will be deleted after the following step (see section 4.2.3).

The remaining variables, which are considered of being susceptible of influencing the load profile, will be included in the model after applying the necessary preprocessing transformations.



Data set	Attribute	Type
Boiler	🟥 Code Name	Categorical
	🟥 Description	Categorical
	🟥 Building	Categorical
	🟥 Room	Categorical
	🟥 Manufacturer	Categorical
	🟥 Type	Categorical
	🟩 Number	Numeric
	🟩 Size (l)	Numeric
	🟩 Age (years)	Numeric
	🟩 Pump power (kW)	Numeric
Heating	🟥 Code Name	Categorical
	🟥 Description	Categorical
	🟥 Building	Categorical
	🟥 Room	Categorical
	🟥 Manufacturer	Categorical
	🟥 Type	Categorical
	🟥 Type of pump	Categorical
	🟩 Type of heat generation	Categorical
	🟩 Heat power (kW)	Numeric
	🟩 Age (years)	Numeric
	🟩 Number of pumps	Numeric
	🟩 Pump power (kW)	Numeric
	🟩 Number of radiators	Numeric
Cold Water Supply	🟥 Code Name	Categorical
	🟥 Description	Categorical
	🟥 Type of supply	Categorical
	🟥 Building	Categorical
	🟥 Room	Categorical
	🟥 Manufacturer	Categorical
	🟥 Type	Categorical
	🟥 Type of pump	Categorical
	🟩 Age (years)	Numeric
	🟩 Number of pumps	Numeric
	🟩 Pump power (kW)	Numeric

Data set	Attribute	Type
Cooling	🟥 Code Name	Categorical
	🟥 Description	Categorical
	🟨 Type of device	Categorical
	🟨 Building	Categorical
	🟨 Room	Categorical
	🟨 Manufacturer	Categorical
	🟨 Type	Categorical
	🟩 Age (years)	Numeric
	🟩 Power (kW)	Numeric
Ventilation	🟥 Code Name	Categorical
	🟥 Description	Categorical
	🟨 Building	Categorical
	🟨 Room	Categorical
	🟨 Manufacturer	Categorical
	🟨 Type	Categorical
	🟩 Age (years)	Numeric
	🟩 Type of ventilation	Categorical
	🟩 Type of device	Categorical
	🟩 Volume flow (m^3/h)	Numeric
	🟩 Power (kW)	Numeric
	🟩 Cooling power (kW)	Numeric
	🟩 Heating power (kW)	Numeric
	🟩 Heat recovery power (kW)	Numeric
Rooms	🟨 Room	Categorical
	🟨 Building	Categorical
	🟩 Room type	Categorical
	🟩 Area	Numeric

🟩 Green: attribute was selected to be used in the model
🟥 Red: attribute was deleted
🟨 Yellow: attribute was not included for the model,
but was used to handle missing data

Table 4.1: Variables of the six original data sets, colored depending on their use for the model.

4.2.3 Missing Values Handling

The data from the JRC was not collected expressly for this project and thus, an important lack of information had to be faced, resulting in a demanding step of the preprocessing.

In the following, missing values will be referred as information that is missing for any reported appliance or room. This is, the existence of the appliance or room has been reported, but some information about it is not provided. These cases are discussed in detailed within this section. However, a close inspection of the resulting input set after preprocessing shows multiple buildings without any information about appliances, or only a very small proportion of them. The reason could be that the mentioned buildings have a small size, as it was the case for some of them. Nonetheless, some other buildings showed unexpectedly large energy consumptions, suggesting that several appliances have not been reported and could lead to a significant decrease in the performance of the model. However, quantifying this lack of information is not possible and so is not to deal with it. Therefore, they will not be explored in this section.

When dealing with missing data, the simplest practice is to ignore missing values and remove samples containing them. Nevertheless, this should only be done in the case of knowing that the missing values do not follow a pattern, because otherwise their deletion would inappropriately alter the input set. This situation is often named as *missing at random*, a necessary condition for deletion of data [44].

However, deletion may not be affordable for small data sets, or ones lacking a high amount of information (which are situations that apply to this project) but, as most of predictive models are not able to handle missing values, something needs to be done. The main method to overcome this situation is called *imputation*, which means that missing values are replaced by a specified value, that can be understood as their *expected* value. The simplest imputation technique is the single imputation, consisting on the assignment of a single value (generally the mean, median or mode of a feature) to all the missing values of the same feature [45]. Even though this can work for some applications, it has the disadvantage of decreasing the variance of the input data, and treating equally all samples, even if they are completely different. For this reason, more complex methods, defined under the term of multiple imputation, are preferred as they try to extract information from other features in order to replace different missing values depending on the sample [45]. This technique has been explored within this work.

For the given data set, only numeric attributes belonging to the appliances data sets presented missing values and will consequently be the only ones discussed in this section. These numeric features presented a total of 58% of missing values, which is a significant amount of missing information.

Age missing values

The imputation of the age attribute of each appliance type followed a simple method, that can be observed in the flow diagram of Figure 4.4. If the value of an age attribute is missing, the mode of the appliance ages of that building is imputed or, if there is not any repeated age, the median is imputed. The reason for selecting this process is that for the same building, the age of an appliance is more likely to be the same as the other appliances of the building, because of the influence of the year of construction and building modifications. For the worst case, when the age of all the appliances of the same type of a building are missing, the best that can be done is to impute the overall median. It must be noted that here, and in the following, the median is used instead of the mean. This has been done because it provides a more robust imputation since it is not affected by outliers. In this case, outliers would be, for instance, very new appliances.

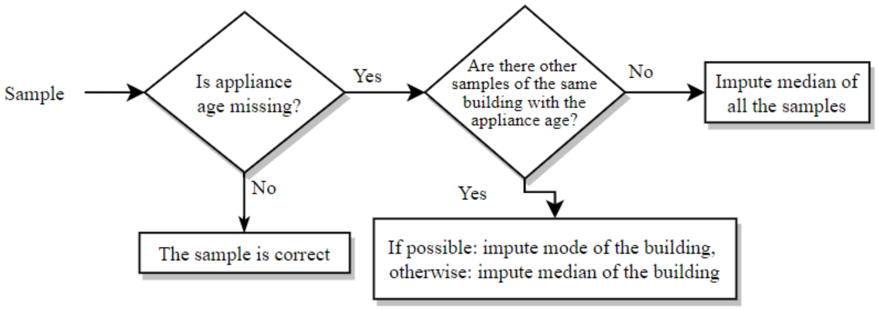

Sample
Is appliance age missing?
Yes
No
The sample is correct
Are there other samples of the same building with the appliance age?
No
Impute median of all the samples
Yes
If possible: impute mode of the building, otherwise: impute median of the building

Figure 4.4: Imputation process for missing values in age attributes.

Further boiler, heating and cold-water supply missing values

For other attributes, such as the number of pumps or the power, the presence of missing values lays out a problem. The difficulty regarding this lack of data comes from the fact that, for the appliances that are reported in the original sets, the given information does not differentiate between missing and zero values. This does not arise as a problem for cooling and ventilation devices, which can be assumed to have a non-zero rated power and thus, we assume that the value is missing. However, a problem does arise for the boiler, heating and cold-water supply devices, where zero values are completely plausible. For instance, reported appliances do not necessarily have pumps, so a missing value could mean that the appliance does not have any pump. Since it is not possible to know if the value is missing, for these cases, they are assumed to be zero.

Further cooling and ventilation missing values

As it has been mentioned, the absence of a value for the power of cooling appliances and the power and volume flow of ventilation appliances can be assumed as a missing value. The following process will be explained with the power, but also applies for the volume flow.

For these cases, a first imputation was carried out, by looking at the manufacturer and the type of the appliance. If a power value was missing but not its manufacturer and type, then the corresponding value of the other appliances of the same type was imputed. Then, for the remaining missing values, an imputation by regression was proposed. However, as can be seen in the example of Figure 4.5, a clear relation between the power of the appliance and the room area was not found, neither by splitting or aggregating all the room types or all the device types. The latter means that simply imputing a constant would work equally well, so a multiple imputation method as used for the age will be used here.

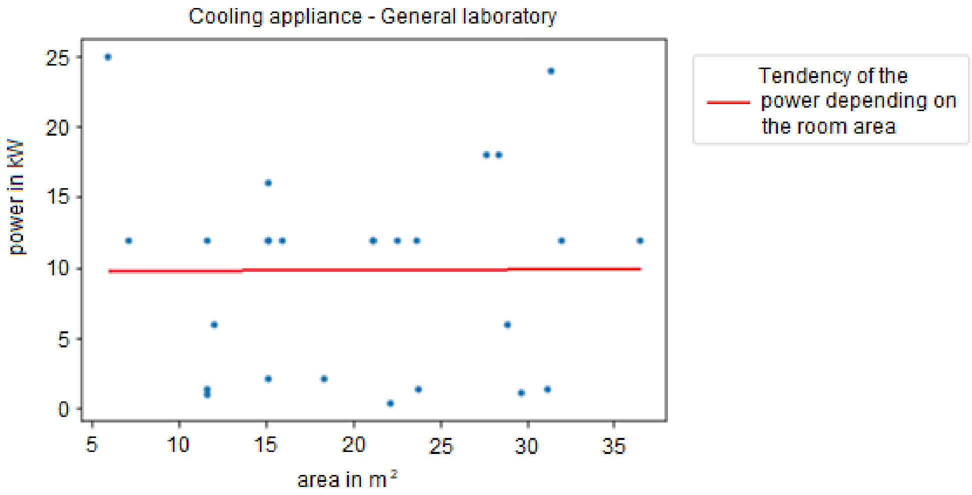


Figure 4.5: Scatter plot relating power and room area of cooling appliances in general laboratory rooms.

Figure 4.5 also shows a high variance in the power of the appliances. Assigning an estimated value can lead to an important error for a device. However, as the appliances will then be aggregated by building, the assigned values will offer a good estimation of the *size* of the building, in terms of appliances, and will lead to an improvement on the performance of the model.

Following the previous approach, the median power of the different subgroups (combinations of device types and room types) is imputed. If this is not possible, the median power of the immediately higher-level group is imputed, and this process is successively repeated until a suitable value has been assigned. To illustrate these processes, Figure 4.6 exposes the imputation of the power for the cooling appliances and Figure 4.7, the imputation of the power as well as volume flow of the ventilation appliances.

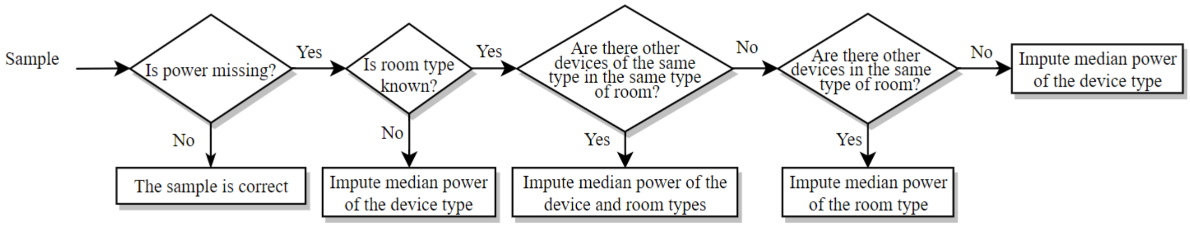

Sample
Is power missing?
Yes
No
The sample is correct
Is room type known?
Yes
No
Impute median power of the device type
Are there other devices of the same type in the same type of room?
No
Yes
Impute median power of the device and room types
Are there other devices in the same type of room?
No
Yes
Impute median power of the room type
Impute median power of the device type

Figure 4.6: Imputation process for the power in cooling appliances.

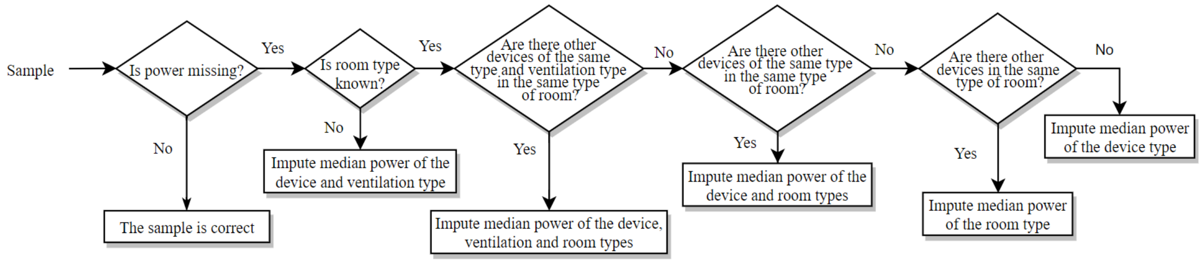

Sample
Is power missing?
Yes
No
The sample is correct
Is room type known?
Yes
No
Impute median power of the device and ventilation type
Are there other devices of the same type and ventilation type in the same type of room?
No
Yes
Impute median power of the device, ventilation and room types
Are there other devices of the same type in the same type of room?
No
Yes
Impute median power of the device and room types
Are there other devices in the same type of room?
No
Yes
Impute median power of the room type
Impute median power of the device type

Figure 4.7: Imputation process for the power and volume flow in ventilation appliances.

Finally, for the cooling, heating and heat recovery power, missing values are filled with zero, since it is more likely that the device does not have cooling or heating rather than the value is missing.

4.2.4 Data aggregation and feature engineering

Although data from the original data sets come sampled by appliance and room, respectively, the input of the model must be sampled by building, since the given output are the building load profiles. This implies that the data must be aggregated, and in some cases, features will need to be transformed, known as *feature engineering*.

Aggregating numeric features can be, generally, as simple as adding the values for the same building. However, when it comes to categorical features, aggregation involves a trade-off between the loss of information and the rapidly increase of the number of features. Consequently, as well as for the handling of missing values, the data aggregation needs to be treated in different ways depending on the features.

Numeric features aggregation

Starting with the simplest case, the numeric variables size, pump power (boiler, heating and cold-water supply appliances appliances), number of radiators, power (cooling appliances), cooling power, heating power and heat recovery power, shown in Table 4.1, can be aggregated by adding all the feature values for the same building. For the case of the boiler's size and pump powers, the values must be first multiplied by the respective number and number of pumps to obtain the total value. The number of attributes can then be removed.

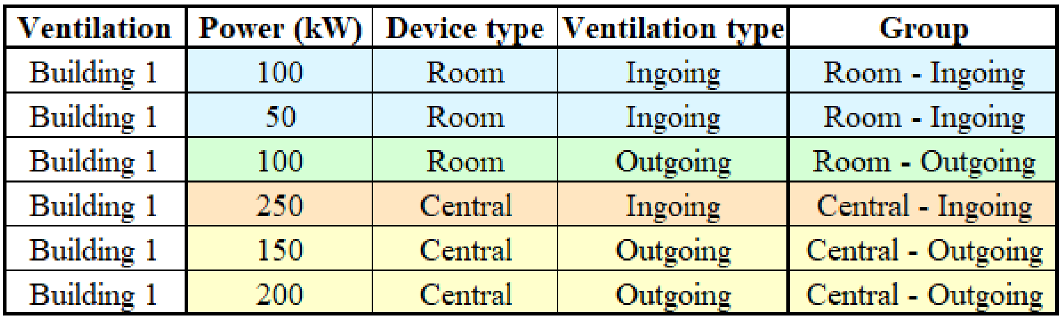
Age aggregation

Age attributes should evidently not be added, but the increase of the number of age features is additionally not desired, so the aggregation will be performed by assigning the mode of the appliances age of each building or, otherwise, the median.

Categorical features aggregation

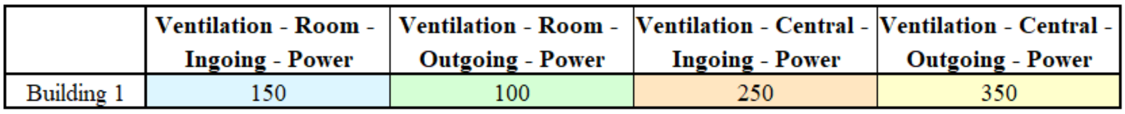
Aggregating categorical features by selecting the mode would result in a significant loss of information and hamper the purpose of determining the dependencies of the profiles and the different types of rooms and appliances. In consequence, a more complex method must be implemented for these cases to guarantee that the desired information is fed into the model.

To illustrate the method, a simplified example of the ventilation power will be presented in Table 4.2 and Table 4.3, and the same procedure is followed for the volume flow. For ventilation appliances, there are two categorical variables of interest: Device type, divided into Room, Central and Wall categories; and Ventilation type, divided into Ingoing, Outgoing and Circulating. Both variables are combined into 9 groups, which are then transformed to new features, and the power of each building's appliance is summed into the corresponding transformed feature. The transformation method can be understood as an extension of dummy encoding [46], since it involves the creation of new columns in the same way. The difference is that, rather than using dummy variables (0,1), the appliance power is used instead of 1, and then an aggregation is made.



Ventilation	Power (kW)	Device type	Ventilation type	Group
Building 1	100	Room	Ingoing	Room - Ingoing
Building 1	50	Room	Ingoing	Room - Ingoing
Building 1	100	Room	Outgoing	Room - Outgoing
Building 1	250	Central	Ingoing	Central - Ingoing
Building 1	150	Central	Outgoing	Central - Outgoing
Building 1	200	Central	Outgoing	Central - Outgoing

Table 4.2: Simplified example of the ventilation appliances of a building before aggregation.



	Ventilation - Room - Ingoing - Power	Ventilation - Room - Outgoing - Power	Ventilation - Central - Ingoing - Power	Ventilation - Central - Outgoing - Power
Building 1	150	100	250	350

Table 4.3: Resulting structure of ventilation features after aggregation.

Heat power attribute of heating appliances also need to be transformed, but as there are only two type of heat generation (electrical and non-electrical power source), only two features are created by using the same process as ventilation devices.

The last features to be aggregated are the room type and area. The total area of the different room types is expected to provide important information to the model. Though it is desired to preserve the maximum information, the classification according to DIN 277 [47] derives in 165 different types of room, an unmanageable number of classes for the developed models. Moreover, such a granular division of the room types is not needed to analyze the dependencies of the profile and the room types, since different rooms, which have similar uses, will have a similar power consumption behavior, so grouping them would simplify the aggregation process. As a result of grouping the room types, 37 classes have been selected which, after a transformation equivalent to the one applied to the ventilation power, 37 features summing the areas of the corresponding classes are created. Table 4.4 shows all the variables resulting from the aggregation process, which will serve as input to the model, and where the selected room classes can be appreciated.

Feature
Boiler - Age
Boiler - Size
Boiler - Pump Power
Heating - Age
Heating - Electrical Power Source
Heating - Non Electrical Power Source
Heating - Pump Power
Heating - Radiators
Cold Water Supply - Age
Cold Water Supply - Pump Power
Cooling - Age
Cooling - Power
Ventilation - Age
Ventilation - Room - Outgoing - Power
Ventilation - Room - Circulating - Power
Ventilation - Room - Ingoing - Power
Ventilation - Wall - Outgoing - Power
Ventilation - Wall - Circulating - Power
Ventilation - Wall - Ingoing - Power
Ventilation - Central - Outgoing - Power
Ventilation - Central - Circulating - Power
Ventilation - Central - Ingoing - Power
Ventilation - Room - Outgoing - Volume Flow
Ventilation - Room - Circulating - Volume Flow
Ventilation - Room - Ingoing - Volume Flow
Ventilation - Wall - Outgoing - Volume Flow
Ventilation - Wall - Circulating - Volume Flow
Ventilation - Wall - Ingoing - Volume Flow
Ventilation - Central - Outgoing - Volume Flow
Ventilation - Central - Circulating - Volume Flow
Ventilation - Central - Ingoing - Volume Flow
Ventilation - Cooling Power
Ventilation - Heating Power
Ventilation - Heat Recovery Power

Feature
Technical Equipment Room
Technical Equipment Room AU
Technical Equipment Room AW
Technical Equipment Room ET
Technical Equipment Room HA
Technical Equipment Room LA
Technical Equipment Room TG
Biological Laboratory
Changing Room
Chemical Laboratory
Conference
DV-Room
Dining Room
Electrical Laboratory
Electronic Laboratory
Elevator
Garage
General Laboratory
Isotopic Laboratory
Kitchen
Lecture Room
Library
Living Room
Office
Office Kitchen
Other
Physical Laboratory
Reception
Refrigerated Room
Rest Area
Server Room
Shower
Storage
Technological Laboratory
Traffic Area
WC
Workshop

Table 4.4: Input variables for the predictive model

4.2.5 Normalization

The use of such a varied type of features results in a correspondingly varied scaling of the features. Features need to be normalized because otherwise, the difference in the scales can mislead the predictive algorithms, which expect input data to be normalized, and lead to a decrease of the performance [48]. However, due to the sparsity and nonnegativity of the input set, only scaling has been applied dividing by the maximum value, as centering is not recommended for this situation, resulting in variables ranging from 0 to 1.

In addition, the input data set must be modified when used for classification. In classification, proportions of power and area are more significant than absolute values, since they are what characterizes the building. In consequence, the power, volume flow and room areas need to be normalized separately for each building, by dividing by the total power, volume flow and area, respectively, in order to improve the classification accuracy.

4.2.6 Profile selection

The aim of the project is to model the load profile buildings and therefore, the last step of preprocessing is to select the target profiles. Even though each building has a similar profile for each working day, variations from one day to another can be observed due to randomness and other time-dependent factors. Since the input data is not time-dependent, these variations cannot be modeled. It is necessary to average the load profiles of different working days in order to obtain a representative profile of the building, which can be modeled. Moreover, it must be noted that the power consumption has a high dependency on the temperature, and very important seasonal variations [9]. Consequently, as the purpose is to perform cross-sectional predictions, the most appropriate procedure is to use the same time periods for all the buildings, which will result in a better performance of the model.

Figure 4.8 shows the power consumption of an exemplary building for two weeks. The assigned load profile is depicted in Figure 4.9, obtained by averaging the power consumption of the working days of those weeks.

As load profiles may vary considerably during the year, it might be useful for a better building planification to model the load profile of different seasons. To that end, representative load profiles have been selected in three different times of the year (April, July and December), to model the power consumption in different seasons.

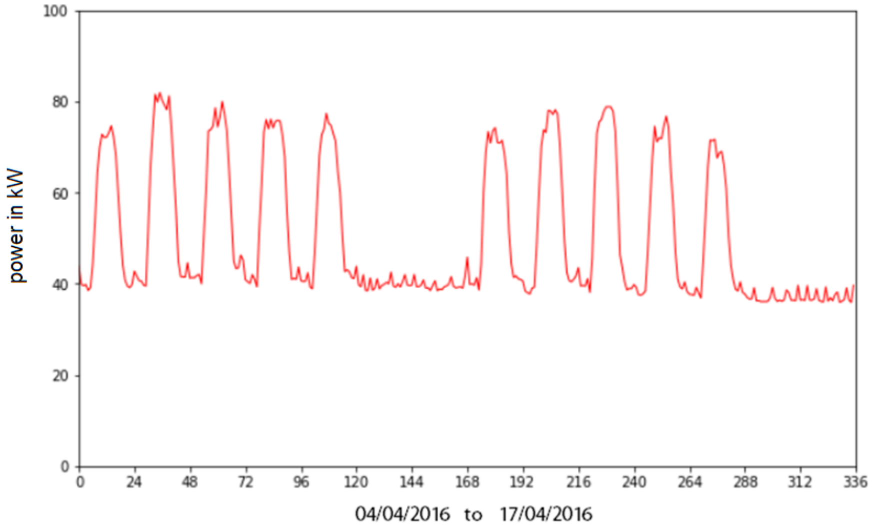


Figure 4.8: Energy consumption of a building for two weeks.

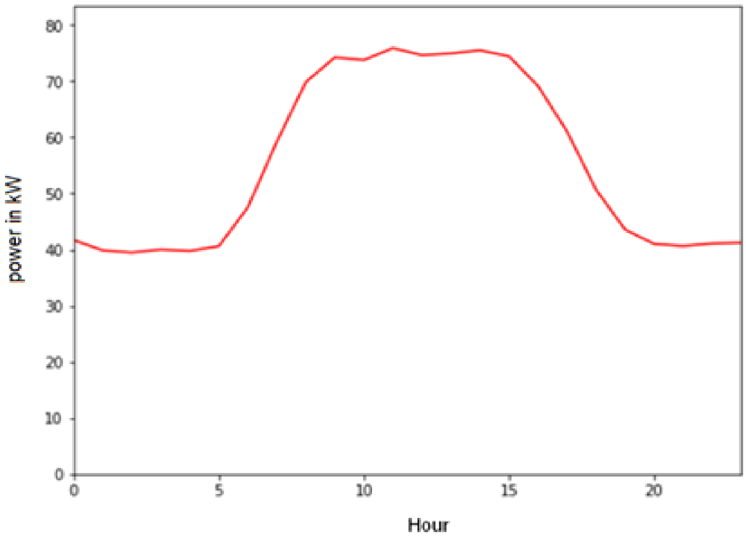


Figure 4.9: Averaged load profile of the previous building to be used in the model.

4.3 Modeling approach

Several top-down approaches of load profile modeling have been introduced in chapter 2, which can be divided into two main groups, depending on the type of algorithms used: On the one hand, regression analysis and, on the other hand, clustering and classification.

Regarding regression approaches, only basic models have been implemented. For instance, Ge et al. [10] simplify the task by predicting 15 parameters instead of 24, and only use two variables as input, and McLoughlin et al. [16] limit their study to the prediction of the peak load and the total energy consumption. The reason for using such reduced procedures is mainly because modeling a load profile only with regression algorithms involves training a model for each hour, which is not a very elegant approach, since a similar task is done for every hour.

However, a regression approach is still suitable for the purpose of modeling load profiles and, moreover, for the purpose of analyzing their dependencies on rooms and appliances. In addition, only one model needs to be implemented since the prediction of each hour works equally. Therefore, a regression model is proposed in this work with a more complex approach than the aforesaid studies, as more algorithms and parameters will be considered.

Unlike for regression techniques, more ambitious models have been implemented for clustering and classification approaches. For instance, McLoughlin et al. [17] do not only cluster load profiles for each building, but also differentiate between each day of the week to assign them to the most appropriate profile class. The main disadvantage is the need of large data sets since clustering involves segmentation of the studied population and, in order to achieve good classification performance, enough samples of each cluster are needed. Still, such approaches will generally involve a reduction of the computational time compared to regression approaches by avoiding the repetitive process of predicting the energy consumption for each hour.

Considering the advantages of clustering and classification approaches and the fact that the method and algorithms used for the regression model can be applied here with only few modifications, a second model consisting of clustering and classification combined with regression is proposed to complement the results obtained with the previous approach.

4.3.1 Regression analysis

The regression analysis proposed in this section follows a similar method as the forecasting approach implemented by Divina et al. [22] since the main idea is also the use of stacking, which was introduced in section 3.1.5.6. As the latter technique has proven to be an effective way of improving the performance of a model [32], this ensemble learning method will be implemented with the aim of obtaining the best achievable results. In order to facilitate the understanding of the applied method, a flow diagram of the complete process is illustrated in Figure 4.10.

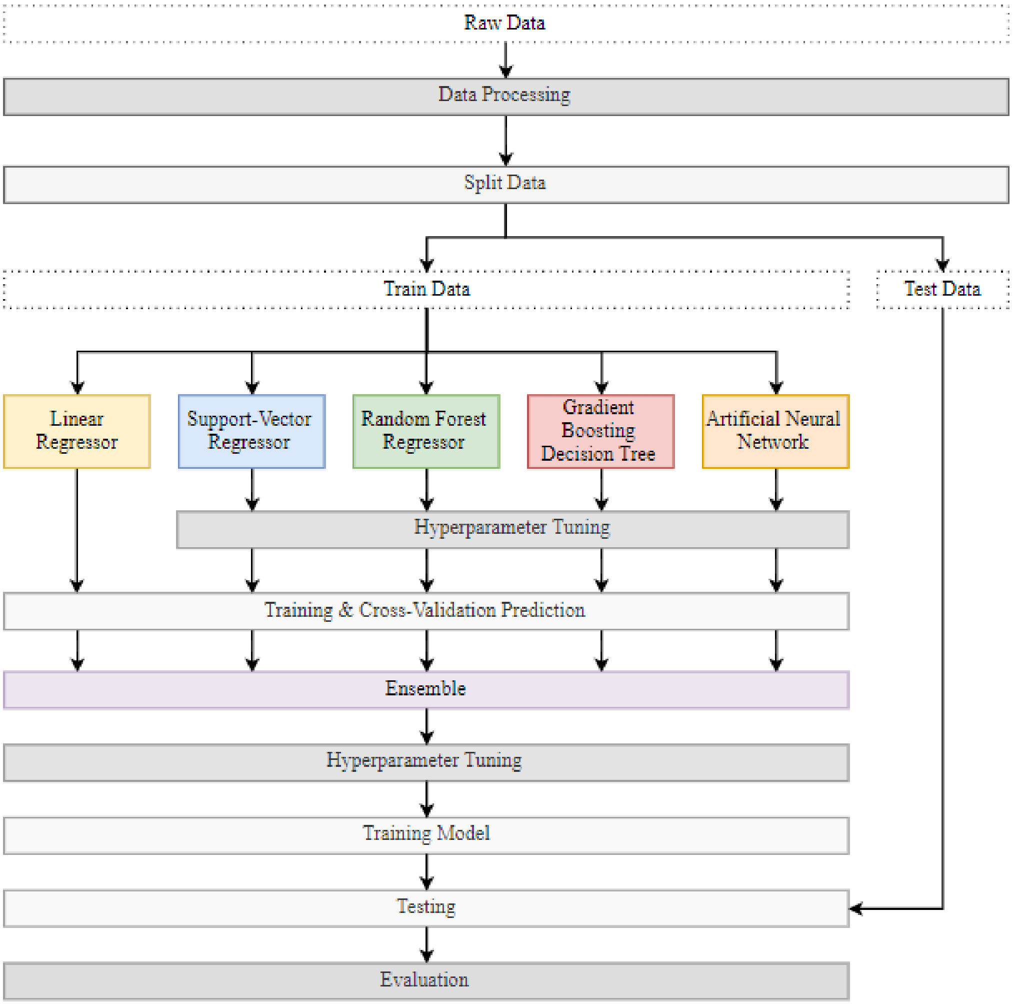
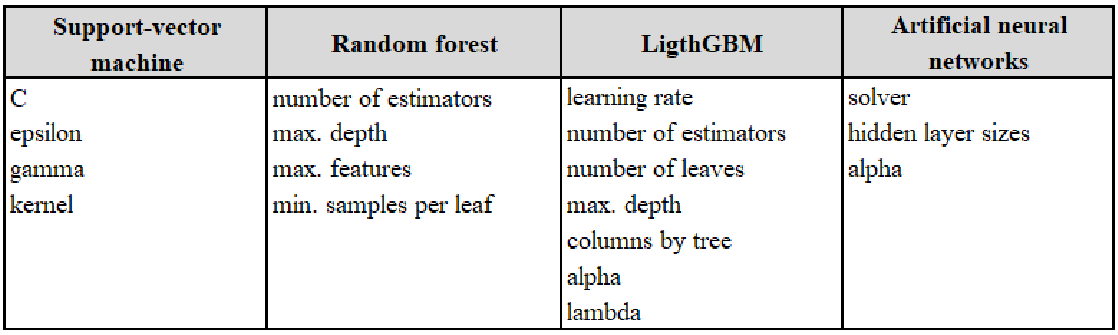


Figure 4.10: Flow diagram of the regression analysis approach.

First, preprocessed data is divided into the training set (80%) and the test set (20%), used to evaluate the model, as introduced in section 3.1.1. Then, training data is fed into the base learner algorithms.

The five selected base learners were the supervised learning algorithms introduced in section 3.1.5. The reason for using such a variety of algorithms is to maximize the variety of predicted load profiles since, on the one hand, it increases the probability of finding the most appropriate algorithm for the task and, on the other hand, the ensemble learner will benefit from this additional information, resulting in more accurate predictions. All the algorithms were implemented with the *Scikit-learn* library [27] except from the GBDT, implemented with *LightGBM* [49], which offers a very powerful tool for supervised learning tasks.

In the second step, the hyperparameters shown in Table 4.5 are tuned for each model. It should be noticed that, since linear regression does not have hyperparameters, this step is skipped. Before this stage, a selection of possible values of the hyperparameters is done. Then, using grid search, all possible combinations are used to train the algorithms for each hour and the one with the lowest mean cross-validation score, this is, the one with the lowest MAPE for the load profiles, is selected.



Support-vector machine	Random forest	LigthGBM	Artificial neural networks
C epsilon gamma kernel	number of estimators max. depth max. features min. samples per leaf	learning rate number of estimators number of leaves max. depth columns by tree alpha lambda	solver hidden layer sizes alpha

Table 4.5: Hyperparameters tuned for each supervised learning algorithm.

Once hyperparameter tuning is finished, the best combinations of hyper parameters are used to train the respective models and the cross-validation predictions are obtained. These predictions are used as input to the stacked algorithm. For this application, the stacked algorithm was chosen to be a RF as it offered the best results. In general, tree-based algorithms show good performances when used for ensemble learning tasks [22]. The hyperparameters of the meta learner are tuned in the same way as before, as a different combination of them is expected to perform better for such a different task, and again, the best combination is used to train the algorithm with the cross-validation predictions of the base learners.

Finally, the implemented model is used to predict the load profiles of the test set and the results are compared to evaluate the performance of the model.

4.3.2 Clustering and classification

To implement the clustering and classification approach, a model such as the proposed by McLoughlin et al. [17] should be sufficient to accomplish the intended task. However, two difficulties arise regarding the clustering of load profiles. The number of samples is very small, but the power range of the existing buildings is relatively large. The results are the necessity of normalizing the load profiles and a tradeoff between selecting more clusters to minimize the inherent error of clustering and selecting a limited number of them to achieve a better classification accuracy.

In consequence, profiles can be normalized dividing by the maximum power. This has the additional advantage that buildings with similar energy consumption behavior will have very similar normalized profiles, paying special attention on the normalization of power and area features mentioned in section 4.2.5. The drawback of normalizing profiles is that the output of the subsequent classification will no longer be the modeled absolute profiles but normalized ones. Therefore, an additional regression sub model has to be implemented in order to predict the maximum load which, multiplied by the assigned normalized profile, will result in the predicted load profile of a building.

Here, the concept of stacking is also used to improve the performance of the complete model and hence, the procedure of each of the sub-models is analogous to the presented for the regression analysis. The process diagram is illustrated in Figure 4.11 and it is explained in the following. For simplicity, ANNs have not been used for this approach.

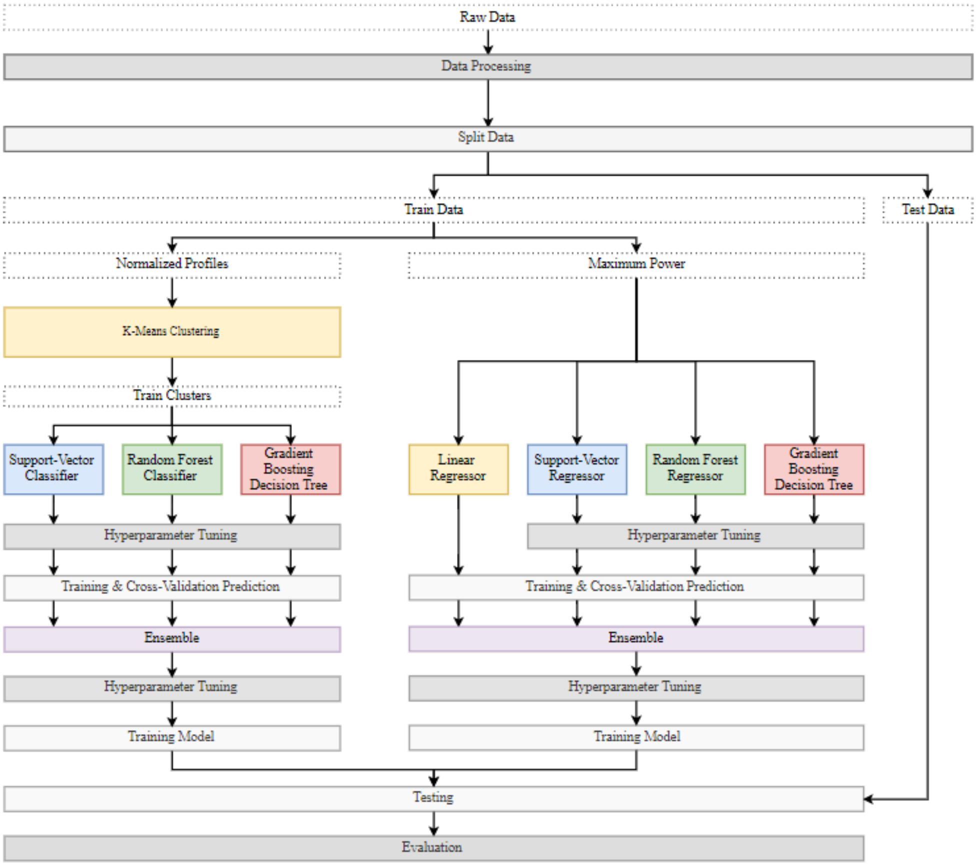


Figure 4.11: Flow diagram of the clustering and classification approach.

Again, preprocessed data is divided into the training set (80%) and the test set (20%), used to evaluate the model. Then, training profiles are normalized and a k-means algorithm, introduced in section 3.2.1. is used to cluster them into five clusters. More clusters would be preferred in order to minimize the error related to the cluster assignment. However, the number of clusters is limited by the data set size since with more samples, the number of load profiles per cluster would be excessively small. As mentioned, more clusters would be desired, but the size of the data set is a limiting factor. The resulting clusters of the load profile will serve as output labels to train the classification algorithms.

The selected algorithms for the classification base learners were SVM, RF and LightGBM since ANNs were discarded and linear regression was not suitable for the classification task. The implementation of such algorithms did not require complicated modifications in relation to the regression algorithms, demonstrating to be very flexible algorithms. In this stage, an analogous hyperparameter tuning to the one in the regression model is performed for the base learners but, instead of minimizing the MAPE, hyperparameters are selected to minimize the cross-validation precision. Precision is a common classification performance metric calculated as the proportion of correctly labelled samples and the total number of samples [30]. Anew, the best hyperparameters are selected and cross-validation predictions of the profile labels are made.

The meta learner, also selected to be a RF, is trained with the resulting predictions and the normalized test profiles are generated.

At this point, the maximum power of the original training profiles is extracted, and a regression analysis of the maximum power needs to be made. However, implementing a new model is not necessary since the previous regression model can be used by only reducing the number of hours parameter to one, as only one value is being predicted. With this model, the maximum power of the test profiles is predicted.

The final stage is to multiply the predicted normalized profiles by the predicted maximum power to obtain the modeled load profiles for the test set, which can be compared to the real profiles and the obtained in the previous section to evaluate the performance of this alternative approach.

5 Analysis of results

The proposed models were subsequently applied to the buildings from JRC. Both were trained with data from 56 buildings (80% of the buildings). The fitted models were finally used to model the load profiles of the other 14 buildings (remaining 20%) from the test set and compared with the real profiles to evaluate the performance of the models. Target profiles were selected from year 2016, averaging workday load profiles from the months April, July and December separately, with the aim of modeling seasonal variations of load profiles. The following chapter will present and analyze the obtained results. In general, the modeling of April profiles will be considered the base case and will be discussed with more detail.

First, results from both models will be analyzed separately in sections 5.1 and 5.2, in order to show model-specific details that have an influence in the final predictions. The performance of both models will be compared in section 5.3. Then, dependencies between the profiles and the different rooms and appliances will be explored in section 5.4. Finally, the modeling of the energy consumption of a building for an entire week will be presented in section 5.5.

5.1 Regression analysis results

Section 4.3.1 discussed the implementation of the regression model. It was shown that in a first step, different base learners are used to predict the entire load profile, and in a second step, a stacked predictive algorithm is used. Consequently, evaluating the performances of the base learners, as well as the performance of the stacked algorithm (ensemble) will give a good insight of how the model is working. First, an exemplary profile is depicted in Figure 5.1, where the predicted profiles of the different algorithms can be observed and compared to the real profile of the building. The accuracy of the algorithms for this building should not be assumed as general for all the test buildings, so no accuracy measure is shown for such a single case. However, it can be observed that the ensemble prediction is the most accurate, slightly better than the one of the RF, something that will shortly become apparent.

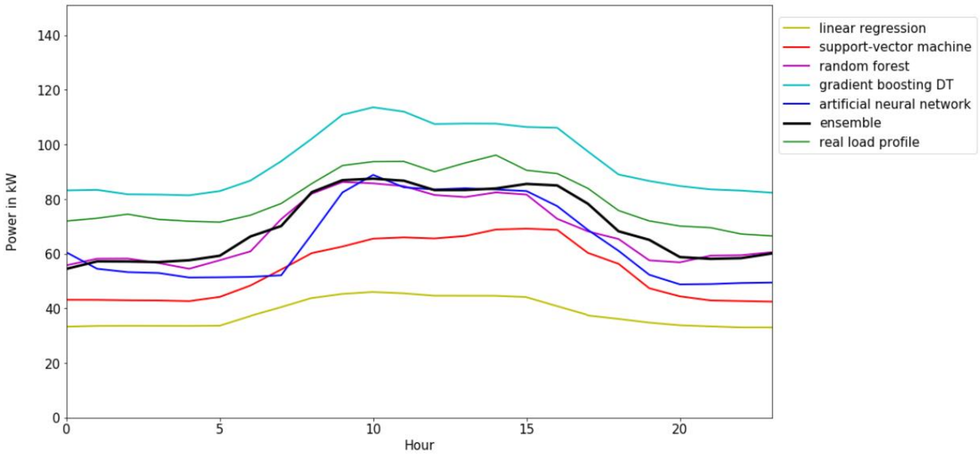


Figure 5.1: Comparison of the predicted profiles of all the predictive algorithms and the real profile of a test building.

To evaluate and compare the prediction performance of the models, all the test buildings need to be considered at the same time, since they represent the unknown buildings of the studied population. In addition, the comparison needs to be made using an accuracy metric, as introduced in section 3.1.7. The mean MAPE and mean sMAPE will be used to compare the different performances, complemented with their corresponding standard deviation (SD). Table 5.1 shows these results.

	mean MAPE [%]	SD MAPE [%]	mean sMAPE [%]	SD sMAPE [%]
Linear regression	90.64	61.36	74.38	29.84
Support-vector machine	56.47	40.76	46.72	20.60
Random forest	44.24	33.51	36.04	20.55
Gradient boosting decision tree	115.03	122.53	57.13	32.90
Artificial neural network	63.73	51.14	58.98	43.25
Ensemble	39.36	28.98	32.79	17.90

Table 5.1: Comparison of algorithm performances using MAPE and sMAPE for test profiles in April.

Focusing on the base learners, we can see that no matter which metric is used, the random forest is the one that performs the best for this modeling task. This situation is underlined by the fact that RFs also obtain the best results when modeling July and December loads, shown Appendix A. In contrast, linear regression does not seem to be suitable for the task, as the problem is unlikely to be linear. Despite the potential of ANNs, they also show bad performances for the three studied cases. However, the problem is likely to be the fact that the training size is very small, making ANNs unsuitable for almost any task. Ultimately, SVMs and GBDTs obtain in general acceptable results. However, their performance is far from the one of the RF.

The following aspect to discuss is the performance of the ensemble learning model. It can be appreciated in Figure 5.1, as well as in Table A.1 and Table A.2, that the performance of the stacked algorithm is, in all the studied cases, better than the best performance of the base learners, the RF. For the base case, the improvement of the mean sMAPE goes from 36.04% to 32.79% and of the mean MAPE from 44.24% to 39.36% which means a relative improvement of 9% and 11% respectively. These results are coherent with the results obtained by Divina et al. [22], demonstrating that these kind of ensemble algorithms can lead to significant improvements of a model performance. Besides, as it was observed in Figure 5.1, the ensemble prediction is similar to the one of the RF, which could be understood in the way that the stacked algorithm mainly uses the predictions from the RF, since it leads to the best performance, but the prediction of the remaining algorithms provide useful information to improve the ensemble prediction.

In addition to a comparison of the accuracy of the five base learners, the convenience of using sMAPE instead of MAPE can also be appreciated. In general, the standard deviation of both metrics is generally big, which could be due to a bad quality of the input data set, as will be explored in a later section. Nevertheless, both the mean and the SD MAPE increase with respect to the corresponding mean and SD sMAPE. An examination of the individual predictions and scores of the test buildings show that for two of the buildings, the predicted profiles are greater than the real profiles. This is severely penalized by the MAPE metric, and could lead to misinterpretation of the results, as it is the case of the GBDT in Figure 5.1. Predictions of the GBDT are generally closer to the real profile than LR for example, but the positive errors in two buildings result in an extreme increase of the MAPE. For this application, sMAPE proves to be a robust metric that can be used to adequately compare the model performance, without losing the interpretability of the MAPE.

5.2 Classification and clustering results

The second model proposed is slightly more complex than the previous regression model since it has more steps. In the following, each step will be analyzed, and the errors associated to each step will be exposed. In the first step, load profiles are clustered as depicted in Figure 5.2. Although the clusters with more variance such as the top right one could benefit from the increase of the number of clusters, it would have a negative effect on the classification task. Therefore, five clusters were selected for the data set.

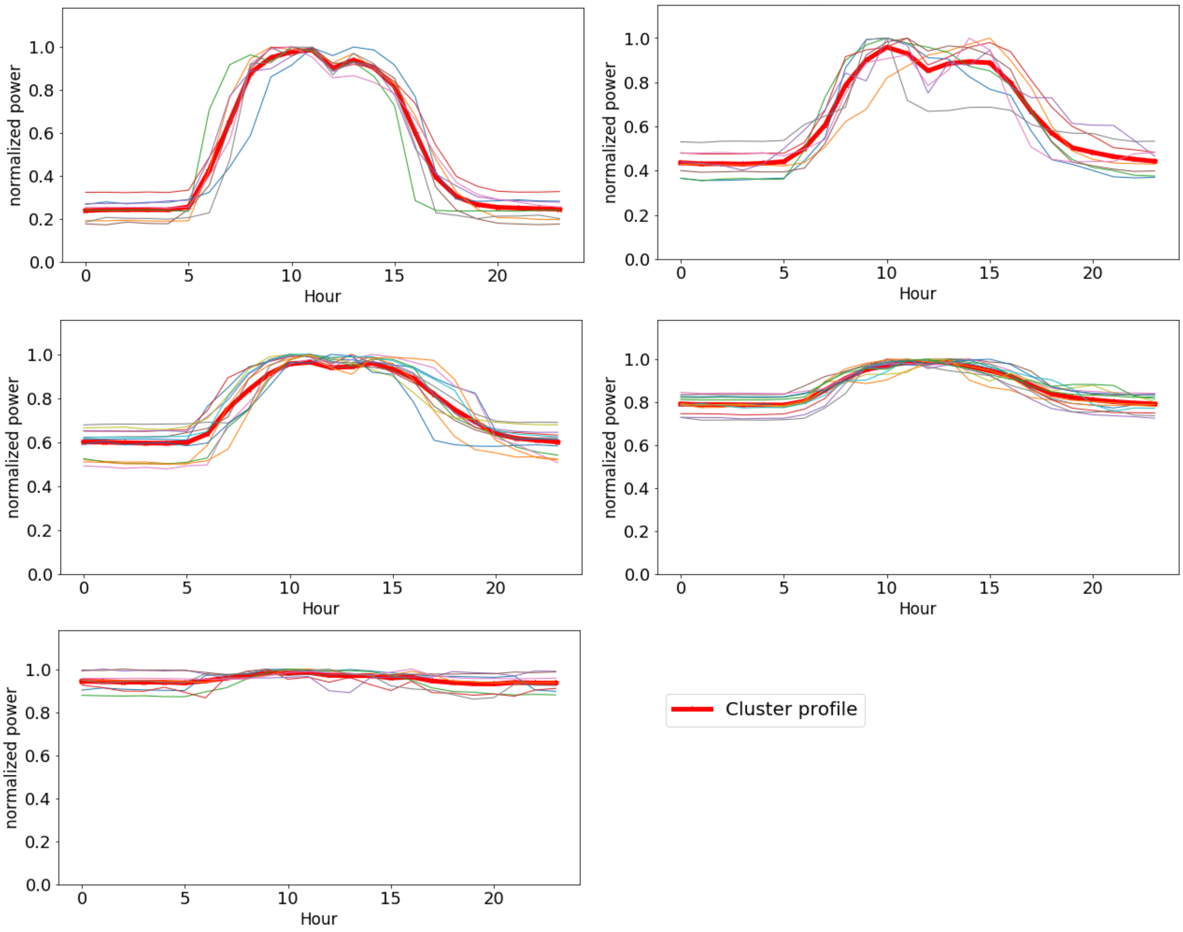


Figure 5.2: Representation of the load profiles from the five clusters and their corresponding cluster profile.

An interesting metric relating the clustering process is the error of assigning a cluster profile to the given buildings. For the base case, the mean MAPE of the normalized training profiles with respect to the normalized cluster profile is 7.20% and the mean sMAPE is 7.29%, obtaining very similar errors for the test profiles (7.17% and 7.68%). This means that, even if the classifier had an ideal performance, and the regressor was able to predict the actual maximum load, these would be the errors of the estimations. If a larger data set was available, the errors could easily be reduced by assigning more suitable cluster profiles.

In the second stage, buildings are classified into the previous clusters. It can be appreciated in Table 5.2 that the accuracy of the classifiers is relatively low. In addition, though the validation accuracy increases for the three studied cases with the ensemble predictions, only in the base case does the test accuracy actually improves. With such a misclassification, the error of the normalized profiles rises to 23.49% in the case of the mean MAPE and 20.40% in the case of the mean sMAPE.

	Gradient boosting decision tree	Random forest	Support-vector machine	Ensemble
Accuracy [%]	35.57	42.86	14.29	50.00

Table 5.2: Accuracies of the classification of normalized profiles for the modeling of load profiles in April.

The third stage corresponds to the prediction of the maximum load, which follows an analogous process to the previous model and the results are presented in Table 5.3. In the same way, similar insights can be obtained from the results, such as that the LR still shows bad performances while the best base learner is the RF, and the stacked algorithm derives in an important decrease of both errors.

	MAPE [%]	sMAPE [%]
Linear regression	80.59	69.37
Support-vector machine	58.20	48.62
Random forest	34.96	33.09
Gradient boosting decision tree	41.34	34.93
Ensemble	30.63	29.36

Table 5.3: Performances of the maximum load regressions for the modeling of load profiles in April.

In the final stage, the predicted normalized profiles and the predicted maximum load are combined to obtain the predicted load profiles. The resulting prediction errors are presented in Table 5.4.

	mean MAPE [%]	SD MAPE [%]	mean sMAPE [%]	SD sMAPE [%]
Clustering and classification	45.17	33.27	38.45	19.53

Table 5.4: Performances of clustering and classification model for the modeling of load profiles in April.

An additional interesting reference value of the both errors are the corresponding errors in the case that the classification was perfect, and all the buildings were assigned to their real cluster. For the mentioned case, the mean MAPE would decrease to 39.96% and the mean sMAPE to 37.20%, which is not a significant improvement. This comparison illustrates that the critical stage of the second proposed model is the regression analysis. To finish with this section, Figure 5.3 illustrates two predicted load profiles as well as the real profiles of the corresponding buildings.

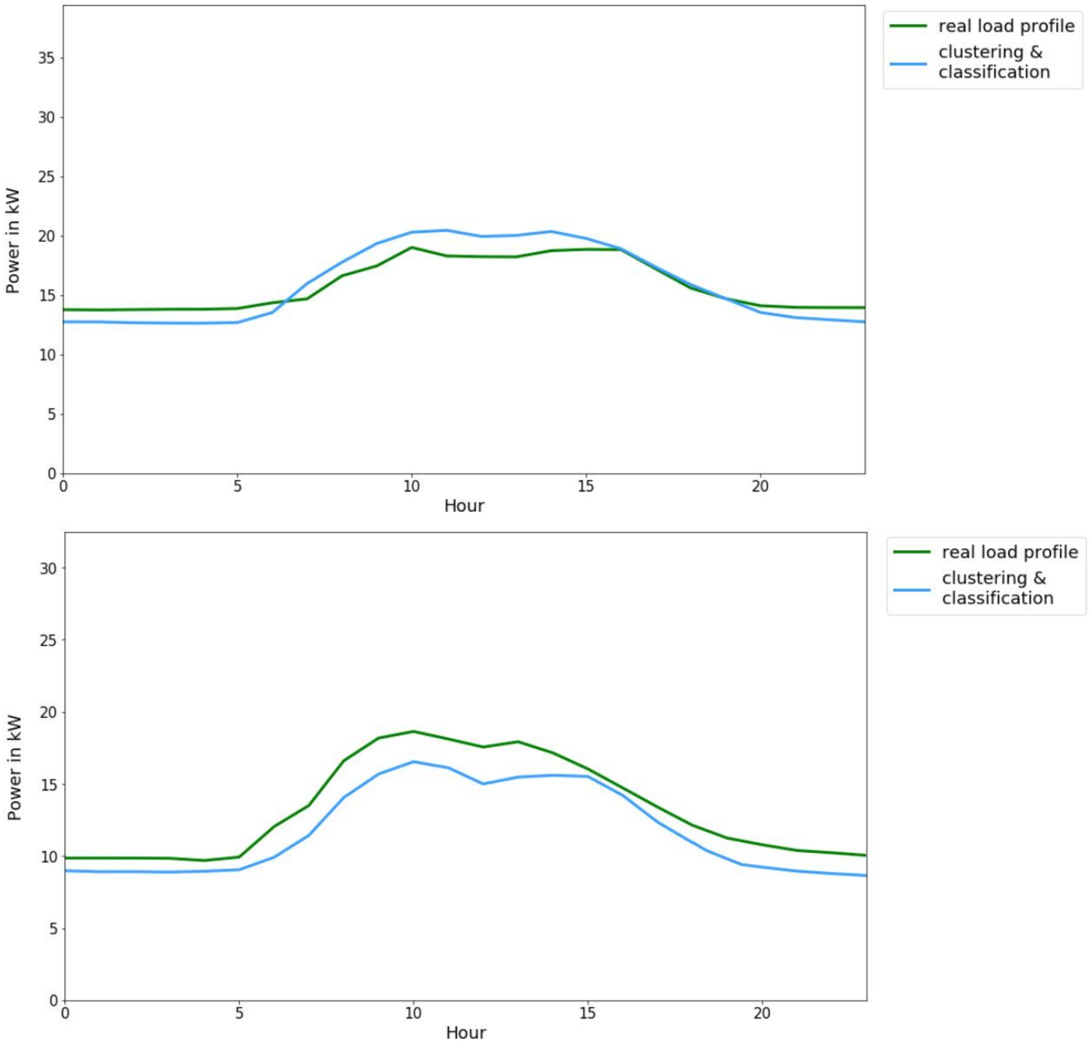


Figure 5.3: Comparison of the predicted profiles of the second model and the real profiles of the test buildings.

5.3 Comparison of models

Considering the results obtained for both proposed models, it is possible to compare the performances in order to determine the best model. Table 5.5 presents the performance metrics for both models. It can be appreciated in the base case, as well as in the other two cases (Table A.9 and Table A.10), that the regression model performs better than the clustering and classification model, improving the performance in these cases of about 10 to 25% compared to the latter model, which is a significant improvement.

	mean MAPE [%]	SD MAPE [%]	mean sMAPE [%]	SD sMAPE [%]
Regression	39.36	28.98	32.79	17.90
Clustering and classification	45.17	33.27	38.45	19.53

Table 5.5: Comparison of performances of the two proposed models for the modeling of load profiles in April.

It should be noted that the presented results are very dependent on the data set, and an increase of the data set size would lead to better performances. For instance, as it was mentioned in section 4.3.2, the clustering and classification model suffers more heavily the reduced number of samples of the training set, increasing the error associated to clustering and classification. Therefore, an increase of the number of buildings used to train the models could derive in important improvements of both but, especially, this second one. However, even if the number of samples increases drastically, the drawback of this second model is that errors of each step are accumulated, while the regression model can only have one source of error. The problem of this situation is that the error associated to regression, which leads to the total error of the first model, is also present in the second model, with only small improvements, meaning that the error of the second model is almost determined to be greater than the error of the first one. The unique way of obtaining a competitive model implemented with clustering and classification would be an implementation that avoids the regression step, as it was proposed by McLoughlin et al. [17], although an excessive number of buildings would be needed to train the model, which is not likely to be possible or worthwhile.

For these reasons, it can be concluded that the proposed model in the first instance is the most suitable to model the load profiles of non-residential buildings since it generates more accurate load profiles than the second one, showing that ensemble learning methods are a robust tool to improve the accuracy of a model.

Even though the presented results can be considered fairly good, more accurate predictions, in terms of mean MAPE and mean sMAPE, would be needed in order to obtain a useful tool for the presented real-world applications. However, a close examination of the individual predicted load profiles can offer a very interesting insight of the sources of error of the models. For instance, very good predictions are obtained for some buildings, as depicted in Figure 5.4, where the MAPE and sMAPE do not exceed 15%. These are very good predictions, which suggest that the predictive models proposed in this work can achieve very accurate results and could indeed be reliably used for the intended purpose.

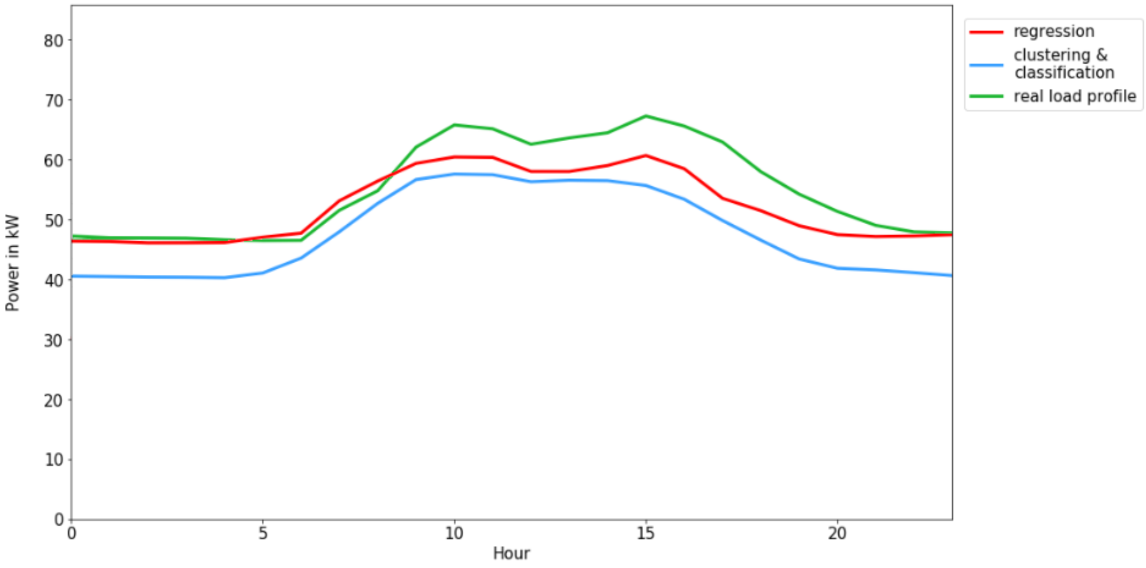


Figure 5.4: Load profiles predicted by both models compared to the real profile from the test set (I).

In contrast, for several buildings, the predicted profiles are far from the real profiles, which are the ones leading to the relatively large mentioned errors. However, for most of these buildings, none of the models is able to obtain a good prediction and, moreover, the predicted profiles are systematically below the real profiles in all these cases. Figure 5.5 illustrates an example of this situation. A further examination of the original data sets shows that, for such situations, almost no appliances were reported, resulting in an extremely large proportion of zero values regarding appliance features in the resulting input set. Although the situation of having a very limited number of appliances is perfectly possible, it becomes unlikely for buildings as the one presented in Figure 5.5, where the building has a high power consumption, indicating that the source of error in these situations could be the bad quality of the given data. For these circumstances, the models have no chance to quantify the relation between the energy consumption and the appliances. In addition, the missing values which were extensively discussed in section 4.2.3 could also have a negative effect on the predicted load profiles, since imputed values have an intrinsic error.

The obtained results illustrate the relevance of the quality of data when performing a predictive task. Incomplete or erroneous input data will lead, not only to a worse performance for the samples affected by the lack of information as it has been explained, but also for any other building with complete information, since the model will have been trained with the erroneous samples that will hinder the modeling of all the buildings. It is probably the fact of having such an incomplete input data set what results in high nonlinearities that make linear algorithms unsuitable for the proposed task, in favor of other algorithms such as random forests that are able to handle this problem.

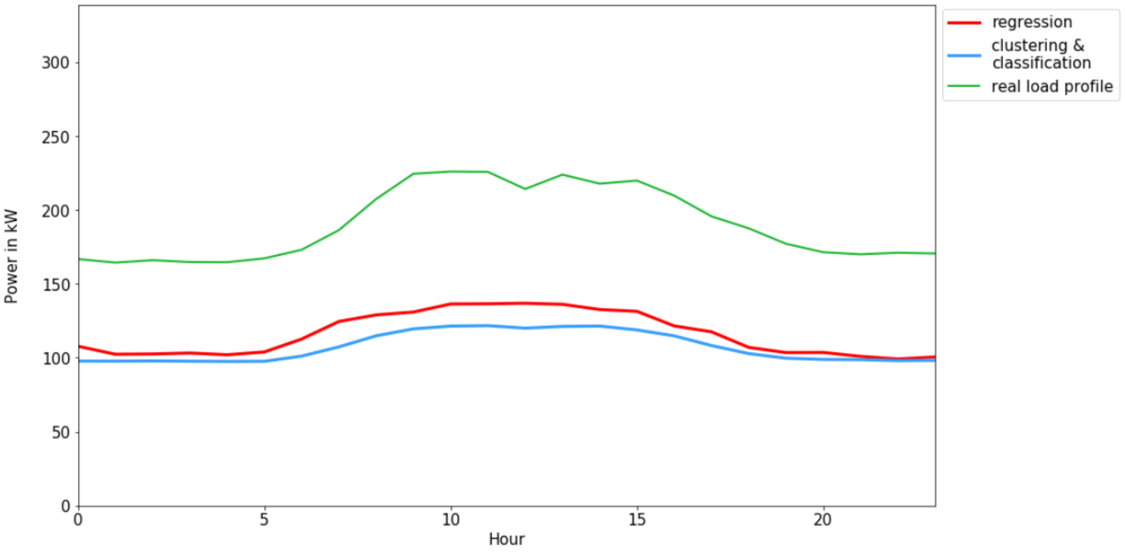


Figure 5.5: Load profiles predicted by both models compared to the real profile from the test set (II).

5.4 Analysis of dependencies between load profiles and features

Besides the generation of load profiles, this project aims to obtain the main dependencies between the load profiles of the studied non-residential buildings and their selected attributes regarding the appliances and rooms. To achieve this purpose, certain algorithms such as linear regression or tree-based algorithms can be used to assess the importance of the input features and therefore will be presented in this section. In order to provide the most detailed information regarding the mentioned dependencies, the regression model will be used for this task since a time decomposition of the dependencies can be made.

Tree-based algorithms offer useful tools to evaluate the importance of features to fit the model. These algorithms provide the fraction of the total divisions that each feature was used to split the input space during training. Therefore, the provided information should be understood as how important each feature for the model was to fit the data. In this sense, even if the importance of a feature is normally related to a major contribution to the output, it does not necessarily imply it. For instance, the positive or negative effects of the features in the output cannot be evaluated. Figure 5.6 and Figure 5.7 illustrate the relative importance of the given appliances (aggregating individual features) and of the most important room types respectively. The calculation of each feature importance is made by averaging the relative importance obtained with the RF and the GBDT.

Regarding appliances, ventilation devices are clearly the most relevant for the model, accounting for almost half of the total importance. The fact of being the appliance which included more information as different features also contributes to this situation. It is not surprising to see that the importance of features has two differentiated time intervals, corresponding to the working (8h-18h) and non-working hours. Figure A.1 summarizes the information of each appliance for three time intervals of the day, related to working and non-working hours. It should be noted, for instance, that the decrease of the relative importance of the ventilation does not mean that it becomes less relevant in the sense that it derives in a lower increase of the output power. In fact, since the total output power increases more intensively, ventilation devices derive in a greater output power than for the previous hours. The only reason for the mentioned decrease is the increase of the importance of other features such rooms and heating which did not had a high influence in other hours. The obtained results are equivalent to the ones obtained for the modeling of load profiles in July (Figure A.2) and December (Figure A.3), standing out the significant increase of the importance of the ventilation for the first case, and the heating for the second.

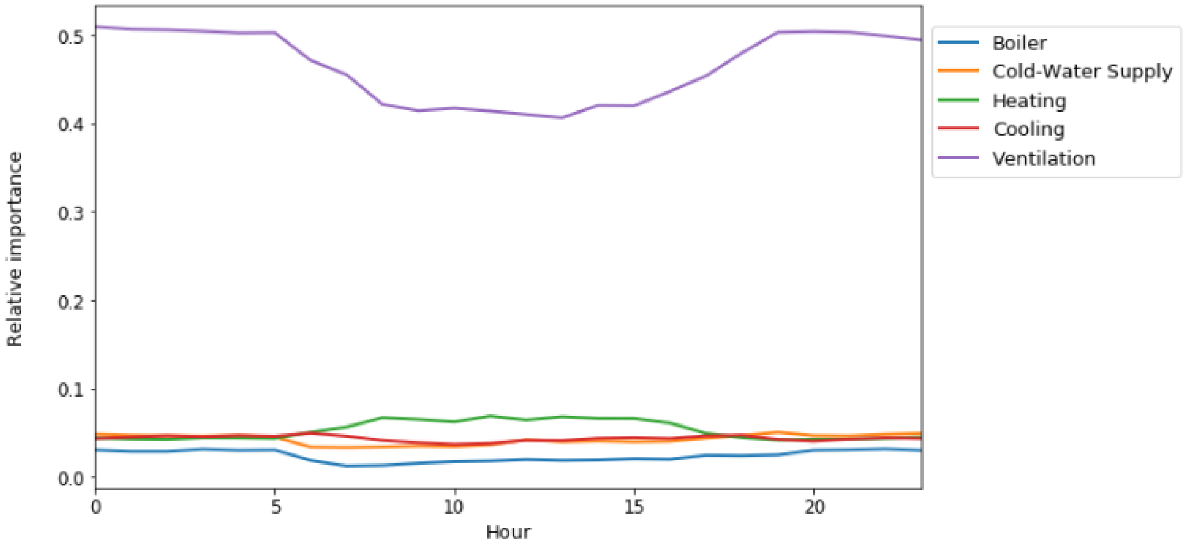


Figure 5.6: Relative importance of the 5 appliance types for the modeling of load profiles in April.

Regarding room types, Figure 5.7 shows that technical equipment rooms, where most of the building equipment is located, are the most relevant for the model. However, their relevance is far from the ventilation appliances. An important insight can be obtained from features such as heating devices or traffic area, office and WC rooms. In contrast to ventilation devices or technical equipment rooms, their importance drastically increases during the working hours. This can be understood in the way that, while the latter contribute more as a shift of the average power, the first ones have a greater effect on the shape of the profile. The mentioned variation of the shape usually corresponds to the working hours. Therefore, the relevance of office, WC or traffic area rooms is likely to come from their direct relation to the number of people working in the building, which has a considerable effect on the increase of power consumption during these hours. As with appliances, Figure A.4 summarizes the information of each of these rooms for the mentioned three time intervals, while Figure A.5 and Figure A.6 show the relative importance of the room types for July and December, respectively.

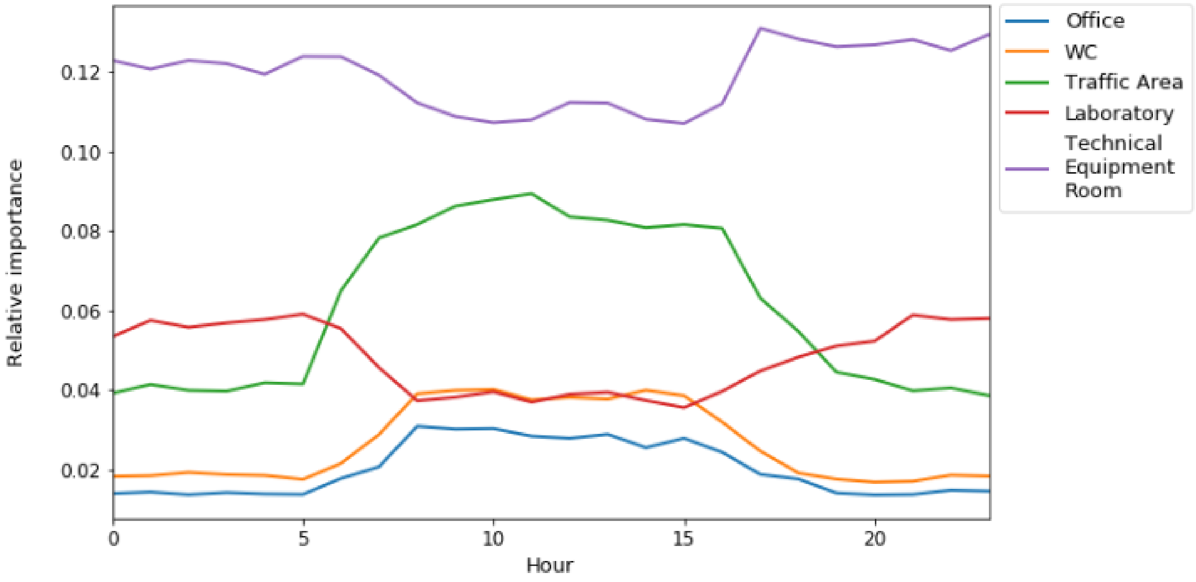


Figure 5.7: Relative importance of the 5 most relevant room types for the modeling of load profiles in April.

Given the importance of ventilation appliances, a further study of the relative importance of the selected features should be done in order to analyze the importance of each type of ventilation appliances, which is illustrated in Figure 5.8. It can be appreciated that central and room devices are the most influent on the energy consumption, while wall devices are almost neglectable. In contrast, the division between ingoing, outgoing and circulating devices does not derive in a significant difference of the importance. In addition, heating and cooling power features outstand as significant features. Ultimately, the same insights can be obtained from load profiles in July (Figure A.7) and December (Figure A.8), perceiving in the latter a relevant increase of the heating power importance.

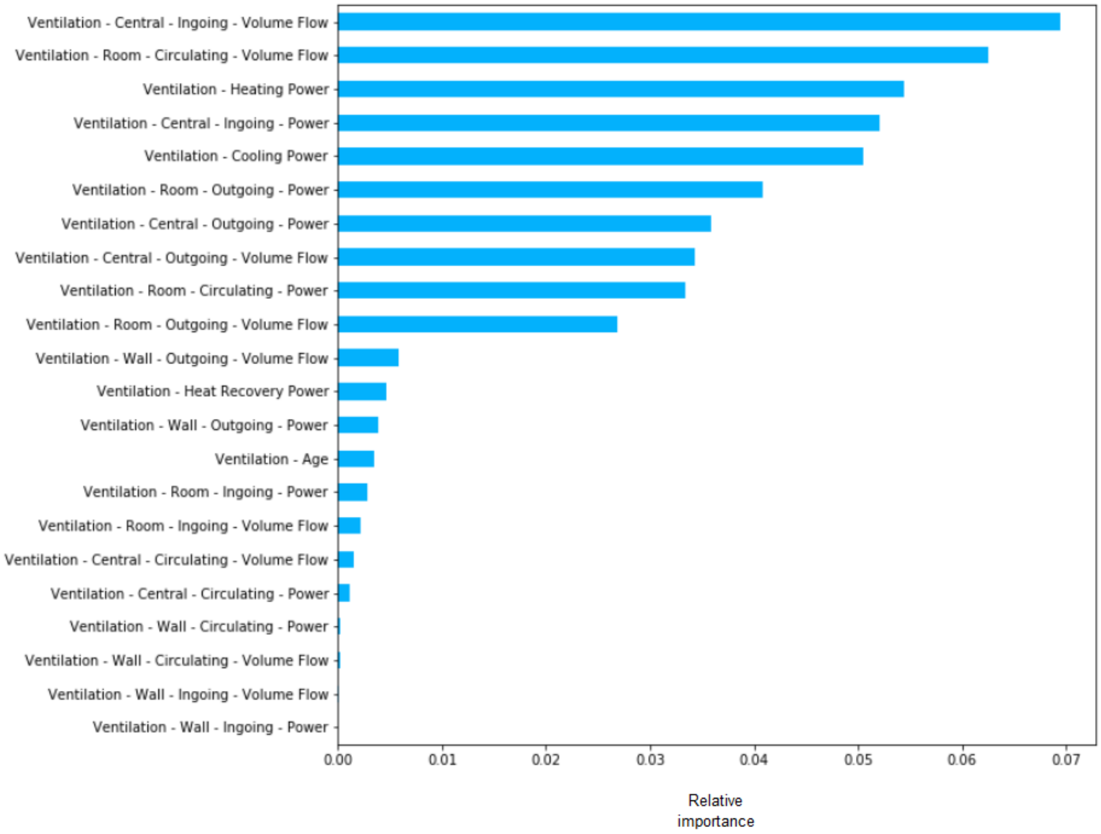


Figure 5.8: Mean relative importance of ventilation features for the modeling of load profiles in April.

The main disadvantage of using the presented tool for the dependency analysis is the impossibility of assessing the positive or negative effects of the attributes to the power consumption. To infer this kind of relation, linear algorithms can offer a good solution. The implementation of linear regression algorithms was explained in section 3.1.5.1, where the parameters corresponding to the slope of the function with respect to each variable are estimated. In consequence, these parameters mean how much does the output vary when the respective feature is varied. Although this would be a suitable way of extracting the desired dependencies, the aforementioned fact that the problem is not linear means that the extracted relations are not as reliable as the ones already shown in this section. In addition, due to the presence of negative values, they cannot be aggregated, leading to an undesired complexity of the results. Therefore, the main insight to be taken from the linear regression coefficients is the direction of the relation between power consumption and features. To illustrate this, only an exemplary result is shown in Figure 5.9, corresponding to a non-working hour of the base case. For simplicity, features without influence in the model (their coefficient is 0) are excluded. It can be observed that, while most of the features have a positive effect on the output power, others such as appliances ages and the heating non-electrical power source feature show a negative effect on the output.

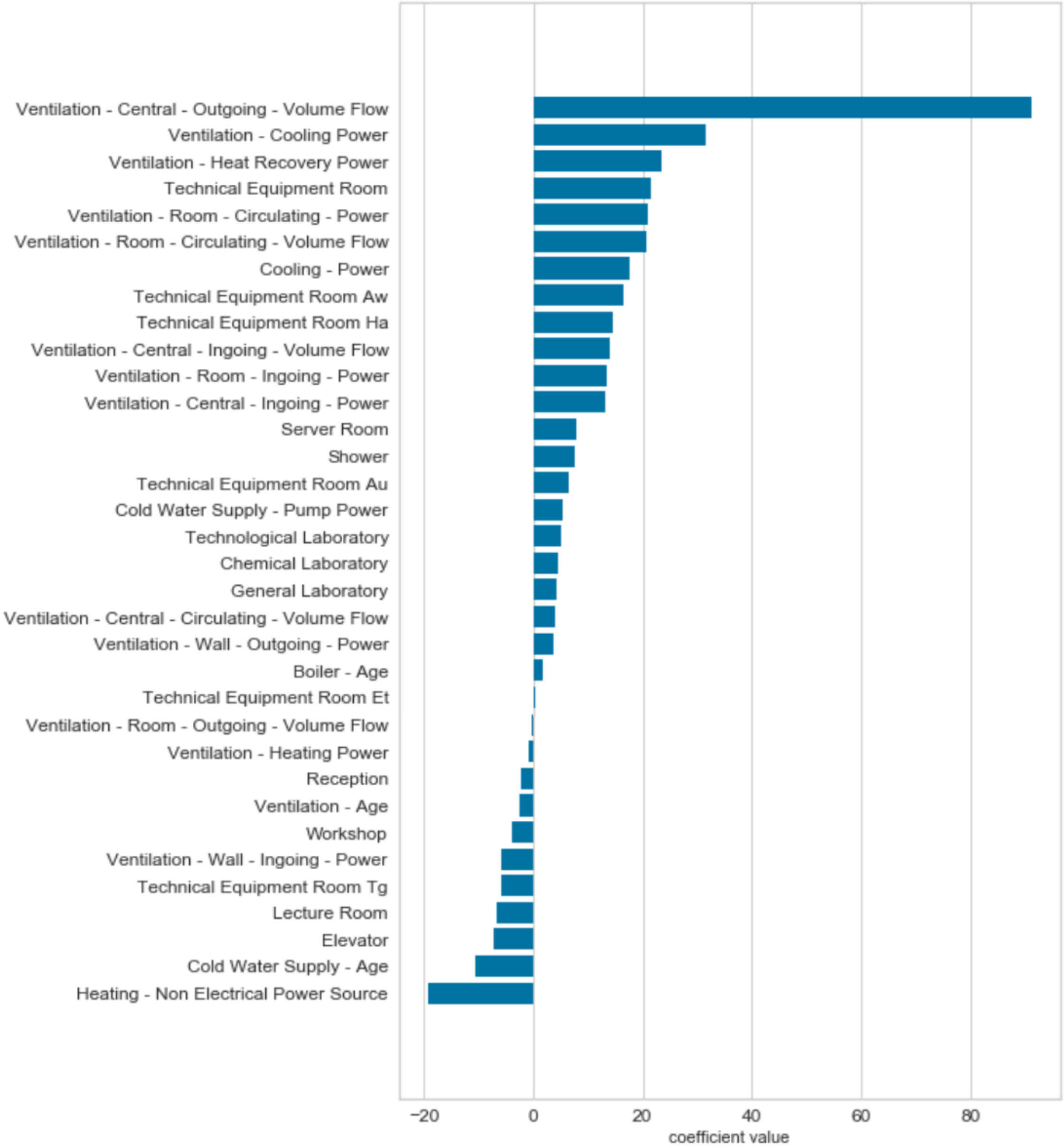


Figure 5.9: Linear regression parameter values of a non-working hour for the modeling of load profiles in April.

5.5 Modeling of the weekly energy consumption

This last section of the chapter will present the modeling of the load profile of a complete week for the studied buildings. The implemented models can be used to model the load profiles of different seasons using the same input data. In the same way, they can be trained with load profiles from different days, for instance, working and non-working days. Therefore, these types of load profiles can be modeled separately and then be combined to generate the load profile of a week. An exemplary profile is depicted in Figure 5.10. For simplicity, it has been assumed that the load profile does not vary significantly from one day to another of the same type, which is the common case.

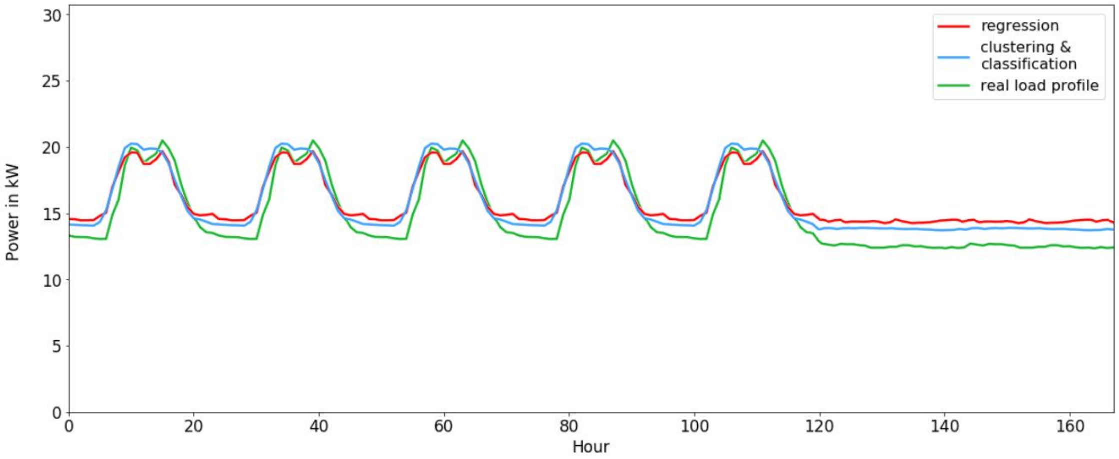


Figure 5.10: Example of the modeling of the weekly load profile of a building with both approaches.

6 Conclusions and outlook

6.1 Conclusions

In this work, two predictive models for the top-down modeling of load profiles for non-residential buildings are developed. The first model consists on a regression analysis for the energy consumption of every hour while the second model is based on the clustering and classification of normalized load profiles combined with the regression of the maximum load. Both models are composed of several machine learning algorithms, for instance, random forests, support-vector machines, gradient boosting decision trees, artificial neural networks and linear regression, and then they use a stacking method, to improve the prediction performance. Stacking is an ensemble learning method that consists on using a predictive algorithm which uses the predictions from the previous models as input to predict the final output.

Despite the presented models were developed to be trained with a set of data from non-residential buildings in order to predict the load profiles of other buildings, they can be used to model any type of buildings, since the number of input parameters is flexible. Results demonstrate that the implemented models can accurately predict load profiles, proving to be useful tools for the intended purpose of modeling load profiles. However, performances suffer from the lack of quality of the input data and thus, a proper evaluation should be made with higher quality data. For all studied cases from the given buildings, the regression model obtained the best performance. This is due to the fact that both approaches perform a regression analysis which derives in similar errors, but the second approach includes the classification and clustering steps, resulting in a decrease of the performance. Regarding the algorithms, random forests proved to perform better than any other for this specific data set. Moreover, the use of an ensemble method, also implemented with a random forest, results in an increase of the performance with respect to the other algorithms.

In addition, the algorithms composing the regression model are used to determine the dependencies between the load profiles and the appliances and rooms of the buildings. Ventilation devices show to be more important for the model than any other room or appliance, accounting for about half of the total importance of the features. The time decomposition of the dependencies show that they can generally be divided into three different time intervals corresponding to the working and non-working hours. During the working hours, the dependency between load profiles and heating devices and rooms such as offices, traffic areas and WCs increases significantly, meaning that they have a high influence on the variations of the energy consumption during the day. This situation is probably due to the relation of the latter room types to the occupancy of buildings, which has a direct effect on energy consumption.

6.2 Outlook

Even though the implemented models, and especially the regression model, proved to be suitable to accomplish the intended task, the proposed methodology could be improved in different aspects.

In any case, the improvement of the data quality is of major importance in order to do a proper evaluation of the models and obtain significantly better results. It would be preferable to have a smaller number of measured features if this enables to have a more complete and correct input data set.

The inferred dependencies between the load profiles and certain room types suggest that people-related room types can have a high influence on the variations of the energy consumption during the day. This implies that the inclusion of general information about the building and particularly occupancy information could provide extremely useful information for the model and derive in drastic increases of accuracy, without needing to make any modification of the model.

Regarding the data preprocessing, further feature selection techniques should be explored. The current methodology was implemented in order to extract all the dependencies between load profiles and features and hence, feature selection was not performed. However, the current data set has a high dimensionality, and results have demonstrated that many features are not relevant for the predictive models. Therefore, feature selection techniques such as greedy search would allow an important reduction of the dimensionality of the data, simplifying the prediction process, and could even lead to an improvement of the performance.

The implementation of the model could also be modified to achieve a wider modeling capability. The inclusion of seasonal or mean temperature variables could allow the modeling of load profiles for specific times of the year without the need of modeling them separately. This is a similar approach to the one proposed by Wang et al. [9]. The implemented approach would be essentially the same as proposed in this work, but the load profiles of different seasons could be trained at the same time and therefore, they would be treated as different samples.

A last ampliation of the model, even more ambitious than the previous one, would involve the increase of the number of buildings that can be modeled. With the actual implementation, only load profiles of buildings from the same population, which in this case are the buildings from JRC, can be modeled. However, the collection of building data coming from a wider geographic area would radically increase the potential of the proposed models, being able to predict profiles of non-residential buildings even at a country level. To achieve such purpose, the inclusion of geographic variables would be necessary.

References

[1] V. Bürger, T. Hesse, B. Köhler, A. Palzer and P. Engelmann, "German Energiewende—different visions for a (nearly) climate neutral building sector in 2050", *Energy Efficiency*, vol. 12, no. 1, pp. 73-87, 2018.

[2] A. Mickaitytė, E. Zavadskas, A. Kaklauskas and L. Tupėnaitė, "The concept moel of suistainable buildings refurbishment", *International Journal of Strategic Property Management*, vol. 12, no. 1, pp. 53-68, 2008.

[3] Deutsche Energie-Agentur, "Der dena-Gebäudereport: Statistiken und Analysen zur Energieeffizienz im Gebäudebestand", 2016.

[4] European Union Energy Initiative Partnership Dialogue Facility (EUEI PDF), "Energy and Climate Change Adaptation in Developing Countries", 2017.

[5] S. Weitemeyer, D. Kleinhans, T. Vogt and C. Agert, "Integration of Renewable Energy Sources in future power systems: The role of storage", *Renewable Energy*, vol. 75, pp. 14-20, 2015.

[6] A. Facci, V. Krastev, G. Falcucci and S. Ubertini, "Smart integration of photovoltaic production, heat pump and thermal energy storage in residential applications", *Solar Energy*, 2018.

[7] M. Döring, "Prediction vs Forecasting", *Datascienceblog.net*, 2019. [Online]. Available: https://www.datascienceblog.net/post/machinelearning/forecasting_vs_prediction/.

[8] J. Massana, C. Pous, L. Burgas, J. Melendez and J. Colomer, "Short-term load forecasting in a non-residential building contrasting models and attributes", *Energy and Buildings*, vol. 92, pp. 322-330, 2015.

[9] C. Wang, G. Grozev and S. Seo, "Decomposition and statistical analysis for regional electricity demand forecasting", *Energy*, vol. 41, no. 1, pp. 313-325, 2012.

[10] Y. Ge, C. Zhou and D. Hepburn, "Domestic electricity load modelling by multiple Gaussian functions", *Energy and Buildings*, vol. 126, pp. 455-462, 2016.

[11] A. Kipping and E. Trømborg, "Modeling Aggregate Hourly Energy Consumption in a Regional Building Stock", *Energies*, vol. 11, no. 1, p. 78, 2017.

[12] B. Gao, X. Liu and Z. Zhu, "A Bottom-Up Model for Household Load Profile Based on the Consumption Behavior of Residents", *Energies*, vol. 11, no. 8, p. 2112, 2018.

[13] A. Capasso, W. Grattieri, R. Lamedica and A. Prudenzi, "A bottom-up approach to residential load modeling", *IEEE Transactions on Power Systems*, vol. 9, no. 2, pp. 957-964, 1994.

[14] I. Richardson, M. Thomson and D. Infield, "A high-resolution domestic building occupancy model for energy demand simulations", *Energy and Buildings*, vol. 40, no. 8, pp. 1560-1566, 2008.

[15] C. Sandels, D. Brodén, J. Widén, L. Nordström and E. Andersson, "Modeling office building consumer load with a combined physical and behavioral approach: Simulation and validation", *Applied Energy,* vol. 162, pp. 472-485, 2016.

[16] F. McLoughlin, A. Duffy and M. Conlon, "Characterising domestic electricity consumption patterns by dwelling and occupant socio-economic variables: An Irish case study", *Energy and Buildings,* vol. 48, pp. 240-248, 2012.

[17] F. McLoughlin, A. Duffy and M. Conlon, "A clustering approach to domestic electricity load profile characterisation using smart metering data", Applied Energy, vol. 141, pp. 190-199, 2015.

[18] L. Pedersen, J. Stang and R. Ulseth, "Load prediction method for heat and electricity demand in buildings for the purpose of planning for mixed energy distribution systems", *Energy and Buildings,* vol. 40, no. 7, pp. 1124-1134, 2008.

[19] C. Bennett, R. Stewart and J. Lu, "Autoregressive with Exogenous Variables and Neural Network Short-Term Load Forecast Models for Residential Low Voltage Distribution Networks", *Energies*, vol. 7, no. 5, pp. 2938-2960, 2014.

[20] A. Lahouar and J. Ben Hadj Slama, "Day-ahead load forecast using random forest and expert input selection", *Energy Conversion and Management,* vol. 103, pp. 1040-1051, 2015.

[21] S. Touzani, J. Granderson and S. Fernandes, "Gradient boosting machine for modeling the energy consumption of commercial buildings", *Energy and Buildings,* vol. 158, pp. 1533-1543, 2018.

[22] F. Divina, A. Gilson, F. Goméz-Vela, M. García Torres and J. Torres, "Stacking Ensemble Learning for Short-Term Electricity Consumption Forecasting", Energies, vol. 11, no. 4, p. 949, 2018.

[23] T. Mitchell, *Machine Learning*. McGraw Hill, 1997, p. 2.

[24] T. O. Ayodele, “Types of machine learning algorithms,” in *New Advances in Machine Learning. InTech*, 2010.

[25] T. Hastie, R. Tibshirani and J. Friedman, *The elements of statistical learning*. New York: Springer, 2004.

[26] O. Balci, "Validation, verification, and testing techniques throughout the life cycle of a simulation study", *Annals of Operations Research,* vol. 53, no. 1, pp. 121-173, 1994.

[27] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[28] Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Wermter S, Riloff E, Scheler G, editors. The Fourteenth International Joint Conference on Artificial Intelligence (IJCAI)San Francisco, CA: Morgan Kaufman; 1995.

[29] J. R. Rice. The algorithm selection problem. Advances in Computers, 1976.

[30] G. James, D. Witten, T. Hastie and R. Tibshirani, An introduction to statistical learning. New York [etc.]: Springer, 2017.

[31] L. Auria and R. Moro, "Support Vector Machines (SVM) as a Technique for Solvency Analysis", *SSRN Electronic Journal,* 2008.

[32] D. Wolpert, "Stacked generalization", *Neural Networks*, vol. 5, no. 2, pp. 241-259, 1992.

[33] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, *The Journal of Machine Learning Research*, 13, p.281-305, 2012

[34] A. Botchkarev, Performance Metrics (Error Measures) in Machine Learning Regression, Forecasting and Prognostics: Properties and Typology. 2018.

[35] B. Walther and J. Moore, "The concepts of bias, precision and accuracy, and their use in testing the performance of species richness estimators, with a literature review of estimator performance", *Ecography,* vol. 28, no. 6, pp. 815-829, 2005.

[36] S. K. Morley, "Alternatives to accuracy and bias metrics based on percentage errors for radiation belt modeling applications," Los Alamos National Laboratory report, 2016.

[37] J. Tayman, Swanson DA: On the validity of mape as a measure of population forecast accuracy. *Popul Res Policy Rev* 1999

[38] A. Jain, M. Murty and P. Flynn, "Data clustering: a review", *ACM Computing Surveys*, vol. 31, no. 3, pp. 264-323, 1999.

[39] C. Deb and S. Lee, "Determining key variables influencing energy consumption in office buildings through cluster analysis of pre- and post-retrofit building data", *Energy and Buildings*, vol. 159, pp. 228-245, 2018.

[40] W. McKinney, "Data structures for statistical computing in python," in Proceedings of the 9th Python in Science Conference, S. van der Walt and J. Millman, Eds., 2010, pp. 51–56.

[41] S. van der Walt, S. C. Colbert, and G. Varoquaux, "The numpy array: A structure for efficient numerical computation," Computing in Science & Engineering, vol. 13, no. 2, pp. 22–30, 2011.

[42] S.B. Kotsiantis, D. Kanellopoulos, P.E. Pintelas, Data preprocessing for supervised learning, *Trans. Eng. Comput. Technol*. 12, 277–282, 2006.

[43] J. Tukey and F. David, "Exploratory Data Analysis", *Biometrics*, vol. 33, no. 4, p. 768, 1977.

[44] D. Heitjan and D. Rubin, "Ignorability and Coarse Data", *The Annals of Statistics*, vol. 19, no. 4, pp. 2244-2253, 1991.

[45] YC.Yuan. Multiple Imputation for Missing Data: Concepts and New Development. Rockville, MD: SAS Institute Inc, 2001.

[46] A. Daly, T. Dekker and S. Hess, "Dummy coding vs effects coding for categorical variables: Clarifications and extensions", Journal of Choice Modelling, vol. 21, pp. 36-41, 2016.

[47] DIN 277-2 Grundflächen und Rauminhalte von Bauwerken im Hochbau. Berlin: Beuth, 2005.

[48] L. Shalabi, Z. Shaaban and B. Kasasbeh, "Data Mining: A Preprocessing Engine", Journal of Computer Science, vol. 2, no. 9, pp. 735-739, 2006.

[49] G. Ke et al., “Lightgbm: A highly efficient gradient boosting decision tree,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 3149–3157.

Appendix A

	mean MAPE [%]	SD MAPE [%]	mean sMAPE [%]	SD sMAPE [%]
Linear regression	118.98	107.96	78.04	30.84
Support-vector machine	82.22	74.11	57.65	24.97
Random forest	50.00	43.96	37.49	22.62
Gradient boosting decision tree	61.91	46.32	44.34	21.02
Artificial neural network	139.37	149.74	69.42	29.17
Ensemble	46.88	38.96	36.05	19.64

Table A.1: Performances of the regression model for the modeling of load profiles in July.

	mean MAPE [%]	SD MAPE [%]	mean sMAPE [%]	SD sMAPE [%]
Linear regression	50.78	39.45	44.05	26.64
Support-vector machine	47.77	33.10	51.00	29.44
Random forest	34.33	26.05	35.69	24.90
Gradient boosting decision tree	65.89	41.02	56.43	32.08
Artificial neural network	91.35	137.37	60.75	41.24
Ensemble	32.41	24.94	32.75	22.97

Table A.2: Performances of the regression model for the modeling of load profiles in December.

	Gradient boosting decision tree	Random forest	Support-vector machine	Ensemble
Accuracy [%]	30.77	23.08	30.77	30.77

Table A.3: Accuracies of the classification of normalized profiles for the modeling of load profiles in July.

	Gradient boosting decision tree	Random forest	Support-vector machine	Ensemble
Accuracy [%]	28.57	42.86	28.57	42.86

Table A.4: Accuracies of the classification of normalized profiles for the modeling of load profiles in December.

	MAPE [%]	sMAPE [%]
Linear regression	43.66	43.43
Support-vector machine	38.76	46.76
Random forest	27.91	30.34
Gradient boosting decision tree	41.15	43.65
Ensemble	25.96	28.67

Table A.5: Performances of the maximum load regressions for the modeling of load profiles in July**.**

	MAPE [%]	sMAPE [%]
Linear regression	71.06	65.96
Support-vector machine	45.01	45.09
Random forest	27.93	29.64
Gradient boosting decision tree	34.90	33.53
Ensemble	23.69	25.73

Table A.6: Performances of the maximum load regressions for the modeling of load profiles in December.

	mean MAPE [%]	SD MAPE [%]	mean sMAPE [%]	SD sMAPE [%]
Clustering and classification	42.33	41.08	42.36	25.98

Table A.7: Performances of clustering and classification model for the modeling of load profiles in July.

	mean MAPE [%]	SD MAPE [%]	mean sMAPE [%]	SD sMAPE [%]
Clustering and classification	60.48	45.15	50.57	19.16

Table A.8: Performances of clustering and classification model for the modeling of load profiles in December.

	mean MAPE [%]	SD MAPE [%]	mean sMAPE [%]	SD sMAPE [%]
Regression	32.41	24.94	32.75	22.97
Clustering and classification	42.33	41.08	42.36	25.98

Table A.9: Comparison of performances of the two proposed models for the modeling of load profiles in July.

	mean MAPE [%]	SD MAPE [%]	mean sMAPE [%]	SD sMAPE [%]
Regression	46.88	38.96	36.05	19.64
Clustering and classification	60.48	45.15	50.57	19.16

Table A.10: Comparison of performances of the two proposed models for the modeling of load profiles in December.

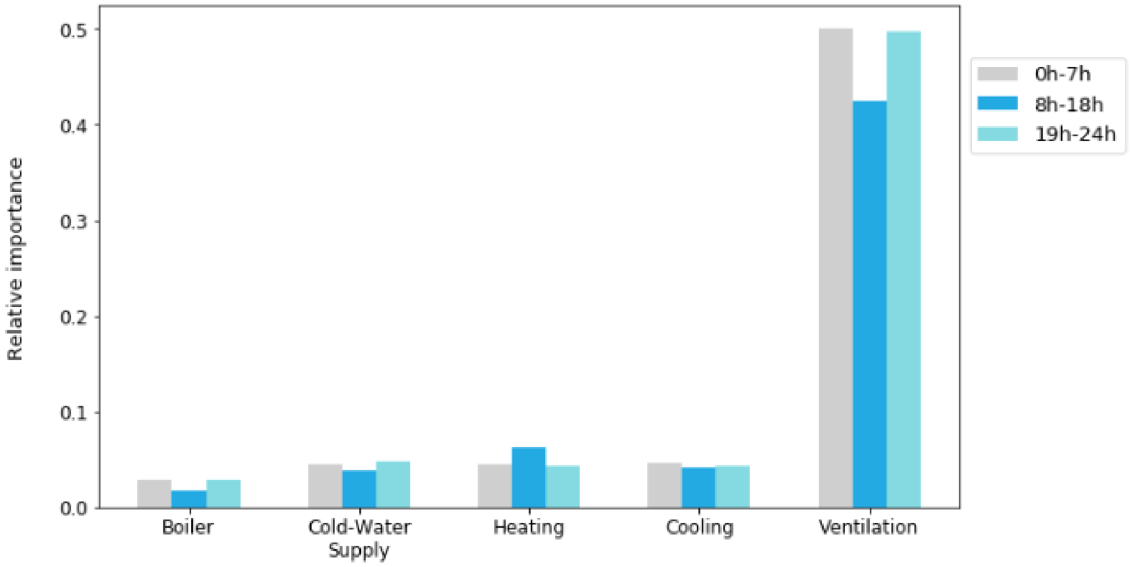


Figure A.1: Relative importance of appliances during the three intervals of a day for the modeling of load profiles in April.

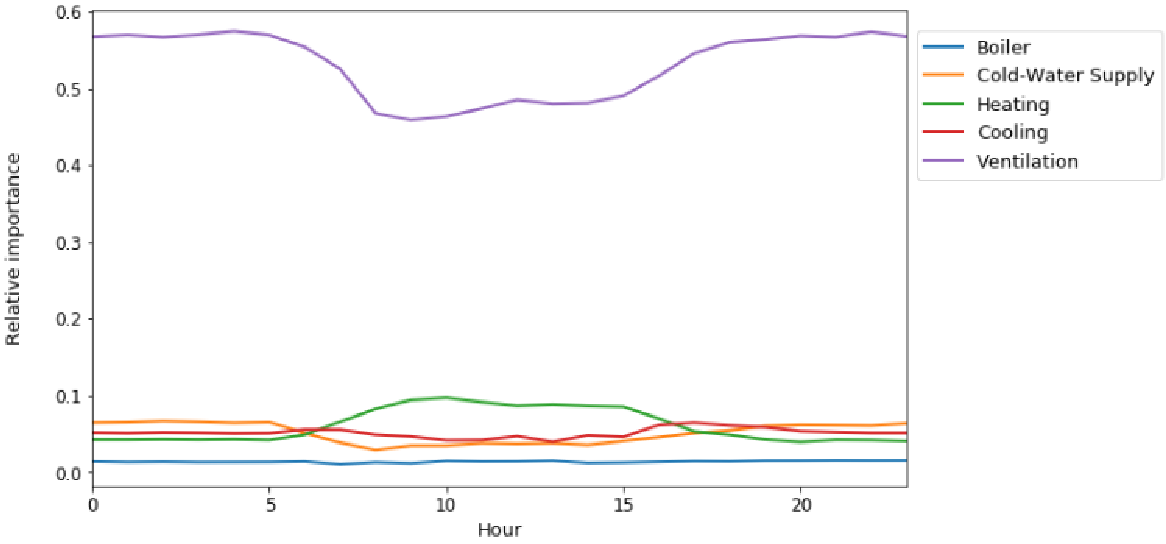


Figure A.2: Relative importance of the 5 appliance types for the modeling of load profiles in July.

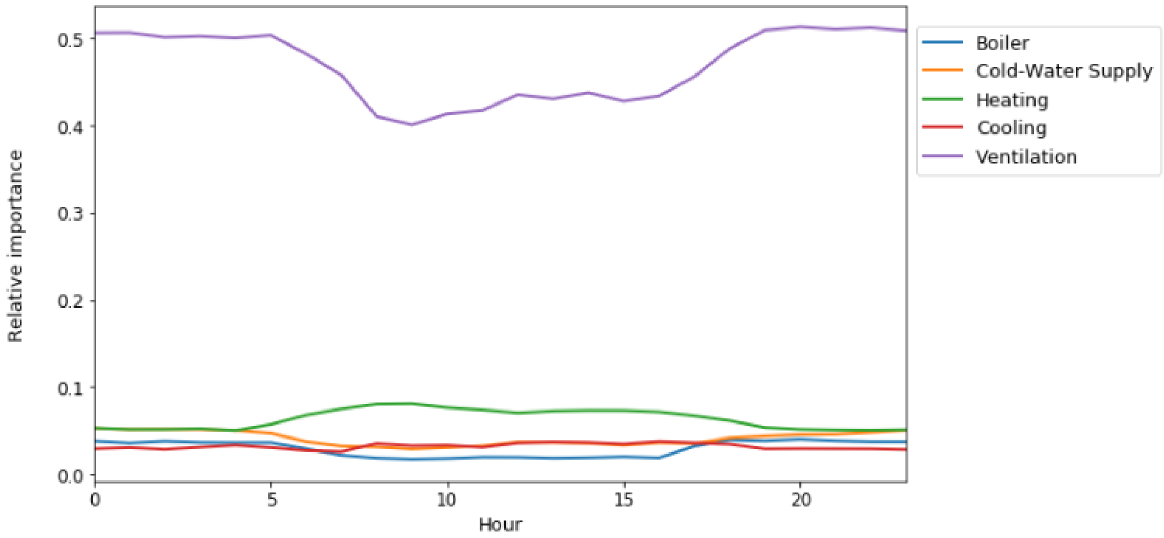


Figure A.3: Relative importance of the 5 appliance types for the modeling of load profiles in December.

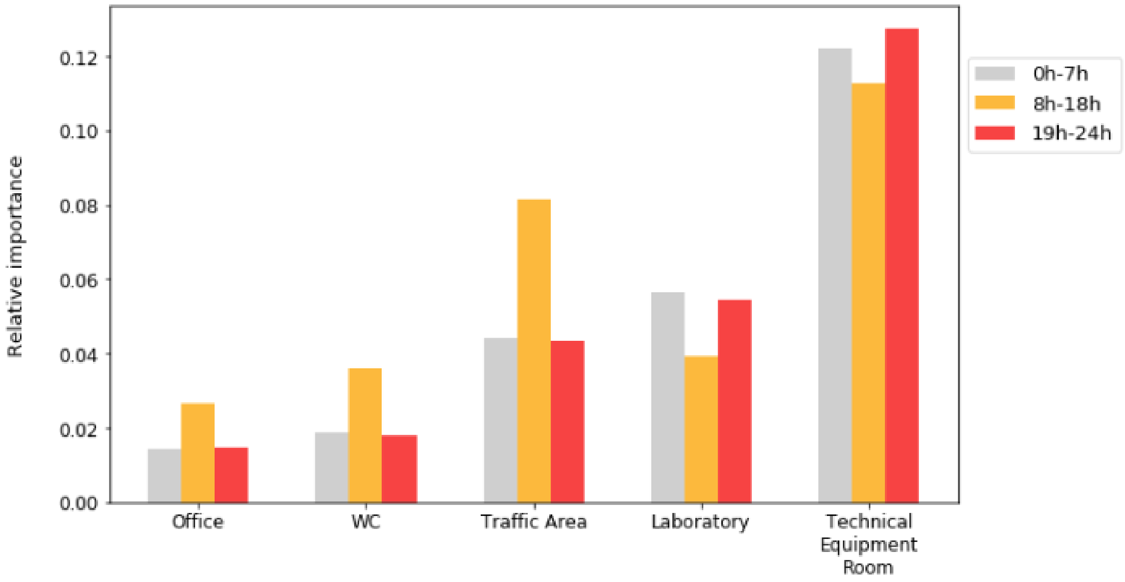


Figure A.4: Relative importance of most relevant room types during the three intervals of a day for the modeling of load profiles in April.

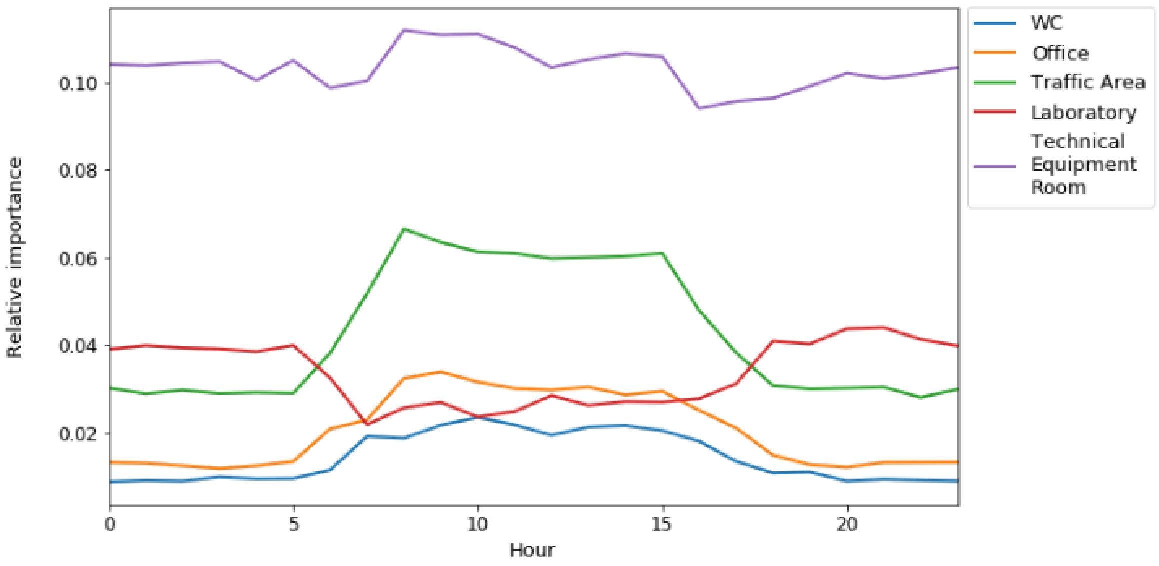


Figure A.5: Relative importance of the 5 most relevant room types for the modeling of load profiles in July.

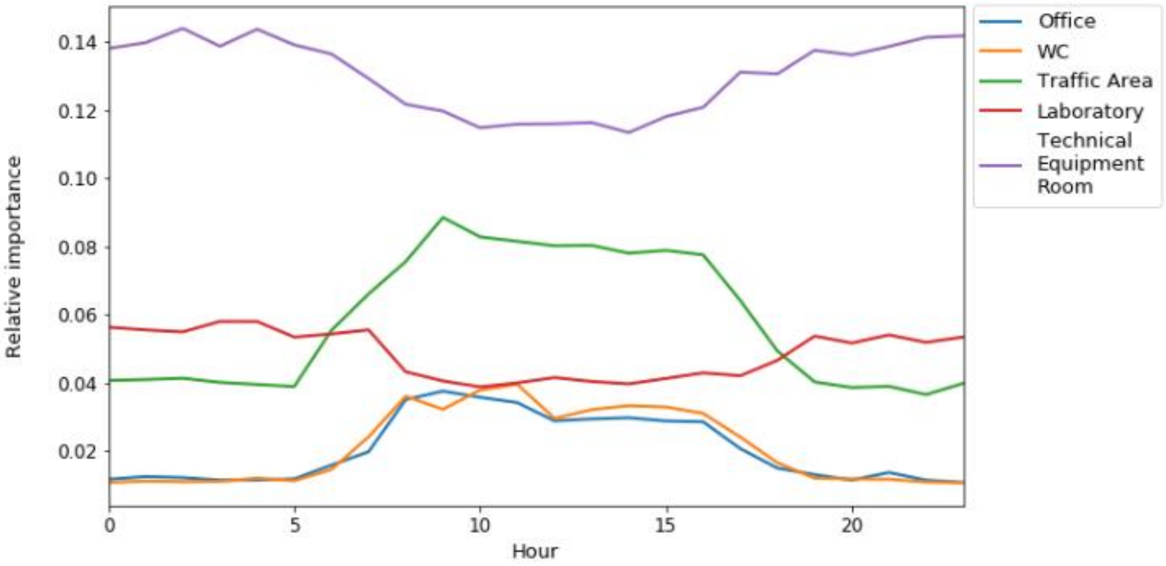


Figure A.6: Relative importance of the 5 most relevant room types for the modeling of load profiles in December.

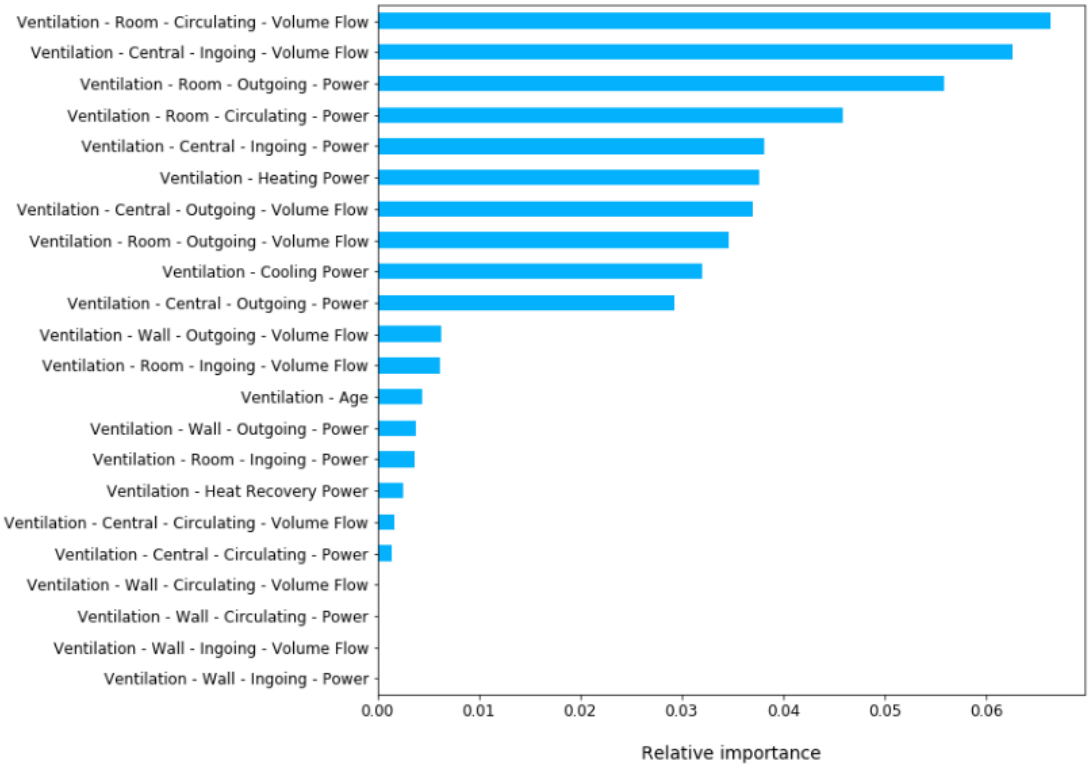


Figure A.7: Mean relative importance of ventilation features for the modeling of load profiles in July.

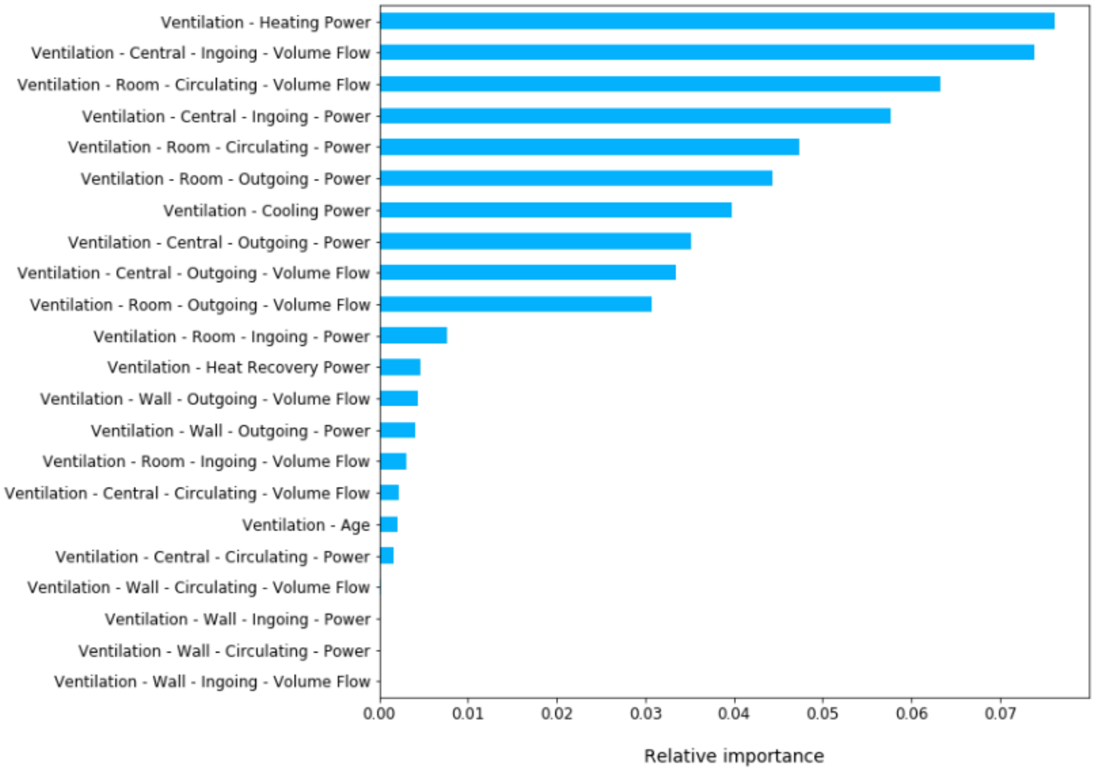


Figure A.8: Mean relative importance of ventilation features for the modeling of load profiles in December.