A Multimodal and Transfer Learning Approaches to Residential Building Operational Energy Estimation

Yulan Sheng

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

The University of Sheffield

Faculty of Engineering

Department of Civil and Structural Engineering

August 2023

Declaration

All work presented within this thesis is the author's own work except where specific reference has been made to the work of others.

Some research material produced during the course of this project has been published and/or is currently undergoing peer-review for publication. Where a chapter includes reproductions of such material, which may with the explicit permission of relevant co-author(s) be featured wholly or in parts, these are indicated at the outset of the chapter.

Signiture:

Date: 28 August 2023

Acknowledgement of collaborative work within the thesis

The candidate confirms that the work submitted is her own, except where work that has formed part of jointly authored publications has been included. The contribution of the candidate to this work has been explicitly indicated below.

1. Sheng, Y., Ward, W.O., Arbabi, H., Álvarez, M. and Mayfield, M., 2022, September. Deep multimodal learning for residential building energy prediction. In IOP Conference Series: Earth and Environmental Science (Vol. 1078, No. 1, p. 012038). IOP Publishing.

   In this publication, the candidate contributed to the design of the study, undertook the study and wrote the manuscript.

   Direct reproductions of this published material are included in Chapter 1, 2, 3, 5, 7, 8 of this thesis.

2. Sheng, Y., Arbabi, H., Ward, W.O., Álvarez, M. and Mayfield, M., City-scale estimation on residential building energy consumption and evaluation on multimodal housing features. (In Prep.).

   In this publication, the candidate contributed to the design of the study, undertook the study and wrote the manuscript.

   Direct reproductions of this published material are included in Chapter 1, 2, 3, 4, 7, 8 of this thesis.

3. Sheng, Y., Arbabi, H., Ward, W.O., and Mayfield, M., Knowledge transfer for efficient residential energy estimation and retrofit prioritisation. (In Prep.).

   In this publication, the candidate contributed to the design of the study, undertook the study and wrote the manuscript.

   Direct reproductions of this published material are included in Chapter 1, 2, 3, 6, 7, 8 of this thesis.

The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

Acknowledgements

Many thanks to all my supervisors, Martin, Hadi and Mauricio, my mum, my friends, and everyone from the RISE group for their supports, guidance and feedbacks. Especially Ling Min, for sharing these opportunities to me that made all these good things possible.

This journey would not be as pleasant and peaceful as it is without all of you...

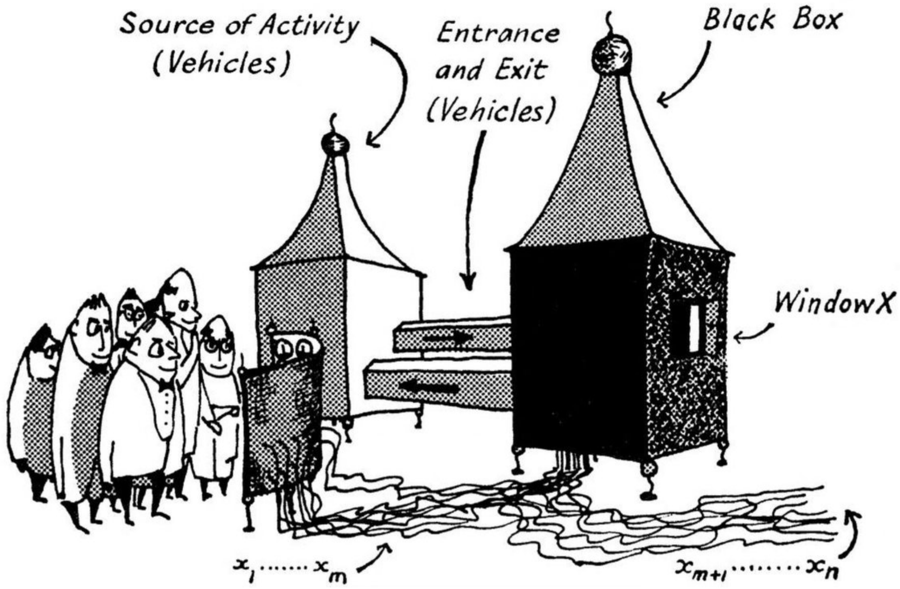


Illustration of a black box by Pask [1]

Abstract

Decarbonising the housing stock is one of the critical mechanisms for the UK to meet the net-zero target by 2050. To identify effective retrofit measures, a data-driven urban retrofit modelling approach is commonly used, involving operational energy estimation and retrofit scenario exploration, which are heavily dependent on the quantity and quality of data. However, accessibility to adequate housing data varies across regions, which poses challenges to establishing a robust understanding of the existing housing stock. In response, this thesis aims to investigate the potential of the application of multimodal and transfer learning approaches, to enhance the trustworthiness and adaptability of operational energy consumption estimations in residential buildings, especially in cases with limited data quality or quantity. Three studies were conducted to answer the following research questions: 1) Can incorporating multi-modalities improve the trustworthiness of operational energy estimation? and 2) Can transfer learning techniques improve the adaptability of operational energy estimation?

To address these questions, this thesis started with an initial study using a machine learning model for operational energy estimation. With the help of an automated machine learning tool, the housings in Sheffield were examined based on Energy Performance Certificates (EPC) and map data from the Ordnance Survey. The results were explained by permutation feature importance and partial dependence, which revealed the critical factors in estimation, and identified the key building elements to guide retrofit. Applying the trained model to Barnsley and Merthyr Tydfil revealed decreased performance in the latter, emphasising the need for a more adaptable approach acknowledging the spatial heterogeneity.

The second study examined the capability of using street view images for energy estimation. The case study found that using images alone is not able to offer accurate estimation. Therefore, to answer the first research question, the idea of multimodal learning was introduced, where both EPC and Google Streetview (GSV) data were incorporated to reduce the bias caused by relying on a single source of data. The models trained using multiple modalities were compared with models built on a single modality, using statistical metrics and the SHapley Additive exPlanations (SHAP) to examine the effectiveness. Statistic metrics R^2 and mean absolute percentage error were employed to evaluate the multimodal network, which shown improvements in prediction accuracy. SHAP was used to examine the changes in feature importance and correlations with and without multiple modalities. The changes demonstrated the model is able to identify and connect key features from both modalities to perform the prediction.

The third study was designed for the second research question. The potential of transfer learning was investigated to address the challenge of adapting a trained model to changing local contexts when only limited data is available for training. Three cities were examined, Barnsley, Doncaster and Merthyr Tydfil. They are either limited in data quantity or quality, to represent possible challenges in model implementation. Developing upon the multimodal network trained on the second study, the layers in the network were adjusted with different train-ability and learning rates, so it is capable of leveraging knowledge from cities with sufficient data to assist in accurate estimation for cities with poor data. The predictions performed with and without transfer learning were evaluated, which demonstrated significant improvements among all the examined cities. Each region was explained with their respective feature importance ranks, to present that, although the models were all developed upon one multimodal network, it is able to leverage key information from each region and adapt the model accordingly.

Overall, this thesis examined the utilisation of deep multimodal network and transfer learning methods, contributing to enhancing the trustworthiness and adaptability of energy estimation models, applied explainable AI to analyse the feature importance and partial dependence of the housing features to offer guidance on effective retrofit prioritisations and achieve the net zero targets.

Contents

List of Figures

List of Tables

List of Abbreviations

ANN Artificial neural networks.

AP Average precision.

AutoML Automated machine learning.

CNN Convolutional Neural Network.

CSH Code for Sustainable Homes.

cv Coefficient of variance.

DNN Deep neural network.

EPC Energy Performance Certificates.

GHG Greenhouse gas.

GIS Geographical Information System.

GSV Google Street-view.

LIME Local interpretable model-agnostic explanations.

LoD Level-of-Detail.

LSTM Long Short-Term Memory.

MAE Mean absolute error.

MAPE Mean absolute percentage error.

MARVel Multi-Spectral Advanced Research Vehicle.

MLP Multi-layer perceptrons.

NAS Neural Architecture Search.

NHM National Housing Model.

NPI Normalised perimeter index.

OLS Ordinary least squares.

ONS The Office for National Statistics.

OS Ordnance Survey.

PD Partial dependence.

PFI Permutation feature importance.

RdSAP Reduced Data Standard Assessment Procedure.

ReLU Rectified Linear Unit.

ResNet Residual Network.

SAP Standard Assessment Procedure.

SDG Sustainable Development Goals.

SHAP SHapley Additive exPlanations.

std Standard deviation.

SVM Support vector machine.

UPRN Unique Property Reference Number.

XAI Explainable artificial intelligence.

Chapter 1

Introduction

1.1 Climate goals and current trends of energy consumption

The UK government has committed to an ambitious target to achieve net-zero emissions by 2050 to tackle climate change. The concept of 'net-zero' means any Greenhouse gas (GHG) emissions produced would be balanced by an equivalent amount removed from the environment. In order to achieve the net-zero target, it is equally important, to promote the use of renewable energy technologies: the UK government has been working on replacing heavy-emitting fuels, e.g., coal, with relatively cleaner and renewable energy, e.g., natural gas, and at the same time to reduce the amount of emissions: according to the Department for the Department of Business, Energy & Industrial Strategy (BEIS), a reduction of at least 68% of GHG emissions at 1990 levels is required by 2030 [2].

The amount of emissions is closely associated with the daily activities among the sectors. When estimating the total amount of GHG emissions, the vast majority are based on energy use data, supply and consumption. Among all the sectors, the residential sector has become one of the largest energy consumers around the world. The recent years have witnessed the growing pressure residents feel in paying energy bills, caused in part by the worldwide COVID-19 pandemic and further magnified by the rapid increase in energy prices as a result of the Russian-Ukrainian war [3, 4].

In the UK, the residential sector was the only sector that saw a continuous increase in energy consumption since 2018, while the other sectors: transport, industry and services, all experienced a decrease, Figure 1.1. According to the official statistics, the average energy use for all existing properties in England in 2022 is approximately 41,115 kWh per year, with a 5% increase per household compared with the 2021 level [5].

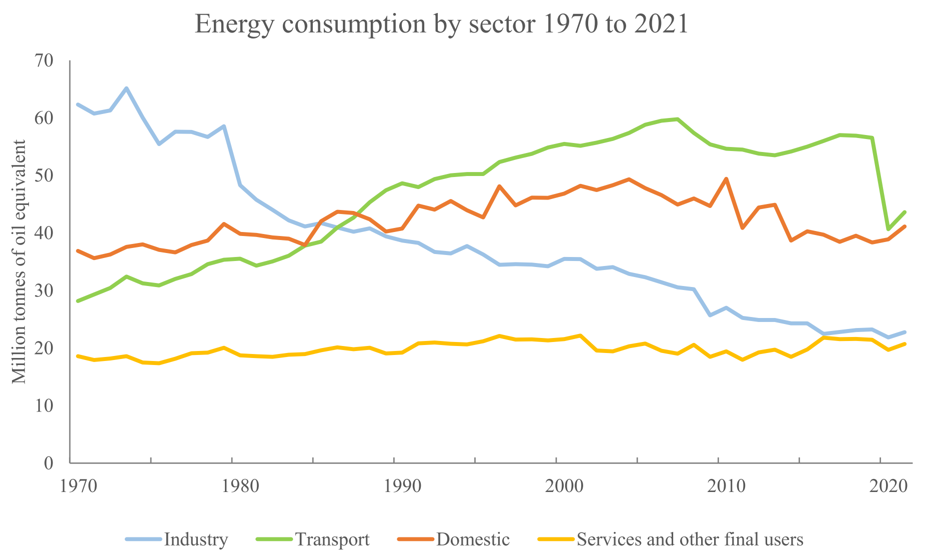


Figure 1.1: Energy consumption distribution in the UK from 1970 to 2021. The domestic sector has become the second largest energy consumer since around 1990 and the only sector witnessed a continuous increase since 2018. Data used in this graph is collected by DESNZ and BEIS [5].

This upward trend in energy consumption by all sectors highlights the challenges the UK government is currently facing to achieve its emission reduction goals by 2050. Addressing the energy consumption of residential buildings is crucial, which leads to an ongoing debate about how to decarbonise the built environment. In 2013, the Treasury conducted a review of carbon reduction options and suggested four main strategies with respect to the carbon reduction curve [6]: 'build nothing', 'build less', 'build clever' and 'build efficiently'. While 'build nothing' may address the root cause of the climate crisis, it is not a practical solution considering the growing world population. Therefore, it is important to build efficiently and intelligently.

1.2 Route to decarbonising the built environment

From the EcoHomes, the Code for Sustainable Homes (CSH), to the upcoming Future Homes Standard, the UK government has been attempting to shape the residential sector to build efficiently. The Code for Sustainable Homes introduced a rating system from Level 1 to Level 6 to incorporate sustainable features into new homes. While it was withdrawn in 2015 due to government policy shifts towards streamlining regulations, it was considered as one of the first frameworks that provide a systematic assessment of residential buildings beyond their physical characteristics [7, 8]. The principles of the CSH provided valuable insights to inform the new regulations. Although still in the early stage of preparation, homes built under the Future Homes Standard will be mandated to achieve at least 30% less energy consumption and GHG emissions compared to homes built following the current standards to avoid the need for costly retrofits in the future [9].

However, the effects of regulating new developments alone may be subtle compared with the vast amount of emissions produced by existing buildings. In recent years, residents have faced growing pressure to manage energy bills, a challenge exacerbated in part by the worldwide COVID-19 pandemic when more time was spent at home and the rapid increase in energy prices. The average domestic energy bills in 2022 are estimated to be £2,316, representing a 74% increase comparing to the 2021 level [5]. Statistics indicate that around 64.5% of this domestic energy consumption is attributed to thermal costs, such as heating and water boiling [5] which are essential to residents, especially considering the projected increased frequency of extreme climate conditions due to climate change.

To support net-zero policies and tackle fuel poverty, in addition to the direct support for energy bill payment, such as the *Energy Bills Support Scheme* and the *Energy Price Guarantee*, renovating existing homes is one of the critical drivers in decarbonising the UK. Home renovation directly addresses key issues outlined in the framework of the Sustainable Development Goals (SDG): SDG 3 Good Health and Well-being, SDG 7 Affordable and Clean Energy, SDG 10 Reduced Inequalities, SDG 12 Responsible Consumption and Production, and SDG 13 Climate Action [10]. The government has committed to raising the minimum energy rating for existing homes in the market from the current E to C. This means that, according to the most recent English Housing Survey for 2020 to 2021, over a half of the assessed homes in England and Wales will need to be retrofitted to meet the new standard [11]. Figure 1.2 mapped the distribution of the percentage of homes currently with an EPC rating D or below, which need to be renovated to a higher standard once the minimum rating been raised. Generally, local authorities located in the Northern UK tend to have more properties with EPC ratings below C.

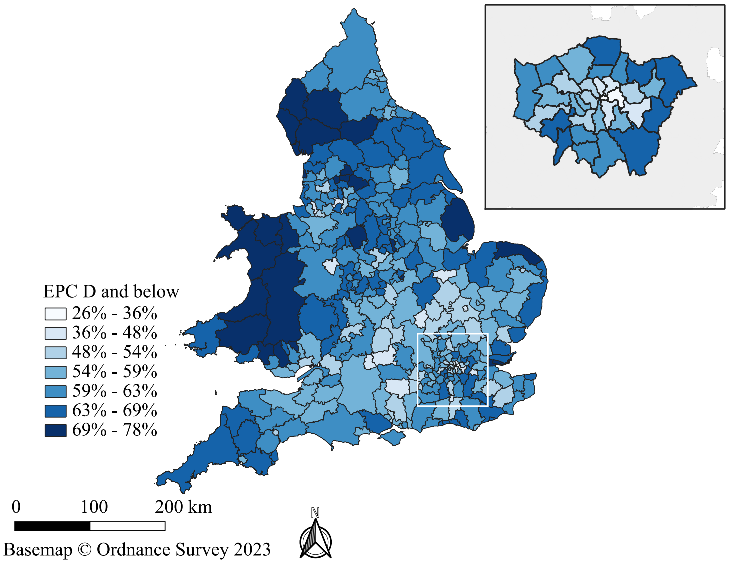


Figure 1.2: Spatial distribution of proportion of homes with a EPC rating D or below, zoom-in map on the right for London boroughs. EPC data from [12] and aggregated to the local authority level. Darker colours suggests more homes are below band C and require to be retrofitted once the new housing standard regulation comes into effect.

A series of regulations and incentives, such as the *Social Housing Decarbonisation Fund*, the *Home Upgrade Grant*, the *Boiler Upgrade Scheme* and the *Public Sector Decarbonisation Scheme* have been introduced to improve home energy efficiencies, especially for low-income residents. These incentives primarily support residents exposed to fuel poverty or in the least energy rating equal to or below D by installing energy efficiency measures including insulation improvement and low-carbon heating technologies. A survey for all the Home Upgrade Grant applications is conducted monthly. In the recent survey, to the end of March 2023, the number of measures installed is indicated in Figure 1.3a. The bar chart shows an aggregated distribution of all the measures installed for three phases of the home upgrading scheme. Measures being coloured the same suggests they belong to the same category of retrofit measures. While Solar PV installation is the most commonly implemented measure, accounting for approximately 37% of all upgraded properties, it is important to note that the primary effect of such installation is to generate renewable energy and save on electricity bills, rather than improving home energy performance, such as the insulation improvements for loft and wall conditions.

The benefits of these installations were further evaluated, and the estimated annual energy savings were calculated using the National Housing Model (NHM), a common tool used by the UK government to project the policy and legislative impacts on environmental and energy usage. Although not all measures were analysed for their potential savings, the estimated annual energy savings were compared to the total installation costs, as shown in Figure 1.3b. According to the NHM projection results, solar PV installation and insulation improvements for the loft, solid walls, and cavity walls are the most cost-effective energy-saving measures encouraged by the home upgrading scheme.

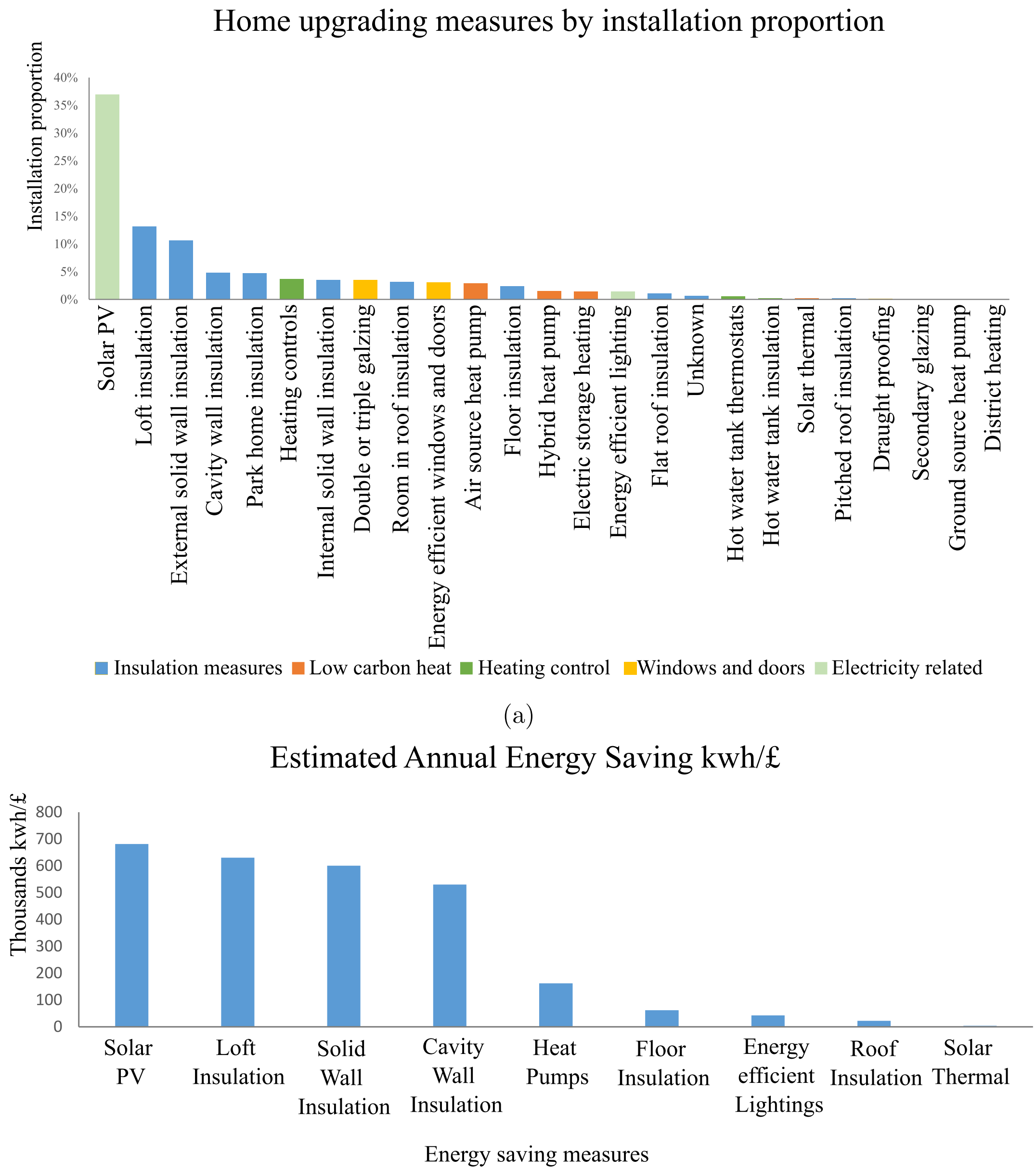
However, despite all the benefits residents and the society may gain from retrofitting homes, statistics indicate that a third of the total £6.6 billion budget allocated for these energy-saving incentives remains unspent within the target time frame of 2020 to 2025 [13]. The *English Housing Survey* sheds some light on the reasons for this underspending. According to the survey conducted between September 22 and October 3, 2021, among the properties and residents surveyed regarding the *Opinions and Lifestyle Survey*, less than a fifth of people in Great Britain (19%) were considering improving their home's energy efficiency [14]. Among those who were not considering any improvements, the most common reason was the belief that their home was already efficient enough (35%), followed by not owning their own home (29%) and the perceived high cost of making changes (28%) [11].

While it is important to raise the awareness and willingness of residents to retrofit their homes and choose more sustainable ways of energy usage, the UK government recognised there is still work remaining to improve their evidence base to help identify the most effective measures for different buildings and regions [9]. Existing studies have conducted assessments for different buildings and regions using urban retrofit modelling. Typically, the urban retrofit modelling involves two stages of estimation, the first stage is to understand the current energy performance, and then integrate with the economic costs of each measure to determine the most effective retrofit measures. However, these estimations rely heavily on the quantity and quality of the input data, which vary from region to region, posing challenges for accurate energy consumption estimation and retrofit scenario determination. Surveys were also conducted to examine the implementation results of the retrofit schemes, and revealed that although the residents confirmed their living standards have been improved, it has not achieved its designed level [15]. These emphasised the need for a better understanding of energy usage that reflects actual energy performance respective to homes in each region.

1.3 Research Aim, Questions and Objectives

The continuously increasing energy consumption in the residential sector and the disappointing retrofit results, highlights the necessity for a more robust and comprehensive understanding of the energy efficiency of existing housing stock to facilitate effective home retrofit and energy reduction strategies. Numerous studies have been conducted to explore energy estimation approaches, including data-driven, physics-based, and hybrid methods, utilising data at various spatial scales. At a large-scale assessment of residential properties, data-driven approaches employing machine learning models are commonly utilised. Data-driven approaches rely heavily on the quantity and quality of input data. It is important to acknowledge that the data itself is not neutral and can contain embedded biases. Especially for data collected in situ, it can be challenging to determine whether biased data or extreme cases represent outliers due to mistakes or reflect real circumstances.

Therefore, the primary aim of this study is **to investigate a potential framework to enhance the trustworthiness and adaptability of data-driven energy estimation methods**.



(a)

(b)

Figure 1.3: Statistics of the Home Upgrading Scheme deployed. Statistics for the three phases untill the end of May 2023 were included in the chart production. (a) A total number of measures deployed in three phases of home upgrading schemes. Solar PV, insulation improvements are the primary measures used. (b) Estimated annual savings by the measures deployed in the three phases of home upgrading schemes projected by the National Housing Model by BEIS. The energy savings were further examined against the costs used in installations.

This aim will be achieved by addressing the following research questions:

1. **Can incorporating multi-modalities improve the trustworthiness of operational energy estimation?** and

2. **Can transfer learning techniques improve the adaptability of operational energy estimation?**

In this thesis, the word **trustworthiness** does not necessarily implies that the designed framework should be compared with smart meter data as a measure of error rate. As the real time data is not publicly available, the trustworthiness means that the machine learning models developed are able to offer a robust estimation based on the predicted housing conditions and energy consumption data, in this case, the EPC recorded values. And this cannot be achieved without a **transparent** understanding of how the estimation is made, which is achieved by implementing explainable AI tools as a way of interpreting the model behaviour. And the word **adaptability** means the estimation is able to acknowledge the changing context, both the spatial scales and housing characteristics, to perform adequate estimation that is able to reflect the housing conditions in the target cities.

This research aims to answer these two questions through exploring the applications of different machine learning techniques, including traditional machine learning, automated approach, multimodal learning and transfer learning in estimating the operational energy consumption of residential properties in four case study areas, using visual and tabular database.

Specifically, the research aim will be achieved by the following objectives:

- Identify the key features for energy estimation and understand their contributions to the estimation process.

- Evaluate the effectiveness of introducing multiple modalities in the prediction model to mitigate potential biases in the data.

- Evaluate the effectiveness of introducing transfer learning to accommodate changing spatial contexts, even with limited data availability.

1.4 Thesis Outline and Structure

The thesis is structured as follows to address the research questions and objectives. The studies conducted in Chapter 4 to Chapter 6 are structured in a hierarchical progression, with each chapter leveraging and advancing the knowledge established in the previous chapter.

Chapter 2 provides a comprehensive literature review of existing studies on residential energy consumption. The review summarises the methods and data used in previous research, identifies their limitations, and highlights the research needs for improving the trustworthiness and adaptability of operational energy estimation procedures.

Chapter 3 outlines the general methodology and data applied to all the studies in this thesis.

The primary data used in this thesis is explained, including how the data is collected and pre-processed, with the statistics of these data used in each study.

Chapter 4 presents a case study conducted on 142,756 residential properties in Sheffield. This serves as a baseline study where common approaches used in existing studies are evaluated and compared with an automatic model development approach. The limitations related to data trust-ability and model adaptability are further examined through the application of the developed model in Sheffield to different cities, Barnsley, and Merthyr Tydfil.

This leads to **Chapter 5** where additional modalities are introduced for a case study for 9,015 residential properties in Barnsley. This case study addresses the first research question, improving the trustworthiness by minimising the bias in energy estimation caused by unreliable mono-modality input.

Chapter 6 addresses the adaptability issue of traditional data-driven methods by proposing a methodology that leverages knowledge from regions with reliable data, Barnsley, to assist in energy estimation for regions with limited data, Doncaster and Merthyr Tydfil, allowing the model to adapt to new spatial contexts.

Chapter 7 presents a discussion on the key findings, policy implications, limitations and recommendations for future studies.

Chapter 8 concludes the thesis by summarizing the main contributions of this work and providing recommendations for future research.

The first objectives are investigated in all the three studies presented in **Chapter 4**, **Chapter 5** and **Chapter 6**. These studies examined the contribution of the key housing features to energy consumption and asses how different sources of inputs can affect the importance rankings. The second objective is achieved by the second study presented in **Chapter 5**. And the third objective is accomplished in the study presented in **Chapter 6**.

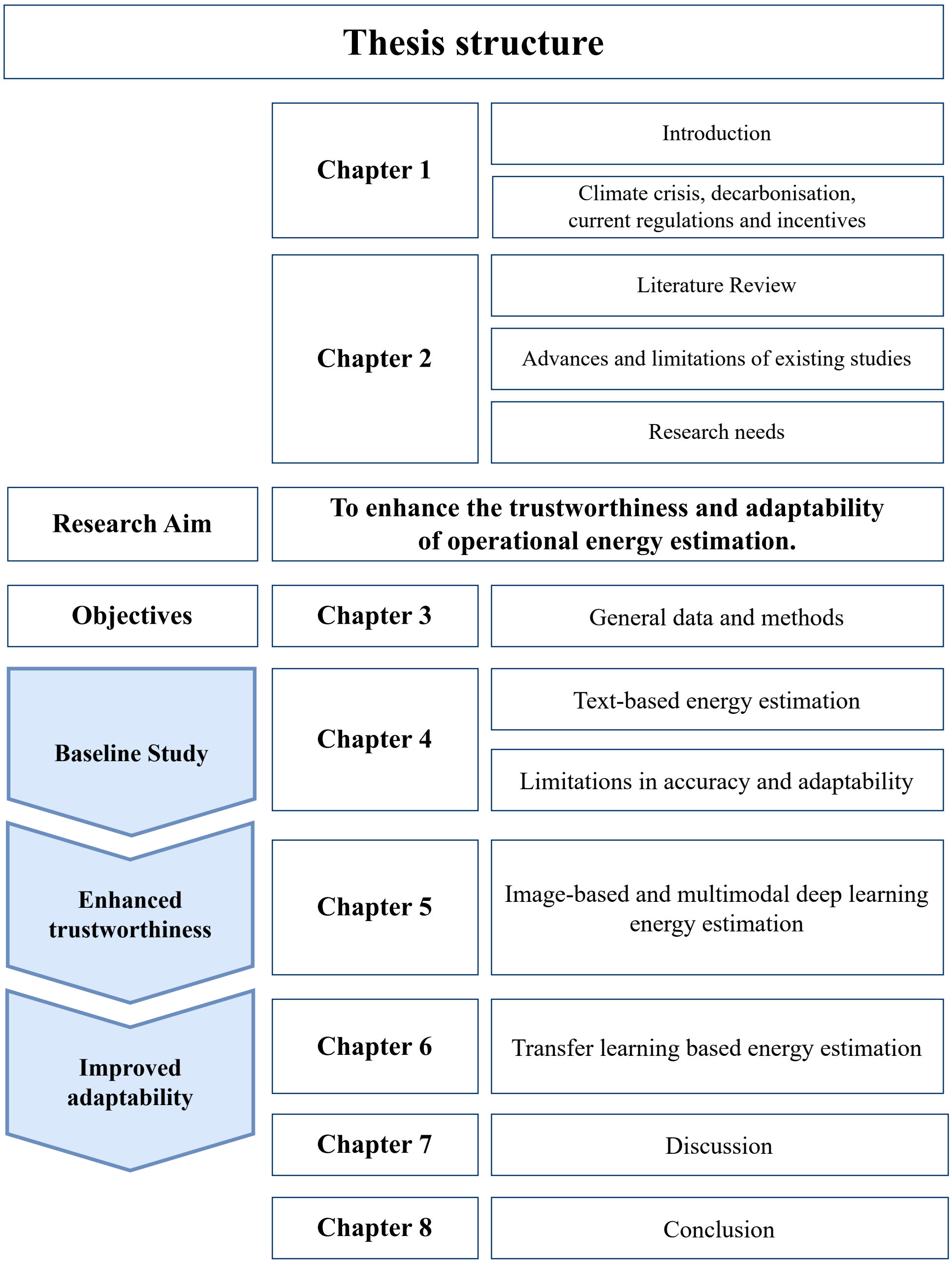


Figure 1.4: The structure of the thesis, and the corresponding objectives each chapter addresses.

Chapter 2

Literature Review

2.1 Chapter Introduction

When estimating residential energy performance, the existing approaches can be divided into three subgroups based on the main methodologies used: data-driven (black-box); physics-based (white-box); and hybrid approaches (grey-box) that combine the previous two approaches [16, 17, 18, 19]. Each approach has different requirements regarding the types and levels of details of the input data.

Physics-based approaches typically rely on detailed information on buildings' thermal characteristics, such as the thermal transmittance of the material. Such information collection usually requires access to the properties' internal space for detailed examinations. This data is then input into a physical model developed based on theories of heat transfer. One of the popular building simulation tools is EnergyPlus. When using EnergyPlus for energy consumption simulations, the process usually starts with constructing a 3D model of the property, and then inputting a range of detailed parameters into submodules designed in EnergyPlus for simulation, including the construction material, orientation, and HVAC systems. The simulation accuracy is highly reliant on the level of detail of the 3D building models and parameter inputs [20]. Benefiting from using only the physical characteristics of the property, this approach can be applied without knowledge of historical consumption data. Thereby the simulation can be conducted during the design phase before the construction is completed. However, this process involves large uncertainties in data processing, parameter assumptions and model settings and can be time-consuming. These limitations have hindered physics-based approaches from applying to large-scale studies.

When the detailed information and internal access are limited, or for large-scale studies, data-driven approaches are commonly adopted to develop statistical or machine learning models, based on historical energy consumption and building morphology. The primary method of the data-driven approach is to develop machine learning models for estimation.

Machine learning is a branch of artificial intelligence that involves training algorithms to learn patterns from data and make predictions. When predicting the operational energy consumption for residential properties, machine learning algorithms usually analyse historical energy usage data, along with various other relevant factors such as building characteristics.

The strong link between energy consumption and machine learning is also demonstrated from a bibliometric study, as shown in Figure 2.1a, suggesting machine learning is a popular method widely used in this discipline: large number of keywords appeared are machine learning-related, with relatively bigger dots referring to higher number of occurrence in the existing literature. The bibliometric study analysed 39,009 records returned for citation search using 'Residential building energy consumption' or 'Residential building energy analysis' for the last five years. Each node in the network represents one keyword used in existing literature, and the edges denote the co-citation relationship among the nodes. The nodes are coloured based on the closeness of citation relationship, and sized based on the usage frequencies. Figure 2.1b visualises a filtered bibliometric ego-network centred on the keyword 'Energy Consumption', further highlighting keywords directly linked with energy consumption. Nodes representing data-driven approaches, e.g. machine learning, are shown with a significant proportion of direct citation links.

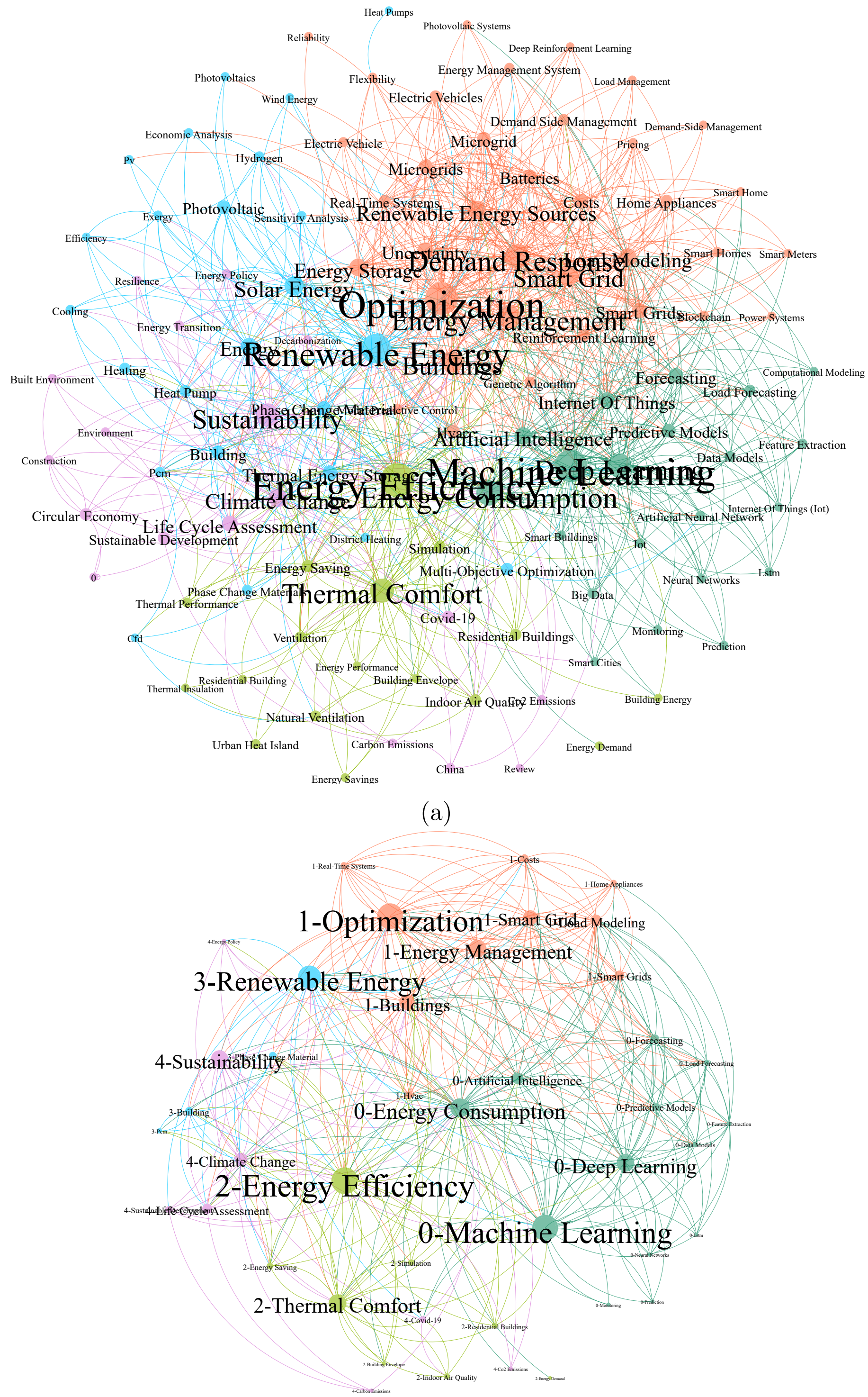


Figure 2.1: Bibliometric study of 39,009 citation records searched based on 'Residential building energy consumption' or 'Residential building energy analysis'. Each colour represents one cluster of keywords that were calculated as densely connected. (a) Co-citation map of keywords used in existing literature. (b) A filtered ego-network of literature keywords directly linked with **energy consumption.**

From the existing literature, it has been found that, in general [21, 22]:

- Buildings constructed in a similar space and time tend to have similar building characteristics; and
- Buildings with similar characteristics tend to have similar energy needs.

These two rules verified the capability of such a data-driven method in estimating residential building energy consumption, and introduced the features that have close associations with building energy performance which have been intensively used in data-driven energy analysis.

Hybrid approaches aim to minimise the limitations data-driven and physics-based approaches have. Studies suggest that using a grey-box approach significantly increased the model forecasting capability [17]. There are mainly two directions of integration. The first is to replace part of the white-box algorithm with machine learning techniques. Xu et al. (2021, as cited in [23]) designed an Artificial Neural Network that accepts the thermal model processed by EnergyPlus as input to perform housing energy prediction. The second approach is to use the data-driven approach as a data preparation tool for physics-based models. For instance, Chen et al. [24] proposed a methodology to automatically generate data to simulate the energy usage of retail and office buildings in Boston. Similar to this thesis, the case study was conducted with no access to buildings' internal zoning and HVAC systems details. They first created 3D models for properties at a city scale by extruding the arbitrary property footprint with their recorded number of floors using the Geographical Information System (GIS). Then an algorithm is applied to develop arbitrary thermal zones that meet engineering standards. The HVAC systems are also automatically determined for these properties based on the prototype building provided by the Commercial Building Energy Saver Toolkit. For instance, the HVAC system for large office buildings is designed as '...a central plant with chillers and boilers provides chilled and hot water for the central VAV with reheat systems' [24].

Considering this work focuses on properties at a city scale, and no access to individual properties was granted, a data-driven approach was selected. The remaining literature review will provide a general overview of existing data-driven works, focusing on what features have been considered important, what data were utilised to provide representatives to such features, and the methodologies employed for energy performance prediction from these features. Followed by a brief discussion on the limitations and potential solutions for the existing literature.

2.2 Housing Features and Energy Performance

The two general rules listed in the introduction of this Chapter provided insights into the main factors that have close associations with building energy performance. The first rule follows the first law of geography: 'Near things are more related than distant things'[25]. In the study of residential energy analysis, the first rule indicates that the geographical location and the year of construction are important factors to infer housing characteristics. One of the potential reasons why location and age mean certain housing features is that, housing legislation changes regularly to comply with the housing needs and environmental concerns at that time and also what might be needed in the future in that area. For instance, the Town and Country Planning Act issued in 1947 [26] prioritised developing single apartment blocks. The construction sector then develops homes accordingly [26], hence the second rule, which suggests that energy needs are predictable from these characteristics.

Therefore, accurate descriptions of housing conditions are important for understanding and estimating the operational energy.

2.2.1 Representations of building features and relative importance in estimation

A variety of data describing the building features are utilised in the existing energy studies. Table 2.1 provides a summary of building features from selected existing studies. The number of properties examined in the study, the selected algorithms and the type of output of the prediction are included in the table. The tick mark in each column represents that such data is used in the study, and a star mark is used to indicate the data is considered as the key feature in energy estimation if analysed.

Generally, the housing features used in existing studies can be categorised into three groups: 1) the building morphology, such as the total floor area and the number of floors, that mainly describe the physical form, structure and design, 2) the thermal features, such as the conditions of walls, and roof, that mainly describe the elements directly relating to the thermal performance, and 3) others, such as the climate data. From the list, we are able to see that there is an overlap between the features used in the estimation, but no agreed conclusion has been made on the most important features. This inconsistency in the finding of the dominant housing features might be resulted from the fact that these studies are conducted at different locations and different spatial scales. It is also possible that, when the data is aggregated to a certain scale, i.e. at the national level, some of the characteristics in the subregional levels may be omitted and hidden. This leads to one of the main concerns in spatial analysis, which will be discussed later in Section 2.5. These feature classifications were used as a reference when deciding what data to include when designing the studies in this thesis.

Apart from the different combinations of features used, different algorithms were also employed. This is largely dependent on the data used, the type of outputs (categorial, e.g. rating, or numeric, e.g. consumption) and the input size. Some of these were conducted as comparative studies, where multiple algorithms were tested to find the best performance [22, 27, 28]. This selection of algorithms used in existing studies is also used as reference when designing the methodology in this thesis.

Table 2.1: A selective summary of housing features used in existing recent studies, including the study location, size of input, algorithm used or found best, and output. The existence of a check mark means the variable is used in the study. The variable is marked with an asterisk star if it is considered as the key housing feature in the corresponding energy analysis. The table is ordered according to the year of publication.

Author	[22]	[29]	[27]	[28]	[14]	[30]	[31]
Study location	New York, USA	Austria	Ireland, UK	Germany	UK	UK	Glasgow, UK
Input size	20,000	3,865	850,000	25,000	All EPCs	5,000	165,318
Algorithm	OLS	CNN	Neural Network	SVM	Logistic regression	DNN	CNN+MLP
Output	Energy intensity	EPC rating	EPC rating	Consumption	EPC rating	Consumption	EPC rating
Total floor area	✓		✓	✓*	✓	✓*	✓
Number of floors	✓				✓	✓	✓
Walls condition			✓	✓	✓	✓	✓
Windows condition			✓	✓	✓	✓	✓
Roof condition			✓	✓	✓	✓*	✓*
Floor condition	✓		✓		✓	✓	✓
Lighting condition			✓		✓	✓	✓
Fuel type			✓	✓	✓		✓
Main heat			✓	✓			✓*
Heating control							✓
Room Count					✓	✓*	✓
Year of construction	✓	✓	✓*	✓	✓*		✓
Building type			✓*	✓	✓	✓	✓
Location					✓		✓
Climate data						✓	
Images		✓					✓*

Age or year of construction

The Office for National Statistics (ONS) examined the housing stock in England and Wales using EPC records and reached a conclusion emphasising the important role of the 'year of construction' acts in energy estimation, suggesting that 'Age of the property is the biggest single factor in the energy efficiency of homes' [14]. They utilised logistic regression to measure the odds ratio as the indicator of whether properties with target housing features are more likely to be energy efficient. However, ONS did not provide a detailed methodology of how the statistical model is developed, especially what parameters were used and how well the model fits the data. Similar conclusions were made by Ali et al. [27], where age and building type were found as the key features for energy estimation, by using a neural network for properties in Ireland.

Other building features

There are also studies that have suggested opposite findings on what are the dominant features for energy estimation. Olu-Ajayi et al. [30] used random forest trained on 5,000 residential properties and ranked the feature importance according to the trained trees. Their results suggest the size of the total floor area is the dominant feature in energy estimation, followed by roof conditions, number of rooms with heating facilities and monthly average wind speed. Sun et al. [31] developed a multimodal network using both EPCs and images for energy estimation. They implemented the *KernelExplainer* from the SHapley Additive exPlanations (SHAP) to evaluate the rank of importance of their data. Their feature importance rank suggests that properties with 'Roof: Pitched, no insulation' and 'Hotwater: Electric immersion, standard tariff' have the highest SHAP value and are thereby associated with lower energy efficiency.

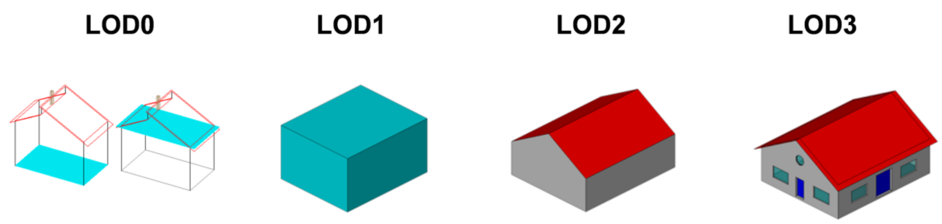
The above results hence lead to the second rule, the relationship between building characteristics and energy needs, emphasising how housing features can be used to estimate energy. Optimal descriptions or representations of such features are crucial in building energy estimation. Existing literature has experimented with a wide range of different data inputs providing such information, including data either in 3D, e.g. CityGML model and LiDAR point cloud [32, 33], or image-based, e.g. streetview images and real estate evaluation report[29] or text-based, e.g. EPC [19, 34].

2.2.2 Sources of data

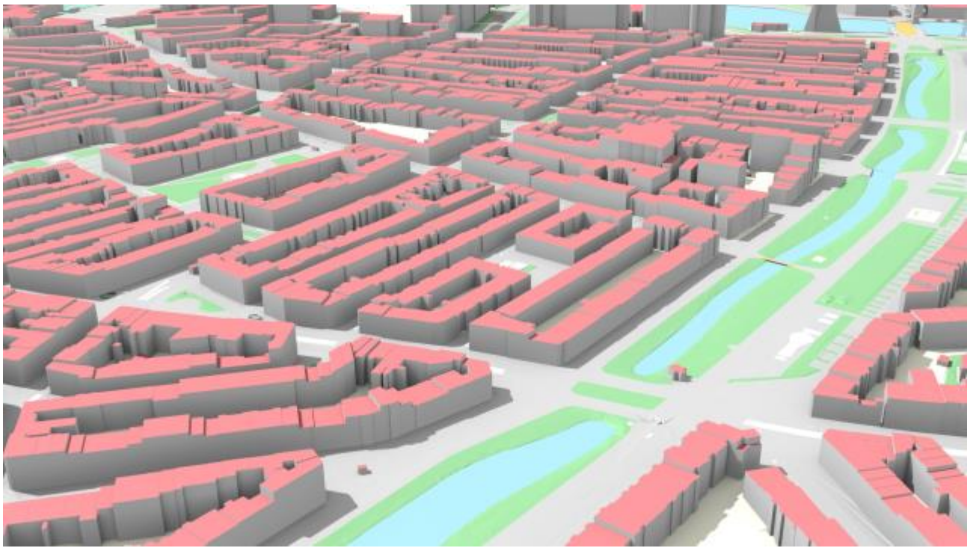
Despite the importance of building features in inferring building energy needs, easily accessible complete databases are often unavailable, especially for databases that contain information about the year of construction [21, 32]. Existing studies have attempted to infer building age from properties' physical features [22, 32, 35]. Rosser et al. [21] proposed a methodology to predict the year of construction using map data and historical satellite images. Their machine learning model used random forests and achieved 77% prediction accuracy [21]. Rosser et al. [21] further evaluated the contribution of the housing features towards estimating the age, and suggested that the average pitch of the roof and the measure of footprint complexity are the strongest predictors. However, their model was trained based on a relatively small number of properties (1,096) in Nottingham to predict five aggregated age bands covering a rather wide time span. The test samples they used were also derived from a single neighbourhood, which tends to have similar building features and construction age.

3D dataset as building representations

Biljecki and Sindram [32] attempted to estimate the building age from a 3D Level-of-Detail1 city model (LOD1) in Rotterdam accompanying 9 other attributes describing the building appearance, (e.g. ceiling and building height, and building use). 3D GIS building models are popular datasets for studies focusing on European cities. Depending on the level of completeness, as shown in Figure 2.2a, the 3D city model ranges from LoD1 to LoD3, from simple extrusions of 2D footprint polygons to detailed building models with accurate visualisation showing specific roof types, locations of windows and walls, etc. However, due to the lack of completeness, most of the existing European studies use the LoD1 models [32, 36], as visualised in Figure 2.2b. This dataset is usually adopted in physics-based or hybrid energy simulations.



(a) Example cityGML LoD models, from LoD0, a footprint polygon, to LoD3, a 3D mesh model. Figures adopted from [37].



(b) Example visualisation of the CityGML LoD1 model at a city scale, widely used in studies focused on European cities. This is an example of properties in Rotterdam, adopted from [32].

Figure 2.2: Demonstrations of 3D GIS modes widely used in statistical studies for European cities.

In the best scenario, when all attributes are available, Biljecki and Sindram [32] are able to predict the year of construction with a root mean square error of 11 years. However under more realistic situations, when data availability is limited, the prediction error increases to 26 years. The model developed by Biljecki and Sindram [32] suggests the average property age in the nearby neighbourhood is the key feature in accurate age estimation. When such information is not available, the building height and footprint shape complexity play the dominant role in the prediction.

Nouvel et al. [36] implemented a LoD1 model with a multiple linear regression model to predict the heat demand and energy saving potentials for around 1,000 buildings in Bospolder, Netherlands. Their model achieved accuracies ranking from 5% to 25%, higher error was found at disaggregated level. Nouvel et al. [36] suggests that the variations in accuracy are mainly caused by the limited data availability for individual properties, especially about occupants' behaviours and unoccupied buildings.

Image-based energy analysis

Image-based energy analysis which integrates computer vision with data-driven energy consumption prediction is emerging. Researchers have recognised images are good visual representations showing rich information about the building and its surroundings [29, 38]. Despotovic et al. [29] developed a workflow to extract representative patches for building elements (e.g., doors, roofs, windows) to classify housing energy ratings. They developed a CNN-based network to 127,945 image patches extracted using SIFT from 2,065 Austrian detached houses, and achieved a classification accuracy of 62% in energy rating prediction. This case study demonstrated the potential of applying building photographs in housing energy prediction. However, these images have restrictions that only the properties' external physical characteristics can be seen by the prediction model, and may lack representative data for actual insulation conditions.

While the existing studies using real estate images validated the use of photographs in building energy prediction, the real estate datasets are often not publicly available. Google Street-view (GSV) can be a potential alternative source of image data for building energy prediction. Similar to real estate reports, GSV has been widely used in computer vision studies. Example studies have been conducted with GSVs to derive neighbourhood socio-demographic patterns and traffic auditing [39], but not much in energy prediction. When the GSV van drives around the city, the captured scenes are publicly available and can be downloaded with API. Yuan and Cheriyadat [40] conducted building height and facade estimation using OSM building footprint and GSV. They proposed a methodology to estimate camera position and project street scenes to 2D maps. This study showcases the potential of GSV images as descriptions for building characteristics, such as building height and facade material, which is critical in building energy estimation.

Although the street view image database are largely enriched by the recent advances in technology, studies using street view images are prone to general challenges. Common obstacles include heterogeneous image quality, presence of irrelevant objects, and variations in spatial coverage and update frequency [41]. Taking the images by driving through neighbourhood also suggests that only the characteristics of the front facades of properties are considered in the model training, which largely neglected the features of the rear. Using this source of data only may lead to a biased representation of the properties' energy performance.

Energy Performance Certificates

One widely used tabular dataset is the Energy Performance Certificates (EPC). The EPC is an official document that assesses buildings' energy performance, mandatory for every property in the UK, similar to the *Energy Star score* in the USA and *Diagnostic de Performance Energétique* in France [42]. The assessment can be considered as a simplified physics-based approach presented in the form of a worksheet.

For new buildings or buildings undergoing major renovations, the EPC is created using the Standard Assessment Procedure (SAP), while existing buildings use the Reduced Data Standard Assessment Procedure (RdSAP), which requires less data input. The primary difference between SAP and RdSAP is that RdSAP, when building information is missing, utilizes a series of default values that align with building regulations, such as BS EN ISO 13790, the Standard for Energy Performance of Buildings, instead of in-situ data collection used in SAP. However, there is no clear indication in the EPC records which version of SAP was used in production.

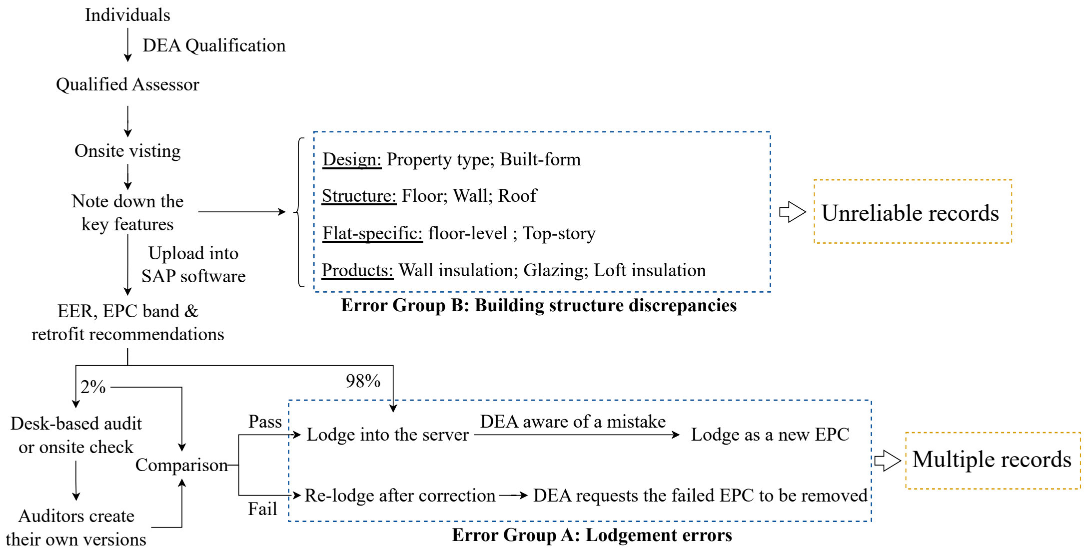


Figure 2.3: Demonstration of the standard workflow of EPC making. The possible errors during the procedure are marked in groups with their potential outcomes. This chart is produced based on the work by Hardy and Glew [43].

The overall procedure of producing an EPC is illustrated in Figure 2.3. To become an energy assessor, a person needs to complete a short-term training to be qualified (for residential assessors, the qualification is named as 'Domestic Energy Assessor'). The assessors are required to visit the property, and record building characteristics based on their inspection on-site with both texts and images. Specifically, the SAP calculation considers factors that contribute to energy efficiency [44]:

- Materials used for construction of the dwelling
- Thermal insulation of the building fabric
- Air leakage ventilation characteristics of the dwelling, and ventilation equipment
- Efficiency and control of the heating system(s)

- Solar gains through openings of the dwelling
- The fuel used to provide space and water heating, ventilation and lighting
- Energy for space cooling, if applicable
- Renewable energy technologies

All of these should be assessed and recorded by the assessors in situ and then uploaded to a designated software to calculate the Energy Efficiency Rating (EER). For new builds, these values are calculated during the design phase and construction, for existing buildings, on the other hand, standard values set in the respective building construction standards are used as default in the calculation (e.g. BS EN ISO 6946 is used to calculate the U-values for walls and roofs). In the current SAP 10 since June 2022, the score or energy cost factor (ECF) is calculated by the following equations [44]:

$$\begin{aligned} \text{ECF} &= \text{deflator} \times \text{total cost} \div (\text{TFA} + 45) \\ &= \text{deflator} \times (\text{Fuel costs for heating, cooling and lighting}) \div (\text{TFA} + 45) \end{aligned} \tag{2.1}$$

$$\begin{aligned} &\text{if ECF} \geq 3.5, \quad \text{SAP 10} = 108.8 - 120.5 \times log_{10}(\text{ECF}) \\ &\text{if ECF} < 3.5, \quad \text{SAP 10} = 100 - 16.21 \times \text{ECF} \end{aligned} \tag{2.2}$$

In the above equations, TFA is total floor area, 45 was added to the TFA as a standardised adjustment made to account for the average size of additional spaces in the property, such as communal areas, that may not be directly heated [44]. The deflator is a price indicator for electricity if uses an off-peak tariff. Once a SAP score is calculated to a scale of 0 to 100 [44], it is rounded to the nearest integer and converted to the EPC rating, ranked from G, the least efficient, to A, the most efficient.

Table 2.2: The conversion table between SAP scores and EPC ratings.

EPC rating	G	F	E	D	C	B	A
SAP score	1 -20	21 -38	39-54	55-68	69-80	81-91	92 or more

Following the EU Commission's guidance [43], 2% of the produced EPC documents are selected to conduct quality assurance procedures, from simple desk-top audits of the input data to detailed checks including on-site visits by accredited third parties. Once checked, the certificates are ready to be issued with the remaining 98% of non-audited certificates. Accompanying with certificates, SAP also offers retrofit recommendations. The assessors are responsible for comparing their records with a table of possible housing conditions, and selecting the corresponding improvement measures as retrofit recommendations.

With technological advancements, a wide range of mobile applications were developed to assist the assessment process. To gain a deeper understanding of the making of EPCs, the access to one of the government-approved software was obtained to trial the making procedure [45]. The portal allows the assessors to either choose from pre-defined options (e.g. for wall materials) or enter values manually (e.g. for wall thickness if measured), and offers automatic SAP calculation while information is being recorded. While these pre-defined options may offer an efficient and standardised way of EPC making, they may lack the ability to reflect complex building structures.

These produced EPCs have been used in energy prediction studies. Ali et al. [19] developed a workflow that uses existing EPC data to predict buildings' energy ratings when such information is not available. Their best-performing machine learning model has achieved 88% accuracy in predicting building EPC ratings for properties in Ireland. However, apart from the major drawbacks of EPC being a standardised representation of the property's energy usage, there are other issues with EPCs that the above studies did not take into consideration.

Hardy and Glew [43] examined the EPCs from its production to the lodgement. As illustrated in Figure 2.3, error group A is potential mistakes that might happen during the lodgement of the certificates. While error group B is issues with assessors' disagreements on building features. Crawley et al. [46] reported that around 1.6 million properties were found to be associated with multiple valid EPCs in the system, highlighting the potential issues in the data lodgement caused by error group A.

Further to this, a 'mystery shopper' study was carried out by Jenkins et al. [47] and Department of Energy and Climate Change [48]. The study was primarily focused on the error group B. Four assessors and an external organisation were asked to inspect the same 29 houses in the UK and then compare their notes and the resulted EPCs. The resulted EPC ratings by the five assessors are represented in the box plots in Figure 2.4. Although the number of assessments is limited (because of the lack of qualified energy assessors [47]), the distribution of resulted ratings still provides insights into the significant inconsistencies in EPC calculation caused by the inspections. Notably, almost two-thirds of the assessed properties have had ratings varied across two EPC bands.

2.3 Housing Features and Retrofit Scenarios

The relative importance of housing features in energy consumption and estimation also plays a critical role in identifying optimal retrofit measures. Retrofit schemes can generally be categorized into three types based on potential energy savings and costs: no-cost changes, shallow retrofit, and deep retrofit [10, 27]. No-cost changes, such as adjusting the tariff, typically result in an energy reduction of up to 25%. Shallow retrofit, which involves one or two improvements, e.g., installing double-glazed windows, aims to improve energy use by 25% to 45%. Deep retrofit, on the other hand, involves multiple renovation measures and aims for energy reductions exceeding 45%. It should be noted that although deep retrofit can result in a larger energy reduction, it often requires significant alterations to the building structure and systems.

The retrofit grants promoted by the UK government primarily fall into the category of shallow retrofit. Alongside the upcoming new housing standard, a 'fabric first' approach is promoted to focus on improvements in insulation conditions of building fabric, e.g. walls, and windows. A study was conducted by Elsharkawy and Rutherford [15] to evaluate the implementation results

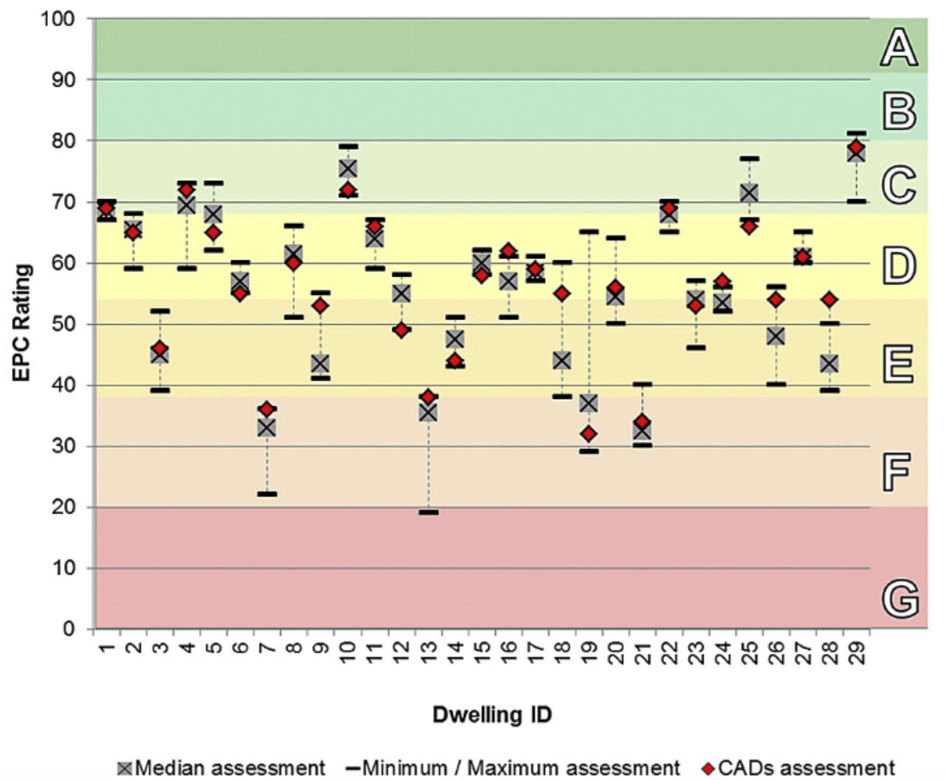


Figure 2.4: Values of EPC ratings for the 29 dwellings, created by five different assessors (Figure adopted from Department of Energy and Climate Change [48])

of the Community Energy Saving Programme (CESP) in Nottingham, where the scheme focused on improving the insulation of solid external walls, installing heating controls, and replacing boilers. The study surveyed 150 properties, before upgrading, most of the homes in the study area at that time were built from solid brick walls and pitched roofs, both uninsulated. The survey found that participants reported significant improvements in living standards, particularly in relation to cold, condensation, dampness, and mould. Around 52% of the participants reported less heating was used, while 44% reported a similar usage pattern as before. Elsharkawy and Rutherford [15] argued that the CESP scheme did not meet its projected goal of energy savings and bill reduction.

In order to achieve more effective retrofit, various studies have attempted to identify optimal combinations of retrofit measures to achieve the highest energy reduction. Typically, a data-driven approach is used, where a prediction model is first developed to estimate property energy performance based on selected housing features. From the energy performance model, researchers identify the key building features and use these to design the retrofit scenarios accordingly. The weight or value of features is then recalibrated to reflect the changes made by the designed retrofit scenarios. In this case, the trustworthiness and transparency of the energy estimation model are critical to provide optimal design of retrofit scenarios.

Gabrielli and Ruggeri [49] designed a decision support tool that tested 18 retrofit scenarios on 25 buildings in North Italy. Gabrielli and Ruggeri [49] utilised data describing the building geometry, techniques installed and the occupants' usage patterns to develop a multiple linear regression model for energy estimation before and after retrofit. The regression model was validated by comparing against the simulation results using EnergyPlus. In their example case study, the selected building was tested under different scenarios, where not surprisingly, implementing more retrofit measures leads to higher energy reduction results and more net savings (savings in bill deduct retrofit costs).

Ali et al. [27] selected housing features recorded in the EPC as home retrofit guidance. Among all the variables assessed by the EPC, Ali et al. [27] selected 16 features as retrofit recommendations, including fabric renovation and heating system upgrading and developed a machine learning model to perform energy estimation. They tested multiple popular algorithms and neural networks for the best prediction result. They tested the model with properties in Dublin and the results suggest that *year of construction*, *roof type* and *whether the second wall is semi-exposed* are the most important housing features in their model. The important features identified by the initial energy estimation model were selected as retrofit recommendations. They integrated the results with respective costs associated with each retrofit scenario gathered by the Sustainable Energy Authority of Ireland, to create a generic retrofit model to provide retrofit recommendations. Ali et al. [27] recognised the limitations of their study are mainly due to the use of EPCs.

These studies have shown the potential of utilising data-driven techniques to determine the optimal combinations of retrofit measures for the target property. They also recognised in their study the limitation caused by data scarcity and unreliability, which remains to be addressed.

2.4 Machine Learning in Residential Building Energy Analysis

2.4.1 Machine Learning

As discussed in the overview of this Chapter, machine learning is the primary method in data-driven approaches. By definition, machine learning is 'a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decision making under uncertainty' [50]. In general, machine learning can be categorised as supervised and unsupervised learning according to the existence of response variables, classification or regression tasks depending on the desired outputs, instance-based or model-based by the generalisation methods employed.

In the scope of this study, as the aim is to uncover the patterns of change in energy consumption by the variations in building characteristics, using energy consumption data from EPC as the ground truth value, therefore, this thesis will mainly use supervised regression models. Existing energy estimation studies have tested and compared a wide range of algorithms, including: K-nearest neighbours (KNN) [30], linear regression [22, 30, 51], random forest [18, 19, 21, 22, 30, 51], decision tree [30] and gradient boosting [30, 51]. The following sections will provide a brief description of these algorithms.

K-nearest neighbours

K-nearest neighbours is an example of instance-based models. As the name suggests, instance-based models rely on the inputs, and compare the unseen variables to these inputs using a similarity measure, commonly a distance metric such as the Euclidean distance $\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$. In the context of energy consumption regression, the prediction is made by taking the weighted average of the nearest training data based on the distance measured. The number of nearest points considered during the prediction process is defined by the K.

Linear regression

A linear regression model assumes that the output y can be expressed by linearly combining the input features, which can be plotted as a single straight line. The function can be expressed as $y\left(x\right) = w^T x + \epsilon = \sum_{j=1}^{D} w_j x_j + \epsilon$, where $w^T x$ denotes a scalar product between the input feature x and the weight vector w, and ϵ is the error between the predicted y and the ground truth value[50]. The objective of this function is to map the relationship while minimising the residual error ϵ.

Tree-based machine learning

Tree-based machine learning models are commonly used in existing energy estimation, which are: random forest, decision tree and gradient boosting. The decision trees recursively split the input data into binary branches, until only a single leaf node remains. Each split can be considered as a conditional step, that ensures a maximum decrease in the impurity of the child node. When making predictions for numeric outputs, the input x moves down the tree following the learned hierarchy, and uses the weighted average of trained y in the single remaining leaf as prediction results. Developed upon this, the random forest models perform the prediction task by collectively learning from multiple trees built and evaluated independently. On the other hand, although gradient boosting is also an ensemble of decision trees, it builds the trees based on the ones previously built to improve the deficiencies.

2.4.2 Deep learning

A subfield of machine learning, named deep learning, is also widely used in energy consumption analysis. At the very core of deep learning lies the Artificial neural networks (ANN). The ANN draw inspiration from the brain's architecture of 'how biological neurons might work together to perform complex computations using propositional logic' [52]. Similar to a biological brain, ANN consist of neurons with distinct roles, enabling machines to learn and assign specific weights to each neuron based on its function. A typical neural network includes an input neuron (multiple if the inputs are multimodal), an output neuron (multiple if it is for a multi-output task), and hidden layers in between.

A substantial amount of applications can be found using neural networks in energy consumption estimation, where Multi-layer perceptrons (MLP) is usually used for tabular features [31], Convolutional Neural Network (CNN) for the visual components [29, 31, 38], and Long Short-

Term Memory (LSTM) is frequently adapted to forecast time-series energy load [53]. The following sections will elaborate a bit more on the details of MLP and CNN. Since there is no time-series energy consumption data available to access, such as smart meter data, LSTM may not be an adequate algorithm and, thereby, will not be discussed in this work.

Multi-layer perceptrons

A Multi-layer perceptrons (MLP) is usually adopted for tabular or text-based inputs. An MLP consists of a series of perceptrons stacked into a fully connected feed-forward network. When training, information flows through all the hidden layers to calculate the output, hence the term 'feed-forward'. The mathematical equation of an MLP can be expressed by $y = \varphi\left(\sum_{i=1}^{n} w_i x_i + b\right)$. The output y is computed by passing a weighted $w_i x_i$ through a non-linear activation function φ. One widely used activation function in MLP and other neural networks is the Rectified Linear Unit (ReLU). The ReLU function computes and outputs either 0 if the original value is negative, or retains the original value for positive inputs [52]. In other words, the ReLU function exhibits a linear relationship with positive input values.

Convolutional neural network

When dealing with image inputs, Convolutional Neural Network (CNN) are commonly applied. Figure 2.5 provides a simplified illustration of how a convolution layer deals with visual inputs. When inputting an image with a specified width, height and number of channels, a convolutional layer processes the data by sliding or convolving a size-defined filter to every unique place in the image where the size of the filter can fit, and results in a dot product $w^T x + b$, where w is the filter, and b is bias. The stride value, denoted as s, determines the number of pixels a filter shifts at a time. Combining these results, a feature map is produced. The size of the feature map is determined by the input image and filter defined, where the height is calculated by $(H - h + 2p)/s + 1$, width is $(W - w + 2p)/s + 1$ and number of channel is determined by the number of filters.

Example feature maps created by three sequentially connected convolutional layers are visualised in Figure 2.6. The feature maps are displayed in the order of appearance. As the network becomes deeper, the input image was processed with more convolutional layers, the resulted feature maps become more abstract and capture higher-level representations of the image.

Following the convolutional layers, a pooling layer is usually added. This layer acts as a filter to subsample the feature maps, by finding the average value or maximum value for each patch in the feature maps. It is used to reduce the number of parameters and computational load to lower the risk of over-fitting.

When the process ends, similar to other neural networks, CNN includes a fully connected layer. It performs the prediction as the weighted sum of all the outputs plus bias. Mathematically, a

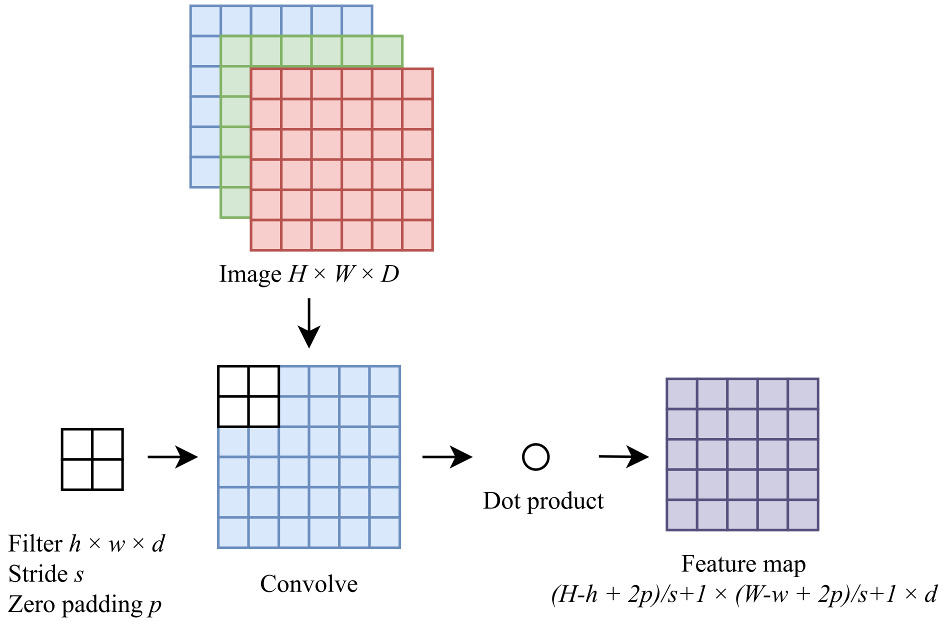


Figure 2.5: Graphical demonstration of a single convolutional layer, where an image of size $H \times W \times D$ being processed by sliding filter of size $h \times w \times d$, stride s and zero padding p, and resulted in a feature map.

CNN can be expressed as the following equation [52]:

$$z_{i,j,k} = b_k + \sum_{u=0}^{f_h-1} \sum_{v=0}^{f_w-1} \sum_{k'=0}^{f_{n'}-1} x_{i',j',k'} \cdot w_{u,v,k',k} \quad \text{with} \begin{cases} i' = i \times s_h + u \\ j' = j \times s_w + v \end{cases} \tag{2.3}$$

Where the output is the neuron z in row i, column j in a feature map k, and $x_{i',j',k'}$ is the output of the previous neuron, and $w_{u,v,k',k}$ is the connection weight.

Researchers have developed advanced architectures deriving from the basic CNN structure, such as the AlexNet model, which improves the parameter optimisation by introducing the Rectified Linear Unit (ReLU) which significantly improved the training time, and the ReLU, sustains efficient parameter learning when the network grows deeper. These architectures have proven effective performance in benchmarking datasets and many existing studies [29, 31, 54].

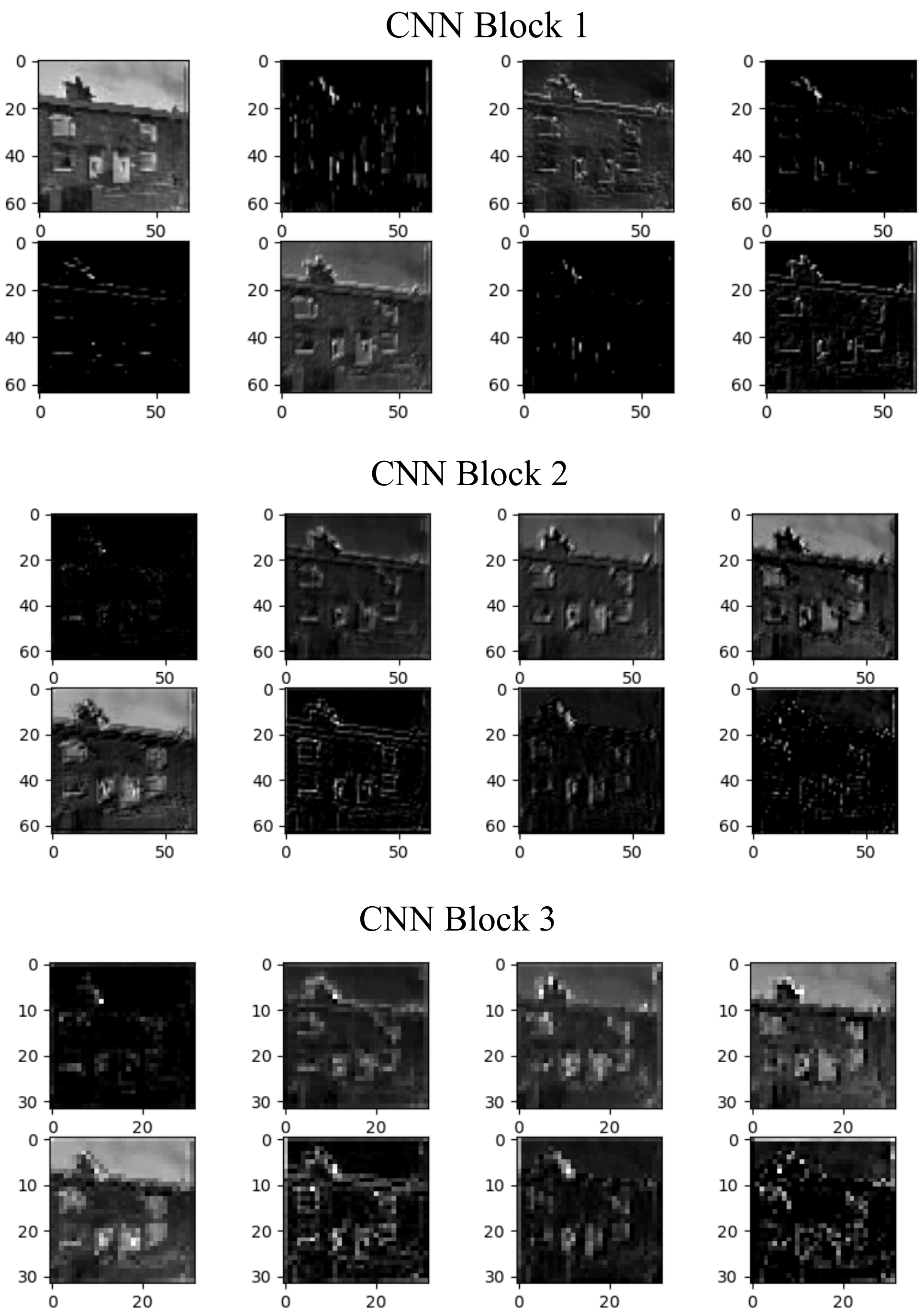


Figure 2.6: Series of example feature maps of a detached house in Barnsley. From top to down is the feature map produced from first, second and third convolutional blocks. The feature map becomes more abstract after learning by the CNN blocks.

2.5 Limitations of black-box approach

Despite all the existing studies have returned good prediction results, the utilisation of machine learning is prone to two main limitations: trustworthiness and adaptability.

2.5.1 Can we trust the model and the prediction?

One of the main challenges with data-driven approaches is the uncertainties involved when training the black box. There are two main concerns in this regard. The first concern is: *Is the model trustworthy?* The suitability of the model for the specific task and whether an adequate performance has been achieved is usually evaluated statistically, using metrics such as R-square[30], F1 score[31], Mean absolute error (MAE) [22, 30] and Mean absolute percentage error (MAPE)[30] to measure the model robustness. These evaluation metrics are usually chosen based on the prediction task, i.e. classification or regression, and the goals of the analysis. The following will provide a brief overview of these metrics, including definitions, value ranges and optima [55, 56].

R^2 is the coefficient of determination measuring the degree of fitness. For a machine learning model trained based on the independent variable x and dependent output y with a mean $\bar{y}$, the predicted outcome is denoted as $\hat{y}$. The R^2 score for this model is calculated using Equation 2.4. In the fraction, the denominator is the *variance* of y distributed around the mean, calculated by the total sum of squares $\sum (y - \bar{y})^2$. The numerator $\sum (y - \hat{y})$ is the residual sum of square, which is the summed and squared difference between actual y and predicted $\hat{y}$. Although the value range for R^2 is usually between 0 and 1, where close to 1 suggests the model is a good fit for the data inputs, it is possible to be arbitrarily negative, presenting worse performance. A negative R^2 suggests the difference between $\hat{y}$ and y is significantly large, indicating the model is not explaining better than just a single line plotted on the mean value. This evaluation metric is usually not used alone, as it does not indicate the quality of individual predictions.

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{2.4}$$

The F1 score is usually used to evaluate classification models. It calculates a harmonic mean of precision and recall. F1 score can range between 0 to 1, close to 1 the model is making accurate positive predictions (high precision) while also capturing a significant portion of the actual positive instances in the dataset. True negative predictions are not considered in this evaluation.

$$\begin{aligned} \text{F1} &= \frac{2 \times precision \times recall}{precision + recall} && (2.5) \\ &= \frac{2 \times TruePositive}{2 \times TruePositive + FalsePositive + FalseNegative} && (2.6) \end{aligned}$$

MAE calculates the mean absolute error between the predicted values and the expected ground truth. As it computes an absolute value, the value range is $[0, +\infty]$. Higher MAE suggests lower prediction accuracy. MAE treats all errors equally and does not provide information on the direction of errors.

$$\text{MAE}\,(y,\hat{y}) = \frac{1}{n}\sum\left(|y_i - \hat{y}_i|\right) \tag{2.7}$$

MAPE calculates the average percentage difference between the predicted values and the expected ground truth values. It is usually presented in a percentage format, ranging from 0% to 100%, but can be over 1 suggesting a higher error rate. Although it tells a percentage error that is easily understandable, it can be problematic when dealing with small actual values.

$$\text{MAPE}\,(y,\hat{y}) = \frac{1}{n}\sum\left(\frac{|y_i - \hat{y}_i|}{y_i}\right)\times 100 \tag{2.8}$$

Another aspect of the trustworthiness of a model is that, existing studies often employ statistical models designed using data and algorithms based on researchers' knowledge or previous studies, which may not be suitable for different case study region. This may be the reason why the majority of existing research were conducted on a comparative basis, where multiple algorithms were tested to select the most optimal one. This leads to the second concern: *Is the prediction trustworthy?* This refers to whether the model behaves in a reasonable manner that aligns with the requirements of the task.

Ribeiro et al. [57] set up an interesting experiment regards this concern. They trained a logistic regression on 20 images to determine whether a photo is showing huskies or wolves. Specifically, they intentionally selected photos of wolves with snow backgrounds for training. They tested the model performance on 60 additional images, the prediction is not as trustable as it should be. As shown in Figure 2.7, a husky is misclassified as a wolf in the prediction. This faulty prediction is because the black box model weighted the presence of snow or light background more than any visual features of the animals themselves.

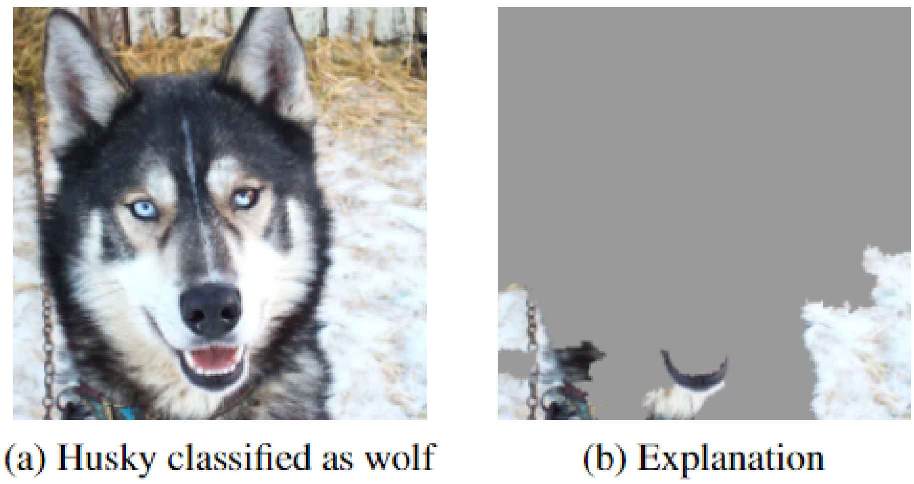


Figure 2.7: Example of a unreliable classification showing the uncertainties of the black box approach, figure adopted from Ribeiro et al. [57]. Photo (a) is wrongly predicted as wolf because the presence of snow, as explained by LIME in image (b).

These uncertainties of the black box have led to research into the interpretability and explainability of machine learning to increase the transparency of the black box approaches. The need to increase the model transparency is particularly high in sectors implementing AI for high-stakes applications, for example, medical diagnostics [58, 59, 60, 61].

While the terms 'interpretability' and 'explainability' are sometimes used interchangeably, this work considers these two ideas to be at different levels of one taxonomy. Interpretability refers to the capability of a model to be understood by humans, about why the model made such prediction [59]. One example model with high interpretability is the decision tree. In decision trees, each split is based on a clear condition, making it relatively easy to understand how the prediction processes. These conditions, such as the gini value calculated when the tree splits into the next hierarchy, can be used to rank their relative feature importance towards the model prediction. Therefore, we can say the random forest has high interpretability. Existing studies have used the gini value calculated to identify the dominant housing features which can be further used to guide retrofit options [21, 30, 31].

However, as models grow deeper and more complex, such as the use of deep neural networks, the model becomes challenging to interpret. This difficulty in understanding complex algorithms emerges in the development of Explainable artificial intelligence (XAI). The explain-ability mainly concerns the *post hoc* understanding of the input-output relationship of the model. Popular XAI tools are Local interpretable model-agnostic explanations (LIME) and SHAP.

LIME and SHAP offer different approaches to explain a model. LIME is a kind of local surrogate model. It trains a local surrogate or substitute model, such as a linear model or random forest, using weighted sample data, to mimic similar prediction results as the complex model [59]. Mathematically, the local surrogate model can be expressed as Equation 2.9. Where the explanation LIME model tries to develop a local surrogate $G \approx f$ the original model, by minimising the error L between G and f, at the same time use minimum model complexity $\Omega(g)$, while π_x controls the number of instances x to use in the local model [59]. Because LIME uses local surrogate models, it mainly focuses on local interpretation, rather than features' global contributions. In other words, it focuses on providing explanations for each prediction, which may not capture the overall global behaviour of the model. Figure 2.7 is an example of using LIME to explain image inputs.

$$\text{LIME}(x) = argmin_{g \in G} L(f, g, \pi_x) + \Omega(g) \tag{2.9}$$

SHAP, on the other hand, offers a global approach. The main value associated with this tool is the 'shapely values'. SHAP considers the prediction as a result of coalition among the feature inputs. By definition, shapely values are 'the average marginal contribution of a feature value across all possible coalitions.' [59, 61, 62]. Mathematically, the shapely value or the SHAP value for a single feature is the sum of all the possible coalitions the feature can join, which can be computed by Equation 2.10. In this equation, $M!$ is the number of ways to form a coalition, and $|S|$ is the number of features in the coalition S. The shapely value assumes all the feature joins the coalition sequentially, which results in $(M - |S| - 1)!$ number of ways for a feature to join the coalition after feature i joins. $\frac{|S|!(M-|S|-1)!}{M!}$ is the weight for the marginal contribution of feature i to coalition S, which is computed by the latter part of the equation.

$$\text{Shapley values} \quad \phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(M-|S|-1)!}{M!}\left[f_x\left(S \cup \{i\} - f_x\left(S\right)\right)\right] \tag{2.10}$$

For example, in the study by Sun et al. [31], SHAP was used to examine the importance of building features in a multimodal prediction model. As shown in Figure 2.8, most of the pink dots identified by their model are clustered around structural elements such as windows and doors, which suggests that the model is effective to use meaningful areas in the building facade for estimation.

Figure 2.8: Example streetview images being explained using SHAP, produced by Sun et al. [31]. The XAI tool uses dots to highlight pixels where the trained model considers important for the prediction. Red dots indicate positive contributions and blue indicate negative.

2.5.2 Is the designed model applicable to changing contexts?

Secondly, traditional machine learning requires the target region to have the same feature distribution as the source region when applying the trained model[25, 63, 64, 65, 66]. Meeting this condition can be challenging, which restricts successfully trained models from wider applications.

This concern is related to the scale problem when working with spatial data. By definition, the scale can be referred to as the size or spatial extent of the study[67]. Within the determined geographical space, the pattern observed is a result of a complex system with activities such as interaction, dispersion, diffusion and exchange which resulted in spatial heterogeneity [25]. Therefore, the scale of a study is important because it informs about the sampling for model training.

Geographical scales used in existing energy estimation studies range from neighbourhood level to city-wide, and national scale. Generally, a neighbourhood-level study allows the model to capture local variations specific to the area, but the model may lack generalisation capabilities that fully capture the diversity and complexity present in larger regions. Using city-scale data offers a balance between local variations and a broader representation of the area. A larger sample size compared to neighbourhood data can also lead to more robust models. But the variability between neighbourhoods within the city may be neglected. On the other hand, although using data at a national scale may generate a comprehensive view of energy usage patterns across the entire country, it may overlook local nuances and regional variations that have a significant impact on energy consumption. The model trained using national-scale data may not be able to capture specific features of individual cities or neighbourhoods.

However, there is limited investigation into how geographical scales selected for modelling may impact the energy estimation performance. Therefore, choosing the appropriate scale of data depends on the research objectives, the level of generalisation required, and the availability of data.

2.6 Potential Methodologies

2.6.1 Deep Multimodal Learning

The lack of accurate data is not unique to building energy analysis. There is a growing interest in the potential of developing deep learning models that use multiple inputs [68, 69], named deep multimodal learning, to reduce the bias. The nature of this approach is to accept heterogeneous cues from different modalities to offer additional and potentially more comprehensive knowledge for a given task. Similar to the five senses human beings have, data can be classified under different modalities, such as image (visual), text (word), audio (sound), and physiological signals. Existing applications are in the field of face recognition, medical diagnosis, and self-driving systems [68, 69], but few applications are in building energy prediction. This study considers that describing a property using different modalities with a multimodal deep learning approach may to some extent reduce the level of bias compared with unimodal approaches where only one type of data is used (i.e. EPC).

Sun et al. [31] experimented with the multimodal approach combining Scottish EPC data and GSV to estimate the energy efficiency rating. 165,318 properties in Glasgow were examined, and their modelling accuracy increased from 79.7% to 86.8% when including the GSV facade images compared to using only the EPC. This case study validated the benefits of including additional visual components is beneficial to energy efficiency modelling.

2.6.2 Automated Machine Learning (autoML)

This study will apply Automated machine learning (AutoML) and Neural Architecture Search (NAS) to conduct an exhaustive but effective search on optimal algorithms and hyperparameter settings, and reduce the reliance on the knowledge from machine learning experts and existing literature. A comparison between existing algorithms used in existing studies and the ones selected using the automatic tool will be made to verify the robustness of such method.

The autoML offers a combined algorithm selection and hyper-parameter optimisation tool to reduce the costs of machine learning model development [70]. It takes care of raw data input from the beginning to the final step, which reduces development costs, and at the same time provides optimal estimation accuracy [71, 72]. Similarly, neural architecture search can be done automatically to find the best-fit neural network structure. A flowchart of the common procedure of automatic model construction is visualised as Figure 2.9 [71]. An automatic approach usually goes through four stages, data preparation, feature engineering, model generation and evaluation. Similar preparation stages are required for both autoML and NAS to extract features from the input data to feed into the model generation stage. This stage can be further separated into *search space*, where an optimal machine learning model or neural network structure is found, and *optimisation methods*, where the found model is fine-tuned. Hyperparameter refers to the parameters such as learning rate, and architecture refers to the elements such as the number of layers in a neural network. Then the optimised model is evaluated to find the most robust solution for the given task. On the other hand, such an automatic procedure means that the developers may have less control over the model structure and fine-tuning process.

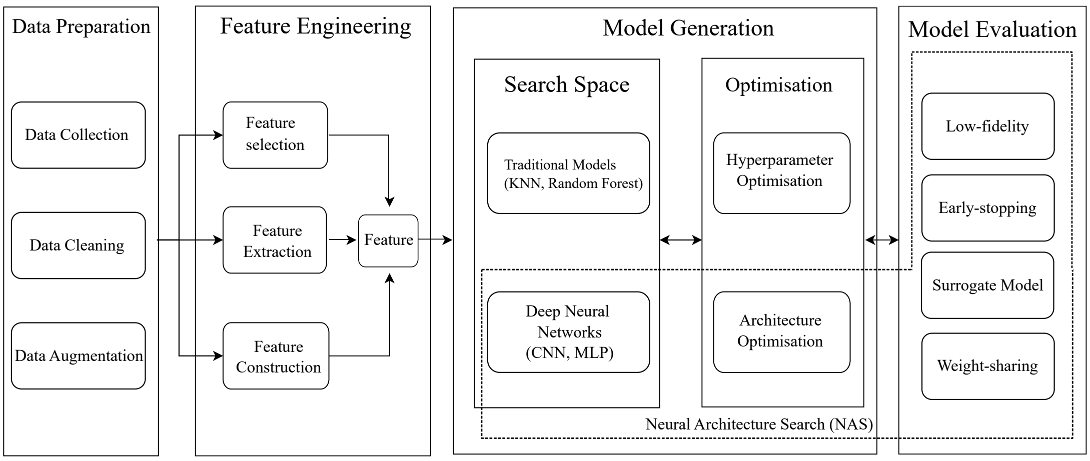


Figure 2.9: A flowchart inside the 'black box' of autoML and NAS. [71])

A wide range of open-source autoML tools is available to choose from. Ferreira et al. [73] analysed six recent autoML libraries: Auto-Sklearn, AutoGluon, H2O AutoML, rminerAutoML, TPOT and TransmogrifAI. Their performance were tested and compared on binary and multi-class classification and regression tasks using thirteen benchmark datasets. Small differences in prediction accuracy were found among the inspected tools, 3% to 16% difference for binary classification tasks, 4% to 8% for multi-class classification, and only 1% difference was found when training with all regression data [73]. Such little difference suggests that the selection of the autoML tool should not impact much on the overall prediction accuracy which made the tool selection convenient. Since different autoML tools only have limited difference in prediction accuracy, the tool used in this thesis was selected based on the availability of such a tool, coverage of existing machine learning algorithms and familiarity of the package itself. Auto-sklearn was selected for the study presented in Chapter 4.

Auto-sklearn

Auto-sklearn is an autoML tool developed based on the Scikit-learn, a popular Python library offering a wide range of machine learning algorithms [70]. As illustrated in Figure 2.10, Auto-sklearn can be considered as a pipeline with three main steps. The first step is meta-learning, where the input data is compared with pre-stored benchmark data [70]. The algorithms that performed well on the benchmark data that is similar to the user inputs are selected as target algorithms. The second stage then trains, fine-tunes and evaluates all target algorithms. The Bayesian optimisation simultaneously calculates the correlations between the hyper-parameter settings and the prediction accuracy. This correlation is the main criterion the Auto-sklearn used for algorithm selection. The pipeline also tests whether building an ensemble of multiple algorithms will achieve better prediction performance. The performance of all the trained algorithms is evaluated and only the best one proceeds to produce prediction.

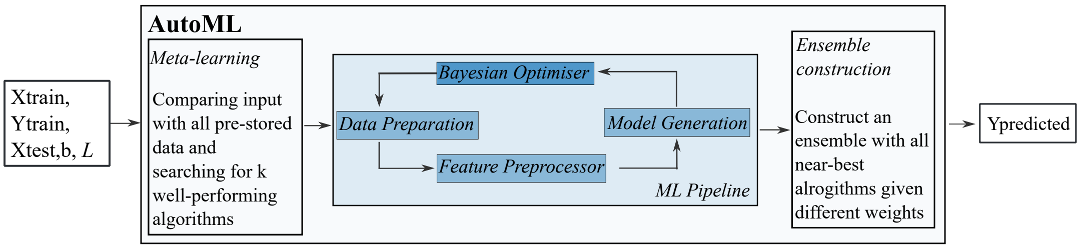


Figure 2.10: An overview of the Auto-sklearn system. The input data follows the pipeline to construct the most optimal model and then perform prediction. The pipeline involves meta-learning, data preparation, feature preprocessor, model generation, Bayesian optimisation and ensemble construction.

2.6.3 Neural architecture search (NAS)

Similarly, a subfield of autoML is the automatic search for the best neural network structures named Neural Architecture Search (NAS). The general workflow of searching for the best architecture is visualised in Figure 2.9. Typically, NAS algorithms can be classified based on their architecture optimisation methods, including reinforcement learning, evolution-based, gradient descent, surrogate model-based and hybrid methods. While autoML is largely dependent on the pre-sets of algorithms stored in the selected developed packages, for example, Auto-sklearn uses the algorithms stored in Scikit-learn libraries, a NAS would have more flexibility in the architecture search.

He et al. [71] conducted a systematic overview of the state-of-the-art of NAS, and compared the performance of NAS algorithms on classification tasks using the benchmark image data of CIFAR-10 and ImageNet. The prediction accuracy ranges from 91.12% to 97.87% and 70.62%

to 84.4% respectively [71]. Although there is only a minor difference between the prediction accuracy, the resources used, calculated by the number of GPU utilised and the days spent to complete the calculation, ranges from 0.2 to 22,400. Therefore, the trade-off between prediction accuracy and resource budget becomes the key consideration when selecting the NAS tool to use.

2.6.4 Transfer Learning

Transfer learning has emerged as a promising approach to address the second key limitation of traditional machine learning, as discussed in Section 2.5. Traditionally, machine learning models require the prediction data to have the same feature distribution as the training data, otherwise, a new model should be trained from scratch [63, 64, 65, 66]. However, in many cases, data availability is limited, such as the age prediction conducted by Biljecki and Sindram [32] which the prediction accuracy is largely impacted by data availability. The lack of adequate data can be of low quality, or expensive to collect. Transfer learning, on the other hand, reduces the reliance on data by leveraging the knowledge gained from one task to help other relevant prediction tasks. Benefiting from the flexible structure of neural networks, we are able to intrude in the middle of the structure, and connect pre-training layers where the knowledge has been stored to learning layers for prediction.

The idea of transfer learning mainly revolves around three elements, the domain, D, the prediction task T, and the marginal probability distribution $f(\cdot)$ or $P(y|x)$. Transfer learning considers the base model, or in this case, the region where comprehensive housing data is available, as the source domain D_S. And the region where less data is available, as the target domain D_T. According to the variations of these three elements between D_S and D_T, transfer learning can be classified as [63, 65]:

- $T_S \neq T_T$: When only the prediction tasks differ, the transfer learning model is known as *Inductive transfer learning.* Depending on the availability of labelled data in D_S, inductive transfer learning can be further divided into self-taught learning and multi-task learning.

- $D_S \neq D_T$ and $T_S = T_T$: This scenario is referred as the *Transductive transfer learning* or *Domain adaptation.* Our study falls into this scenario.

- $T_S \neq T_T$, Y_S and Y_T unknown: Similar to unsupervised machine learning, when no labelled data in either domain, *Unsupervised transfer learning* should be applied.

One significant constraint of transfer learning is the assumption of domain similarity between the source and target domain. When the domains diverge in terms of data distribution, or features, the transferred knowledge may not be relevant, leading to poor performance on the target domain. Additionally, negative transfer may occur, where the knowledge from the source domain hinders performance on the target domain. Despite these limitations, researchers are actively applying transfer learning in real-world scenarios, such as face verification and medical image diagnosis [65, 74], as well as the recent popular language model GPT-4 [75], but its application in energy prediction is relatively new.

Gao et al. [65] utilised transfer learning to predict the electricity consumption of a newly developed office building. As new buildings lack historical energy consumption data, a traditional machine learning model is difficult to train. Gao et al. [65] developed a transfer learning model to learn the energy behaviour from two similar buildings in other cities to assist the prediction for the new-build. The network model was first pre-trained using D_S, top layers are frozen as a method to store the knowledge, and fine-tuned with data from D_T. This integration of additional knowledge from similar buildings resulted in a significant improvement of 20% on average for the models[65].

Hooshmand and Sharma [64] developed a transfer learning model to predict the energy demand for the next 24 hours, on the basis of a CNN model and transfer learning. Similar to Gao et al. [65], the convolutional layers are frozen to retain the knowledge learned, and only the last fully-connected layer is retrained. The performance of this transfer learning model is evaluated by comparing with three other algorithms: a classic but simple and linear model using only D_T data, a CNN model trained with only D_T, and the direct application of the model pre-trained with D_S without transferring the knowledge (no fine-tuning). The transfer learning significantly increased the prediction accuracy by 20%, 17% and 30% respectively.

2.7 Summary of Research Needs

This chapter provides an overview of existing literature on residential building energy performance. With a particular focus on data-driven approaches, the popular tabular and image dataset (in Section 2.2) and machine learning techniques were discussed (in Section 2.4). As an intensely explored area, existing studies have demonstrated success records in applying such methodologies to energy prediction with available data. However, there remain gaps that need to be addressed regarding the limits of available data and traditional machine learning methods for energy predictive models.

Firstly, the selection of algorithms and parameters in traditional machine learning models heavily relies on researchers' knowledge and previous studies, leading to potential biases and suboptimal performance. Secondly, traditional machine learning models require the target domain to have the same feature distribution as the source domain, making it challenging to apply the model to unseen contexts. Furthermore, there are also issues with the available data, particularly with EPCs. Despite being one of the predominant databases in data-driven energy predictive models, EPC suffers from reliability issues due to its making procedure.

To aid these needs, this thesis will investigate the application of autoML, multimodal deep learning, transfer learning and explainable artificial intelligence, to enhance the understanding of residential building energy performance. Leveraging these techniques will enable more accurate predictions and broader applicability, even for regions with limited available data. The following chapters 4, 5 and 6 contribute to addressing these limitations in the following ways:

1. Developing a prediction framework that implements automatic algorithm and parameter settings selection, reducing the reliance on subjective choices and improving model performance;

2. Developing and implementing a model incorporating multiple modalities from multiple data sources, minimising the bias caused by relying on only one source of data, therefore achieving better trustworthiness;

3. Developing and implementing of a model that is capable of coping with unknown contexts, by leveraging and transferring knowledge from known properties with well-documented energy performance information to predict properties unknown, so it has higher adaptability to changing contexts;

4. Implementing appropriate interpretable and explainable tools to improve the transparency and understanding to ensure the reliability of the model developed, to improve the trustworthiness and transparency.

Chapter 3

General data and methods

3.1 Chapter introduction

The analysis and prediction this thesis conducted are mainly based on multimodal data representing residential buildings' morphological features and thermal conditions. In general terms, in this thesis, multimodal learning is referred to as research problems that include multiple modalities [76]. Although there is no specific definition of what modalities are, this thesis links the modalities with sensory modalities, such as image (visual), text (word), audio (sound), and physiological signals (e.g., vision, vocal and touch) that stored in different formats (e.g., image, audio, and tabular of texts). As mentioned in the literature review, it has been widely used in the field of face recognition, medical diagnosis, and self-driving systems, but few applications are in building energy prediction. The nature of this approach is to accept heterogeneous cues from different modalities for additional and potentially more comprehensive knowledge of a given task. Therefore, describing a property using different modalities with a multimodal learning approach may reduce the level of bias compared with mono-modal approaches where only one type of data is used.

Spatial, morphological and thermal data, in tabular and image format, are utilised in this work. This chapter provides a summary of general data collection, statistics, preprocessing steps and general methodologies applied in the following Chapters 4 to 6.

3.2 General data explained

Several criteria were considered when selecting the housing features and data sources used in this thesis. To investigate the application of multimodal learning and transfer learning, the data needs to be available at a city scale for different regions, and needs to have a common reference system so it can be elaborated with other data used in this study. The data used in existing studies as summarised Table 2.1 offered insights into the potential databases.

In this thesis, map data and street view image data were trialled as representations of buildings' morphology. EPCs were used to offer descriptions for buildings' thermal conditions. The map data was used in the initial study presented in Chapter 4, and street view image data was used in the work discussed in Chapter 5 and 6.

3.2.1 Map data

The Ordnance Survey (OS) MasterMap Building Height Attribute products [77] were used in this study. It provides a comprehensive map and spatial data covering the entire UK. Table 3.1 has listed all the features extracted from the map data, with a brief description of what building morphology each feature represents. The statistics are provided in Table 3.2.

Variables 1, 3 and 4 are values provided in the OS MasterMap, the rest are calculated using ArcGIS. The perimeters were calculated using the field calculator, and the number of vertices was calculated following $VxCount = !shape!.pointcount$ in ArcMap. Variables 5 and 6 are metrics adapted to describe the complexity of the polygons representing the building shape. The Normalised perimeter index (NPI) is a shape metric measuring the roundness. An NPI value

Table 3.1: List of features based on OS MasterMap, with brief descriptions of what they represent and how they are calculated.

No.	Variables	Description
1	Total floor area	Area of the building footprint (a)
2	Perimeter	Total length of building polygon outline (p)
3	Relh2	Relative height from ground to the base of the roof
4	Relhmax	Relative height from ground to the highest part of the building
5	NPI	Normalised perimeter index (NPI) calculated by $\frac{2\sqrt{a\pi}}{p}$
6	Vxcount	Number of vertices in building polygon
7	Builtrate	Ratio between all property footprint and postcode area

Table 3.2: Statistics of map data used for model prediction, before and after applying the simple random sampling approach in the first case study in Chapter 4.

Variables	All Samples			Subsamples		
	Mean	Std	CV	Mean	Std	CV
Total floor area	81.45	38.16	0.47	81.02	40.11	0.49
Perimeter	41.82	26.01	0.62	45.84	32.83	0.72
Relh2	6.33	3.26	0.52	6.78	3.95	0.58
Relhmax	8.17	3.40	0.42	8.73	4.16	0.48
NPI	0.78	0.04	0.05	0.77	0.05	0.06
Vxcount	12.57	7.29	0.58	9.96	5.00	0.50
Builtrate	0.21	0.28	1.33	0.23	0.36	1.57

further departed from 1 suggests the building has a more complex shape [78]. Three properties are highlighted in Figure 3.1 as examples. Property A is a primary school in Sheffield, while B and C are terraced houses that can be commonly found in the UK. Each property has been marked with its area, total perimeter length and the calculated NPI. By comparing these values, we can see that, buildings with more irregular shapes have smaller NPI values (Property A). On the other hand, B and C are the same type of housing, so similar NPI and perimeter lengths were calculated because they are more similar in building shapes. This measure can offer valuable indications of the shape of the building, which is closely related to its type and design, and tells information on the year of construction and energy consumption.

3.2.2 Street-view image data

The map data can only provide partial representations of the building as it is mostly derived at the floor plan level. Street view images, on the other hand, offer more information from the vertical illustration, including the appearance of the target properties, the allocation of and the ratios between different building elements, and also to some extent the conditions of these elements. It is a potential alternative source of data to map and tabular databases. In this study, two databases were compared and combined for analysis: GSV downloaded online and street-view images captured by the Multi-Spectral Advanced Research Vehicle (MARVel) owned by the research group. The detailed data collection procedure will be presented in the subsequent chapters.

Example street view images are illustrated in Figure 3.2, where Figure 3.2a is an example image

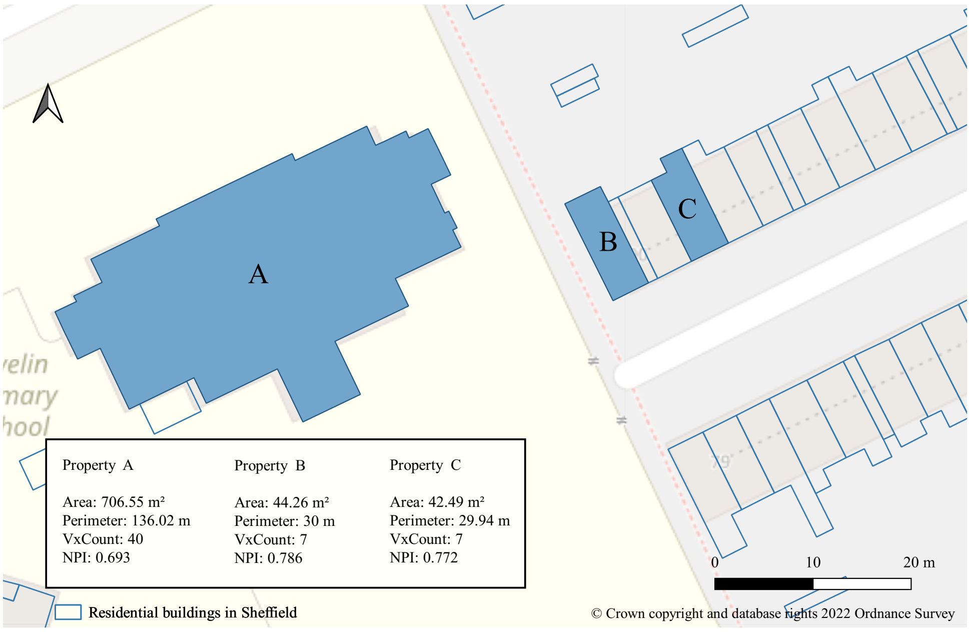


Figure 3.1: Illustration of example map data.

downloaded using the Google service, and Figure 3.2b is an example captured by MARVEL.

Since both images are captured while driving through the neighbourhood, they unavoidably contain visual information irrelevant to this study on building energy analysis. While existing studies such as Despotovic et al. [29] and Zeppelzauer et al. [38] used scale-invariant feature transform to detect pixels of interest and produce individual image patches with only specific building elements for prediction, this study believes that using whole property images can allow the energy estimation algorithm better understand the global context of such housing. Therefore, to reduce the number of irrelevant contexts, an object detection algorithm, called YOLOv5, is applied to the extracted houses from the street view images.

A custom YOLOv5 model is trained specifically for this study using over 800 manually labelled GSV images with bounding boxes. The YOLOv5 is a python algorithm firstly developed by [79]. The algorithm detects objects by dividing an image into a grid and then calculating the weights to help determine the possibility of whether the detected pixels belong to a house feature as a regression problem[79]. The pixels detected as houses are grouped together and bounded with a box, with values on the top of the boxes showing how likely the detected feature is a house. As the detection results shown in Figure 3.3, the trained model successfully detected where houses are located in the street view images. The custom YOLOv5 model has an Average precision (AP) of around 0.8. AP is a commonly used metric in object detection, which compares the ground-truth bounding box with the detected ones and produces a single value ranging from 0 to 1 [79].

Object detection often returns multi-detection in one street view image, it is thereby necessary to select one of them as the target property. This step has taken the patterns found for houses' energy performance in the same neighbourhood into consideration. As discussed in the literature review Chapter 2, residential houses in the same neighbourhood tend to be built under similar

(a)

(b)

Figure 3.2: Example Streetview images collected. (a) was downloaded from Google Streetview, (b) was captured by MARVEL.

(a)

(b)

Figure 3.3: The object detection results of the example street-view images. Objects detected as houses are marked in red boxes, where the values on the box indicated how likely the object is a house. Multiple houses were detected in both images.

(a) (b)

Figure 3.4: By applying the watershed segmentation, pixels inside the image is removed to prevent causing bias during the model training process. (a) Example house images after object detection and cropping. (b) The resulted images after the watershed segmentation, where the sky is removed.

specifications and have similar energy performance. With this assumption, even if the properties next to the actual queried house are selected for prediction, the bias should be limited. These houses detected in various sizes provide the possibility to determine the target house for the following energy performance prediction. If multiple houses are detected in the same image, the largest house detected is used for the following prediction. The first houses on the left in both example street view images in Figure 3.3 are considered as the target properties. All selected images are resized and stretched when necessary for the following machine learning models.

In an experiment setup, the 'sky' in the images was highlighted by the machine learning model as key features to perform prediction. Therefore, to avoid developing a machine learning model similar to the biased example discussed in the literature review (Figure 2.7), these housing crops are further preprocessed to remove the sky using the watershed segmentation. The watershed segmentation considers an image as a topography map, and separates the object and background of the image based on the difference in intensity in grayscale. Brighter pixels have higher values of intensity. For street view images, skies are usually brighter than properties, which makes it easy to apply a watershed segmentation. Figure 3.4 presents examples of street view images being processed with the watershed segmentation.

3.2.3 Thermal information from the Energy Performance Certificates

Energy performance certificates are used to provide texts or tabular variables relating to buildings' energy performance. EPC is a popular resource in energy analysis studies. It is a compulsory document issued for every property in the UK. A typical EPC record is usually valid for 10 years, but properties are required to have an assessment once the property's condition or market status is changed. When a new registration is created, often the procedure fails and leaves the old records undeleted in the system. As studied by Crawley et al. [46], this mistake has resulted in multiple EPC records that can be found associated with the same property. The initial preprocessing step for the downloaded EPCs is to filter these duplications. If the property address or reference number occurred multiple times, it means that the property is associated with multiple EPC records. These redundant EPCs were filtered based on when the record was created. The single latest-issued EPC is used as the data input.

Overall, one EPC contains 92 categories offering building-related information from three perspectives: spatial and reference information to identify where the property is (e.g. the unique property reference number (UPRN) and address); the current property characteristics and energy performance; and potential characteristics and energy performance if recommended retrofit implemented. These data were filtered, and only necessary information was adapted to avoid high costs in time and computational power. The features used for existing studies, as summarised in Table 2.1 were also used as a reference when selecting the features for this study. In this study, the EPCs are mainly used as a source that offers the conditions of properties' elements, which is directly associated with their thermal performance. Apart from these thermal features, UPRN was used to link the data with map data from OS, and detailed address was used for streetview collection. The selected variables and their brief descriptions are listed in Table 3.3.

Table 3.3: List of data extracted from the EPC, with brief description of what the represent of and example classes in categorical data

No.	Variables	Description
8	Property type	Type of property (e.g. house)
9	Built form	Type of built-form (e.g. detached)
10	Number habitable rooms	Number of rooms in the property
11	Number heated rooms	Number of rooms can be heated in the property
12	Roof description	Roof types and insulation conditions (e.g. pitched)
13	Walls description	Wall types and insulation conditions (e.g. filled cavity)
14	Floor description	Floor types and insulation conditions (e.g. solid, insulated)
15	Lighting description	Percentage of low energy lighting used
16	Main heat	Types of heatings used (e.g. Air source heat pump)
17	Ageband	Construction age grouped in 12 bands (e.g. before 1900)
18	Energy consumption	Total energy consumption (kWh/year)

Variables 8 to 11 and 17 are features describing the general characteristics of the buildings, while variables 12 to 16 provide more detailed descriptions of the conditions of specific building elements. The original energy consumption recorded in the EPCs is measured in kWh/m^2 per year. The total floor area for each house is taken into consideration here to produce variable 18, which is used as the ground truth data for training the energy prediction model. The total energy consumption is used in this study instead of energy intensity to enable the exploration of the relationship between energy consumption and floor area, and further allows the outputs to be comparable with real smart meter data in future studies when such data are available.

Inconsistencies and abnormal entries are found for the categorical variables. This may be caused by the fact that the records were created by multiple inspectors and may have also followed different versions of guidance on creating EPCs. All variables are pre-processed following two steps. The first step is to replace blank or abnormal entries. For example, if the entry is marked as 'INVALID!' or 'NO DATA', these entries are combined as 'unknown'. This process also ensures the records only contain English records. The classes in each categorical data (variables 13-17) were also reorganised. Similar descriptions in the categories are found and merged. For instance, 'some double glazing' and 'partial double glazing' used to describe the window insulation conditions are combined into one category. A detailed procedure filtering and preprocessing the EPC data is attached as Appendix 8.3.

3.2.4 Statistics of EPCs used in the following thesis

The following section includes statistics for the EPCs obtained for each studied region. When conducting each designed case study, the EPC data are either pre-processed to filter problematic entries or linked with other data sets to perform multimodal analysis. Therefore, the following statistics are only indications of the data used in their respective case studies, neither cover all the records in the original EPC downloaded online, nor represent the whole property stocks in the region.

The statistics for numeric and categorical features are summarised respectively. Separate tables are provided for regions where the entire database or a subset of it are used for multiple case studies. For numeric data, their average, Standard deviation (std) and Coefficient of variance (cv) are calculated. The coefficient of variance is calculated as the relative variability of a dataset's mean in relation to its standard deviation $CV = \frac{std}{mean}$, it is used as an indicator of variability in data distribution. A higher cv suggests higher variability in the dataset. For categorical data, each class is counted and calculated with the proportion of the grand total.

In total, there are four cities examined in the following studies, Sheffield (Chapter 4), Barnsley(Chapter 4 to 6), Doncaster(Chapter 6) and Merthyr Tydfil (Chapter 4 and 6). Among the four regions, Doncaster has the smallest total floor area per property on average, and the smallest CV. This finding is not surprising, as the Doncaster dataset used in the third case study only contains a small subset of all the existing housing stock captured in a neighbourhood due to data availability. Properties in all regions have 4 habitable and heated rooms on average. Barnsley has the highest average proportion of energy-saving lighting being installed, and the smallest CV. For the amount of energy consumption, on average, according to the EPC recoded estimation, properties in Merthyr Tydfil tend to use the largest estimated regulated energy use among the four regions, while Barnsley has the largest variability across properties (largest CV).

Properties in Sheffield

Properties in Sheffield was examined in the first case study, will be presented in Chapter 4.

Total sample size: **142,75** residential properties in Sheffield

Numeric data

Table 3.4: Statistics of numeric data used for model prediction, before and after applying the simple random sampling approach in the first case study in Chapter 4

Variables	Mean	Std	CV
Total floor area	81.45	38.16	0.47
Number habitable rooms	4.06	1.77	0.44
Number heated rooms	3.96	1.76	0.44
Lighting description	0.53	0.34	0.64
Energy consumption (kWh)	22,219.42	14,149.90	0.64

Categorical data

Table 3.5: Statistics of the categorical features extracted from the Sheffield residential EPCs.

Table 3.6: Property type

Property type	Count	Proportion
Bungalow	6,481	4.54%
Flat	31,428	22.09%
House	101,220	70.84%
Maisonette	3,627	2.54%

Table 3.7: Built form

Built form	Count	Proportion
Detached	24,808	17.40%
Enclosed End-Terrace	1,654	1.17%
Enclosed Mid-Terrace	1,177	0.83%
End-Terrace	20,323	14.22%
Mid-Terrace	41,763	29.21%
Semi-Detached	49,363	34.53%
unknown	3,668	2.66%

Table 3.8: Floor description

Floor descriptions	Count	Proportion
Another dwelling below	23,045	16.20%
Conservatory	2	0.00%
Solid, insulated	4,827	3.37%
Solid, uninsulated	26,698	18.67%
Suspended, insulated	4,172	2.92%
Suspended, uninsulated	68,153	47.67%
To external air, insulated	154	0.11%
To external air, uninsulated	163	0.11%
To unheated space, insulated	1,642	1.15%
To unheated space, uninsulated	6,451	4.51%
Average U-Value 0-1.33	7,399	5.23%
unknown	50	0.04%

Table 3.9: Windows description

Windows descriptions	Count	Proportion
Double	129,050	90.39%
High performance	7,799	5.47%
Multiple	192	0.13%
Secondary	591	0.41%
Single	4,831	3.38%
Triple	195	0.14%
unknown	89	0.08%

Table 3.10: Continued Statistics of the categorical features extracted from the Sheffield residential EPCs.

Table 3.11: Walls description

Walls descriptions	Count	Proportion
Cavity wall, insulated	75,517	52.82%
Cavity wall, uninsulated	18,479	12.92%
Cob, as built	10	0.01%
Granite or whin, insulated	17	0.01%
Granite or whin, uninsulated	186	0.13%
Sandstone or limestone, insulated	648	0.45%
Sandstone or limestone, uninsulated	6,833	4.78%
Solid brick, insulated	1,381	0.97%
Solid brick, uninsulated	23,085	16.15%
System built, insulated	2,479	1.73%
System built, uninsulated	1,485	1.04%
Timber frame, insulated	2,010	1.41%
Timber frame, uninsulated	121	0.08%
Average U-Value 0-2.1	10,458	7.46%
unknown	47	0.04%

Table 3.13: Mainheat

Mainheat descriptions	Count	Proportion
Air source heat pump	198	0.14%
Boiler	124,334	86.96%
Community scheme	5,722	4.00 %
Electric heaters	4,319	3.02%
Ground source heat pump	17	0.01%
Room heaters	7,465	5.22%
Warm air	752	0.53%
Water source heat pump	2	0.00%
Unknown	164	0.11%

Table 3.12: Roof description

Roof descriptions	Count	Proportion
Another dwelling above	20,833	14.63%
Flat, insulated	2,948	2.06%
Flat, uninsulated	1,971	1.38%
Pitched, insulated	83,452	58.37%
Pitched, uninsulated	21,332	14.92%
Roof room(s), insulated	2,566	1.79%
Roof room(s), uninsulated	2,006	1.40%
Thatched	5	0.00%
Thatched, insulated	9	0.01%
Average U-Value 0-2.4	7,560	5.37%
unknown	74	0.06%

Properties in Barnsley

Properties in Barnsley are used in all three case studies. In the first two case studies, which will be presented in 4 and Chapter 5, the total sample size is: **10,897** residential properties in Barnsley.

Numeric data

Table 3.14: Statistics of numeric data used for case study in Chapter 5

Variables	Mean	Std	CV
Total floor area	88.45	44.89	0.51
Number habitable rooms	2.75	2.61	0.95
Number heated rooms	4.39	1.26	0.29
Lighting description	0.67	0.37	0.55
Energy consumption (kWh)	14,509	1,0149	0.69

Categorical data

Table 3.15: Statistics of the categorical features extracted from the Barnsley residential EPCs.

Table 3.16: Property type

Property type	Count	Proportion
Bungalow	575	5.28%
Flat	1,316	12.08%
House	8,969	82.31%
Maisonette	36	0.33%

Table 3.17: Built form

Built form	Count	Proportion
Detached	3,887	35.67%
Enclosed End-Terrace	95	0.87%
Enclosed Mid-Terrace	38	0.35%
End-Terrace	1,506	13.82%
Mid-Terrace	1,517	13.92%
Semi-Detached	3,566	32.73%
unknown	287	2.63%

Table 3.18: Continued Statistics of the categorical features extracted from the Barnsley residential EPCs.

Table 3.19: Floor description

Floor descriptions	Count	Proportion
Another dwelling below	861	7.90%
Solid, insulated	2,867	26.31%
Solid, uninsulated	883	8.10%
Suspended, insulated	1,117	10.25%
Suspended, uninsulated	766	7.03%
To external air, insulated	23	0.21%
To external air, uninsulated	2	0.02%
To unheated space, insulated	33	0.30%
To unheated space, uninsulated	18	0.17%
Average U-Value 0-1.33	4,292	39.39%
unknown	34	0.31%

Table 3.20: Windows description

Windows descriptions	Count	Proportion
Double glazing	7,242	66.46%
High performance glazing	3,564	32.71%
Multiple glazing	2	0.02%
Single glazing	43	0.39%
Triple glazing	10	0.09%
unknown	35	0.32%

Table 3.21: Walls description

Walls descriptions	Count	Proportion
Cavity wall, insulated	5,136	47.14%
Cavity wall, uninsulated	535	4.91%
Cob, as built	3	0.03%
Granite or whin, uninsulated	10	0.09%
Sandstone or limestone, insulated	90	0.83%
Sandstone or limestone, uninsulated	182	1.67%
Solid brick, insulated	48	0.44%
Solid brick, uninsulated	231	2.12%
System built, insulated	17	0.16%
System built, uninsulated	15	0.14%
Timber frame, insulated	109	1.00%
Timber frame, uninsulated	1	0.01%
Average U-Value 0-2.1	4,485	41.16%
unknown	34	0.31%

Table 3.22: Roof description

Roof descriptions	Count	Proportion
Another dwelling above	705	6.47%
Flat, insulated	15	0.14%
Flat, uninsulated	9	0.08%
Pitched, insulated	5,361	49.20%
Pitched, uninsulated	272	2.50%
Roof room(s), insulated	180	1.65%
Roof room(s), uninsulated	8	0.07%
Thatched	2	0.02%
Average U-Value 0-2.4	4,305	39.51%
unknown	39	0.36%

Table 3.23: Continued Statistics of the categorical features extracted from the Barnsley residential EPCs.

Table 3.24: Mainheat descriptions

Mainheat descriptions	Count	Proportion
Air source heat pump	78	0.72%
Boiler	10,220	93.80%
Community scheme	149	1.37 %
Electric heaters	126	1.16%
Ground source heat pump	4	0.04%
Room heaters	247	2.27%
Warm air	5	0.05%
Unknown	67	0.61%

Table 3.25: Construction age band

Construction age band	Count	Proportion
1900-1920	369	3.39%
1930-1949	215	1.97%
1950-1966	270	2.48%
1967-1975	223	2.05%
1976-1982	121	1.11%
1983-1990	210	1.93%
1991-2002	1,890	16.60%
Before 1900	161	1.48%
Post 2002	3,253	29.85%
Unknown	4,265	39.14%

For the transfer learning based study, which will be presented in Chapter 6, significant data loss was experienced when matching EPCs with their respective MARVEL captured building facade. The total sample size used is: **1,547** residential properties in Barnsley. The statistics are what follows:

Numeric data

Table 3.26: Statistics of numeric data used for case study in Chapter 6

Variables	Mean	Std	CV
Total floor area	83.75	33.59	0.40
Number habitable rooms	4.69	1.87	0.40
Number heated rooms	4.64	1.83	0.39
Lighting description	0.52	0.36	0.69
Energy consumption (kWh)	1,6309	9,973	0.61

Categorical data

Table 3.27: Statistics of the categorical features extracted from the Barnsley residential EPCs.

Table 3.28: Property type

Property type	Count	Proportion
Flat	598	38.67%
House	949	61.33%

Table 3.29: Built form

Built form	Count	Proportion
Detached	660	42.67%
Enclosed End-Terrace	89	5.75%
Enclosed Mid-Terrace	25	1.62%
End-Terrace	278	18.00%
Mid-Terrace	198	12.80%
Semi-Detached	297	19.20%

Table 3.30: Floor description

Floor descriptions	Count	Proportion
Another dwelling below	423	27.34%
Solid, insulated	742	47.96%
Solid, uninsulated	32	2.07%
Suspended, insulated	258	16.67%
Suspended, uninsulated	92	5.95%

Table 3.31: Windows description

Windows descriptions	Count	Proportion
Double glazing	1,547	100%

Table 3.32: Continued Statistics of the categorical features extracted from the Barnsley residential EPCs.

Table 3.33: Walls description

Walls descriptions	Count	Proportion
Cavity wall, insulated	1,279	82.66%
Cavity wall, uninsulated	226	14.67%
Timber frame, insulated	42	2.66%

Table 3.34: Roof description

Roof descriptions	Count	Proportion
Another dwelling above	395	25.53%
Flat, insulated	12	0.78%
Pitched, insulated	1,140	73.69%

Table 3.35: Mainheat descriptions

Mainheat descriptions	Count	Proportion
Boiler	1,547	100%

Table 3.36: Construction age band

Construction age band	Count	Proportion
1930-1949	41	2.67%
1976-1982	21	1.33%
1991-2002	464	30.00%
Post 2002	980	63.33%
Unknown	41	2.67%

Properties in Doncaster

The MARVEL capture conducted in Doncaster was only in a neighbourhood, which has limited capture. Further data loss was experienced when linking different modalities. The statistics below only represent the properties examined in this thesis, not the distribution for the entire housing stocks in Doncaster. The total sample size used for the Doncaster domain is: **451** residential properties in Doncaster.

Numeric data

Table 3.37: Statistics of numeric data used for case study in Chapter 6

Variables	Mean	Std	CV
Total floor area	79.13	22.73	0.29
Number habitable rooms	4.25	1.01	0.24
Number heated rooms	4.25	1.01	0.24
Lighting description	1	0	0
Energy consumption (kWh)	8,228	4,431	0.54

Categorical data

Table 3.38: Statistics of the categorical features extracted from the Doncaster residential EPCs.

Table 3.39: Property type

Property type	Count	Proportion
Bungalow	263	58.33%
House	188	41.67%

Table 3.40: Built form

Built form	Count	Proportion
Detached	19	4.17%
End-Terrace	75	16.67%
Mid-Terrace	56	12.50%
Semi-Detached	301	66.67%

Table 3.41: Floor description

Floor descriptions	Count	Proportion
Solid, uninsulated	319	71.7%
Average U-Value 0-1.33	132	29.3%

Table 3.42: Windows description

Windows descriptions	Count	Proportion
Double glazing	38	8.33%
High performance glazing	413	91.67%

Table 3.43: Walls description

Walls descriptions	Count	Proportion
Cavity wall, insulated	198	43.90%
Solid brick, uninsulated	205	45.5%
Average U-Value 0-2.1	48	10.6%

Table 3.44: Roof description

Roof descriptions	Count	Proportion
Pitched, insulated	192	42.6%
Average U-Value 0-2.4	259	57.4%

Table 3.45: Mainheat descriptions

Mainheat descriptions	Count	Proportion
Boiler	451	100%

Table 3.46: Construction age band

Construction age band	Count	Proportion
1976-1982	207	45.90%
Before 1900	244	54.10%

Properties in Merthyr Tydfil

EPCs for Merthyr Tydfil were used in study 1 and 3. All the EPC records were used in case study 1, while, similar to other target regions, data loss due to matching modalities was experienced and resulted in a smaller sample size.

Total sample size: **27,477** residential properties in Merthyr Tydfil.

Numeric data

Table 3.47: Statistics of numeric data used for case study in Chapter 4

Variables	Mean	Std	CV
Total floor area	84.43	35.30	0.42
Number habitable rooms	4.37	1.45	0.33
Number heated rooms	4.28	1.30	0.30
Lighting description	0.52	0.34	0.65
Energy consumption (kWh)	22,835	13,844	0.61

Categorical data

Table 3.48: Statistics of the categorical features extracted from the Merthyr Tydfil residential EPCs for case study in Chapter 4.

Table 3.49: Property type

Property type	Count	Proportion
Bungalow	1,743	6.34%
Flat	2,466	8.97%
House	2,2913	83.39%
Maisonette	355	1.29%

Table 3.50: Built form

Built form	Count	Proportion
Detached	3,025	11.01%
Enclosed End-Terrace	98	0.36%
Enclosed Mid-Terrace	115	0.42%
End-Terrace	4,785	17.41%
Mid-Terrace	11,133	40.52%
Semi-Detached	8,086	29.43%
unknown	235	0.86%

Table 3.51: Floor description

Floor descriptions	Count	Proportion
Another dwelling below	1,724	6.27%
Solid, insulated	1,539	5.60%
Solid, uninsulated	21,499	78.24%
Suspended, insulated	65	0.24%
Suspended, uninsulated	1,130	4.11%
To external air, insulated	12	0.04%
To external air, uninsulated	20	0.07%
To unheated space, insulated	36	0.13%
To unheated space, uninsulated	83	0.30%
Average U-Value 0-1.33	1,357	4.96%
unknown	12	0.04%

Table 3.52: Windows description

Windows descriptions	Count	Proportion
Double glazing	25,006	91.01%
High performance glazing	1,174	4.27%
Multiple glazing	13	0.05%
Secondary glazing	34	0.12%
Single glazing	1,189	4.33%
Triple glazing	42	0.15%
unknown	19	0.07%

Table 3.53: Continued Statistics of the categorical features extracted from the Merthyr Tydfil residential EPCs for case study in Chapter 4.

Table 3.54: Walls description

Walls descriptions	Count	Proportion
Cavity wall, insulated	9,556	34.78%
Cavity wall, uninsulated	3,390	12.34%
Cob, as built	3	0.01%
Granite or whin, insulated	183	0.67%
Granite or whin, uninsulated	3,761	13.69%
Sandstone or limestone, insulated	145	0.53%
Sandstone or limestone, uninsulated	3,570	12.99%
Solid brick, insulated	160	0.58%
Solid brick, uninsulated	1,857	6.76%
System built, insulated	1,202	4.37%
System built, uninsulated	1,727	6.29%
Timber frame, insulated	399	1.45%
Timber frame, uninsulated	10	0.04%
Average U-Value 0-2.1	1,502	5.46%
unknown	12	0.04%

Table 3.55: Roof description

Roof descriptions	Count	Proportion
Another dwelling above	1,429	5.20%
Flat, insulated	141	0.51%
Flat, uninsulated	136	0.49%
Pitched, insulated	20,789	75.66%
Pitched, uninsulated	3,422	12.45%
Roof room(s), insulated	138	0.50%
Roof room(s), uninsulated	47	0.17%
Thatched	1	0.00%
Average U-Value 0-2.4	20	4.96%
unknown	12	0.04%

Table 3.56: Mainheat descriptions

Mainheat descriptions	Count	Proportion
Air source heat pump	42	0.15%
Boiler	26,374	95.99%
Community scheme	351	1.28 %
Electric heaters	361	1.31%
Ground source heat pump	10	0.04%
Room heaters	287	1.04%
Warm air	19	0.07%
Unknown	33	0.12%

Table 3.57: Construction age band

Construction age band	Count	Proportion
1900-1920	6,209	22.60%
1930-1949	1,861	6.77%
1950-1966	4,647	16.91%
1967-1975	3,359	12.22%
1976-1982	2,104	7.66%
1983-1990	1,029	3.74%
1991-2002	1,250	4.39%
Before 1900	3,599	13.10%
Post 2002	1,062	3.87%
Unknown	2,402	8.74%

For the third study in Chapter **??**, the total sample size is **1,345**.

Numeric data

Table 3.58: Statistics of numeric data used for case study in Chapter 6

Variables	Mean	Std	CV
Total floor area	91.64	11.71	0.13
Number habitable rooms	4.32	0.63	0.15
Number heated rooms	4.232	0.63	0.15
Lighting description	0.60	0.20	0.33
Energy consumption (kWh)	14,203	5,304	0.37

Categorical data

Table 3.59: Statistics of the categorical features extracted from the Merthyr Tydfil residential EPCs for case study in Chapter 6.

Table 3.60: Property type

Property type	Count	Proportion
Flat	40	2.94%
House	1305	97.06%

Table 3.61: Built form

Built form	Count	Proportion
End-Terrace	237	17.65%
Mid-Terrace	316	23.53%
Semi-Detached	792	58.82%

Table 3.62: Floor description

Floor descriptions	Count	Proportion
Solid, insulated	521	38.71%
Solid, uninsulated	695	51.65%
Suspended, uninsulated	129	9.64%

Table 3.63: Windows description

Windows descriptions	Count	Proportion
Double glazing	1,345	100%

Table 3.64: Walls description

Walls descriptions	Count	Proportion
Cavity wall, insulated	923	68.52%
Cavity wall, uninsulated	80	5.95%
System built, insulated	342	25.43%

Table 3.65: Roof description

Roof descriptions	Count	Proportion
Another dwelling above	292	21.71%
Pitched, insulated	761	56.58%
Pitched, uninsulated	292	21.71%

Table 3.66: Mainheat descriptions

Mainheat descriptions	Count	Proportion
Boiler	1,345	100%

Table 3.67: Construction age band

Construction age band	Count	Proportion
1930-1949	158	11.76%
1950-1966	119	8.82%
1991-2002	198	14.71%
Unknown	870	64.71%

3.3 General methodology

The entire framework of this thesis is developed in sequence, where each layer of the framework is one case study. At its core, the approach leverages historical energy consumption data and relevant contextual features as inputs for the machine learning model proposed in each case study. In particular, the use of automated machine learning tools and explainable AI techniques will be explained here.

3.3.1 Automated machine learning

Existing studies have been using machine learning based data-driven approaches to estimate buildings' energy performance [21, 22, 30]. However, these models were usually designed using data and algorithms chosen based on researchers' knowledge or the ones previous studies have used, which may not be suitable when local contexts changed[63, 64, 65, 66]. This is potentially one of the reasons why there is no agreed best algorithm from the selected literature summarised in Table 2.1.

This study attempted to limit the errors caused by inadequate machine learning models by using AutoML. AutoML offers a combined algorithm selection and hyperparameter optimization tool to reduce the costs of machine learning model development [70]. It takes care of raw data input from preprocessing to prediction and evaluation, provides a tool that reduces development costs, and at the same time ensures optimal estimation accuracy [71, 72]. As limited variations were found among different AutoML tools [73], this thesis chose AutoSklearn and AutoKeras because both of these tools are developed based on popular Python libraries that offer a wide coverage of pre-learned machine learning algorithms.

On the other hand, as discussed in Chapter 2, such an automated approach means that the users have limited control over the fine-tuning process, and dependent on the package selected, the pre-sets of algorithms the autoML can choose from can also be limited. Comparative studies will be conducted to assess whether the models returned by the automatic search outperform the algorithms existing studies used.

3.3.2 Evaluation metrics and explainable AI

Both manually designed models and the best architecture found by the autoML can be considered as a complete 'black box' [16, 59]. Thereby, to understand how the model is learning from the given house features, the relative feature importance and correlations between predictors and energy consumption were evaluated using statistical metrics and explainable AI approaches.

As discussed in the literature review, there is a range of popular metrics that should be chosen based on the designed purpose. In this study, the R^2 is used to measure the degree of fitness, and then further evaluated using F1 score for classification and MAPE for regression. Generally, in the field of machine learning, to conclude that a model is able to perform accurate estimations, it should achieve a R^2 over 0.7, F1 score over 0.7 and MAPE less than 0.5 [55, 56].

While the evaluation metrics and scoring are important in ensuring the robustness of the models, how each feature contributes to the final prediction is also critical for this study. The relative contributions of the input features will be computed using feature importance rank and their correlations with the prediction will be examined using the partial dependence plots. The feature importance rank can offer valuable insights into the most important housing features when estimating energy consumption. These key features are usually used in the second stage of the urban retrofit modelling, to design the potential retrofit scenarios. The partial dependence is also important, as it can provides suggestions on the possible outcomes in energy efficiency improvements when a potential retrofit measure is implemented. In the subsequent chapters, the approaches selected to provide such model explanations will be discussed in more detail.

Chapter 4

Preliminary study: residential energy consumption prediction from tabular data

4.1 Chapter introduction

Although the close link between housing characteristics and their energy consumption is widely recognised, existing data-driven housing energy analysis often concluded with machine learning models developed that are able to provide decent forecasts on energy consumption and mappings of the spatial distribution of the energy performance. But further studies on how the input housing features correlate with the energy performance are to some extent limited [21, 29, 65, 80].

This chapter attempts to expand the existing studies on statistical energy modelling to explore the correlations between housing features and their energy consumption using machine learning models. It serves as a baseline study for the entire work, and only tabular data was used in this case study. A case study was conducted for all residential housings with an EPC in Sheffield. To model the relationships between characteristics and energy consumption, housing features were extracted from Ordnance Survey Map data and EPC, details on data collection and brief descriptions can be found in Chapter 3. This study then continued to investigate the ranking of housing features in correlation to the building age and energy consumption prediction, utilising autoML and explainable approaches.

This chapter aims to address two main research questions: 1) *What type of properties may need retrofit?* 2) *What building elements or features may be prioritised to be retrofitted to improve energy efficiency?* These are answered by:

- Identifying the most important features for building age and energy consumption estimation;
- Investigating the marginal effects of most important features on building age and energy consumption to guide retrofit measure selection.

4.2 Methodology

In this study, two models were trained, and the overall workflow is presented in Figure 4.1. The first model predicts the construction age bands for properties with no age specified in the EPC using OS map data. This step ensured the data for energy consumption prediction was complete. The OS map data is linked with EPC using the UPRN reference. Before training and performing the prediction, the age bands in the EPCs were aggregated and subsampled to minimise the bias caused by unbalanced distribution. The second model then integrates the completed age data with properties' morphological and thermal characteristics extracted from the EPCs to predict energy consumption. The performance of models trained using different manually trained and fine-tuned was compared with the best algorithm found by Auto-sklearn. The trained model was further explained using the permutation feature importance and partial dependence to investigate the relationship between the housing features and energy consumption. The subsequent sections will provide detailed explanations of the individual steps in the proposed workflow.

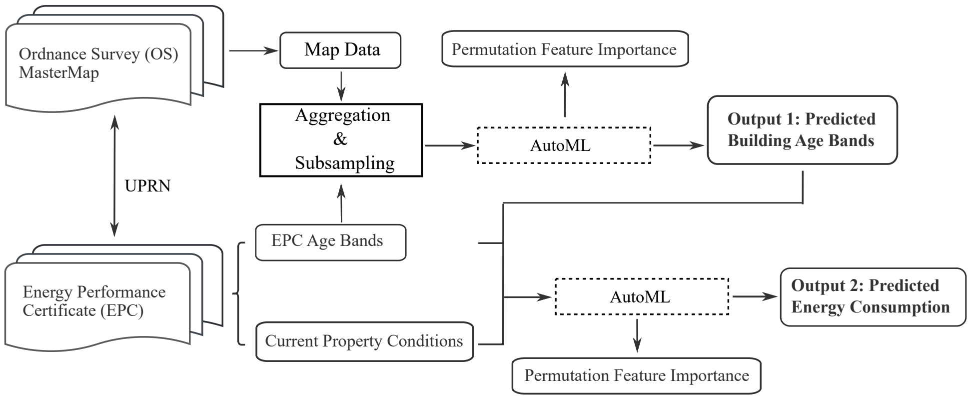


Figure 4.1: The designed workflow this study follows, including data inputs (OS and EPC), information extraction and pre-processing, model training by autoML and outputs.

4.2.1 Age bands aggregation and subsampling

The ground truth data used in training the age prediction model is variable 19 in Table 1.2, the age band recorded in the EPC. The EPC has 12 age bands in total: before 1900; 1900-1929; 1930-1949; 1950-1966; 1967-1975; 1976-1982; 1983-1990; 1991-1995; 1996-2002; 2003-2006; 2007-2011; and 2012 on-wards. These age bands are classified following the changes in regulations for building construction, which mainly are amendments for the conservation of fuels and power [44]. The way the age bands are classified suggests it may not be the best representation of how buildings' physical shapes and designs change over time. Relatively lower prediction accuracy is expected when conducting the age detection. However, this is the only open-sourced data that can be found offering adequate spatial coverage and level of detail for property age. There are other age data, such as the products from Verisk [81], which interprets building age from imagery, but classified the age in a very generic way (i.e. historic, postwar and modern).

The data also distributes unevenly across different age groups. Although the uneven distribution is a representation of the number of properties constructed in the real world, it can negatively affect the performance of machine learning models. Machine learning models usually try to maximise the prediction accuracy by assigning more weights to classes with more occurrences [82]. To reduce the bias caused by the imbalanced distribution, age bands with fewer records are aggregated into one class, and then a simple random sampling method is used to randomly select 4,000 properties from each age band for prediction. The results of the age band aggregation will be discussed in the Chapter results (see Section 4.3.1).

4.2.2 Automated machine learning: Auto-Sklearn

Auto-sklearn was selected as the automated model development tool for this case study. As discussed in the literature review chapter, little difference was found between different autoML applications.

Two models were separately trained using Auto-sklearn, a classification model for age band prediction, and a regression model for energy consumption prediction. To minimise the effects of multi-collinearity, the input data were divided into two sets based on the rules stated in Section

1.2. Building age bands were predicted primarily based on the spatial and morphological features of buildings, and energy consumption was predicted with more thermal-related features. When training, all the input data was randomly split, 80% was used for training and 20% for testing. The trained model performance on the new dataset was examined using the testing data.

The performance of all the trained algorithms was evaluated. Model accuracy score and F1-Macro score were used for the age classification model. The accuracy score calculates the proportion of predicted labels that exactly match the 'true' labels. The most optimal algorithm for age band prediction was then used to predict the construction year band for the incomplete age records. Regression models for energy consumption prediction were evaluated by R^2 and the mean absolute percentage error.

Comparison study between Auto-sklearn and traditional ML pipeline

A comparison study was conducted as a robustness test to examine whether Auto-sklearn outperforms the traditional machine learning pipeline, one algorithm selection and fine-tuning are conducted in separate steps. Similar to how Auto-sklearn behaves, the input data was preprocessed. As presented in Chapter 3, numeric data, variables 1-7 (Table 3.1), 11, 12, 16 and 19 (Table 3.3), was normalised to be unit invariant. Categorical data, variables 8-9, 13-15, 17 and 18 (Table 3.3), was processed using the one-hot encoding. This encoding process converts each class in the categorical data into separate features in a binary format. If the sample falls into this feature, then 1 is marked, otherwise 0.

A list of algorithms used by existing studies, discussed in Chapter 2 was selected: linear regression [22, 30], K-nearest neighbours [30], random forest [18, 21, 22, 30], decision tree [30] and gradient boosting [30], were tested on both age and energy consumption predictions as comparison study. F1-Macro score and R^2 score were also used for evaluating the models and comparing them with the models trained using Auto-sklearn.

As shown in Table 4.1, the traditional pipeline provided a result different from what the Auto-sklearn concluded. Among the five algorithms, random forest estimators achieved the best performance for both prediction tasks. It is also the algorithm that most of the existing studies have applied for residential building energy estimation [18, 21, 22, 30]. The resulting predictions are also less accurate than the Auto-sklearn computes.

Table 4.1: Comparison among model training scores for all predictions to check the robustness of using autoML. Different algorithms and better training accuracy were concluded by applying autoML.

		Agebands classification		Energy consumption regression	
	Algorithms	Model Score	F1-Macro	R^2	MAPE
AutoML	**Gradient Boosting**	**0.543**	**0.540**	**0.828**	**0.18**
Manual	Linear Regression			0.753	0.24
	K-Nearest Neighbours	0.412	0.583	0.758	0.19
	Decision Tree	0.445	0.901	0.554	0.23
	Random Forest	**0.468**	**0.991**	**0.776**	**0.19**
	Gradient Boosting	0.446	0.473	0.767	0.21

4.2.3 Explaining the model and feature correlations

This section offers a description of the explainable tool used in this study for text-based tabular data: permutation feature importance and partial dependence.

Permutation feature importance

Permutation feature importance (PFI) was used to rank how each variable can affect the overall model performance. The PFI is calculated by randomly shuffling or permutating each input data. The resulting prediction accuracies before and after the shuffling are calculated and compared. A larger difference in accuracy score suggests the variable is relatively more important to the model [59]. Compared with the gini feature importance used in the existing study [21], the PFI performs better in dealing with categorical variables, especially if they are processed with the one-hot encoder. For example, after the one-hot encoding procedure, the feature class 'Property type', will be expended into four separate variables: property type: bungalow, property type: flat, property type: house, and property type: maisonette. The gini feature importance can only provide individual measures on the four sub-classes; while the PFI is able to store and permute before they are processed with the one-hot encoding system. More useful hints on what input data in their original class are necessary for the predictions can be offered.

Partial dependence

To further investigate how the building features contribute to the prediction of each age band and overall energy consumption, Partial dependence (PD) are adopted. The PD calculates the average marginal effects a target feature has towards the prediction outcomes [59, 83]. For a machine learning model $F(...)$ trained with features x_i, each x produces an estimation result y_k, where $i = 1, 2, 3..., p$ and $k = 1, 2, 3, ..., N$. The output of this machine learning model can be written as $\hat{y}_k = F(x_{1,k}, x_{2,k}, ..., x_{p,k})$. The PD $\Phi(x)$ of target numerical variable x_j can be calculated using the following equations, where the predictions made by all the covariates apart from x_i are averaged, denoted as $\bar{x}_i$ [84]:

$$\Phi j(x) = \frac{1}{N}\sum_{k=1}^{N} F(x_{1,k}, ..., x, ..., x_{p,k}) \tag{4.1}$$

$$\Phi j(x) = a_j x + \frac{1}{N}\sum_{k=1}^{N}\sum_{i\neq j} a_i x_{i,k} \tag{4.2}$$

$$= a_j x + \sum_{i\neq j} a_i \bar{x}_i \tag{4.3}$$

For categorical variables, the PD replaces all the input features with the target feature and then calculates the average results [59, 83]. This value suggests, when all other elements remain similar, how the average energy consumption prediction would change relatively when the variable changes to the target feature.

4.3 Residential houses in Sheffield

4.3.1 Overview

The focused area selected for this baseline study is Sheffield, UK. Following the steps explained in the data and methodology sections, EPC records for all residential buildings in Sheffield available as of December 2021 were downloaded and preprocessed. All these records were first filtered so every property only contains the latest record. Among all EPCs downloaded, there were 23.5% of properties found to be associated with multiple records which add up to 34.3% of Sheffield EPC records. The resulting dataset comprised 142,756 homes and their associated EPC records. According to the EPC, the residential properties in Sheffield have an average energy consumption of around 274.50 kWh/m^2 per year or 22,219.42 kWh per year, if the footprint area for each property recorded in the EPC is used for calculation.

As illustrated in Figure 4.2, before aggregation, the original records from EPCs show that most of the residential buildings in Sheffield were developed between 1900 and 1966, and few were built after 2012. There are also 10,392 (7.3%) of properties' construction age remains unknown. Without pre-processing, this uneven distribution will lead to a biased model. Based on the number of properties each age band contains, the age band '1991-1995' and '1996-2002' were combined into the new class '1991-2002'; '2002-2006', '2007-2011' and '2012 on-wards' were aggregated into the new class 'post-2002'. The aggregation process ensured all age bands had enough data to follow the sampling process for model training.

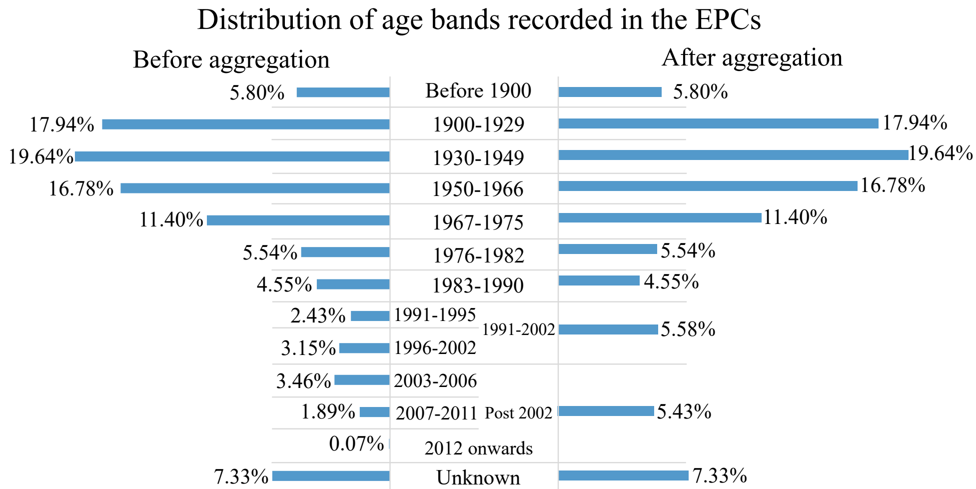


Figure 4.2: Distribution of construction age recorded in the EPCs before (left) and after aggregation (right)

A summary of the basic statistics of the data is presented in Chapter 3. For numeric data, statistics for the data and their subsets used in the predictions are listed, including their mean std and CV. The coefficient of variance is calculated as the ratio between the std and the mean $\frac{std}{mean}$. The last four variables in Table 3.4 are only used for energy prediction so no subsamples were generated.

Among all the numerical data used in this study, it is not surprising to find that, except for the built rate, all the variables have CV less than 1. As more than 70% of residential properties in Sheffield are houses, they tend to have relatively similar physical features, the same as the example map illustrated in Figure 3.1. The only variable that has a CV larger than 1 is the built rate, this is also common because properties in the more rural areas of the city are less densely built than neighbourhoods around the city centre. By comparison, the subsets generated using the sampling method can to some extent be considered representative of all the data collected, as there is no significant difference between the statistics of original and subsampled data.

4.3.2 Results of model training and prediction

Age detection

The age detection model was firstly trained on the processed dataset. The auto-Sklearn detected 37 algorithms that might be optimal for predicting building age bands. The most optimal model used a gradient boosting algorithm, which trains the model by sequentially adding input variables to the ensemble of decision trees and refitting the model based on the errors made by the previously added inputs [50].

For testing data, the most optimal model Auto-Sklearn trained achieved an accuracy score of 0.543 and an F1-Macro score of 0.540. The model performance was further evaluated by comparing the predicted age bands for the test data with their true class in EPC records, Figure 4.3. Although the majority of the age bands were correctly predicted, especially for the aggregated age bands, as expected, a few remain mispredicted. Apart from the reason that the age bands were originally grouped to reflect the change in energy regulations instead of design and appearance, another potential reason for this misprediction might be because developers tend to design houses that fit into the general building styles nearby which may not be recently constructed.

	Age Band	Predicted								
		Before 1900	1900-1929	1930-1949	1950-1966	1967-1975	1976-1982	1983-1990	1991-2002	Post 2002
Labelled	Before 1900	38.00%	32.50%	5.70%	4.70%	3.40%	2.90%	3.20%	3.90%	5.70%
	1900-1929	24.60%	49.80%	9.50%	3.70%	2.60%	3.20%	1.70%	1.70%	3.20%
	1930-1949	2.60%	3.30%	58.30%	18.00%	2.60%	3.90%	5.20%	5.50%	0.50%
	1950-1966	2.60%	2.10%	29.30%	39.90%	11.70%	6.60%	4.20%	2.40%	1.20%
	1967-1975	2.50%	2.10%	10.20%	13.60%	51.10%	10.60%	6.40%	3.30%	0.10%
	1976-1982	2.10%	1.70%	9.30%	5.90%	9.80%	50.90%	14.90%	3.90%	1.60%
	1983-1990	3.10%	3.00%	8.20%	4.80%	4.40%	9.30%	56.00%	10.10%	1.10%
	1991-2002	3.50%	3.50%	4.30%	2.70%	3.50%	4.10%	11.10%	55.00%	12.40%
	Post 2002	1.00%	0.60%	0.00%	0.50%	0.60%	0.40%	0.70%	6.10%	90.10%

Figure 4.3: Heatmap table showing the resulted between the true (columns) and predicted age bands (rows) using the random forest classification.

Energy consumption prediction

The energy consumption prediction was then conducted after age bands were classified for each property. The age prediction results from the first model were used to train the model. Auto-sklearn determined the best-performing algorithm using **data preprocessors** based on feature type, **feature agglomeration** as feature processors and **gradient boosting** as the regressor. The data preprocessors selected are the same as what were used for the traditional ML in the comparison study: one-hot encoding for categorical data and normalisation for numeric data. The feature agglomeration process clusters highly correlated features together during the feature process phase [85].

The trained model achieved a R^2 score of 0.828, and a MAPE of 18.1%. The results suggest that overall, around 82.8% of the test data can be explained by the trained algorithm; and the prediction results based on the test data have an average difference of 18.1% compared with the ground truth.

4.3.3 Feature importance

The PFI plotted in Figure 4.4 ranked how important each input feature is in both models towards the prediction. The x-axis is plotted in its log form, to offer clearer visualisation for variables with less feature importance.

The features used for the age prediction model are ranked in Figure 4.4a, detailed description was presented in Table 3.1 and Table 3.3. The importance rank suggested that, the built-up rate is the most important feature when predicting the age bands of residential buildings in Sheffield, floor area and property types are also relatively important. Excluding the variable builtrate caused a 23.9% decrease in model accuracy score, and a 25.6% decrease in F1-Macro score.

The NPI and the number of vertices are found relatively less important. As the example properties illustrated in Figure 3.1, when predicting the age of residential buildings, buildings tend to have little difference in shapes and thereby less sparsity in values can be found. Excluding NPI and the number of vertices only caused a decrease in accuracy score and F1-Macro by 0.37% and 0.56% respectively. In overall, when data availability is limited, the age band of the housing can be estimated by understanding the housing size, the building type, and how densely the postcode area is developed.

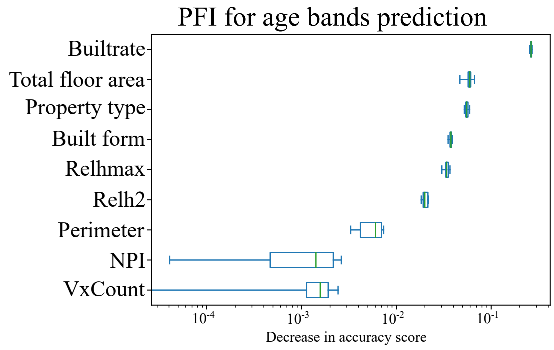
Figure 4.4b ranked how the input data affect the model performance when estimating energy consumption for Sheffield. The total floor area is the dominating feature in this estimation, followed by building materials, which is also the most common retrofit target. Excluding total floor area from model training led to a 15.3% decrease in R^2 score and a 26.0% increase in MAPE value.

On the other hand, the type of property and number of habitable rooms are the least important in estimating housing energy consumption, excluding these features only resulted in a 2.80% decrease in R^2 score and a 3.26% increase in MAPE. The model also suggested a different result to ONS's conclusion. The house age bands ranked seventh among all features, which indicates that it has relatively less impact on energy consumption prediction.

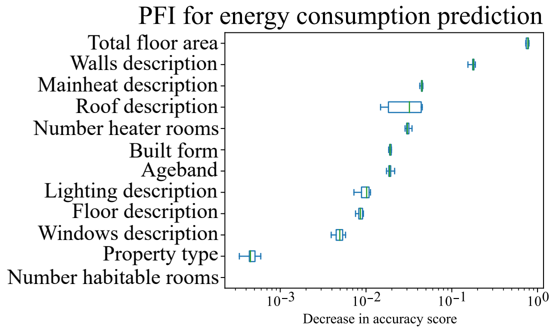
4.3.4 Partial dependence

Age detection

According to the feature importance calculated in the last section, the built rate is the key feature when estimating housing age. Figure 4.5 illustrates the complex relationships between the built rate and each age band in Sheffield. The trendlines showing positive correlations are



(a) PFI for age detection



(b) PFI for energy detection

Figure 4.4: PFI for variables used in the two machine learning models, x axis in log form. (a) is for age detection and (b) is for energy consumption prediction

highlighted while negative correlations are light-coloured. In general, postcode areas with a built rate less than 30% tend to have a combination of houses built in different eras. If the area has a built rate higher than 30%, houses in the area are more likely to be built before 1900 or between 1950 and 1966.

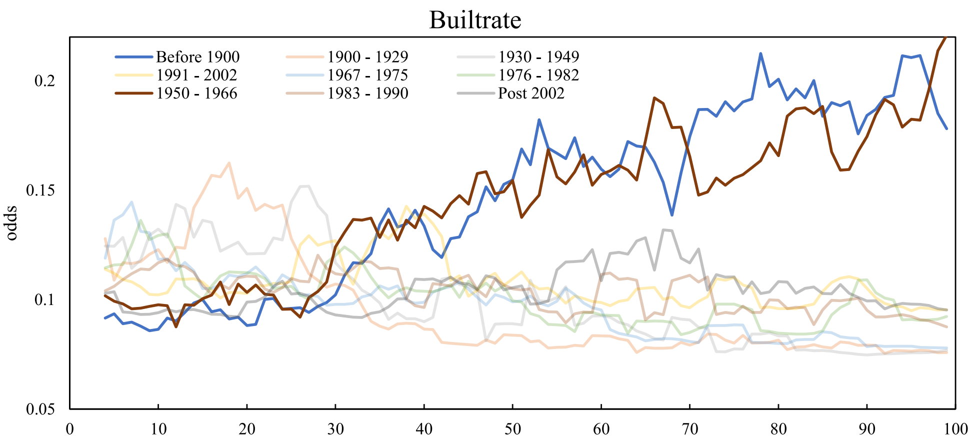


Figure 4.5: Partial dependence plot of builtrate (x-axis) against possibility of being built in the target ageband (y-axis).

Energy consumption prediction

The partial dependence plots in Figure 4.6 show how each building feature affects the energy consumption. The charts are ordered according to their ranked permutation feature importance. For features referring to the building fabric: walls, floors, and roofs, separate charts are produced to indicate whether insulation is installed and the corresponding energy consumption variations.

The dominant feature ranked by PFI in Figure 4.4b is the total floor area of the house. A positive linear relationship can be found between house sizes and energy consumption, as shown in Chart 4.6a. In general, larger houses in Sheffield usually have higher energy consumption. A similar correlation can be found for the number of habitable rooms in chart 4.6o, as it is also an indicator of how large the property is, its importance is weakened in the overall energy prediction and ranked lowest in the calculated PFI. On the other hand, the number of heated rooms in chart 4.6g is less obvious. The energy consumption firstly slightly decreases when more rooms can be heated in the property, then stays relatively stable. As this number only refers to the number of rooms with heating facilities, it does not necessarily suggest the number of heatings in use, thereby the energy demands do not change significantly.

The building fabric is the next key feature in estimating housing energy needs. How different types and conditions of housing material may affect the housing energy needs are intensively researched [86].

Charts 4.6b, 4.6c, 4.6e, 4.6f, and 4.6j to 4.6m provided a comparison to show how different material used for each building element may affect the energy consumption in Sheffield. These correlations may be referred to answer the following questions to determine what retrofit measures should be applied to reduce energy consumption: 1) whether the type of material can be or should be changed, 2) if the material used stay unaltered, whether insulation should be installed, and 3) if the material should be and can be changed, what material and insulation may be a more energy efficient options. The correlations of uninsulated building material against energy consumption are placed on the left of the page and insulated ones on the right. Together these charts suggest that, in general, with insulation in place, a large drop in energy consumption can be expected.

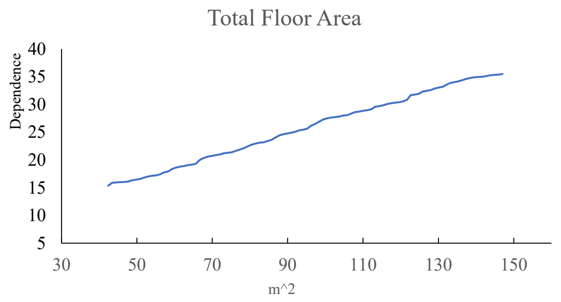
Closer inspection of Figure 4.6b and Figure 4.6c shows that, in general, insulated walls perform significantly better in energy saving than uninsulated ones. Before insulation, houses with walls built in timber frame, granite or whin or cob may have less energy consumption. Although all uninsulated walls should be upgraded to reduce heat loss, among the seven types of material, walls built in sandstone or limestone and solid brick may be prioritised for retrofit. After insulation, houses built with cavity walls and sandstone or limestone walls may witness a relatively significant reduction in energy consumption. This mapped correlation to some extent agrees with what the home upgrade grant primarily focused on retrofitting, where improving the insulation conditions of solid walls and cavity walls has taken a large proportion of the funded project.

The overall energy consumption can also be reduced significantly by applying insulations to the roof and floor. Among the five categories in Figure 4.6e, pitched roofs tend to have higher energy needs. After insulation, different roof types are likely to perform similarly. For the floor, if the floor is connected to unheated space or suspended should be prioritised for upgrading and retrofitting.

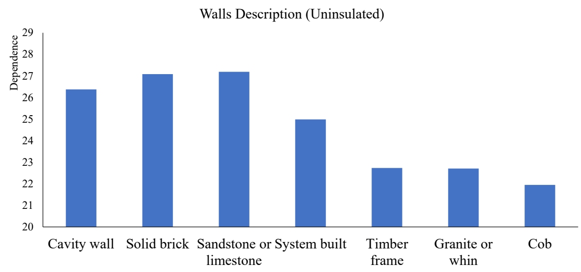
The energy consumption of residential buildings in Sheffield also has a positive relationship with the proportion of low-energy lights and energy-effective windows. In general, more low-energy lights installed, and better window material (e.g. double and more glazing) used means less energy consumption.

How different heat sources may affect the energy consumption are compared in Figure 4.6d. According to the chart, houses that use electric heaters tend to have the highest energy consumption in Sheffield, while houses with heat pumps use less energy.

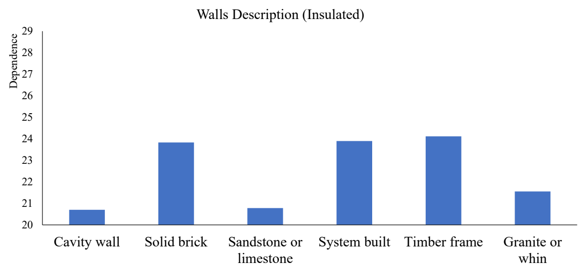
Charts for the remaining housing features: the built form in Figure 4.6h, the age band in Figure 4.6i and the property type in Figure 4.6n, may provide preliminary guidance on the potential housing that should be targeted for home upgrade projects. Together with the relationship shown in Figure 4.6a, the model suggests that larger and older detached houses in Sheffield may be prioritised for retrofit.



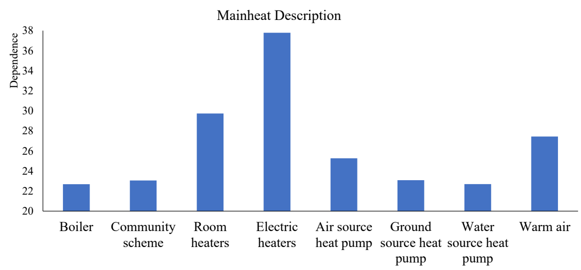
(a) PDP for total floor area



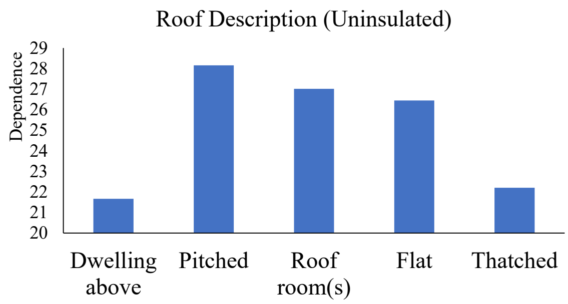
(b) PDP for uninsulated walls*



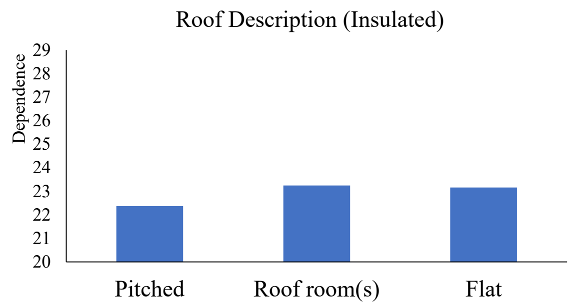
(c) PDP for insulated walls*



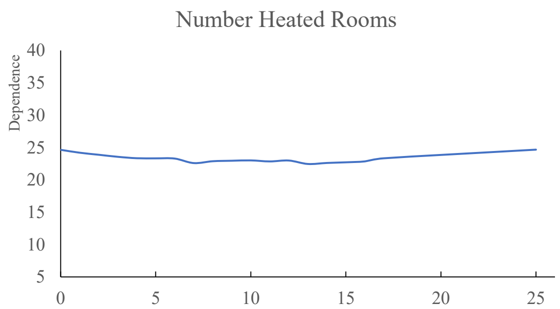
(d) PDP for main type of heating option used



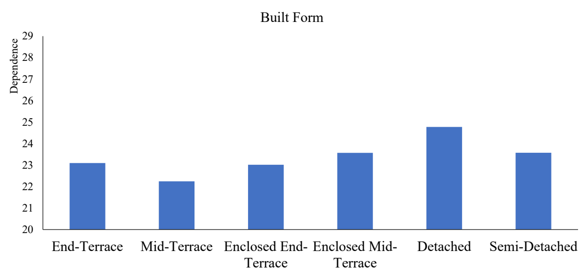
(e) PDP for uninsulated roof



(f) PDP for insulated roof

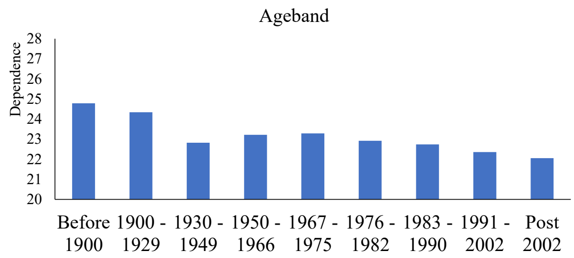


(g) PDP for number of heated rooms

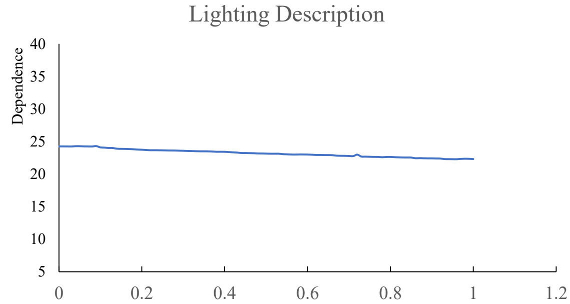


(h) PDP for the type of built form

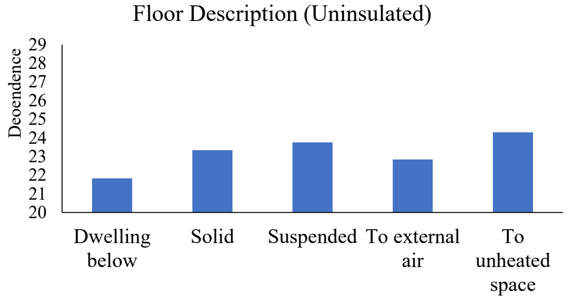
Figure 4.6: PDP for the marginal effects of building features (the x-axis) towards residential energy consumption in kWh (the y-axis) in Sheffield. (*In EPC, the system built wall is treated as masonry [12].)



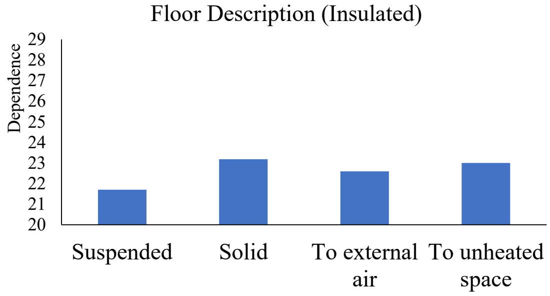
(i) PDP for age bands



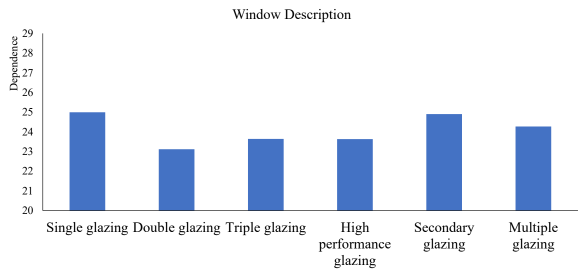
(j) PDP for percentage of low energy lighting installed



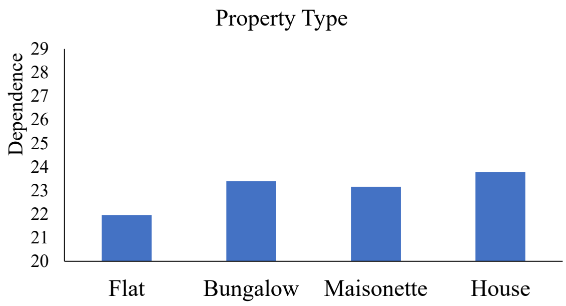
(k) PDP for uninsulated floor



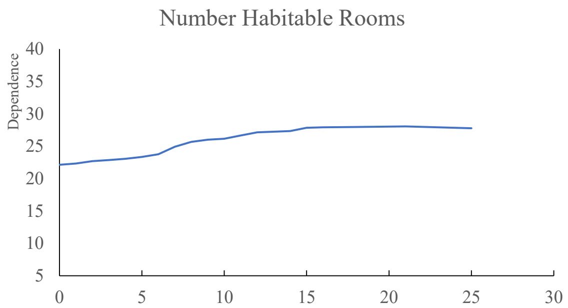
(l) PDP for insulated floor



(m) PDP for window type



(n) PDP for property type



(o) PDP for number of habitable rooms

Figure 4.6: (Continued) PDP for the marginal effects of building features (the x-axis) towards residential energy consumption in kWh (the y-axis) in Sheffield.

4.4 Testing the model adaptability: application in other regions

The methodology developed in this case study for Sheffield was applied to other regions to assess the model's adaptability. The performance of the trained models in predicting the energy consumption of residential housing stock in Barnsley, England and Merthyr Tydfil, Wales were assessed. The following results represent several experimental tests conducted with the trained model.

4.4.1 Housing stock overview in Barnsley and Merthyr Tydfil

These two cities, Barnsley and Merthyr Tydfil respectively represent a city with similar housing characteristics and spatially close to Sheffield, and one city with different housing characteristics and geographically far away from Sheffield. As illustrated in Figure 4.7, Merthyr Tydfil are geographically far away from Sheffield. Based on the aforementioned general patterns for housing stocks, properties in neighbouring regions mean similar energy performance, and regions further away mean they are less likely to have similar energy performance.

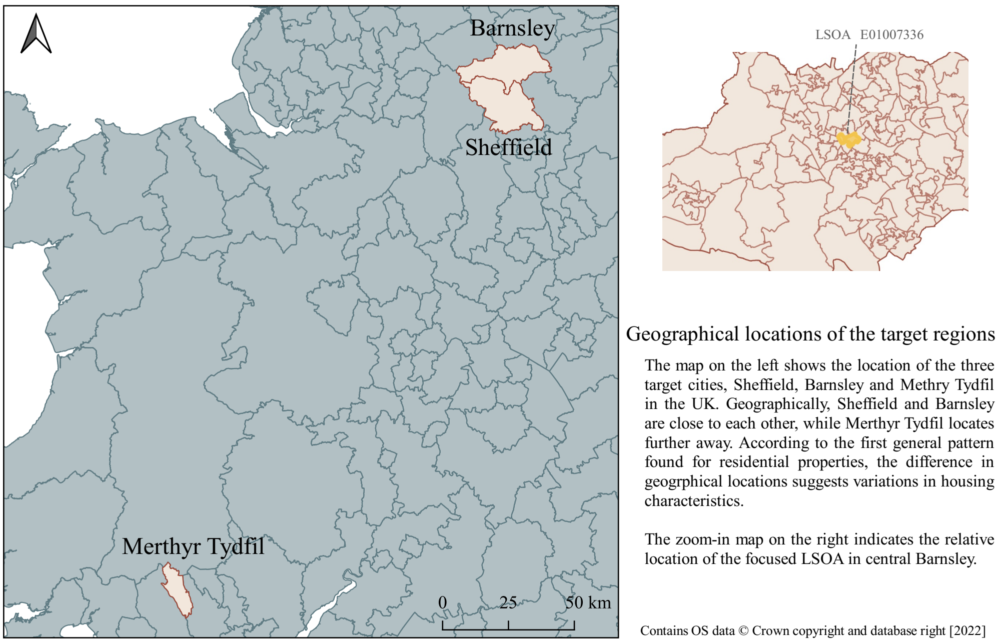


Figure 4.7: Illustrative map showing where the three cities are in the UK and the focused LSOA are in Barnsley.

The Table 4.2 presents an overview and comparison for the properties in the three cities based on their EPCs records. The table presents both categorical and numeric features, showing the most frequently observed classes for categorical features and average values for numeric features. More detailed statistics were provided in Chapter 3. Both similarities and variations exhibits among the three cities. All three cities have the same average number of habitable and heated rooms, a common window and roof type, similar heating options. But the most common built form, conditions of floor, roof and year of construction are different among the three cities. It is difficult to determine the degree of difference among the cities solely based on these tabular data.

Table 4.2: Housing feature comparison among the three cities. The table indicates the most frequent feature class if the feature is categorical, and an average value if the feature is numeric.

Features	Sheffield	Barnsley	Merthyr Tydfil
Property type	House	House	House
Built form	Semi-detached	Detached	Mid-terrace
Total floor area	81.4	88.8	84.4
Habitable rooms	4	4	4
Heated rooms	4	4	4
Floor	Suspended, uninsulated	Solid, insulated	Solid, uninsulated
Window	Double glazing	Double glazing	Double glazing
Wall	Solid brick, uninsulated	Cavity, insulated	Cavity, insulated
Roof	Pitched, insulated	Pitched, insulated	Pitched, insulated
Efficient Lighting	52.8%	67.5%	51.9%
Heating	Boiler	Boiler	Boiler
Ageband	1900-1929	1991-2002	Before 1900

4.4.2 The adaptability of the trained model

The first attempt is to directly apply the models trained in this case study using Sheffield OS data and EPCs to both cities. This attempt failed due to the fact that, the nature of traditional machine learning requires feature attributes for predictions to be exactly the same as what the model was originally trained on [63]. However, this is not usually the case in the real world, Barnsley and Merthyr Tydfil have different housing features compared with Sheffield. For instance, there are properties in both cities that were recorded as built with 'park home wall', and one property is heated using 'solid fuel' in Barnsley. Although it is to some extent feasible to recategorise 'solid fuel' as wood for prediction, the 'park home wall', a specific type of exterior wall for residential caravan or motor home, usually made from plasterboard [87], remains a feature that the Sheffield model has never seen before.

The second attempt is to exclude these unseen features in both regions so the Sheffield model can be directly used. When predicting for Barnsley, the model resulted in an R^2 of **0.78**, and a **MAPE** score of **24.8%**. For Merthyr Tydfil, the model performance decreased to an R^2 of **0.518** and a **MAPE** score of **22.6%**. As discussed in Chapter 3, a model with a R^2 score over 0.7 would usually be considered as good. The decrease in R^2 suggests the model trained in Sheffield is less suitable to predict the energy consumption for properties in Merthyr Tydfil, as expected. The deficiency in model performance to some extent suggests that energy modelling at a national level may not be as accurate as ones modelled respectively for the city, as the feature distribution may be skewed to suit the vast majority. It is also the local authorities, who play the critical role in identifying the worst performing homes and residents at risk of fuel poverty [9].

In collaboration with Barnsley Council, a third experiment was conducted with a smaller amount of data. Since the model trained with Sheffield is not prone to unseen contexts, a new energy consumption estimation model is built following the same methodology. 687 properties in one of the LOSA (ref no.E01007336), as shown in the following Map 4.7, in central Barnsley were selected. The morphological and thermal features for these properties were obtained to train the model tailored for this region. The Barnsley model predicts building energy consumption with a MAPE of 37.59% compared with EPC recorded energy consumption. The fact that the Barnsley model performed poorer than the model for Sheffield may be because the fact less data were available for training, which emphasised the importance of the quantity of the data input in model training.

4.5 Chapter conclusion

Spatial, morphological and thermal characteristics of residential houses contribute to housing age and energy consumption prediction in various degrees. The feature importance rank and PDP plots also offered valuable insights into the relationships between housing elements and energy performance for properties in Sheffield. The results from the autoML and explainable approach applied suggested that, the size of the property, the material and insulation conditions of the walls are key features affecting buildings' energy consumption. Agebands, on the other hand, ranked 7th in all the housing features included in the energy consumption prediction. Together these plots suggest that, energy savings may be largely made by retrofitting relatively older (built before 1930) and larger detached houses. For houses of similar sizes and ages, improving the insulation conditions of the building fabric will lead to the most significant improvement in residential energy efficiency.

4.5.1 Limitations

Although existing studies have reviewed the EPC as neither reliable nor accurate, it is the only comprehensive data for all residential buildings available at the national level [46, 88]. There are existing studies that introduced potential alternative data sources to describe the building's thermal and physical conditions. For instance, photos of the target properties and scanned LiDAR 3D models. Further to this, in other fields of interest (e.g. medical diagnosis), multi-modal deep learning that combines predictors from a variety of data sources in different formats is widely adopted as a solution to overcome the limitations caused by using only one source of input. This leads to the first research question: *Can the multimodal approach enhance the reliability of the estimation?* The following studies Chapter 5 and 6 aim to answer this question by implementing a multimodal approach for energy prediction. Explainable AI will also be implemented to explore whether the key features identified may change when other sources of data are included in the machine learning models.

The limitations of adaptability of such machine learning methods are presented by applications in Barnsley and Merthyr Tydfil. The direct application of such a model is unsuccessful due to unseen variables in the target cities, or in other words, different data distributions. After removing these variables prohibits the direct application, the resulting prediction suggests less suitability for properties in Merthyr Tydfil than Barnsley. This is not a surprising result, since Merthyr Tydfil is further away from Sheffield and the properties there are expected to have vast differences with properties in Barnsley and Sheffield. The section also emphasised the importance of the quantity of inputs when building a machine learning model for accurate estimation results.

Improvement of the model adaptability to different spatial contexts is needed, which refers to the second research question of this thesis: *Can transfer learning assist in improving model performance for regions with poorer or fewer data?*. Chapter 6 attempts to address this question. Instead of adding more variables into the model for training, Chapter 6 adapts the recent advances in machine learning, the transfer learning. As discussed in the literature review, the transfer learning approach understands the target task by knowledge learned from other tasks and domains [63, 89], so it has advantages in helping to build models when the target region has a different variable distribution and is limited in data quality and quantity.

Chapter 5

Multimodal deep learning energy prediction from EPC and Google Streetview

5.1 Chapter introduction

In the last chapter, EPC and map data are employed, and demonstrated the capability of these tabular data in accurately estimating energy usage. However, the limitations of tabular data are also discussed, especially with EPC, which can be inaccurate or out-of-date.

The literature review has demonstrated the use of various data types in existing studies for energy estimation. This chapter aims to explore whether the use of image data as visual descriptions of the target property can be an alternative to the tabular data used in the last case study. Several literature have demonstrated the capability of such data in estimating the EPC ratings [29, 31]. Example image data sources are from real estate reports, the TABULA database and Google Street View.

Considering the volume of data required for training a data-driven model, the Google Street View (GSV) images are selected. Although the capturing and updating schedule of the GSV database varies across regions and sometimes varies at the neighbourhood level, it may still be considered as providing more up-to-date information for the enquired property compared to the EPCs. As one EPC is usually valid for 10 years unless the homeowners decide or are required to obtain a new one to reflect the changes in their homes, whereas an image can be captured at convenience.

Despite the successful case studies presented in existing literature [29], the performance of the model trained and validated solely on images is not sufficient for accurate energy estimation. Therefore, the concept of deep multimodal learning is introduced, aiming to reduce the biases resulting from mono-modality. The concept behind multimodal learning is to leverage heterogeneous features from multiple data inputs to mitigate biases arising from mono-modality. As discussed in Chapter 2, the use of multimodal learning for deep learning tasks has been gaining attention in various domains, particularly in areas such as expression recognition and medical diagnosis [68, 69].

This case study considers the images may provide more up-to-date information about the properties that are unknown to the tabular data EPC, enabling cross-validation between the two modalities. Similar to the approach employed by Sun et al. [31], a common practice in multimodal learning is to use a Multilayer Perceptron (MLP) for tabular data and a Convolutional Neural Network (CNN) for image data. In this chapter, the study is extended to test the capability of the automatic NAS tool, AutoKeras, to explore and compare different architectures employed in existing studies and those developed through NAS. The ultimate goal of this study is to answer: **Is image data alone sufficient for energy estimation? If not, does incorporating multiple modalities improve the prediction accuracy?** The main objectives of this Chapter are as follows:

1. To investigate whether employing a Neural Architecture Search (NAS) tool may yield different network structures compared to existing studies and if the resulting algorithm exhibits improved performance.

2. To examine whether multimodal learning may improve the prediction performance.

3. To examine whether incorporating multiple modalities may lead to changes to the relative importance of housing features.

5.2 Data and methodology

This study was conducted as a comparative analysis. The first comparison was drawn between machine learning models developed following a traditional approach or using an automatic neural network search tool. The model performance was evaluated using statistical metrics R^2 and MAPE. The evaluation results can provide insights into whether GSV itself is sufficient to provide a robust energy estimation. The second comparison was conducted to examine the resulting changes in model prediction by introducing multiple sources of data. The best-performing algorithms from the first comparison were used to build the multimodal network for this second comparison. The multimodal network incorporates both textual descriptions and visual representations of residential housing characteristics, as depicted in Figure 5.1. The two modalities were linked using a shared geospatial reference, and were processed with their individual stream of pre-processing steps to produce tabular and image inputs respectively. After being pre-processed, each modality was trained with the best algorithm found in the first comparison study respective to their data type, before being merged together for the final prediction.

The subsequent subsections will provide a comprehensive overview of the collection process for Google Street View images, as well as the pre-processing and modelling approaches employed for both data modalities. Details regarding the general data collection and pre-processing steps can be found in Chapter 3.

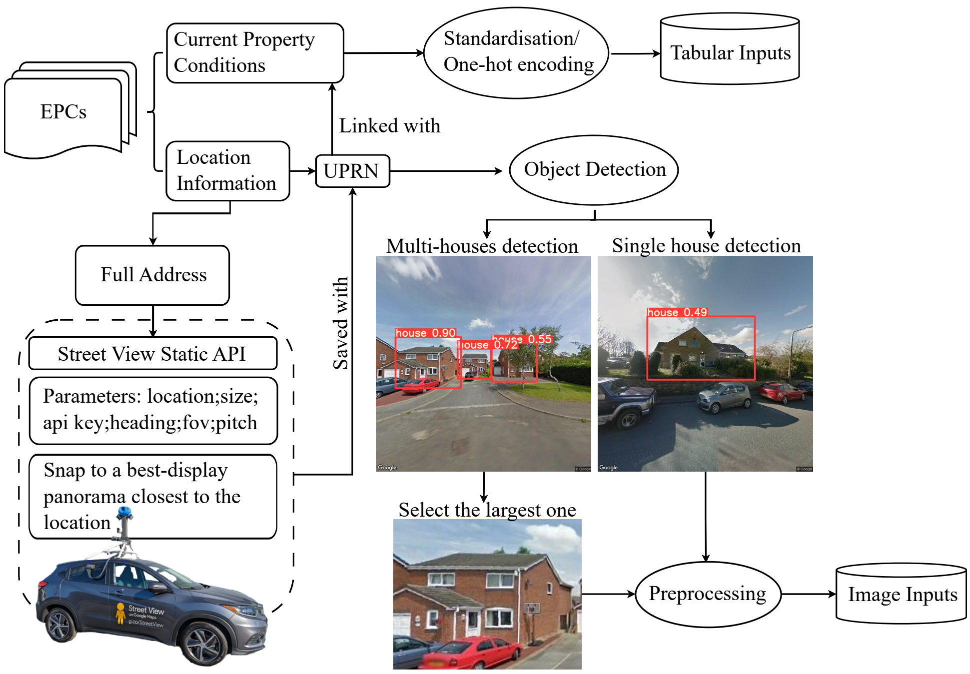


Figure 5.1: A flow chart demonstrating the data collection and feature preprocess for the multimodal deep learning. In this case study, EPC and GSV are collected and saved with UPRN so they can be matched. All data went through the preprocess according to their data type.

5.2.1 Google Streetview images through Street View Static API

The Google Street View API allows users to extract panoramic images according to specified parameters. The primary search for GSVs is based on the location parameter. There are two methods to query the images for a specific property: using geographical latitude and longitude values or using a postal address down to the house number level. When an address text string is used as a location reference, the API automatically selects the closest panorama that provides the best display of the specified location. If coordinates are provided, the API conducts a 50-meter radius search. In this study, the full postal addresses recorded in the EPC were used as a reference for the target properties in the image search.

When the location is defined, the next step is to define image-related parameters, such as the output image size, and the heading of the vehicle when capturing. When no heading is specified, the API automatically calculates the best viewing angles relative to the optimal viewing location. Therefore, the heading and view angle calculation procedure used in the study by Sun et al. [31] is no longer required in our case.

These collected street view images are saved with the UPRN references, to match with their corresponding EPC records to facilitate the following deep multimodal learning.

5.2.2 Model development

Two sets of comparative studies were conducted, to evaluate and validate the robustness of different approaches.

The first comparative study compares the performance of models manually developed, which are selected based on common algorithms used in existing literature, versus the models developed using the automatic NAS approach. Existing literature has deployed Convolutional Neural Network (CNN) or CNN-based architectures, ResNet for image data and Multi-Layer Perceptron (MLP) for the tabular data [29, 31]. To examine whether these models are the most efficient for the designed purpose, and if not, to find a better-performing architecture, AutoKeras (AK) was used to perform the automatic neural network search. AK is a tool developed based on Keras and TensorFlow, which are fundamental deep learning libraries [90], and the tool is employed due to its capabilities and familiarity. Similar to the NAS framework discussed in Chapter 2, AK performs the architecture search in three main steps. *Data preparation*, where AK analyse the raw data input to determine specific preprocessing tools for implementation. *Search space*, where a series of suitable neural architecture and hyper-parameters are tested. And *Optimisation*, where the best-performing settings are evaluated and fine-tuned. The optimisation method used in AK can be considered as reinforcement learning, the network search always mutates the prior best-performing configuration for the next search [90]. According to the type of data input, the AK image regressor and AK Structured data regressor were deployed. The comparison was made by evaluating and comparing the effectiveness of such a model in providing an accurate understanding of buildings operational energy performance, using the estimated energy consumption data provided by EPC, through two metrics: the goodness of fit to the data inputs R^2, and the error rate between prediction and actual MAPE.

Such evaluation is then further used to advise whether using images only is sufficient for the estimation, which leads to the second comparison: between models developed sorely on a single modality and models built using two modalities. The multimodality model was developed using the best architecture concluded from the first comparison. Figure 5.2 illustrates the workflow of a multimodal network, where a concatenate layer is added after each stream of data is processed and computed, to merge both modalities to produce the final outputs.

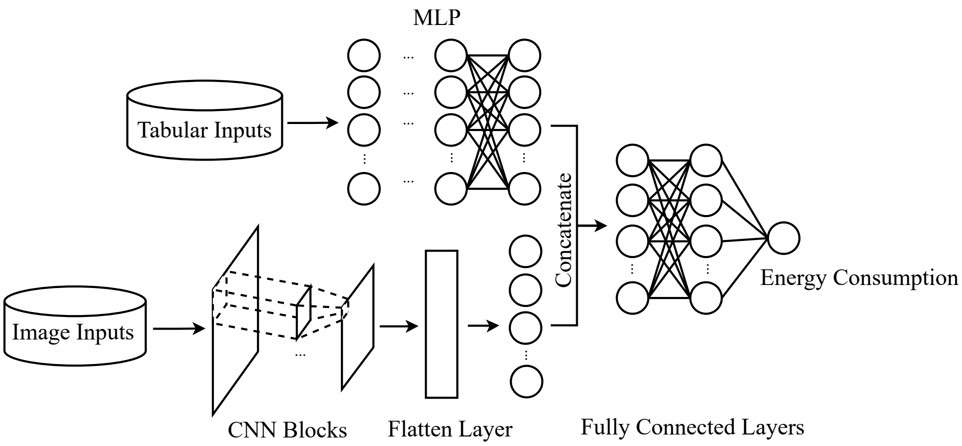


Figure 5.2: A flow chart demonstrating the data collection and feature preprocess for the multimodal deep learning. In this case study, EPC and GSV are collected and saved with UPRN so they can be matched with. All data went through the preprocess according to their data type.

5.2.3 Explainable AI: SHAP

In this case study, the SHapley Additive exPlanations (SHAP) is employed to explain how the inputs contribute to the network prediction. As introduced in Chapter 2, popular explainable AI tools are LIME and SHAP. LIME computes a local interpretable model around individual predictions, while SHAP measures the contribution in a more globally coherent manner. This thesis is more interested in the input global contributions to the network, therefore, the SHAP is employed.

SHAP is developed by Lundberg and Lee [62] based on the concept of cooperative game theory and assigns a value to each feature in a prediction based on its contribution to the prediction's outcome [59, 61, 62]. As discussed in the literature review, the shapley values or the SHAP values are calculated using Equation 2.10. As the values represent the average marginal contribution of each feature to all possible coalitions of features, it serves a similar explanation role as the PDP method used in the work presented in Chapter 4. Positive SHAP values indicate that the inclusion of the feature positively contributes to increasing the prediction value, while negative values indicate a negative contribution. To distinguish with the method used in Chapter 4, the SHAP computed partial dependence will be named as 'SHAP PD' for what is followed. These contributions of each feature towards one prediction results are also an indicator of their feature importance, this is considered as a local feature importance rank. Averaging the absolute SHAP values (mean|SHAP values|) for all the features used can thereby produce a global feature importance rank for the entire model.

$$\text{SHAP values} \quad \phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(M - |S| - 1)!}{M!} \left[f_x\left(S \cup \{i\} - f_x\left(S\right)\right)\right]$$

5.3 Investigation of the application of the deep multimodal network

5.3.1 Overview

Due to data availability, this case study was conducted for residential properties in Barnsley, UK. A total number of 13,384 EPC records were downloadable for Barnsley up till 31 January 2023. However, more than 13% of properties are found to be associated with multiple entries, so filtering and reorganising the data is essential. Some records were referenced with the same value of UPRN, which are considered problematic and emitted from the analysis.

The overall statistics for the EPC data used can be found in Table 3.14 and Table 3.15 in Chapter 3. Among these properties, approximately 72.9% are houses (H), 20.4% are flats (F), 6.1% are bungalows (B), and the remaining 0.7% are maisonette (M). The mean estimated energy consumption for these properties was found to be 180.05 kWh/m^2 per year. When considering the household area, the average energy consumption amounted to approximately 15,214 kWh per year. Overall, the average energy usage in Barnsley is lower than the national average. Some extreme cases were observed where recorded energy consumption exceeded 1,000 kWh/m^2 per year, but no clear associations with specific housing characteristics could be identified for these extreme consumption levels.

Table 5.1 provides a summary of the main building elements for each property type. There are similarities across different building types, particularly in terms of heating systems and fuel sources. This pattern aligns with the prevailing trend in existing properties in England, as highlighted in a recent report published by DESNZ and BEIS [5], gas is and has remained the predominant source of heating since around 1990. The main differences among building types in Barnsley are observed in elements related to property structures, specifically the floors and roofs.

Table 5.1: Most frequent building features in EPC for properties in central Barnsley

Type	Floor	Window	Wall	Roof	Heat
B	Suspended, uninsulated	Double	Cavity, insulated	Pitched, insulated	Boiler
F	Dwelling below	Double	Cavity, insulated	Dwelling above	Boiler
H	Solid, insulated	Double	Cavity, insulated	Pitched, insulated	Boiler
M	Dwelling below	Double	Cavity, insulated	Pitched, insulated	Boiler

5.3.2 Streetview image and object detection

After data collection and preparation, the input data for this case study consists of 9,050 Google Street View (GSV) images and their corresponding EPC records. Out of the recorded

(a) (b)

Figure 5.3: Collage of example GSV downloaded and preprocessed. The largest crop in one GSV is selected as the target image representative. To ensure the input images are all in one size, paddings are added and resized. (a) Selected examples of GSVs of properties in Barnsley. (b) The images after detection, crop, resizing and sky removal.

addresses in the EPCs, 2,074 were unable to match with a GSV image and were excluded from further analysis. These images were processed into raw data input by using object detection, crop, padding and resizing. A collage of selected street view images and houses detected and preprocessed are shown in Figure 5.3. As illustrated. although a large amount of irrelevant visual features are filtered during the object detection stage, there are still some remains close to the property itself, e.g. vehicle park in the front, greenery and pavements. An XAI is important in this case, to evaluate whether the algorithm has used reasonable visual features in estimation.

5.3.3 Results: energy prediction and key feature identification

Following the methodology designed, six different models were developed respectively. Table 5.2 presents the evaluation results of the models developed.

Table 5.2: Model inputs, structure and evaluation results of all deep learning models developed in this case study, using models built either in a traditional or automatic manner.

Model	Algorithm	Input data	R^2	MAPE
1	CNN	Image	0.76	1.73
2	RestNet	Image	0.84	1.15
3	MLP	Tabular	0.90	0.86
4	AK image regressor	Image	0.79	1.74
5	AK structured regressor	Tabular	0.67	1.72
6	**Multimodal MLP+ResNet**	**Tabular & Image**	**0.97**	**0.43**

Comparison between conventional and automatic model development approach

The first comparison can be drawn by comparing Models 1,2,3 and Models 4,5. AK image regressor returned a CNN model in the form of two CNN blocks, and AK structured regressor returned an MLP model with three dense layers, which are the same as what has been implemented in existing literature and also what was used to develop Model 1,2 and 3.

Although a similar base structure was used to develop these five models 1 to 5, different evaluation results were obtained. The main difference between the searched algorithm and the manually designed model, and also the reason for different evaluation results, is the hyper-parameters being used for the layers. In this case, the difference is made due to the number of layers in each model and in image models, the number of filters used in the convolutional layers and the number of perceptrons in each layer in the MLP. It is also surprising that, in this study, the models designed by AutoKeras did not outperform the conventional model development. Models created in a conventional style achieved a R^2 score of more than 85%, meaning that the trained models have a high capability of explaining the variance in the data inputs, while the AK models are found to be less fit to the data. AK models also resulted in more bias in the prediction accuracy, where all MAPE scores are shown more than 1, meaning quite large offsets exist between the predicted energy consumption and the actual recorded. Although slightly better results were obtained, the conventional models trained on images only also did not achieve adequate prediction accuracy, which should be at least below 1. Model 1 MLP network obtained an adequate accuracy, where 79% of difference were found between the predicted results and actual results.

Based on these evaluation results, this work may argue that, first, automated tools may not always provide the best algorithms for the designed purpose. The reason for such finding can be complex, and there are a substantial amount of different NAS tools being developed which may yield different algorithm results than AK and the manually developed CNN. Second, models built using a single source of data may not be sufficient to provide the best prediction accuracy. This study therefore moved on to develop a multimodal network and examine whether using multiple modalities can improve the estimation accuracy.

Comparison between mono-modality and multi-modality

After incorporating both sources of data, the model performance has been significantly improved. The MAPE value has dropped from over 100% to 46% with a R^2 value of 0.95, suggesting the multimodal network is able to provide a significantly better understanding of the property energy performance using the inputs.

The impacts of incorporating multiple streams of data are also reflected in the changes in features' relative importance to the estimation. Figure 5.4 presents the relative importance elements in both data inputs resulting from mono- and multi-modal networks. SHAP was used to provide explanations on how the inputs contribute to the final prediction of the designed network.

A random selection of images is presented in Figure 5.4a. The first column in the figure is the original image input, the second column is the result identified by the model built by using only images, and the third column is the result of the multiple modalities. The scale is visualised at the bottom, darker red suggests higher SHAP values, which means the pixels have higher positive impacts on the model performance, while darker blue suggests the opposite. From these randomly drawn sample images, it can be seen that both models were able to use features from the property facade for estimation.

Comparing the images in the same row shows the changes in key pixels identified by the respective model. It can be seen that, fewer pixels were detected by the mono-modal network than the multi-modal network, suggesting that it has a relatively weaker capability in extracting informative features from the facade. In the mono-modal network, pixels surrounding the edges of the property, especially the roof of the property (the second subplot), tend to have a higher contribution to estimating energy consumption. In comparison, after incorporating EPC data, the multimodal network is able to extract more useful features from the building facade. However, the key pixels identified by the multimodal network relatively show less clustering effects, and no clear evidence can be seen that if certain building features contribute more to the model prediction. This means that it is difficult to identify key features and design respective retrofit scenarios, emphasising the important role of the tabular modal included in the estimation network.

The change in feature importance is also reflected in tabular modality, the relative importance of the EPC features, as presented in Figure 5.4b. For the single modality model, the total floor area is identified as the most important feature for predicting the operational energy consumption of properties in Barnsley, followed by the properties' age band and the conditions of the wall. After adding image as another source of data, the rank changed. The 'floor descriptions' becomes the dominant feature, followed by the conditions of walls and roof. Total floor area and year of construction dropped to rank seventh and eighth respectively. This change may indicate that, after integrating visual modality, the network may have found that 'Floor Description' carries more information for the predictions, which might not have been apparent in the purely tabular setup.

Partial Dependence

The evaluation results demonstrated that a multimodal network can provide a better understanding of the properties' energy performance. The ranked feature importance for multiple modalities can then be used for retrofit scenario exploration. To explore the effects in energy consumption when renovating each housing element, the respective correlations to energy consumption are measured using SHAP, as presented in Figure 5.5. As explained, the shapely values calculated using SHAP are similar to the traditional partial dependence measure employed in Chapter 4, mapping the SHAP value can provide indications of the feature's partial dependence.

The series of subplots in Figure 5.5 is ordered by the relative feature importance rank computed using the deep multimodal network. It needs to be noted that, the features used in the prediction are pre-processed, where categorical features were encoded with numeric labels, and numeric features were normalised. For categorical features, the class each numbered label referred to is indicated on the right hand of each plot. The x-axes in the plots for numeric features were reverted to their original value.

The limitation of using SHAP values for partial dependence plots is that, each scatter or dot in the plot indicates a shapely value calculated for each feature joined the prediction, the resulting scatter plot therefore exhibits a distribution around the mean, where darker blues and wider lines can be seen. For the purpose of interpreting the general trends across different feature classes, these darker areas will be considered as the centroid of the distribution and used to

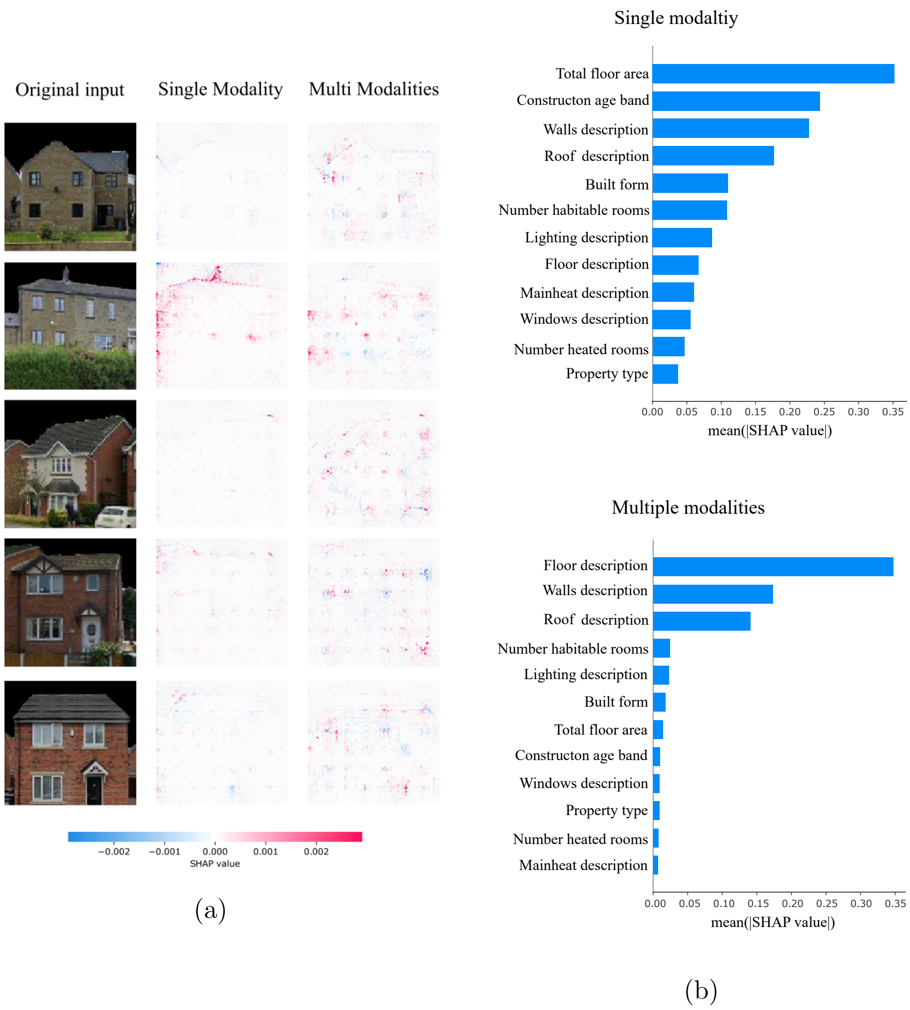
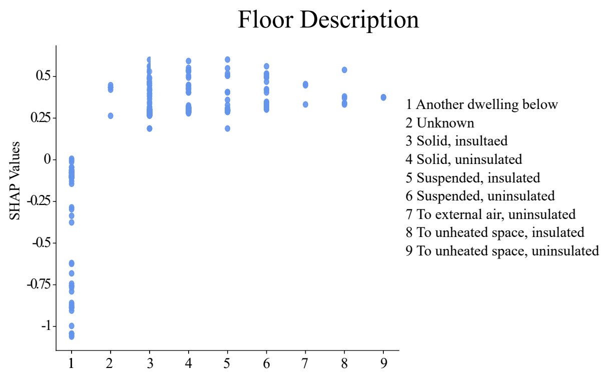


Figure 5.4: Comparison of the relative importance of data inputs to the energy estimation, reflecting changes in key features because of integration of additional visual modality. (a) Presented the changes in images. The first column is the original image input into the model, the second column is the key pixels identified by the model built using image only, and the third column identifies the key pixels by the model built using multi-modalities. The multimodal network is able to extract more information from the street view images than the model built using image only. (b) The rank of relative feature importance of tabular inputs to the model, the upper bar chart is mono-modality, and the bottom chart is multi-modalities. Incorporating images has caused changes in the feature importance rank.

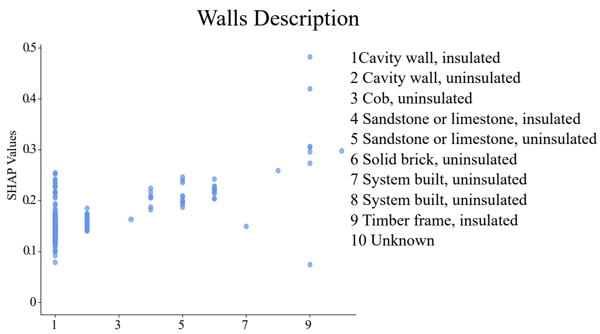
offer explanations. It is also worth noticing that it is natural that certain housing features have a smaller number of samples than others, which resulted in fewer dots on the plot and more clustered effects.

The top 3 key features identified by the feature importance rank, floor, walls and roof also suggested the same correlations with the first case study conducted in Sheffield. When there is another dwelling below, insulated cavity walls, and with roof rooms presented, the property tends to consume less energy.

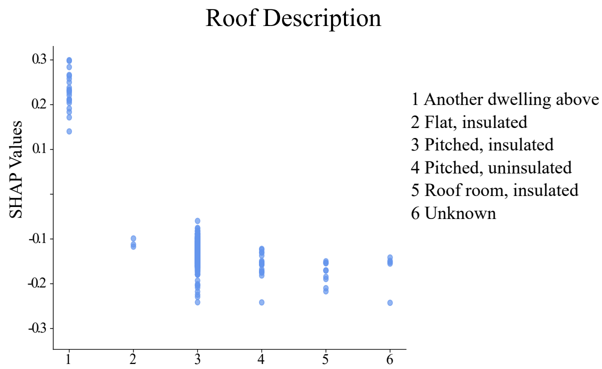
In general, the same findings can be found with the PDP mapped in the first case study for Sheffield in Chapter 4. Larger properties, more habitable and heated rooms suggests higher energy need. Insulated building elements have better performance than properties constructed in the same material but uninsulated. The higher proportion of energy-efficient installations, in other words, more energy-efficient lighting being installed, means lower SHAP values and less energy consumption. Relatively old and large detached houses with floors connecting to unheated space may be prioritised for renovation.



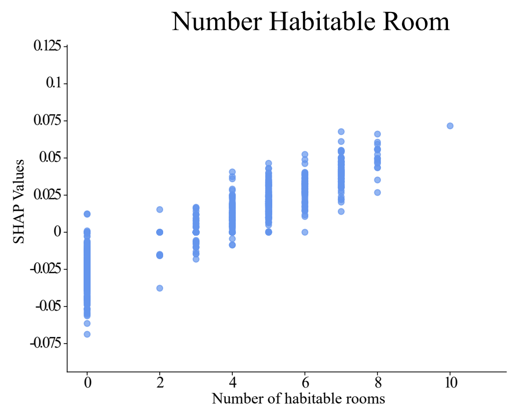
(a) SHAP PD for floor conditions.



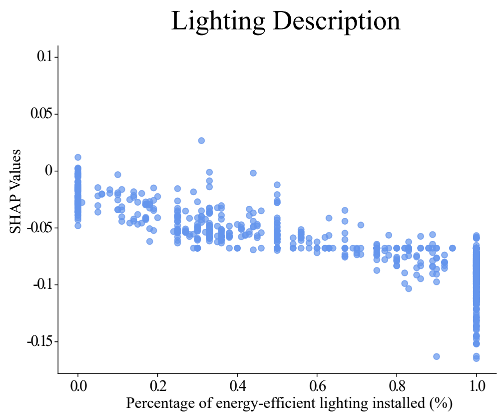
(b) SHAP PD for wall conditions.



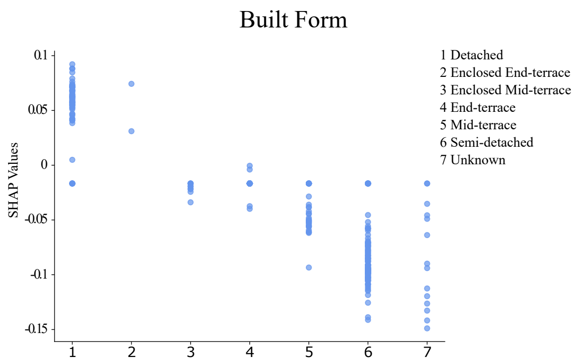
(c) SHAP PD for roof conditions.



(d) SHAP PD for number of habitable rooms.

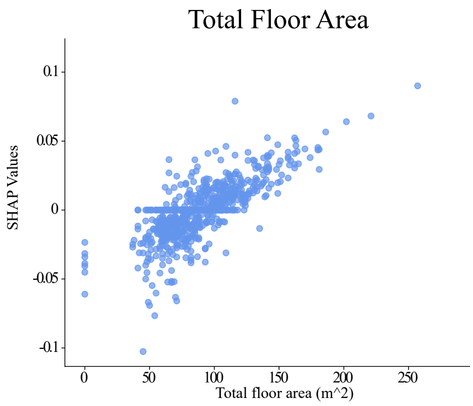


(e) SHAP PD for the proportion of energy-saving lights.

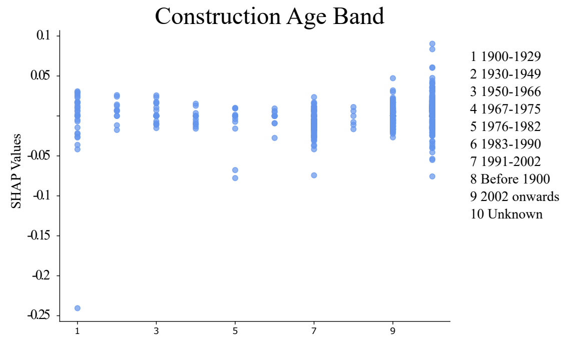


(f) SHAP PD for built form.

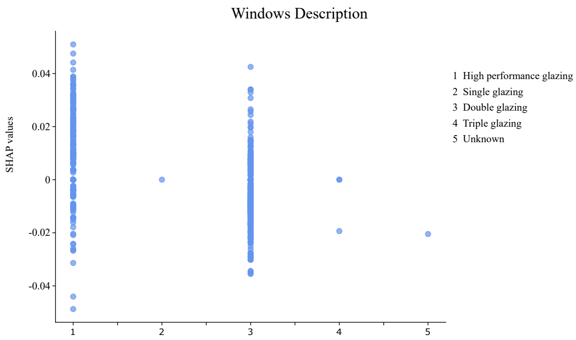
Figure 5.5: SHAP PD plots showing the marginal effects (SHAP values y-axis) of building features (the x-axis) in Barnsley.



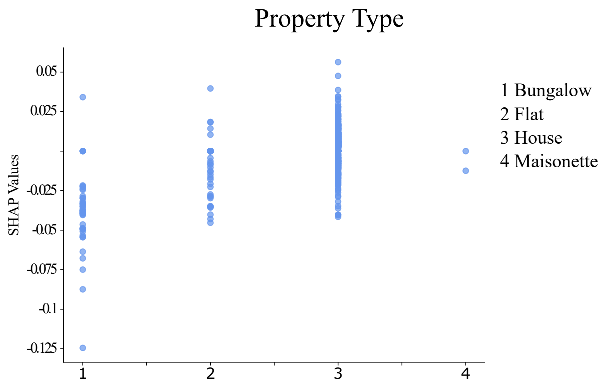
(g) SHAP PD for total floor area.



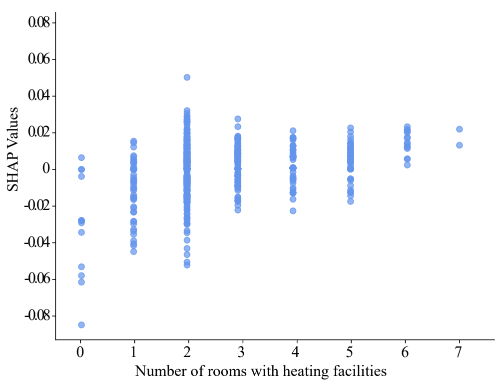
(h) SHAP PD for the construction age band.



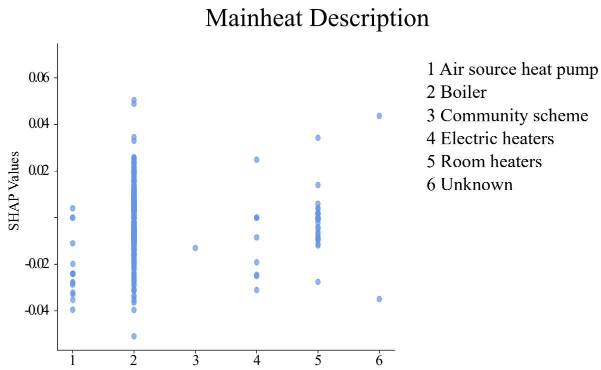
(i) SHAP PD for window conditions.



(j) SHAP PD for the type of property.



(k) SHAP PD for number of rooms accessible to heating facilities.



(l) SHAP PD for the main type of heating used.

Figure 5.5: SHAP PD plots showing the marginal effects (SHAP values y-axis) of building features (the x-axis) in Barnsley.

5.4 Chapter conclusion

Despite existing case study discussed in the literature review successfully using real estate images to predict the respective energy ratings, this work found that mono-modality models with images can only achieve limited accuracy in estimating the operational energy consumption. Therefore, this study adopted the idea and examined the potential of employing multimodal deep learning to improve the accuracy of energy consumption estimation. Deep learning models that accept multiple streams of data inputs have become popular due to their capability to process complex unstructured data. Deriving from the best neural network architecture concluded from the first comparative study made between NAS and manually designed models, this study was able to create a multimodal network that accepts both tabular and visual inputs from EPCs and GSV images. The prediction performance of the multimodal network has proven to be significantly better than models built using mono-modality, comparing the resulted MAPE.

The improvement in model performance is also revealed by using the XAI tool SHAP. By comparing the explanation results, the multimodal network is able to detect and extract a larger number of features from the image data contributing to a higher prediction accuracy. The relative feature importance of the tabular input EPC is also changed. The dominant property feature changed from property size to floor conditions.

The results of this case study offered insights into the benefits of utilising multiple data inputs and the XAI tool to achieve more transparent, robust and accurate predictions of residential building energy consumption on a large scale. However, it should be noted that the spatial coverage of these potential modalities may be limited. For instance, the spatial coverage and updating schedule of the Google Streetview images used in this case study are location-dependent. The next chapter will aim to address this limitation, by leveraging knowledge from cities with comprehensive datasets to assist in analysing cities with poor or less data.

Chapter 6

Transfer learning based efficient residential energy estimation for regions with limited data

6.1 Chapter introduction

The last chapter demonstrated the application of a multimodal network can achieve significant improvement in energy consumption estimation. However, it is very common that a target region can only access to limited data or poor-quality data, which makes it difficult to provide accurate knowledge to train a data-driven model. For instance, the collection and updating schedule of GSV varies across regions. From the results presented in the preliminary study in Chapter 4, the direct application of a model successfully trained on one region to another region with different housing characteristics will lead to a decreased prediction capability.

This chapter seeks to address the limitation of data availability across regions by overcoming the spatial heterogeneity. In this chapter, the idea of transfer learning will be introduced and investigated its capability to address this limitation. The concept of transfer learning is inspired by how humans learn from one thing to another. It is capable to break the barriers between cities and leverage knowledge learned from a city with a relatively comprehensive housing database to assist in energy consumption estimation for cities with limited or poorer housing data.

By employing transfer learning, this chapter aims to achieve the second aspect of the research aim, to improve the energy estimation model so it is adaptive to changing spatial contexts, by answering the research question: **Can transfer learning techniques improve the adaptability of operational energy estimation for regions with limited data accessibility?**.

The concept of transfer learning indicates that, to transfer knowledge, it is required to have the following elements: a) a source domain with a comprehensive housing database, b) a target domain that needs to learn from the source domain, in this study, cities with limited property data accessibility, and c) a network that was able to effectively learn from the source domain and apply to the target domain.

Overall, the research question is answered by:

- Developing a model that is able to perform transfer learning upon the best-performing network developed in Chapter 5;
- Performing prediction on target domains and evaluating its performance with and without learning from the source domain;
- Comparing the evaluation result to conclude whether transfer learning is an effective approach for cities with less adequate data.

6.2 Methodology and MARVEL data

For the purpose of this study design and data availability, three cities are selected: Barnsley and Doncaster, England, which have similar building characteristics, and Merthyr Tydfil, Wales, which exhibits differences from the other two cities. The statistics of these cities were included in Chapter 3. Three scenarios are designed to evaluate the proposed transfer learning method:

- **Scenario 1 Same city, different data source:** Transferring knowledge from Google

Street View to assist prediction using MARVEL captures in Barnsley. This will be conducted by using the base model trained on GSV and EPC for Barnsley, and transfer learning with MARVEL captures for Barnsley.

- **Scenario 2 Cities with similar building features, same data source:** Transferring knowledge from Barnsley to assist estimation in Doncaster. This will be conducted by using the base model trained on GSV and EPC for Barnsley, and transfer learning with MARVEL captures for Doncaster.
- **Scenario 3 Cities with different building features, same data source:** Transferring knowledge learnt in Barnsley to assist estimation in Merthyr Tydfil. This will be conducted by using the base model trained on GSV and EPC for Barnsley, and transfer learning with MARVEL captures for Merthyr Tydfil.

The overall workflow is illustrated in Figure 6.1. EPCs and GSVs for Barnsley are considered as the source domain. Target domains use EPCs and images captured by the MARVEL. The multimodal network trained and tested in Chapter 5 using the source domain is used as the base network for this study. The data collection and modelling procedure will be detailed in the remainder of this section.

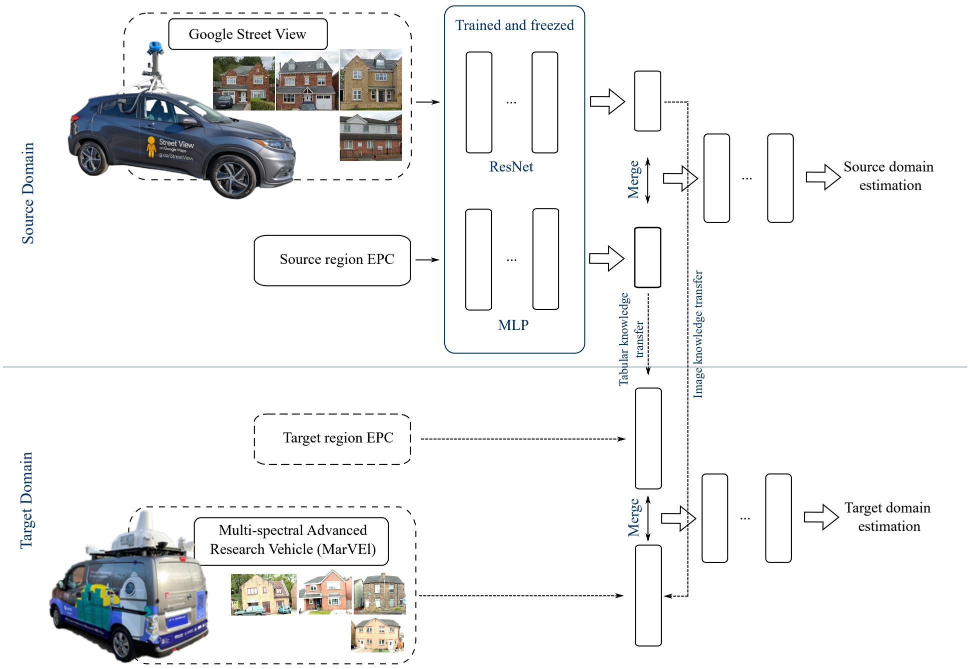


Figure 6.1: Overall workflow. Knowledge is firstly learned from the source domain by training and freezing the shallow layers. Then transfers to new contexts from the target domain before two modalities merge to perform the final prediction.

6.2.1 Transfer learning

The transfer learning approach used in this study is traditional transfer learning, which involves pre-training the base network, and then fine-tuning the top layers, as indicated in Figure 6.1. The transfer of knowledge is achieved by freezing the layers pre-trained by the source domain,

Figure 6.2: The MARVEL. This data-capturing system is used for collecting street view images for Barnsley, Doncaster and Merthyr Tydfil in this thesis. Detailed design of this data capturing system can be found in Meyers et al. [91], Dai et al. [92]

and setting different learning rates for the networks. In this study, the learning rate for the trainable layers is set as ten times higher than the untrainable layers. The base network used in this study is the multimodal network developed in Chapter 5. The model has been evaluated using R^2 and MAPE and proved its capability of performing effective energy estimation, and further evaluated using SHAP to confirm that it is able to capture hierarchical patterns and representations from the inputs that are meaningful to energy estimation, which ensures that useful information will be transferred.

6.2.2 Street view image captured by MARVEL

MARVEL, similar to a GSV van, collects data by driving through the neighbourhoods, with more controllable capturing schedules and routines. It is a vehicle-mounted data-capturing system designed by the RISE group for city-scale building stock data collection. As demonstrated in Figure 6.2, the electric vehicle is equipped with a data capturing platform on top of the van.

The data-capturing platform contains sensor units that allow simultaneous collection of various types of data, including visual, thermal, and hyperspectral images, as well as point clouds, along with their corresponding geographical locations and driving paths, through a combined National Addressable Spatial Systems system and Inertial Measurement Unit [93]. This collection of data enables several streams of research opportunities. The visual image is a common format for street view images, similar to GSV; thermal data is one of the primarily source of data for diagnosing building defects; the hyperspectral images allow assessment of material stock; and the point clouds are an important data source for 3D construction and benefit the development of digital twins.

The primary data used in this study are the visual images captured as Ladybug stream files. Ladybug cameras on MARVEL are installed facing different directions, forming a hexagonal shape. Among the cameras, the two at the back are positioned perpendicular to the street, while the two at the front have different angles. As a result, the street view images that capture the views directly facing the street are between the two cameras. Therefore, the captured images from different cameras were firstly fused into panoramic images and then sliced into four images facing directly towards the properties alongside the streets. These slices are mapped based on their geographical information to identify the best views.

Several captures have been performed around the UK, including Sheffield, Barnsley, Doncaster, Merthyr Tydfil and Glasgow. The author acted as a member of the survey team for the capture in Merthyr Tydfil, while the remaining endeavors were executed by fellow colleagues. Since the captures are conducted for different projects at different times, the image quality and quantity for each city varies. Table 6.1 provides a summary of the image data captured by MARVEL used in this study, including the sample size and issues. Comparing the three target domains, Barnsley has the largest sample size, but the worst image quality among the three. Doncaster has the best quality data, but the smallest sample size. The variations in quantity and quality of the captured data among the cities provide an interesting setting for this case study to showcase the potential of the transfer learning approach.

Table 6.1: A summary of the MARVEL image data used in this study for the three case study cities.

City	Sample size	Limitations
Barnsley	1,547	Images being over exposed
Doncaster	451	Limited sample size
Merthyr Tydfil	1,345	Images affected by raindrops

In comparison to GSV images, MARVel captures data has constraints in sample size, due to storage availability during each capture. Linking the captured image with EPC also suffered from significant data loss, further reducing the amount of image data available to use.

MARVEL images also varied in quality, this is primarily due to weather conditions. During the data capture stage, the driving speed and routes were carefully planned to ensure comprehensive coverage. However, the weather conditions remain less predictable. Example captures for the selected cities are presented in Figure 6.3. Compared with GSV images, the images captured in Barnsley are over-exposed (Figure 6.3b), and some important features, especially the roof, become similar to the sky. Therefore, the usual sky-removal preprocess step was not able to proceed for these images. The capture for Merthyr Tydfil was affected by the rainy weather (Figure 6.3d). On the other hand, the weather conditions were considerably optimal during the capture in Doncaster (Figure 6.3c), it primarily served as a test drive after the vehicle was initially set up and for a rather different research purpose, which resulted in a smaller sample size of captures.

6.3 Results of model training

Following the proposed methodology, this study performed two modelling approaches. for each case study city. The model performance without using transfer learning is used as the baseline to compare against the model performance with knowledge transferred. Table 6.2 presents the evaluation results of the model training.

The multimodal network trained in Chapter 5 has demonstrated its capability of providing an accurate understanding of the operational energy performance for properties in Barnsley, using EPC and GSV. But before adding the transfer learning component to the multimodal network, all three predictions returned relatively low fitness and accuracy. Not surprisingly,

(a) Barnsley GSV

(b) Barnsley MARVEL

(c) Doncaster MARVEL

(d) Merthyr Tydfil MARVEL

Figure 6.3: Example street view images captured by two different sources used in this study. The quality of the image varies across regions, mostly affected by weather conditions.

Table 6.2: Results of the model training for three different case study cities, without and with using the transfer learning approach proposed in this study.

Case study cities	Without transfer		With transfer		
	R^2	MAPE	R^2	MAPE	Improvement in MAPE
Barnsley	0.28	7.12	0.70	2.92	192%
Doncaster	-41.09	1.29	0.42	0.58	42%
Merthyr Tydfil	-0.39	1.84	0.83	0.67	33%

since the base model is trained on GSV images obtained in Barnsley, when predicting the energy consumption using MARVEL captured in Barnsley has the highest R^2 score of 0.28. The R^2 scores for Doncaster and Merthyr Tydfil are extremely low. Negative R^2 scores were obtained, as discussed in Section 2.5, these negative values suggest that there are significant errors between the predicted $\hat{y}$ and y and that the model is not suitable to perform energy estimation using MARVEL captured images and EPC for Doncaster and Merthyr Tydfil. Nevertheless, the prediction made for Barnsley has obtained the highest MAPE error among the three cities. This may suggest that the MARVEL captures are more different than the GSV images captured when weather conditions are more suitable for imaging.

After adding the transfer learning component, we can see that the prediction performance for all the target regions have been significantly improved. The prediction error for the Barnsley domain has the largest improvement, the MAPE error has dropped by 192%, while the Doncaster domain and Merthyr Tydfil domain experienced a drop of 42% and 33% respectively. All these improvements in evaluation metrics indicate that, after transfer learning, the model is more suitable for the target regions and can perform more accurate predictions.

6.3.1 Feature importance

Although only limited data for the target domain was used to train the prediction model, the relative feature importance rank still offers valuable insights for these examined properties. The ranking is presented in Figure 6.4.

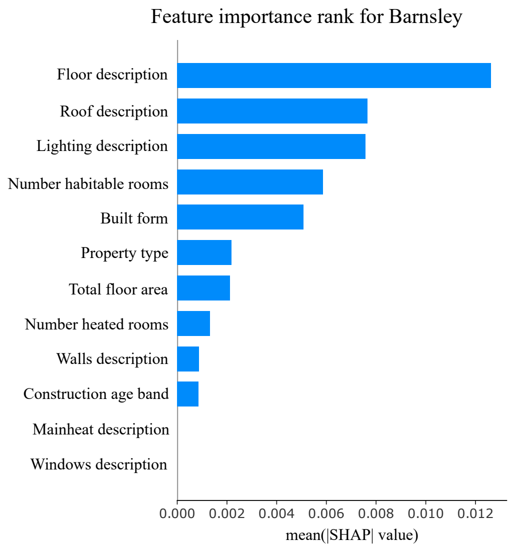
In Barnsley, as illustrated in Figure 6.4a, slightly different ranking results were shown for these selected samples compared to the ranks produced using the residential EPCs in the second case study in Chapter 5. The conditions of the building's structural elements, floor and roof, remain the dominant features for estimation. Surprisingly, the proportion of energy-saving lighting being installed is ranked third for the transfer learning based prediction. This change could be attributed to various factors. Within the constrained sample, the proportion of energy-saving lighting installed in the property may gain prominence in influencing the model predictions. We can see from the statistics provided in the general data Chapter 3, compared with the entire database used for training the base network, the reduced samples have a larger value of CV (69%, the entire database has a CV of 55%). This larger CV means that the properties in $D_T Barnsley$ have higher variability in their lighting conditions, which may be the reason for the increased importance being identified by the transfer learning model.

For the Doncaster domain, Figure 6.4b, the condition of the walls of the property plays a dominant role in the estimation, followed by property type and roof conditions. It is worth noting that, the EPCs used to tune Doncaster's model in this case study only comprise a very small number of distinct classes for roof conditions, but it resulted in a high rank of importance. There are two possible interpretations for this rather contradictory finding. One is that, these classes of roof conditions have a pronounced effect on energy consumption estimates, so the model found higher importance for this feature. Another possibility is that, because the sample size for Doncaster is small, the data is less able to alter the 'knowledge' the base network learned from $D_S Barnsley$. As the roof is considered as the third important feature for estimating $D_S Barnsley$, the transfer learning model, before training with any target domain, may have naturally assigned a higher importance to the roof to start with.

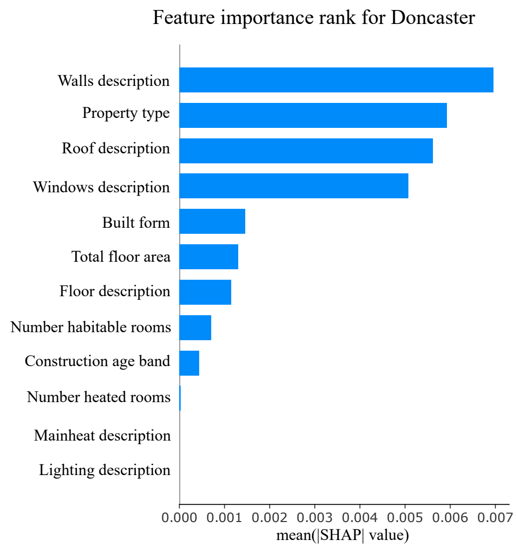
The Merthyr Tydfil domain also exhibits different ranking results, as shown in Figure 6.4c, where the conditions of walls, roof and year of construction are found as the most important features. The last four features, total floor area, main heating used, type of window and built form were found to be not important in this prediction. The reason is that, as statistics shown in Table 3.58, all properties in this subset of the dataset contain quite similar classes. The total floor area for this subset of properties only has a CV of 0.13, suggesting a small variation around the average size. All of the properties examined are using 'Boiler', and 'Double glazing' windows.

Comparing the three feature importance ranks, as shown in Figure 6.5, all three study regions exhibit different orders of ranking. The initial ranking is based on what the base network produced using Barnsley data. Features are divided into four tiers, with every three features considered as one tier and marked using one colour, from the most important to the least important. Typically, features considered to belong to the top tier in the base network remain in the top tier for the transfer learning models. The same applies to features in the bottom tier. However, there are also some exceptional cases where a feature with a high initial rank may be less important for another region. For instance, 'Walls descriptions' is initially ranked second but drops to ninth when using MARVEL captures instead of GSVs in Barnsley. Additionally, a feature with relatively less importance in the base network can be considered one of the dominating features in another region. For instance, 'Property Type' is only ranked 10th by the base network but identified as the second most important feature for the Doncaster domain.

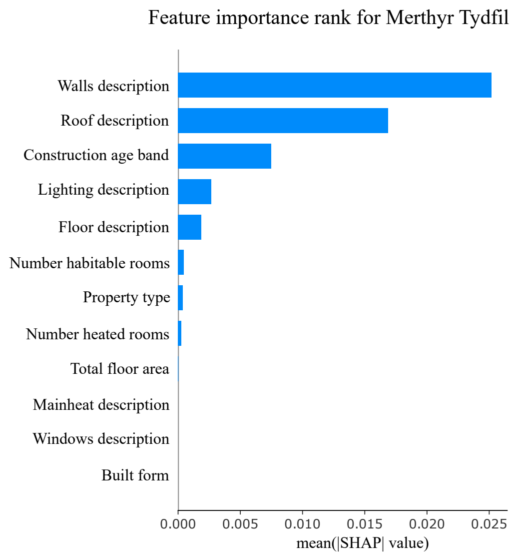
One potential explanation for the changes in importance rank is the data distributions of the sample properties used in each domain. Based on the tables included in Section 3.2.3, the Barnsley source domain has a larger variability in building features. This variation suggests that there are more significant differences in floor types, making floor description a crucial factor for the model. The model learns that differences in floor attributes can lead to substantial variations in energy consumption, thus ranking it higher in importance. In contrast, most of the sample properties used as the target domain in Merthyr Tydfil have relatively more uniform characteristics, which reduces the significance of this feature. Consequently, the final model with a transfer learning element does not find the floor description as significant and ranks it lower. Similarly, due to data loss, the properties remained for the Barnsley transfer learning model has less different types of walls than the properties used for training the Barnsley base model. Hence the rank for 'Walls descriptions' dropped, although both models are trained for Barnsley.



(a) Feature importance rank for D_SBarnsley.



(b) Feature importance rank for D_SDoncaster.



(c) Feature importance rank for D_SMerthyr Tydfil.

Figure 6.4: The feature importance ranks for the three target domains, computed by the transfer learning models.

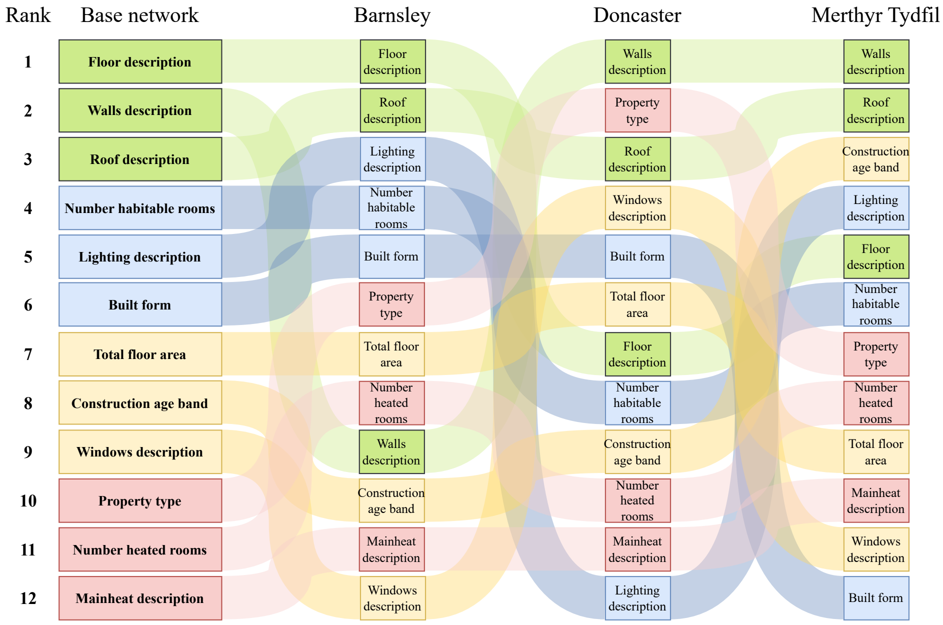


Figure 6.5: The changing ranking of the features used in the prediction for three target domains. Although all the three model is based on the base model trained on Barnsley, the feature importance rank varied across three target regions. Indicating it is able to adapt to the changed local contexts to reflect the feature importance rank accordingly.

The variation in ranking may suggest that, although all the models are trained using the Barnsley source domain, the final transfer learning models are able to pick up what is important in the data added representing the new regions and proceed with corresponding changes, indicating that the proposed framework can adapt to new city contexts. However, the limited data sample for the target domain may lead to a less accurate indication of the actual feature importance. Further studies may be conducted after collecting more data, especially Doncaster and Merthyr Tydfil, in both visual and tabular formats, to further examine the robustness of the transfer learning model and provide more understanding of the building energy performance for the cities.

However, unfortunately, because only a subset of properties in each region were used for this study, not all the classes of housing features were able to be examined. Further examination of the partial dependence plots may not provide as much meaningful information as the last two studies presented as only a few classes were seen by the transfer learning models developed in this study.

6.4 Chapter conclusion

This chapter examined the potential of utilising transfer learning to improve the model performance for changing spatial scales and contexts, when limited data is available. Developed from the best-performing multimodal network designed for Barnsley, by setting specific train-ability and learning rates to each layer of the network, it is able to store and transfer knowledge learned from the source domain, which is the EPC and GSV for Barnsley, to facilitate more accurate prediction for the target domains, which are EPC and street view images captured by MARVEL for Barnsley, Doncaster and Merthyr Tydfil. The proposed study and results demonstrated that utilising transfer learning is able to contribute to a more adaptable understanding of the relationship between housing characteristics and energy consumption, providing valuable insights into the feature importance with limited data.

The database for the three case study regions represented three different limitations in data accessibility. The Barnsley domain has the largest sample size, but the worst data quality. The Doncaster domain has the best data quality, but the smallest sample size. And the Merthyr Tydfil, has an adequate sample size and data quality, but it has the most different building characteristics with the source domain, which is Barnsley in this study. By comparing the evaluation results with and without transfer learning, this work demonstrated that the proposed transfer learning approach is able to overcome the limitation of model performance caused by limited data availability. Comparing the results for Barnsley, we are able to see a significant improvement in the model performance, the largest improvement reduced the MAPE by 192%. Comparing the evaluation results for the three cities suggests that, when facing changing spatial context, the model is able to perform an adequate and stable prediction.

Furthermore, the model evaluation results support the statement that both quality and quantity are important for training an effective and accurate data-driven model. When data accessibility is limited, the quality of data may be more critical than data quantity. However, when the quantity of data and coverage of features are limited, the model may be less capable of providing accurate indications of the relative contributions and correlations with energy consumption.

Chapter 7

Discussion

7.1 Chapter introduction

The thesis has conducted three distinct case studies to address the inherent limitations in data-driven operational energy consumption estimations. These case studies were designed to investigate the applications of deep multimodal learning and transfer learning to address the research questions initially set at the beginning of the thesis.

The results derived from the applications have demonstrated significant improvement in enhancing the trustworthiness and adaptability of operational energy consumption predictions for residential buildings. This section will provide a discussion of the results attained across the three case studies. It takes into account the constraints encountered during the research, while recommending potential future investigations, the practical implications of integrating the proposed approaches within the realms of energy-efficient retrofit practices and the formulation of environmental policies will be discussed.

7.2 Enhancing the trustworthiness and adaptability of energy estimation modelling

The conducted series of studies holds significant implications for achieving the research aim, which is centred on **enhancing the trustworthiness and adaptability of energy estimation modelling**. In this aim, the **'trustworthiness'** of the modelling means a *robust* and *transparent* understanding of the correlations between housing features and properties' energy performance. The **'adaptability'** is referred to as the ability to acknowledge the changing spatial scale and social contexts of different regions.

The first study presented in Chapter 4 served as the first step of this work. The study primarily used EPC, which although flawed, is the most comprehensive database available for existing English housing stock. The utilisation of spatial, building morphological features, and thermal characteristics facilitated the construction of a robust random forest model. This model effectively delivered accurate energy consumption estimates for properties in Sheffield. The subsequent revelation of feature importance ranks and partial dependence plots offered deeper insights into the predictive role of individual features. These findings not only agreed with the general patterns outlined in the literature review but also affirm the contributions of **spatial, morphological, and thermal** attributes to estimating energy consumption.

The following application of the designed model to different regions also achieved an intriguing investigation - the relative significance of these attributes varies regarding to the target regions. These variations correspond with the diverse algorithm selection and relative importance of the selected features summarised for selective existing research. This finding underscores the criticality of choosing a spatial scale pertinent to the intended analysis and constructing a model that acknowledges the local contexts.

However, the limitations of EPCs are widely recognised. Although the methodology for creating and issuing EPCs is frequently reviewed and updated, the original EPCs downloaded for this work are still exposed to the errors and issues discussed in the literature review. Recognising the limitations of EPC data, the second case study explored alternative data sources, to investigate the potential of using image data as alternatives to EPC to represent the building characteristics. CNN models that are commonly adopted for image data were deployed to perform energy estimation using images. Despite the successful application of such data in predicting EPC ratings, the experiment study for energy consumption prediction in Barnsley did not return satisfactory results. One potential implication that can be drawn from the results obtained so far, is that, models developed using a single source of input may not be sufficient enough to facilitate a robust understanding of the properties. Neither EPC nor GSV can be considered as extremely accurate in providing comprehensive representations of the target properties, hence the trustworthiness of the understanding of operational energy estimation resulting from using these data may be limited. This therefore leads to the first research question proposed at the beginning of this thesis:

> *Can incorporating multi-modalities improve the trustworthiness of operational energy estimation?*

To answer this research question, a deep multimodal network was developed, and its performance was evaluated and compared against the single modal networks. Both EPCs and GSVs were used in the multimodal network. The multimodal network is developed based on the best-performing algorithm structure for each modality and merged after being processed in their own pathways. The model is able to extract the key information from each source of data and create a joint representation that captures the combined information from both EPCs and GSVs. The evaluation results demonstrated a significant improvement in prediction accuracy and fitness to the dataset, where the MAPE error rate was improved from 1.15 for ResNet and 0.86 for MLP to 0.43 using both modalities. Based on these results, we may conclude that incorporating multi-modalities can significantly improve the trustworthiness of operational energy estimation. This conclusion is further supported by examining the relative contributions of both modalities using the explainable AI tool SHAP. When examining the image inputs, compared with the ResNet model trained sorely on image data, the multimodal network is able to extract more useful features from the images contributing to the model in overall.

The primary reason that the first two studies in Chapter 4 and 5 are able to develop models specific for the target regions is that, datasets with adequate quantity and quality are available to develop models from scratch. However, the accessibility to good and large-scale data varies across regions. We know from the literature review and the initial study in Chapter 4 that, applying a model built from one region to others is often unsuitable due to spatial differences. This can be due to varying feature distributions and may lead to significant accuracy reduction. Essentially, a successful model application requires similarity in data distribution in the target region. This leads to the second research question, addressing the model application issues due to the variations in data available across regions:

> *Can transfer learning techniques improve the adaptability of operational energy estimation?*

The third case study was designed to answer this research question by developing a model that can learn and store key features learned from the source domain, usually have better data available, and transfer the knowledge to assist the prediction for the target domain, which has limited data accessibility. In this work, there are in overall three pairs of source domain and target domain:

- D_S Barnsley EPC + GSV $\longleftrightarrow$ D_T Barnsley EPC + MARVEL
- D_S Barnsley EPC + GSV $\longleftrightarrow$ D_T Doncaster EPC + MARVEL
- D_S Barnsley EPC + GSV $\longleftrightarrow$ D_T Merthyr Tydfil EPC + MARVEL

It is evident from the second case study that multimodal networks have better performance than models developed sorely on one source of data, and evaluated the capability of achieving accurate estimation using the multimodal network developed with EPC and GSV obtained for Barnsley. Therefore, the multimodal model built in the second case study was used as the base network. The ability to learn and transfer knowledge from D_S to D_T was achieved by setting train-ability and learning rates for different layers. By comparison with the model evaluation results with and without the use of transfer learning, with the use of transfer learning, the prediction accuracy was significantly improved for all three pairs of domains. The most significant improvement was found in the first set of domains, for the same case study region Barnsley, the image-modal of the source domain was trained using GSV and the target domain was trained using MARVEL captured street view, the MAPE was improved for 192%. For the other two domains in different regions in the UK, the model was still able to achieve rather stable prediction results. These sets of results agreed with the general rules discussed in the literature review, where closer regions have more similar energy needs.

These three studies were designed to complete a thorough framework, that is able to provide accurate estimation of the residential energy estimation for properties, with minimum requirement for data quality and quantity. With the help of multimodal network and transfer learning, the model is able to provide a robust understanding of the housing stocks in terms of their energy efficiency.

7.3 Implications for energy need analysis and retrofit potential identification

In addition to answering the designed research questions and achieving a better understanding of the built environment, the following examinations in feature importance rank and partial dependence plots provide explanations for each model. These examinations contribute to a transparent understanding of the model and offer insights into the key features for energy consumption estimations and their respective correlations with energy usage. The most important housing features can be as the guidance for the retrofit scenario exploration. When designing the potential scenario, more important features should be included to achieve efficient improvements. The mapped partial dependence offers valuable insights into the correlations between each feature class and energy consumption. From the mapped relationship, we can understand which material used or improved insulation may achieve the best improvements.

The relative feature importance rank offers insights into the most important features for each estimation. Table 7.1 presented a summary of all the feature importance rank concluded in this thesis' case studies. The first and second concluded feature ranks only included EPC records in the model training, so the importance rank is built sorely based on tabular entries. The rankings 3 to 6 are produced using both EPCs and images, so the importance rank is mapped based on EPC entries calibrated by street view images. The *roof descriptions*, which describe the type and insulation conditions of the roof, appeared the most number of times in the top 3 features among all the models developed.

For ranking sorely built on EPCs, the *size of the property* is identified as the key estimator for both models. Such a finding is consistent with the inherent relationship between dwelling size and its energy consumption, as larger properties naturally require a larger number of energy-intensive appliances, such as heating facilities, to maintain residents' demand for comfortable living.

For rankings built on multiple modalities, the importance ranks were calibrated by street view images, and the resulted ranks vary across regions. The impact from *property size* becomes less important than the first two models. The changes in ranking demonstrated that, in a multimodal network, the model is able to learn more complex interactions and relationships between features across the modalities, which can lead to the discovery of hidden patterns that are not apparent when individual modality is considered separately.

For properties in Barnsley, rank 3 and 4, have agreed that *floor description*, which describes the material and insulation conditions of properties' floor, is the most important feature for energy estimation. And models for Doncaster and Merthyr Tydfil, rank 5 and 6, both consider *walls descriptions*, which describes the material and insulation conditions for walls, are the key feature in their estimations. The differences between the feature importance ranks 3 to 6 in three studied cities reflect the different housing conditions among cities, furthermore, it approved that the proposed transfer learning model is able to derive patterns specific to the target regions and adapt the model accordingly so it is more suitable and robust for the regions. This finding emphasized that, to accurately identify the retrofit prioritisations for a specific region, the model used should be tailored to acknowledge the local specific energy performance.

Table 7.1: All feature importance ranks produced by the models this study proposed and examined in Chapter 4 to 6.

No.	Region	Rank 1^{st}	Rank 2^{nd}	Rank 3^{rd}
1	Sheffield	Total floor area	Walls description	Mainheat description
2	Barnsley (monomodal)	Total floor area	Age band	Walls description
3	Barnsley (multimodal)	Floor description	Walls description	Roof description
4	Barnsley (transfer)	Floor description	Roof description	Lighting description
5	Doncaster	Walls description	Property type	Roof description
6	Merthyr Tydfil	Walls description	Roof description	Age band

It is worth noting that, from the various rankings mapped by different regions, properties in various regions may suffer from different issues. Therefore, it is important for local authorities to identify the specific needs of residents to achieve optimized improvements to their homes.

The partial dependence plotted following the feature importance rank in this work, to some extent, demonstrates an approach to identifying the optimal retrofit schemes. Although this thesis did not have enough data samples to calculate the partial dependence between housing characteristics and estimated energy consumption for Doncaster and Merthyr Tydfil, the series of partial dependence plots for Sheffield and Barnsley in Figure 4.6 and Figure 5.5, respectively, may be considered as examples. Similar conclusions can be drawn from these two series of visualizations, indicating that residential properties constructed with 'insulated cavity walls' and 'ground or water source heat pumps' will achieve significant reductions in energy consumption.

These findings, to some extent, align with the prioritizations of home upgrading measures currently being implemented by the UK government. Currently, most home renovation projects focus on improving loft, cavity wall, and solid wall insulation, which corresponds with the 'roof description' and 'walls description' features used in this thesis. These two features are commonly considered key elements for energy estimation.

7.4 Limitation of the designed framework and potential future work

The limitations of the designed framework have mainly two perspectives: the paucity of data available to use and the methods examined. This section will discuss these limitations and provide recommendations to address them accordingly.

The main data used in this thesis are EPCs and street view images. One of the primary assumptions this thesis based on is that the features recorded in the EPCs are accurate indications of the housing conditions. As discussed in the literature review, substantial studies have examined the errors in EPC records. This thesis acknowledged this error, but because this is the only publicly available database covering almost all the properties in the UK, the proposed framework attempted to minimise the error by incorporating a second visual modality. However, EPCs still play a key role in the model training and feature importance ranking, more importantly the models trained in this thesis use EPC-estimated energy consumption as the ground truth data. Although the primary aim is to explore the robustness of the proposed multimodal and transferring framework, without access to real-time energy consumption data, it is difficult to examine the errors between the proposed prediction and real-world conditions.

While the official authorities are continuously updating the methodology for creating such a database, beside the bias between assessor-recorded entries and real conditions, one of the major limitation of EPC is that it does not include information describing the occupants. Additional to in-situ measurement to calibrate the entries in EPCs or create a more comprehensive housing database, it would be beneficial to consider including relative socio-demographic information, such as working pattern and household size. For example, agent-based modelling is also a popular tool that mimics the occupants' energy usage patterns to simulate residential demands. By doing so, it offers an opportunity to calibrate the existing EPC assessment procedure. Following the framework this study proposed, using a calibrated or new database, future studies can contribute to a more holistic understanding of the operational energy performance of the existing housing stock, the estimation can be more accurate and trustworthy, the feature rankings and partial dependence can cover more classes of interests, together leading to a thorough knowledge base and evidence for further explorations on retrofit potentials.

Due to limited data availability and accessibility to individual properties, the main limitation of the image data used is that they usually only show the front exterior surface of the property. Unlike the real estate images used in Despotovic et al. [29], the agents might take photos from multiple angles to attract customers, when driving along the street and capture street view images, most of the time only the facade can be scanned. This might cause some key characteristics to be neglected by the model. For example, for properties with a similar front facade, properties with floor-to-ceiling windows at the back, may exhibit a large difference in energy consumption compared with properties with only solid walls. To have a more holistic understanding of the properties, future studies may include more photos from multiple angles for studies based on 2D images, or 3D models created using techniques such as LiDAR and photogrammetry. This will ensure more complete housing characteristics can be taken into account during the model training, and can also lead to more in-depth interaction with other modalities in the model. The use of 3D models may provide more accurate captures of the architectural features, such as roof angles, and window positions which can be challenging to accurately extract from the 2D streetview images.

For the methodology, the models developed in this study are supervised machine learning, which requires a known historical energy consumption to perform the estimation. This thesis uses EPC-estimated energy consumption as a proximity, and the visual modality is required to have a shared reference system with the EPC so the information can be matched. This mismatch and inconsistent reference has caused a significant data loss in the studies, especially for the work in Chapter 6. Future studies may employ an unsupervised approach to overcome this limitation, for instance, the adversarial unsupervised domain adaptation approach. The adversarial unsupervised domain adaptation is a subfield of transfer learning. The same as the proposed model presented in Chapter 6, it can learn from domains with known consumption and align knowledge across domains. But the word 'unsupervised' means that the approach does not require the target property, or the images of the property to be connected with energy consumption values to perform the prediction. Using an unsupervised approach may further improve the model's ability in adapting to various contexts.

While addressing the limitations of this thesis, future work may utilise the findings from the studies on feature importance and partial dependence to perform the second stage of the urban retrofit modelling. The most important features identified in each study can be used to design the potential retrofit measures for implementation. For example, for properties in Sheffield, possible retrofit options can be 1) improving wall insulations, 2) replacing boiler with higher efficiency equipments and 3) improving roof insulations. To implement the renovations according to the ranks given by the partial dependence, there are other critical factors should be considered, particularly the cost and feasibility of the selected techniques. For instance, according to the partial dependence value calculated in this thesis, replacing existing boilers with heat pumps may achieve a signifiant improvement in building energy efficiency. However, not every property meets the requirement of such installation. There are several criteria may be considered to assess the suitability, such as the accessibility to outdoor area and the kind of underground soil, and because installing heat pumps usually involves drilling a number of holes to install the system several meters below the land surface, it usually involves large capital costs upon installation [94]. The work in this thesis, especially the calculated partial dependence plot, to some extend lead the direction of exploration, but there is more work to be done before actual retrofit.

With the recent advances in AI, there are also other tools may help facilitating home retrofit programmes, such as the ChatGPT. Addition to help with explaining the complex data involved with operational energy consumption, it can also help in the process of communicating with relative stakeholders, and informing strategic decision-making.

7.5 Chapter conclusion

This chapter presented a discussion on the key findings of the studies conducted to explore the applications of multimodal network and transfer learning approaches, the implications for retrofit possibilities from the explained feature importance and partial dependence, and limitations and recommendations for future energy estimation studies. From the three studies conducted in four cities across the UK, it is evident that the proposed application of multimodal and transfer learning approaches can enhance the trustworthiness and adaptability of operational energy estimation. Acknowledging the limitations primarily due to limited data accessibility, this framework has the potential to be further developed, by introducing more datasets, such as images from multiple angles, to contribute to a more thorough understanding of housing energy performance, and assist in implementing effective home renovation measures to achieve the net-zero targets.

Chapter 8

Conclusion and recommendation

8.1 Chapter introduction

While there is no single solution for mitigating the climate crisis, decarbonising the existing housing stock plays a critical role. Effective decarbonisation of the existing housing stock, especially the residential sectors, needs an accurate understanding of the current operational usage to guide the selection of optimal retrofit scenarios. To achieve this, this thesis investigated the application of multimodal and transfer learning approaches in estimating residential properties' operational energy consumption. The models were evaluated through statistical metrics and explainable AI tools. In order to achieve a robust, transparent and adaptable energy prediction framework. This chapter will provide an overview of the findings, contributes to knowledge and conclusion of the work entailed in this thesis.

The literature review conducted in Chapter 2 provided an overview of the existing research related to energy estimation and retrofit option exploration. Existing studies that concern relatively large scale, e.g. at city scale, usually employ data-driven approaches that build machine learning models based on data describing the properties' characteristics to understand the operational energy usage of the target properties. These models are considered to have learned a comprehensive understanding of the city. The literature review also discussed the attempts to explore the inside of the 'black box', with the help of explainable AI tools, such as SHAP, the trained models can be explained to suggest the relationship between housing features and energy consumption, which is critical to guide the selection of retrofit prioritisations.

The advances and drawbacks of the data and machine learning models implemented in the existing studies were discussed. Although machine learning is a widely developed field, concerns remain, especially regarding the **trustworthiness** and **adaptability** of the existing methodologies. From these major concerns, the idea of using multimodal learning and transfer learning to operational energy estimation is introduced. This leads to the main research questions of this work:

- ***Can incorporating multi-modalities improve the trustworthiness of operational energy estimation?***
- ***Can transfer learning techniques improve the adaptability of operational energy estimation?***

To address these research questions, this thesis primarily evaluated two data modalities, tabular and visual, from multiple sources for residential operational energy usage estimation. These include data describing the properties' geospatial and topological characteristics, extracted from OS map data, thermal-related features from EPC and visual representations of properties from the street view images captured by Google van and MARVEL. Using these data, three studies were designed and conducted. Each study serves as the foundation for the next, together towards an effective and holistic operation energy estimation framework.

8.2 Summary of findings

Chapter 4 conducted a baseline study for properties in Sheffield. This case study used OS map data and EPC to train a random forest based machine learning model to perform the energy consumption estimation for properties in Sheffield. This work was further evaluated by permutation feature importance and partial dependence. The evaluation results indicated that the housing sizes and conditions of the external walls are the most important features when estimating the energy consumption of residential buildings. Recommending to target relatively larger and older detached houses in neighbourhoods with higher build density for home renovation projects. However, the generalisability of these results is subject to certain limitations.

The first limitation corresponds to the first aspect of the research aim, *the trustworthiness*. The quality of the issued EPCs is found to be problematic, mainly due to the inconsistencies among the EPC assessors, and the fact that it does not need to be updated unless it have been issued for a decade or the properties has been renovated. This limitation is addressed in the second study in **Chapter 5** by the application of **deep multimodal learning network**. By leveraging GSV as an additional modality for residential energy consumption prediction, this research seeks to address the issues within EPC assessments. The inclusion of GSV imagery enables a more comprehensive understanding of the residential properties, thereby enhancing the predictive trustworthiness of the model. However, it is also worth noting that, the author acknowledges that, by using EPC-estimated energy consumption for model training, the predicted energy consumption may not directly correlate with real-world data, such as the smart meter data. However, it is important to emphasise that the primary objective of this thesis is not to provide a comparative analysis between EPC-estimated energy consumption and smart meter readings. Rather, the ultimate goal is to develop a robust predictive framework capable of accommodating diverse data modalities from different sources. By constructing a flexible and adaptable framework, it is possible to facilitate seamless integration and analysis of smart meter data when such data becomes publicly or research available. Through the application of multimodal and transfer learning techniques, this research aims to establish a foundation for future studies to explore and evaluate the efficacy of predictive models in the context of real-world energy consumption patterns. Therefore, although this thesis provides EPC-based predictions, the ultimate aim is to transition towards the utilisation and validation of more precise real world data, thereby advancing the state-of-the-art in residential energy consumption understanding.

The second limitation corresponds to the second aspect of the research aim, *the adaptability*. The prediction accuracy dropped dramatically when applying the trained model to a region where the buildings' features exhibit vast differences, a Welsh city Merthyr Tydfil. The varied housing types and designs in different geospatial locations constraint the adaptability of the model. However, it is very common that regions may have different accessibility to adequate data, and therefore not be able to develop their own prediction model. This limitation is addressed in the third study in **Chapter 6** by the application of **transfer learning**.

Chapter 5 investigated the potential of using image data as an alternative source to EPC for energy consumption estimation. Although image data has been applied in existing studies for energy rating estimation, this case study conducted for Barnsley did not return satisfying prediction results. The potential reasons for the high prediction error are difficult to determine, but from this result, it is possible to argue that using street view images solely may not be sufficient for the neural network to extract enough knowledge to understand the energy usage of the property. Therefore, this thesis introduced the idea of multimodal deep learning. The multimodal network designed in this thesis is able to accept both visual and tabular representations of the property. The first research question is answered, and the evaluation result confirmed that the multimodal network is able to extract common knowledge from both streams and at the same time learn the difference, to achieve a wider understanding of the energy estimation task and better prediction accuracy. By using this multimodal network, structural elements including floor, walls and roof are found to be the dominant features in understanding the operational energy usage of properties in Barnsley.

Chapter 6 employs transfer learning to improve the estimation model's adaptability to changing spatial contexts when there is limited data accessibility. Developed upon the multimodal neural network developed in Chapter 5, each layer was updated with its train-ability and learning rate. The second research question is answered, after transfer learning, the models' performance for regions with data either limited in quality or quantity were significantly improved.

From the explanations made for each network built in the three studies presented, there is no agreed conclusion can be made in terms of what should be considered as the dominant housing features to estimate the operational energy usage. This thesis suggests that this variation is caused by the spatial heterogeneity across the regions, similar to the existing studies discussed in the literature review. Therefore, it is important to develop a network respective to each region to acknowledge such differences. This feature importance may be used to guide possible retrofit measure selections when implementing home renovation schemes. Different strategies may be prioritised for different regions. For instance, properties in Sheffield may expect more upgrades focusing on improving wall insulations, while properties in Barnsley may be prioritised on improving floor conditions.

8.3 Concluding remarks

This thesis presented the applications of deep multimodal learning and transfer learning in improving the trustworthiness and adaptability of energy estimations. At the same time examined the feature importance and correlations using explainable AI, which offers valuable guidance on optimal retrofit prioritisations.

In the realm of residential energy performance analysis, data-driven approaches emerge as pivotal tools. However, conventional machine learning models trained solely on a single modality of input may fall short in providing a comprehensive understanding of the properties' energy performance, especially when data accessibility is limited, as often encountered in real-world contexts.

Through a series of systematic studies, this thesis has rigorously explored the potential of deploying deep multimodal networks and transfer learning for energy estimation tasks. The results demonstrated large improvements compared with the traditional modelling approach that mainly depending on one source of data, underlining the merit of these methodologies. The outcomes also offered insights into prospective retrofit prioritisations for the areas of investigation. By embracing state-of-the-art techniques, this thesis explored the interconnections between housing features and energy consumption, contributing to the pursuit of optimised energy efficiency strategies.

There are also various technique and datasets existing for potential future work to further enhance the effectiveness of the proposed framework. For example, integrating agent-based modelling include the impacts of occupant behaviour into the predictive framework. By incorporating behavioural dynamics, such as daily routines, temperature preferences, and energy usage habits, the proposed framework may able to make prediction more similar to real-world residential consumption. There are also potential to utilise additional data, such as LiDAR, which can offer a richer view of the property, and smart meter data, if obtainable for research purposes, can significantly contributes to the accuracy and applicability of this proposed framework.

In conclusion, this thesis has demonstrated the efficacy of deep multimodal learning and transfer learning in enhancing the trustworthiness and adaptability of energy estimations, while also shedding light on interpreting the correlations between parameters using explainable AI techniques. By delving into the realm of residential energy performance analysis, the study has underscored the limitations of traditional machine learning models reliant on single modalities of input, particularly in scenarios with limited data accessibility. Through systematic studies in different cities across the UK, significant improvements have been observed with the deployment of deep multimodal networks and transfer learning methodologies, offering valuable insights into prospective retrofit prioritizations. Moreover, the exploration of interconnections between housing features and energy consumption has contributed to guide retrofit potentials. Potential future researches include the integration of agent-based modelling to incorporate occupant behaviour dynamics, utilisation of additional datasets such as LiDAR and smart meter data.

References

[1] Gordon Pask. *An approach to cybernetics.* 1961.

[2] BEIS. UK sets ambitious new climate target ahead of un summit. Technical report, BEIS, 2020. `https://www.gov.uk/government/news/uk-sets-ambitious-new-climate-target-ahead-of-un-summit`, visited on 2024-01-13.

[3] BEIS. Energy company obligation - eco4: 2022-2026, 4 . `https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1065823/eco4-government-response.pdf,visitedon2024-02-04`, year = 2022.

[4] Amin Sokhanvar, Serhan Çiftçioğlu, and Chien-Chiang Lee. The effect of energy price shocks on commodity currencies during the war in ukraine. *Resources Policy*, 82:103571, 2023. `https://www.sciencedirect.com/science/article/pii/S0301420723002829?casa_token=6fPMi99Aa4kAAAAA:A60KOw2AN8JBxgEKdfW5eslZREb5tTSko1kHx3kO2Twclppy6MgrD2icASm4lr6C4rXj`, visited on 2024-02-04.

[5] DESNZ and BEIS. National statistics: Energy consumption in the UK 2022, 10 2022. `https://www.gov.uk/government/statistics/energy-consumption-in-the-uk-2022`, visited on 2024-01-13.

[6] HM Treasury. Infrastructure carbon review. Technical report, 11 2013. `https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/260710/infrastructure_carbon_review_251113.pdf`, visited on 2024-01-13.

[7] Ameera Al-Hassan. The introduction of code of sustainable homes for the uk; potentials and problems. In *FORUM Ejournal 2008-09 Editorial Committee*, 2009.

[8] Christopher O'Malley, Poorang AE Piroozfar, Eric RP Farr, and Jonathan Gates. Evaluating the efficacy of breeam code for sustainable homes (csh): A cross-sectional study. *Energy Procedia*, 62:210–219, 2014.

[9] BEIS. The heat and buildings strategy. Technical report, BEIS, 10 2021. `https://assets.publishing.service.gov.uk/government/uploads/system/uploads/`

attachment_data/file/1044598/6.7408_BEIS_Clean_Heat_Heat___Buildings_Strategy_Stage_2_v5_WEB.pdf, visited on 2024-01-13.

[10] Mohammad Saffari and Paul Beagon. Home energy retrofit: Reviewing its depth, scale of delivery, and sustainability. *Energy and Buildings*, 269:112253, 2022. ISSN 03787788. doi: 10.1016/j.enbuild.2022.112253. https://doi.org/10.1016/j.enbuild.2022.112253, visited on 2024-01-13.

[11] ONS. National statistics: English housing survey 2021-22. Technical report, 2022. https://www.gov.uk/government/statistics/english-housing-survey-2021-to-2022-headline-report/english-housing-survey-2021-to-2022-headline-report, visited on 2024-01-13.

[12] DLUHC. Energy performance of buildings data: England and wales. Technical report, DLUHC, 2023. https://epc.opendatacommunities.org, visited on 2024-01-13.

[13] E3G. UK government continues to underspend on home retrofits. Technical report, E3G, 2 2023. https://www.e3g.org/news/uk-government-continues-to-underspend-on-home-retrofits/, visited on 2024-01-13.

[14] ONS. Age of the property is the biggest single factor in energy efficiency of homes. Technical report, 1 2022. https://www.ons.gov.uk/peoplepopulationandcommunity/housing/articles/ageofthepropertyisthebiggestsinglefactorinenergyefficiencyofhomes/2021-11-01, visited on 2024-01-13.

[15] Heba Elsharkawy and Peter Rutherford. Energy-efficient retrofit of social housing in the UK: Lessons learned from a community energy saving programme (cesp) in nottingham. *Energy and Buildings*, 172:295–306, 2018. https://www.sciencedirect.com/science/article/abs/pii/S0378778818306157, visited on 2024-01-13.

[16] Yixuan Wei, Xingxing Zhang, Yong Shi, Liang Xia, Song Pan, Jinshun Wu, Mengjie Han, and Xiaoyun Zhao. A review of data-driven approaches for prediction and classification of building energy consumption. *Renewable and Sustainable Energy Reviews*, 82:1027–1047, 2018. ISSN 18790690. doi: 10.1016/j.rser.2017.09.108. visited on 2024-01-13.

[17] Mathieu Bourdeau, Xiao qiang Zhai, Elyes Nefzaoui, Xiaofeng Guo, and Patrice Chatellier. Modeling and forecasting building energy consumption: A review of data-driven techniques. *Sustainable Cities and Society*, 48, 7 2019. ISSN 22106707. doi: 10.1016/j.scs.2019.101533. visited on 2024-01-13.

[18] Zeyu Wang, Yueren Wang, Ruochen Zeng, Ravi S Srinivasan, and Sherry Ahrentzen. Random forest based hourly building energy prediction. *Energy and Buildings*, 171:11–25, 2018. https://www.sciencedirect.com/science/article/abs/pii/S0378778818311290, visited on 2024-01-13.

[19] Usman Ali, Mohammad Haris Shamsi, Fawaz Alshehri, Eleni Mangina, and James O'Donnell.

Application of intelligent algorithms for residential building energy performance rating prediction. *Building Simulation Conference Proceedings*, 5:3177–3184, 2019. ISSN 25222708. doi: 10.26868/25222708.2019.210232. visited on 2024-01-13.

[20] Wenliang Li, Yuyu Zhou, Kristen Cetin, Jiyong Eom, Yu Wang, Gang Chen, and Xuesong Zhang. Modeling urban building energy use: A review of modeling approaches and procedures. *Energy*, 141:2445–2457, 2017. `https://www.sciencedirect.com/science/article/abs/pii/S0360544217319291`, visited on 2024-01-13.

[21] J. F. Rosser, D. S. Boyd, G. Long, S. Zakhary, Y. Mao, and D. Robinson. Predicting residential building age from map data. *Computers, Environment and Urban Systems*, 73:56–67, 2019. ISSN 01989715. doi: 10.1016/j.compenvurbsys.2018.08.004. visited on 2024-01-13.

[22] Constantine E. Kontokosta and Christopher Tull. A data-driven predictive model of city-scale energy use in buildings. *Applied Energy*, 197:303–317, 2017. ISSN 03062619. doi: 10.1016/j.apenergy.2017.04.005. visited on 2024-01-13.

[23] Aurélie Foucquier, Sylvain Robert, Frédéric Suard, Louis Stéphan, and Arnaud Jay. State of the art in building modelling and energy performances prediction: A review. *Renewable and Sustainable Energy Reviews*, 23:272–288, 2013. `https://www.sciencedirect.com/science/article/abs/pii/S1364032113001536`, visited on 2024-01-13.

[24] Yixing Chen, Tianzhen Hong, and Mary Ann Piette. Automatic generation and simulation of urban building energy models based on city datasets for city-scale building retrofit analysis. *Applied Energy*, 205:323–335, 2017. `https://www.sciencedirect.com/science/article/abs/pii/S0306261917310024`, visited on 2024-01-13.

[25] Behnam Nikparvar and Jean-Claude Thill. Machine learning of spatial data. *ISPRS International Journal of Geo-Information*, 10(9):600, 2021. `https://www.mdpi.com/2220-9964/10/9/600`, visited on 2024-01-13.

[26] Nick Gallent and Mark Tewdwr-Jones. *Decent Homes for All: Planning's evolving role in housing provision.* Routledge, 2007. ISBN 978-0-415-27446-3.

[27] Usman Ali, Mohammad Haris Shamsi, Mark Bohacek, Cathal Hoare, Karl Purcell, Eleni Mangina, and James O'Donnell. A data-driven approach to optimize urban scale energy retrofit decisions for residential buildings. *Applied Energy*, 267:114861, 2020. ISSN 03062619. doi: 10.1016/j.apenergy.2020.114861. `https://doi.org/10.1016/j.apenergy.2020.114861`, visited on 2024-01-13.

[28] Simon Wenninger and Christian Wiethe. Benchmarking energy quantification methods to predict heating energy performance of residential buildings in germany. *Business & Information Systems Engineering*, 63:223–242, 2021. `https://link.springer.com/article/10.1007/s12599-021-00691-2`, visited on 2024-01-13.

[29] Miroslav Despotovic, David Koch, Sascha Leiber, Mario Döller, Muntaha Sakeena, and

Matthias Zeppelzauer. Prediction and analysis of heating energy demand for detached houses by computer vision. *Energy and Buildings*, 193:29–35, 2019. ISSN 03787788. doi: 10.1016/j.enbuild.2019.03.036. https://doi.org/10.1016/j.enbuild.2019.03.036, visited on 2024-01-13.

[30] Razak Olu-Ajayi, Hafiz Alaka, Ismail Sulaimon, Funlade Sunmola, and Saheed Ajayi. Building energy consumption prediction for residential buildings using deep learning and other machine learning techniques. *Journal of Building Engineering*, 45, 1 2022. ISSN 23527102. doi: 10.1016/j.jobe.2021.103406. visited on 2024-01-13.

[31] Maoran Sun, Changyu Han, Quan Nie, Jingying Xu, Fan Zhang, and Qunshan Zhao. Understanding building energy efficiency with administrative and emerging urban big data by deep learning in glasgow. *Energy and Buildings*, 273, 10 2022. ISSN 03787788. doi: 10.1016/j.enbuild.2022.112331. visited on 2024-01-13.

[32] Filip Biljecki and Maximilian Sindram. Estimating building age with 3d gis. In *Proceedings of the 12th International 3D GeoInfo Conference 2017*, pages 17–24, 2017. https://www.researchgate.net/publication/320570658_Estimating_building_age_with_3D_GIS, visited on 2024-01-13.

[33] Ipek Gursel Dino, Alp Eren Sari, Orcun Koral Iseri, Sahin Akin, Esat Kalfaoglu, Bilge Erdogan, Sinan Kalkan, and A. Aydin Alatan. Image-based construction of building energy models using computer vision. *Automation in Construction*, 116:103231, 8 2020. ISSN 09265805. doi: 10.1016/j.autcon.2020.103231. visited on 2024-01-13.

[34] Camilla Hjortling, Folke Björk, Magnus Berg, and Tord af Klintberg. Energy mapping of existing building stock in sweden – analysis of data from energy performance certificates. *Energy and Buildings*, 153:341–355, 10 2017. ISSN 03787788. doi: 10.1016/j.enbuild.2017.06.073. http://dx.doi.org/10.1016/j.enbuild.2017.06.073, visited on 2024-01-13.

[35] Gustavo Sousa, Benjamin M. Jones, Parham A. Mirzaei, and Darren Robinson. A review and critique of UK housing stock energy models, modelling approaches and data sources. *Energy and Buildings*, 151:66–80, 9 2017. ISSN 03787788. doi: 10.1016/j.enbuild.2017.06.043. visited on 2024-01-13.

[36] Romain Nouvel, Alessio Mastrucci, Ulrich Leopold, Olivier Baume, Volker Coors, and Ursula Eicker. Combining gis-based statistical and engineering urban heat consumption models: Towards a new framework for multi-scale policy support. *Energy and Buildings*, 107: 204–212, 11 2015. ISSN 03787788. doi: 10.1016/j.enbuild.2015.08.021. visited on 2024-01-13.

[37] Charles Heazel and Open Geospatial Consortium. Ogc city geography markup language (citygml) 3.0 conceptual model: Users guide. Technical report, 2021. http://www.opengis.net/doc/UG/CityGML-user-guide/3.0, visited on 2024-01-13.

[38] Matthias Zeppelzauer, Miroslav Despotovic, Muntaha Sakeena, David Koch, and Mario Döller. Automatic prediction of building age from photographs. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, volume 3206060, pages

126–134. ACM, 6 2018. ISBN 9781450350464. doi: 10.1145/3206025.3206060. `https://dl.acm.org/doi/10.1145/3206025.3206060`, visited on 2024-01-13.

[39] Timnit Gebru, Jonathan Krause, Yilun Wang, Duyun Chen, Jia Deng, Erez Lieberman Aiden, and Li Fei-Fei. Using deep learning and google street view to estimate the demographic makeup of neighborhoods across the united states. *Proceedings of the National Academy of Sciences of the United States of America*, 114:13108–13113, 2017. ISSN 10916490. doi: 10.1073/pnas.1700035114. visited on 2024-01-13.

[40] Jiangye Yuan and Anil M Cheriyadat. Combining maps and street level images for building height and facade estimation. In *Proceedings of the 2nd ACM SIGSPATIAL Workshop on Smart Cities and Urban Analytics, UrbanGIS 2016*, 2016. ISBN 9781450345835. doi: 10.1145/3007540.3007548. `https://dl.acm.org/doi/proceedings/10.1145/3007540`, visited on 2024-01-13.

[41] Filip Biljecki and Koichi Ito. Street view imagery in urban analytics and gis: A review. *Landscape and Urban Planning*, 215:104217, 2021.

[42] Berangere Lartigue, Laura Biewesch, Flore Marion, Erica Cochran, and Françoise Thellier. Energy performance certificates in the usa and in france—a case study of multifamily housing. *Energy Efficiency*, 15, 6 2022. ISSN 15706478. doi: 10.1007/s12053-022-10036-x. visited on 2024-01-13.

[43] A. Hardy and D. Glew. An analysis of errors in the energy performance certificate database. *Energy Policy*, 129:1168–1178, 6 2019. ISSN 03014215. doi: 10.1016/j.enpol.2019.03.022. `https://doi.org/10.1016/j.enpol.2019.03.022`, visited on 2024-01-13.

[44] DECC and BRE. The government ' s standard assessment procedure for energy rating of dwellings version 10.2 (11-04-2023). Technical report, DECC and BRE, 2023. `https://bregroup.com/sap/sap10/`, visited on 2024-01-13.

[45] BRE. List of sap programs tested by bre for use in connection with building regulations and energy performance certificates for new dwellings. Technical report, BRE, 4 2023. `https://files.bregroup.com/SAP/SAP10_2_software_2023-04-20.pdf`, visited on 2024-01-13.

[46] Jenny Crawley, Phillip Biddulph, Paul J Northrop, Jez Wingfield, Tadj Oreszczyn, and Cliff Elwell. Quantifying the measurement error on england and wales epc ratings. *Energies*, 12, 2019. ISSN 19961073. doi: 10.3390/en12183523. `https://www.mdpi.com//1996-1073//12//18//3523//htm`, visited on 2024-01-13.

[47] David Jenkins, Sophie Simpson, and Andrew Peacock. Investigating the consistency and quality of epc ratings and assessments. *Energy*, 138:480–489, 2017. ISSN 03605442. doi: 10.1016/j.energy.2017.07.105. `http://dx.doi.org/10.1016/j.energy.2017.07.105`, visited on 2024-01-13.

[48] Department of Energy and Climate Change. Green deal assessment mystery shopping research. Technical report, Department of Energy and

Climate Change, 2014. https://www.gov.uk/government/publications/green-deal-assessment-mystery-shopping-research, visited on 2024-01-13.

[49] Laura Gabrielli and Aurora Greta Ruggeri. Developing a model for energy retrofit in large building portfolios: Energy assessment, optimization and uncertainty. *Energy and Buildings*, 202:109356, 2019. ISSN 03787788. doi: 10.1016/j.enbuild.2019.109356. https://doi.org/10.1016/j.enbuild.2019.109356, visited on 2024-01-13.

[50] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective.* MIT Press, 2012.

[51] Abdo Abdullah Ahmed Gassar, Geun Young Yun, and Sumin Kim. Data-driven approach to prediction of residential energy consumption at urban scales in london. *Energy*, 187, 11 2019. ISSN 03605442. doi: 10.1016/j.energy.2019.115973. visited on 2024-01-13.

[52] Aurélien Géron. *Hands-on machine learning with scikit-learn and tensorflow: Concepts, Tools, and Techniques to Build Intelligent Systems.* O'Reilly Media, 2017.

[53] Daniel L. Marino, Kasun Amarasinghe, and Milos Manic. Building energy load forecasting using deep neural networks. In *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 7046–7051, 2016. doi: 10.1109/IECON.2016.7793413. visited on 2024-01-13.

[54] Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J. Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *Journal of Big Data*, 8, 12 2021. ISSN 21961115. doi: 10.1186/s40537-021-00444-8. visited on 2024-01-13.

[55] Andreas Muller and Sarah Guido. *Introduction to Machine Learning with Python: A guide for data scientists.* 3 edition, 2017. ISBN 978-1-449-36941-5.

[56] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf, visited on 2024-01-13.

[57] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016. https://arxiv.org/abs/1602.04938, visited on 2024-01-13.

[58] Diogo V. Carvalho, Eduardo M. Pereira, and Jaime S. Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronics (Switzerland)*, 8, 8 2019. ISSN 20799292. doi: 10.3390/electronics8080832. visited on 2024-01-13.

[59] Christoph Molnar. *Interpretable machine learning.* Lulu. com, 2020.

[60] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23:1–45, 1 2021. ISSN 10994300. doi: 10.3390/e23010018. visited on 2024-01-13.

[61] P.J.G. Lisboa, S. Saralajew, A. Vellido, R. Fernández-Domenech, and T. Villmann. The coming of age of interpretable and explainable machine learning models. *Neurocomputing*, 535:25–39, 5 2023. ISSN 09252312. doi: 10.1016/j.neucom.2023.02.040. visited on 2024-01-13.

[62] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017. `https://arxiv.org/abs/1705.07874`, visited on 2024-01-13.

[63] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22:1345–1359, 2010. ISSN 10414347. doi: 10.1109/TKDE.2009.191. visited on 2024-01-13.

[64] Ali Hooshmand and Ratnesh Sharma. Energy predictive models with limited data using transfer learning. In *e-Energy 2019 - Proceedings of the 10th ACM International Conference on Future Energy Systems*, pages 12–16. Association for Computing Machinery, Inc, 6 2019. ISBN 9781450366717. doi: 10.1145/3307772.3328284. `https://dl.acm.org/doi/10.1145/3307772.3328284`, visited on 2024-01-13.

[65] Yuan Gao, Yingjun Ruan, Chengkuan Fang, and Shuai Yin. Deep learning and transfer learning models of energy consumption forecasting for a building with poor information data. *Energy and Buildings*, 223:110156, 2020. ISSN 03787788. doi: 10.1016/j.enbuild.2020.110156. `https://doi.org/10.1016/j.enbuild.2020.110156`, visited on 2024-01-13.

[66] Abolfazl Farahani, Sahar Voghoei, Khaled Rasheed, and Hamid R Arabnia. A brief review of domain adaptation. *Advances in Data Science and Information Engineering: Proceedings from ICDATA 2020 and IKE 2020*, pages 877–894, 2021. `https://arxiv.org/abs/2010.03978`, visited on 2024-01-13.

[67] Jingxiong Zhang, Peter Atkinson, and Michael F Goodchild. *Scale in spatial information and analysis*. CRC Press, 2014.

[68] Jabeen Summaira, Xi Li, Amin Muhammad Shoib, Songyuan Li, and Jabbar Abdul. Recent advances and trends in multimodal deep learning: A review. *arXiv preprint*, 2021. `http://arxiv.org/abs/2105.11087`, visited on 2024-01-13.

[69] Khaled Bayoudh, Raja Knani, Fayçal Hamdaoui, and Abdellatif Mtibaa. A survey on deep multimodal learning for computer vision: advances, trends, applications, and datasets. *Visual Computer*, 38:2939–2970, 8 2022. ISSN 01782789. doi: 10.1007/s00371-021-02166-7. visited on 2024-01-13.

[70] Matthias Feurer, Aaron Klein, Katharina Eggensperger Jost, Tobias Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. In *Advances in Neural Information Processing Systems*. Cur-

ran Associates, Inc., 2015. https://proceedings.neurips.cc//paper//2015/file//11d0e6287202fced83f79975ec59a3a6-Paper.pdf, visited on 2024-01-13.

[71] Xin He, Kaiyong Zhao, and Xiaowen Chu. Automl: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212, 1 2021. ISSN 09507051. doi: 10.1016/j.knosys.2020.106622. visited on 2024-01-13.

[72] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. *Automated machine learning: methods, systems, challenges.* Springer Nature, 2019.

[73] Luís Ferreira, Carlos Manuel Martins, Pedro Miguel Pires, and Paulo Cortez. A comparison of automl tools for machine learning, deep learning and xgboost. In *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021. doi: https://doi.org/10.1109/IJCNN52387.2021.9534091. https://ieeexplore.ieee.org/document/9534091, visited on 2024-01-13.

[74] Xi Fang, Guangcai Gong, Guannan Li, Liang Chun, Wenqiang Li, and Pei Peng. A hybrid deep transfer learning strategy for short term cross-building energy prediction. *Energy*, 215, 1 2021. ISSN 03605442. doi: 10.1016/j.energy.2020.119208. https://doi.org/10.1016/j.energy.2020.119208, visited on 2024-01-13.

[75] OpenAI. Gpt-4 technical report. Technical report, 2023. https://arxiv.org/abs/2303.08774, visited on 2024-01-13.

[76] Tadas Baltrusaitis, Chaitanya Ahuja, and Louis Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41:423–443, 2 2019. ISSN 19393539. doi: 10.1109/TPAMI.2018.2798607. visited on 2024-01-13.

[77] Ordnance Survey. Os mastermap topography layer – building height attribute – overview. Technical report, Ordnance Survey, 2021. https://www.ordnancesurvey.co.uk/documents/product-support/user-guide/osmm-topography-layer-building-height-attribute-overview-v1.4.pdf, visited on 2024-01-13.

[78] Michael A Wirth. Shape analysis & measurement, 2004. http://www.cyto.purdue.edu/cdroms/micro2/content/education/wirth10.pdf, visited on 2024-01-13.

[79] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. doi: 10.1109/CVPR.2016.91. visited on 2024-01-13.

[80] Tania Cerquitelli, Evelina Di Corso, Stefano Proto, Paolo Bethaz, Daniele Mazzarelli, Alfonso Capozzoli, Elena Baralis, Marco Mellia, Silvia Casagrande, and Martina Tamburini. A data-driven energy platform: From energy performance certificates to human-readable knowledge through dynamic high-resolution geospatial maps. *Electronics (Switzerland)*, 9:1–26, 2020. ISSN 20799292. doi: 10.3390/electronics9122132. https://www.mdpi.

com/2079-9292/9/12/2132?utm_source=researcher_app&utm_medium=referral&utm_campaign=RESR_MRKT_Researcher_inboundwww.mdpi.com/journal/electronics, visited on 2024-01-13.

[81] Verisk Analytics Inc. UK buildings reference guide. Technical report, Verisk Analytics Inc, 2022. https://www.verisk.com/en-gb/3d-visual-intelligence/products/ukbuildings/#form, visited on 2024-01-13.

[82] Annalisa Appice, Pedro Pereira Rodrigues, Vitor Santos Costa, Carlos Soares, and João Gama. *Machine Learning and Knowledge Discovery in Databases*, volume 9284. Springer International Publishing, 2015. ISBN 978-3-319-23527-1. doi: 10.1007/978-3-319-23528-8. http://link.springer.com/10.1007/978-3-319-23528-8, visited on 2024-01-13.

[83] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001. https://www.jstor.org/stable/2699986, visited on 2024-01-13.

[84] Ron Pearson. Interpreting predictive models using partial dependence plots, 7 . https://cran.r-project.org/web/packages/datarobot/vignettes/PartialDependence.html, visited on 2024-01-13.

[85] Matthias Feurer, Katharina Eggensperger, Stefan Falkner, Marius Lindauer, and Frank Hutter. Auto-sklearn 2.0: Hands-free automl via meta-learning. *arXiv:2007.04074 [cs.LG]*, 2020. https://arxiv.org/abs/2007.04074, visited on 2024-01-13.

[86] Danielle Densley Tingley, Abigail Hathway, and Buick Davison. An environmental impact comparison of external wall insulation types. *Building and Environment*, 85:182–189, 2 2015. ISSN 03601323. doi: 10.1016/j.buildenv.2014.11.021. visited on 2024-01-13.

[87] The British Standards Institution. *Residential park homes - Specification.* BSI Standards Limited 2023, ISO 7637-1:2015(E) edition, 2023. https://bsol.bsigroup.com/PdfViewer/Viewer?pid=000000000030434145, visited on 2024-02-13.

[88] Esfand Burman, Dejan Mumovic, and Judit Kimpian. Towards measurement and verification of energy performance under the framework of the european directive for energy performance of buildings. *Energy*, 77:153–163, 12 2014. ISSN 03605442. doi: 10.1016/j.energy.2014.05.102. https://www.sciencedirect.com/science/article/pii/S0360544214006835#:~:text=A%20measurement%20and%20verification%20plan,established%20under%20identical%20operating%20conditions., visited on 2024-01-13.

[89] Asmaul Hosna, Ethel Merry, Jigmey Gyalmo, Zulfikar Alom, Zeyar Aung, and Mohammad Abdul Azim. Transfer learning: a friendly introduction. *Journal of Big Data*, 9:102, 10 2022. ISSN 2196-1115. doi: 10.1186/s40537-022-00652-w. https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00652-w, visited on 2024-01-13.

[90] Haifeng Jin, François Chollet, Qingquan Song, and Xia Hu. Autokeras: An automl

library for deep learning. *Journal of Machine Learning Research*, 24(6):1–6, 2023. `https://www.jmlr.org/papers/volume24/20-1355/20-1355.pdf`, visited on 2024-01-13.

[91] Gregory Meyers, Chengxi Zhu, Martin Mayfield, Danielle Densley Tingley, Jon Willmott, and Daniel Coca. Designing a vehicle mounted high resolution multi-spectral 3d scanner: Concept design. In *Proceedings of the 2nd workshop on data acquisition to analysis*, pages 16–21, 2019.

[92] Menglin Dai, Wil OC Ward, Hadi Arbabi, Danielle Densley Tingley, and Martin Mayfield. Scalable residential building geometry characterisation using vehicle-mounted camera system. *Energies*, 15(16):6090, 2022.

[93] Menglin Dai, Wil OC Ward, Gregory Meyers, Danielle Densley Tingley, and Martin Mayfield. Residential building facade segmentation in the urban environment. *Building and Environment*, 199:107921, 2021. `https://www.sciencedirect.com/science/article/abs/pii/S0360132321003243`, visited on 2024-01-13.

[94] Umberto Desideri, Nicola Sorbi, Livia Arcioni, and Daniela Leonardi. Feasibility study and numerical simulation of a ground source heat pump plant, applied to a residential building. *Applied Thermal Engineering*, 31(16):3500–3511, 2011.

Appendix 1: Data download and preprocess

This appendix demonstrates a detailed procedure of obtaining and preprocessing the data used in this thesis.

Data retrival

There are mainly four database used in this thesis: OS building product, EPC, GSV and MARVEL captures. The OS map data was accessed and downloaded through the Digimap service. The EPC data is publicly available at https://epc.opendatacommunities.org/. The database is updated monthly. The database allows downloads with a user-specified spatial coverage, from downloading a record for a single property using postcode, to the entire database as a bulk file. The downloaded EPCs are saved as csv.

Code for downloading GSV

GSV download through api

[ ]:
```
#import packages required
import pandas as pd
import numpy as np
```

[ ]:
```
import google_streetview.api
```

[ ]:
```
epc = pd.read_csv(r'b_epc_unique.csv')
```

[ ]:
```
#get full address from epc
addresses = epc.ADDRESS+", "+epc.LOCAL_AUTHORITY_LABEL
```

[ ]:
```
uprns = epc.UPRN
```

[ ]:
```
for i in range(9989,11124):
    params = [{
        'size': '640x640',
        'location': addresses[i],
        'fov': '100',
        'return_error_code': True,
        'source': 'outdoor',
        'key':'..', #the paid API key
        'signiture':'GSV downloaded for research only - ysheng'
    }]
    print(i)
    #create result object
    results = google_streetview.api.results(params)

    #Download images to directory 'downloads'
    results.download_links('{}.png'.format(uprns[i])) #save the file with uprn␣
↪for following model use
```

[ ]:
```
```

Data cleaning

The cleaning of EPC data used in this thesis mainly involves two stages: first, to filter the duplicated and abnormal entries, and second, to reorganise the classes so similar words are joined to avoid complex processing.

The recategorising of EPC is performed in excel spreadsheet, as explained in Chapter 3, where similar catogories are merged into one value.

Code for map and tabular data preprocessing

This section demonstrates the code written for this thesis to 1) calculate the built rate, 2) calculate the npi, 3) filter the duplicated entry in EPC, and 4) merge EPC data with map data to ensure both database is referring to the same property when performing the predictions.

Data preprocess

This notebook demonstrates the process of data cleaning and merging for model prediction.

Import packages

[ ]:
```
import pandas as pd
import numpy as np

import math
from datetime import datetime
```

[ ]:
```

```

Import link tables for geospatial references

[ ]:
```
u_toi = pd.read_csv(r'BLPU_UPRN_TopographicArea_TOID_5.csv')
u_upn = pd.read_csv(r'UKBuildings_Edition_11_ABC_link_file.csv')
```

[ ]:
```
u_toi.rename(columns = {'IDENTIFIER_1':'uprn', 'IDENTIFIER_2':'toid'}, inplace =␣
 ↪True)
```

[ ]:
```
u_toi.head(4)
```

[ ]:
```
ref = [1,4]
u_toid = u_toi.iloc[:,ref]
```

[ ]:
```

```

Import data

[ ]:
```
#epc info
epc = pd.read_csv(r'..\data\epc\sheffield_certificates.csv')

#mapinfo
ginfo = pd.read_csv(r'..\data\gis\geomni.csv')
mapinfo = pd.read_csv(r'..\data\gis\map_info.csv') #os map data
pinfo=pd.read_csv(r'..\data\process\postcodeinfo.csv')
```

[ ]:
```

```

Filter EPC duplications

[ ]:
```
du = epc.duplicated(subset = 'BUILDING_REFERENCE_NUMBER', keep=False) #filter␣
 ↪all properties with multiple records
du_p = epc.loc[du] #save then to seperate df
```

[ ]:
```
uni_ = epc.drop_duplicates(subset = 'BUILDING_REFERENCE_NUMBER',keep=False)␣
↪#filter all properties with unique record
print (len(uni_)+len(du_p)-len(epc)) #check whether they are correctly filtered
```

compare the date when the epc is lodged, the latest one is selected for following study

[ ]:
```
def convert_timestamp(stringdate): #function to convert string date into␣
↪timestamp for comparison
    #date_string_format = "%d/%m/%Y %H:%M%S"
    date_string_format = "%d/%m/%Y"
    date_and_time = datetime.strptime(stringdate, date_string_format)
    timestamp = datetime.timestamp(date_and_time)

    return timestamp
```

[ ]:
```
uni_p = du_p.BUILDING_REFERENCE_NUMBER.unique()

output = []
for p in uni_p: #walk through all the unique address
    target_epc = epc[epc['BUILDING_REFERENCE_NUMBER'] == p] #get epc records for␣
↪the same property

    timestamps = []
    for t in target_epc.LODGEMENT_DATE:
        times = convert_timestamp(t) #convert string date to timestamps
        timestamps.append(times)

    target_epc['timestamp'] = timestamps
    sortted = np.sort(target_epc['timestamp'])[0] #sort the timestamps and␣
↪select the latest one
    select = target_epc[(target_epc['timestamp'] == sortted)] #find the axis␣
↪where the timestamps are selected

    output.append(select)
```

[ ]:
```
outputs = pd.concat(output)
```

[ ]:
```
final_ = pd.concat([outputs,uni_]) #save selected records from properties with␣
↪multiple records to those with unique records
```

[ ]:
```
final_.to_csv(r'epc_unique.csv')
```

[ ]:
```

```

Calculate the builtup rate

```
[ ]: epcp = final_.groupby(['POSTCODE']).sum() #group all properties in the same␣
     ↪postcode area together
     built = epcp['TOTAL_FLOOR_AREA'] #the total built up area in the postcode
```

```
[ ]: built1 = pd.DataFrame(built)
     built1.rename(columns = {'TOTAL_FLOOR_AREA':'builtarea'}, inplace = True)
     built1.head(4)
```

```
[ ]: epc1 = final_ #duplicate the df into a new one
     epc1 = final_.merge(built1, left_on = 'POSTCODE', right_on = 'POSTCODE') #merge␣
     ↪built area with epc
     epc1 = epc1.merge(pinfo, left_on = 'POSTCODE', right_on = 'POSTCODE') #merge␣
     ↪postcode area with epc
     epc1['builtrate'] = epc1['builtarea']/epc1['parea'] #calculate builtup rate
```

```
[ ]:
```

Calculate the NPI

```
[ ]: npi = lambda a,b: (2*(math.sqrt(a*math.pi)))/b
```

```
[ ]: #output = npi(mapinfo.area, mapinfo.perimeter)
     npi_ = (2*(np.sqrt(mapinfo['area']*np.pi)))/mapinfo['perimeter']
```

```
[ ]: mapinfo['npi']=npi_
```

```
[ ]:
```

Merge calculated map data with epc

```
[ ]: epc1 = epc.merge(ginfo, left_on='upn', right_on='prop_id') #merged file␣
     ↪containing epc and os map data and age data
```

```
[ ]: epc1.to_csv(r'epc_os_age.csv')
```

Code for watershed segmentation

Remove the sky from the cropped gsvs

Import package

[ ]:
```
import PIL
from PIL import Image, ImageOps
import cv2
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

[ ]:
```
import skimage
```

[ ]:
```
from skimage import filters
from skimage import io as skio
from skimage.segmentation import watershed
```

[ ]:
```
from scipy import ndimage as ndi
```

[ ]:
```
import os,shutil
```

[ ]:
```

```

Watershed segmentation

[ ]:
```
inputfolder = r'cropped' #folder contains cropped image after object detection␣
↪using YOLOv5
outputfolder = r'no_sky' #output folder
```

[ ]:
```
def draw_group_as_background(ax, group, watershed_result, original_image):
    background_mask = (watershed_result == group)
    cleaned = original_image * ~background_mask
    ax.imshow(cleaned, cmap='gray')
    ax.axis('off')
    #ax.imshow(background_mask.reshape(background_mask.shape + (1,)) * np.
↪array([1, 0, 0, 1]))

    return cleaned
```

[ ]:
```
for _,_,filenames in os.walk(inputfolder, outputfolder):
    for filename in filenames:
        url = os.path.join(inputfolder, filename)

        img = skio.imread(url)
        sobel = filters.sobel(img)

        plt.rcParams['image.interpolation'] = 'nearest'
        plt.rcParams['image.cmap'] = 'gray'
```

```
        plt.rcParams['figure.dpi'] = 200

        blurred = filters.gaussian(sobel, sigma=2.0)

        light_spots = np.array((img > 245).nonzero()).T
        dark_spots = np.array((img < 3).nonzero()).T

        bool_mask = np.zeros(img.shape, dtype=np.bool)
        bool_mask[tuple(light_spots.T)] = True
        bool_mask[tuple(dark_spots.T)] = True
        seed_mask, num_seeds = ndi.label(bool_mask)

        ws = watershed(sobel, seed_mask)

        background = max(set(ws.ravel()), key=lambda g: np.sum(ws == g))
        background_mask = (ws == background)

        cleaned = img * ~background_mask

        background_candidates = sorted(set(ws.ravel()), key=lambda g: np.sum(ws␣
↪== g), reverse=True)

        seed_mask = np.zeros(img.shape, dtype=np.int)

        w = img.shape[0]
        h = img.shape[1]
        w1 = w-1
        h1 = h-1

        seed_mask[0, 0] = 1 # background
        seed_mask[w1, h1] = 2 # foreground

        ws = watershed(blurred, seed_mask)
        fig, ax = plt.subplots()
        f = draw_group_as_background(ax, 1, ws, img)

        outputpath = os.path.join(outputfolder, filename)
        plt.savefig(outputpath, format = 'jpg', bbox_inches='tight', pad_inches␣
↪= 0.0001)
```

[ ]:
```
for _,_,filenames in os.walk(inputfolder):
    for f in filenames:
        processed = os.path.join(outputfolder, f)
        if os.path.exists(processed) == False:
            src = os.path.join(original,f)
            shutil.copy(src, processed)
```

```
[ ]:
```

Appendix 3: Collection of prediction models written

Preliminary study: age prediction using autosklearn

Import package

[ ]:
```
import pandas as pd
import numpy as np
```

[ ]:
```
# print autosklearn version
import autosklearn
print('autosklearn: %s' % autosklearn.__version__)
```

[ ]:
```
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from autosklearn.classification import AutoSklearnClassifier
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler, OneHotEncoder
import matplotlib.pyplot as plt
```

Load data

[ ]:
```
print('now loading data...')
age = pd.read_csv('aggregated_age_sample.csv', index_col=0)
```

[ ]:
```

```

Prepare data for model training

[ ]:
```
print('now specifying x and y...')
for_age = ['Total floor area','Perimeter','Relh2','Relhmax', 'NPI',␣
 ↪'Vxcount','Builtrate']
X=age.iloc[:,for_age]
y=age['agg_age']
```

[ ]:
```
print('...now splitting data...')
# split into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)
```

[ ]:
```

```

Model training

[ ]:
```
print('...now defining model search...')
# define search
automl = AutoSklearnClassifier(
                            exclude = {
                'feature_preprocessor': ["no_preprocessing"]
```

```
                             },
                        resampling_strategy='cv',
                        resampling_strategy_arguments={'folds': 5})
```

[ ]:
```
print('...now fitting...')
# perform the search
automl.fit(X_train, y_train)
```

[ ]:
```

```

Model evaluation

[ ]:
```
print('...statistics...')
# summarize
print(automl.sprint_statistics())
```

[ ]:
```
print('...leaderboard...')
board = pd.DataFrame(automl.leaderboard(detailed = True))
board.to_csv(r'age-leaderboard.csv')
print(automl.leaderboard(detailed = True))
```

[ ]:
```
print('...now printing models found...')
ensemble_dict = automl.show_models()
print(ensemble_dict)
```

[ ]:
```
print('...now evaluating...')
# evaluate best model
prediction = automl.predict(X_test)
acc = accuracy_score(y_test, prediction)
f1_macro_score = autosklearn.metrics.f1_macro(y_test,prediction)
print("Accuracy: %.3f" % acc)
print("f1_macro: %.3f" % f1_macro_score)
```

[ ]:
```
print('...now plot performance over time...')
automl.performance_over_time_.plot(
        x='Timestamp',
        kind='line',
        legend=True,
        title='Auto-sklearn accuracy over time',
        grid=True,
    )
plt.savefig(r'plot_age.pdf')
plt.show()
```

[ ]:
```

```

Perform prediction

[ ]:
```
o_epc = pd.read_csv(r'epc.csv', index_col=0)
to_predict = o_epc.iloc[:,for_age]
```

[ ]:
```
print('...now performing prediction...')
o_predict = automl.predict(to_predict)
result = pd.DataFrame(o_predict)
o_epc['predicted_agebandas']=result
```

[ ]:
```
print('...save results to drive...')
o_epc.to_csv(r'age_result_automl.csv')
```

[ ]:
```
```

Preliminary study: energy prediction using autosklearn

Import package

[ ]:
```
import pandas as pd
import numpy as np
```

[ ]:
```
# print autosklearn version
import autosklearn
print('autosklearn: %s' % autosklearn.__version__)
```

[ ]:
```
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from autosklearn.classification import AutoSklearnClassifier
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler, OneHotEncoder
import matplotlib.pyplot as plt
```

Load data

[ ]:
```
print('...now loading data...')
epc = pd.read_csv('processed_epc.csv', index_col=0)
```

[ ]:
```

```

Prepare data for model training

[ ]:
```
print('...now specifying x and y...')
inputs = ['Property type','Built form', 'Number habitable room', 'Number heated␣
↪room', 'Roof description', 'Walls description', 'Floor description', 'Lighting␣
↪description', 'Main heat', 'predicted age band']
X=epc.iloc[:,inputs]
y=epc['ENERGY_CONSUMPTION_CURRENT']
```

[ ]:
```
print('...now splitting data...')
# split into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)
```

[ ]:
```

```

Model training

[ ]:
```
print('...now defining model search...')
# define search
automl = AutoSklearnRegressor(
                            data_preprocessor = True,
```

```
                                resampling_strategy='cv',
                                resampling_strategy_arguments={'folds': 5})
```

[ ]:
```
print('...now fitting...')
# perform the search
automl.fit(X_train, y_train)
```

[ ]:
```

```

Model evaluation

[ ]:
```
print('...statistics...')
# summarize
print(automl.sprint_statistics())
```

[ ]:
```
print('...leaderboard...')
board = pd.DataFrame(automl.leaderboard(detailed = True))
board.to_csv('energy-leaderboard.csv')
print(automl.leaderboard(detailed = True))
```

[ ]:
```
print('...now printing models found...')
ensemble_dict = automl.show_models()
print(ensemble_dict)
```

[ ]:
```
print('...now evaluating...')
# evaluate best model
prediction = automl.predict(X_test)
acc = accuracy_score(y_test, prediction)
print("Accuracy: %.3f" % acc)
train_predictions = automl.predict(X_train)
print("Train R2 score:", sklearn.metrics.r2_score(y_train, train_predictions))
test_predictions = automl.predict(X_test)
print("Test R2 score:", sklearn.metrics.r2_score(y_test, test_predictions))
```

[ ]:
```
print('...now plotg performance over time ...')
automl.performance_over_time_.plot(
        x='Timestamp',
        kind='line',
        legend=True,
        title='Auto-sklearn accuracy over time',
        grid=True,
    )
plt.savefig('plot_energy.pdf')
plt.show()
```

[ ]:
```
plt.scatter(train_predictions, y_train, label="Train samples", c='#d95f02')
plt.scatter(test_predictions, y_test, label="Test samples", c='#7570b3')
```

```python
plt.xlabel("Predicted value")
plt.ylabel("True value")
plt.legend()
plt.plot([30, 400], [30, 400], c='k', zorder=0)
plt.xlim([30, 400])
plt.ylim([30, 400])
plt.tight_layout()
plt.savefig('automl/scatter_energy.pdf')
plt.show()
```

[ ]:

```python

```

Multi-input model for residential energy prediction

Import all packages required

[ ]:
```
import pandas as pd
import numpy as np
import glob
import cv2
import os
import sys
import PIL
from PIL import Image, ImageOps
import argparse
import locale
```

[ ]:
```
from sklearn.preprocessing import StandardScaler, OneHotEncoder,LabelEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.model_selection import train_test_split
```

[ ]:
```
print (PIL.__version__)
```

[ ]:
```
import tensorflow as tf
print('tensorflow: %s' % tf.__version__)
```

[ ]:
```
from tensorflow.keras.preprocessing.image import ImageDataGenerator,␣
↪img_to_array, load_img
import matplotlib.pyplot as plt
```

[ ]:
```
# import the necessary packages
from tensorflow.keras.models import Sequential
from tensorflow.keras.models import Model
from tensorflow.keras.layers import BatchNormalization
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Activation
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Input
from tensorflow.keras.layers import concatenate
from tensorflow.keras.optimizers.legacy import Adam
```

[ ]:
```
from sklearn.metrics import mean_absolute_percentage_error
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import r2_score
```

[ ]:
```python
import autokeras as ak
print('autokeras: %s' % ak.__version__)
```

[ ]:
```python
import shap
```

[ ]:
```python

```

Define the functios

functions for images

[ ]:
```python
#a function to calculate the size of the image
def area_cal(path):
    image = Image.open(path) #get the image
    w, h = image.size #get the width and height
    area = w * h #the cropped image is a rectangle so width X height for size
    return  (area)
```

[ ]:
```python
def crop_select(df, img_dir):
    final=[]
    for i in range(len(df)):
        #walk throgh all building reference number one by one
        building_ref = str(df.UPRN[i])
        print('now doing image number '+ str(i))

        crops = {'paths':[], 'size':[]}
        for m in range(4):  #assuming there won't be more than 5 housings in one␣
↪image
            img_filename = building_ref+'_'+str(m)+'.jpg'
            img_path = os.path.join(img_dir, img_filename)

            if os.path.exists(img_path)==True: #if is does exist
                area = area_cal(img_path)
                print('It does exists, now calculating the area')
                crops['size'].append(area)
                crops['paths'].append(img_path)

        if len(crops['paths']) > 1:
            print('now comparing sizes and selecting')
            sort = np.argmax(crops['size'])
            select = crops['paths'][sort]
            final.append(select)
            print('------------selection complete, move to next one----------')
        else:
            print('------------only one crop, move to next one-------------')
    return final
```

```
[ ]: def resize_img(inputpath, resize_width, resize_height):

         im = Image.open(inputpath)
         resizedim = im.resize((resize_width, resize_height),PIL.Image.Resampling.
      ↪NEAREST)
         rim = np.array(resizedim)

         return rim
```

```
[ ]:
```

Functions for models

```
[ ]: from tensorflow.keras.applications import ResNet50
     def create_resnet(regress=False):
         base_model = ResNet50(weights='imagenet', include_top=False,␣
      ↪input_shape=(224, 224, 3))

         # Freeze the pre-trained layers
         for layer in base_model.layers:
             layer.trainable = True

         # Add your own regression head
         x = base_model.output
         x = Flatten()(x)
         x = Dense(512, activation='relu')(x)
         x = Dense(4)(x)

         # check to see if the regression node should be added
         if regress:
             x = Dense(1)(x)

         resnet_model = Model(inputs=base_model.input, outputs=x)

         return resnet_model
```

```
[ ]: def create_cnn (width, height,depth, regress=False):
     # initialize the input shape and channel dimension
         inputShape = (height, width, depth)
         chanDim = -1
     # define the model input
         inputs = Input(shape=inputShape)

         #block 1
         x = Conv2D(64, (3, 3), activation = 'relu', padding='same')(inputs)
         x = Conv2D(64, (3, 3), strides = (4,4), activation = 'relu',␣
      ↪padding='same')(x)
```

```
    x = MaxPooling2D(pool_size = (2,2), strides = (2,2))(x)

    #block 2
    x = Conv2D(128, (3, 3), activation = 'relu', padding='same')(x)
    x = Conv2D(128, (3, 3), activation = 'relu', padding='same')(x)
    x = MaxPooling2D(pool_size = (2,2), strides = (2,2))(x)

    # flatten the volume, then FC => RELU => BN => DROPOUT
    x = Flatten()(x)
    x = Dense(4096, activation = 'relu')(x)
    x = Dense(4096, activation = 'relu')(x)
    x = BatchNormalization(axis=chanDim)(x)
    x = Dense(16, activation = 'relu')(x)
    x = Dropout(0.5)(x)
# apply another FC layer, this one to match the number of nodes coming out of␣
 ↪the MLP
    x = Dense(4)(x)

# check to see if the regression node should be added
    if regress:
        x = Dense(1)(x)
# construct the CNN
    model = Model(inputs, x)
# return the CNN
    return model
```

[ ]:
```
def create_mlp(dim, regress=False):
    # define our MLP network
    model = Sequential()
    model.add(Dense(128, input_dim=dim, activation='relu'))
    model.add(Dense(64, activation="relu"))
    model.add(Dense(32, activation="relu"))
    model.add(Dense(18, activation="relu"))
    model.add(Dense(4, activation="relu"))
    # check to see if the regression node should be added
    if regress:
        model.add(Dense(1))
        # return our model
    return model
```

[ ]:
```
def create_mlp_better(dim, regress=False):
    # define our MLP network
    model = Sequential()
    model.add(Dense(152, input_dim=dim, activation='relu'))
    model.add(Dense(32, activation="relu"))
    model.add(Dense(128, activation="relu"))
    model.add(Dense(4, activation="relu"))
```

```
    # check to see if the regression node should be added
    if regress:
        model.add(Dense(1))
        # return our model
    return model
```

Load data

Load csv

```
[ ]: epc = pd.read_csv(r'epc.csv', index_col=0)
```

```
[ ]: epc.info()
```

load image

```
[ ]: gsv = r'no_sky' #processed gsv images, cropped and watersheded
```

```
[ ]: #select the largest crop among all if multiple detection
     selected_img = crop_select(epc, gsv)
```

```
[ ]: len(selected_img)
```

```
[ ]: df = pd.DataFrame(selected_img)
     df.to_csv(r'nosky_b_selected.csv')
```

```
[ ]: #get respective epc from these selected epc
     selected = df.merge(epc, left_on = 'uprn', right_on = 'UPRN')
```

Split the spreadsheet here so both text and image are split

```
[ ]: (Train, Test) = train_test_split(selected, test_size=0.2, random_state=42)
```

```
[ ]:
```

Get image and tabular train and test from the split

Image

```
[ ]: train_img_path = Train.path.values
```

```
[ ]: test_img_path = Test.path.values
```

```
[ ]: def resize_img(path):
         resized_img = []
         for i in range(len(path)):
             img_path = path[i]
             img = Image.open(img_path)
```

```
        resize_width = 224
        resize_height = 224

        resizedim = img.resize((224, 224),PIL.Image.NEAREST)
        rim = np.array(resizedim)

        resized_img.append(rim)

    return resized_img
```

[ ]:
```
train_img = resize_img(train_img_path)
train_img = np.asarray(train_img)
train_img = train_img / 255.0
```

[ ]:
```
test_img = resize_img(test_img_path)
test_img = np.asarray(test_img)
test_img = test_img / 255.0
```

[ ]:
```

```

EPC

[ ]:
```
inputs = [11,12,32,43,44,50,53,56,62,65,71,85,92]
    #[12,13,33,44,45,51,54,57,67,72,86,93] #all variables energy related only
train_ = Train.iloc[:,inputs]
test_ = Test.iloc[:, inputs]
```

[ ]:
```
def csv_preprocess(X):
    for n in X.columns:
        if X.loc[:,n].dtype == 'object':
            processed_col = 'processed_'+n
            X.loc[:,processed_col] = LabelEncoder().fit_transform(X.loc[:,n])

        else:
            processed_col = 'processed_'+n
            X.loc[:,processed_col] = StandardScaler().fit_transform(X.loc[:,n].
 ↪values.reshape(-1,1)).astype(np.float32)

    return X
```

[ ]:
```
processed_train = csv_preprocess(train_)
processed_test = csv_preprocess(test_)
```

[ ]:
```
processed_train.to_csv(r'processed_epc.csv')
```

[ ]:
```
# for training data
processed = [13,14,15,16,17,18,19,20,21,22,23,24]
epc_train = processed_train.iloc[:, processed]
yepc_train = processed_train.processed_consumption

epc_test = processed_test.iloc[:, processed]
yepc_test = processed_test.processed_consumption
```

[ ]:
```

```

Training and Prediction

autokeras

autokeras for images

[ ]:
```
automodel_img = ak.ImageRegressor(overwrite=True, loss='mse', directory=r'..
 ↪\outputs\image',
                                  metrics=[tf.keras.metrics.
 ↪MeanAbsolutePercentageError()],
                                  max_trials=25)
```

[ ]:
```
automodel_img.fit(train_img[0:3500], yepc_train[0:3500], validation_data =␣
 ↪(test_img[0:100], yepc_test[0:100]), epochs = 60, batch_size = 16)
```

[ ]:
```
auto_bestimagemodel = automodel_img.export_model()
auto_bestimagemodel.summary()
```

[ ]:
```
pred_autoimg = auto_bestimagemodel.predict(img_test)
r2_autoimg = r2_score(yepc_test, pred_autoimg)
print ('r2_score: %s' % r2_autoimg)

mape_autoimg = mean_absolute_percentage_error(yepc_test, pred_autoimg)
print('mean_absolute_percentage_error: %s' % mape_autoimg)
```

[ ]:
```
try:
    auto_bestimagemodel.save("autoimage_barnsley", save_format="tf")
except Exception:
    auto_bestimagemodel.save("autoimage_barnsley.h5")
```

[ ]:
```
pred_ = auto_bestimagemodel.predict(img_test)
r2i_a = r2_score(yi_test, y_predi_a)
print ('r2_score: %s' % r2i_a)

mapei_a = mean_absolute_percentage_error(yi_test, y_predi_a)
print('mean_absolute_percentage_error: %s' % mapei_a)
```

[ ]:
```
auto_bestimagemodel.summary()
```

[ ]:
```
y_predi_a = auto_bestimagemodel.predict(img_test)
r2i_a = r2_score(yi_test, y_predi_a)
print ('r2_score: %s' % r2i_a)

mapei_a = mean_absolute_percentage_error(yi_test, y_predi_a)
print('mean_absolute_percentage_error: %s' % mapei_a)
```

[ ]:
```

```

autokeras for csv

[ ]:
```
automodel_csv = ak.StructuredDataRegressor(overwrite=True, loss='mse',␣
↪directory=r'..\outputs\epc',
                                    metrics=[tf.keras.metrics.
↪MeanAbsolutePercentageError()])
```

[ ]:
```
automodel_csv.fit(epc_train, yepc_train, validation_data=(epc_test,␣
↪yepc_test),epochs = 100, batch_size = 32)
```

[ ]:
```
auto_bestcsvmodel = automodel_csv.export_model()
auto_bestcsvmodel.summary()
```

[ ]:
```
pred_autocsv = automodel_csv.predict(epc_train)
r2_autocsv = r2_score(yepc_train, pred_autocsv)
print ('r2_score: %s' % r2_autocsv)

mape_autocsv = mean_absolute_percentage_error(yepc_train, pred_autocsv)
print('mean_absolute_percentage_error: %s' % mape_autocsv)
```

[ ]:
```
try:
    auto_bestcsvmodel.save("epc_barnsley", save_format="tf")
except Exception:
    auto_bestcsvmodel.save("epc_barnsley_h5.h5")
```

[ ]:
```

```

Manual

MLP

[ ]:
```
model_mlp1 = create_mlp_better(epc_train.shape[1], regress=True)
opt = Adam(learning_rate=1e-3, decay=1e-3 / 200)
model_mlp1.compile(loss="mse", optimizer=opt)
```

[ ]:
```
# train the model
print("[INFO] training model...")
```

```
model_mlp1.fit(x=epc_train, y=yepc_train, validation_data=(epc_test,␣
↪yepc_test),epochs=100, batch_size=32)
```

[ ]:
```
pred_mlp1 = model_mlp1.predict(epc_test)
r2_mlp1 = r2_score(yepc_test, pred_mlp1)
print ('r2_score: %s' % r2_mlp1)

mape_mlp1 = mean_absolute_percentage_error(yepc_test, pred_mlp1)
print('mean_absolute_percentage_error: %s' % mape_mlp1)
```

[ ]:
```

```

[ ]:
```
try:
    model_mlp1.save("mlp_barnsley", save_format="tf")
except Exception:
    model_mlp1.save("mlp_barnsley_h5.h5")
```

[ ]:
```
mlp_explainer = shap.KernelExplainer(model_mlp1.predict,epc_train)
```

[ ]:
```
mlp_shap_values = mlp_explainer.shap_values(epc_test,nsamples=100)
```

[ ]:
```
features=epc_test.columns.values.tolist()
```

[ ]:
```
fig = shap.
↪summary_plot(mlp_shap_values,epc_test,feature_names=features,show=False)
plt.savefig(r'mlp.png')
plt.savefig(r'mlp.pdf')
```

[ ]:
```

```

resnet

[ ]:
```
model_resnet = create_resnet(regress=True)
```

[ ]:
```
# Compile the model
opt = Adam(learning_rate=1e-3, decay=1e-3 / 200)
model_resnet.compile(loss="mse", optimizer=opt,␣
↪metrics=['mean_absolute_percentage_error'])
```

[ ]:
```
model_resnet.fit(train_img[0:3500], yepc_train[0:3500],
                 validation_data=(test_img[0:200],yepc_test[0:200]),
                 batch_size=32, epochs=100)
```

[ ]:
```
pred_resnet = model_resnet.predict(test_img)
ra_resnet = r2_score(yepc_test, pred_resnet)
print ('r2_score: %s' % ra_resnet)

mape_resnet = mean_absolute_percentage_error(yepc_test, pred_resnet)
```

```
print('mean_absolute_percentage_error: %s' % mape_resnet)
```

[ ]:
```
try:
    model_resnet.save("resnet_barnsley", save_format="tf")
except Exception:
    model_resnet.save("resnet_barnsley_h5.h5")
```

[ ]:
```

```

[ ]:
```
explainer = shap.DeepExplainer(model=model_resnet, data=train_img[0:220])
```

[ ]:
```
for i in range(0,100,5):
    print('first step')
    shap_values = explainer.shap_values(test_img[i:i+6])
    print('second step')
    shap.image_plot(shap_values,test_img[i:i+6],show=False)
    print('save')
    plt.savefig(r'{}.png'.format(i))
    plt.savefig(r'{}.pdf'.format(i))
```

[ ]:
```

```

cnn

[ ]:
```
model_cnn = create_cnn(224, 224, 3, regress=True)
```

[ ]:
```
# Compile the model
opt = Adam(learning_rate=1e-3, decay=1e-3 / 200)
model_cnn.compile(loss="mse", optimizer=opt,␣
 ↪metrics=['mean_absolute_percentage_error'])
```

[ ]:
```
model_cnn.fit(train_img[0:3500], yepc_train[0:3500],
                 validation_data=(test_img[0:200],yepc_test[0:200]),
                 batch_size=32, epochs=100)
```

[ ]:
```
pred_cnn = model_cnn.predict(test_img)
ra_cnn = r2_score(yepc_test, pred_cnn)
print ('r2_score: %s' % ra_cnn)

mape_cnn = mean_absolute_percentage_error(yepc_test, pred_cnn)
print('mean_absolute_percentage_error: %s' % mape_cnn)
```

[ ]:
```
try:
    model_cnn.save("cnn_barnsley", save_format="tf")
except Exception:
    model_cnn.save("cnn_barnsley_h5.h5")
```

[ ]:
```
explainer = shap.DeepExplainer(model=model_cnn, data=train_img[0:220])
```

[ ]:
```
for i in range(0,100,5):
    print('first step')
    shap_values = explainer.shap_values(train_img[i:i+6])
    print('second step')
    shap.image_plot(shap_values,train_img[i:i+6],show=False)
    print('save')
    plt.savefig(r'{}.png'.format(i))
    plt.savefig(r'{}.pdf'.format(i))
```

[ ]:
```
shap.explainers._deep.deep_tf.op_handlers["AddV2"] = shap.explainers._deep.
 ↪deep_tf.passthrough #this solves the "shap_ADDV2" problem but another one will␣
 ↪appear
shap.explainers._deep.deep_tf.op_handlers["FusedBatchNormV3"] = shap.explainers.
 ↪_deep.deep_tf.passthrough #this solves the next problem which allows you to␣
 ↪run the DeepExplainer.
```

[ ]:
```

```

Create the multimodal model based on best models

[ ]:
```
csv_modal = create_mlp_better(epc_train.shape[1], regress=False)
img_modal = create_resnet(regress=False)
```

[ ]:
```
for i, layer in enumerate(csv_modal.layers):
    layer._name = 'csv_layer_' + str(i)

for i, layer in enumerate(img_modal.layers):
    layer._name = 'img_layer_' + str(i)
```

[ ]:
```
combinedInput = concatenate([csv_modal.output, img_modal.output])

#add the final layers
multi_x = Dense(32, activation = 'relu')(combinedInput)
multi_x = Dense(4)(multi_x)
multi_x = Dense(1)(multi_x)

multimodal = Model(inputs=[csv_modal.input, img_modal.input], outputs=multi_x)
```

[ ]:
```
multimodal.summary()
```

[ ]:
```
#opt = Adam(learning_rate=1e-3)
multimodal.compile(loss="mse", optimizer='adam', metrics=[tf.keras.metrics.
 ↪MeanAbsolutePercentageError()])
```

```
[ ]: multimodal.fit(x={'dense_input':epc_train, 'img_layer_0':
     ↪train_img},y=yepc_train,epochs=80, batch_size=32)
```

```
[ ]: pred_multimodal = multimodal.predict({'dense_input':epc_test, 'img_layer_0':
     ↪test_img})
     r2_multimodal = r2_score(yepc_test, pred_multimodal)
     print ('r2_score: %s' % r2_multimodal)

     mape_multimodal = mean_absolute_percentage_error(yepc_test, pred_multimodal)
     print('mean_absolute_percentage_error: %s' % mape_multimodal)
```

```
[ ]: pred_multimodal = multimodal.predict({'dense_3_input':epc_train, 'img_layer_0':
     ↪train_img})
     r2_multimodal = r2_score(yepc_train, pred_multimodal)
     print ('r2_score: %s' % r2_multimodal)

     mape_multimodal = mean_absolute_percentage_error(yepc_train, pred_multimodal)
     print('mean_absolute_percentage_error: %s' % mape_multimodal)
```

```
[ ]: multimodal.save(r"multimodal_b.keras", save_format="tf")
```

```
[ ]:
```

load the models

```
[ ]: from tensorflow.keras.models import load_model
```

```
[ ]: img_best = load_model(r'multimodal_b.keras')
     #img_best.name = 'img_model'
```

```
[ ]: mlp = create_mlp_better(epc_train.shape[1], regress=False)
```

```
[ ]: #rename the layers to avoid error
     for i, layer in enumerate(mlp.layers):
         layer._name = 'csv_layer_' + str(i)

     for i, layer in enumerate(img_best.layers):
         layer._name = 'img_layer_' + str(i)
```

```
[ ]:
```

pop the last layer and add merge

```
[ ]: def slice_model(loaded_model):
         # remove the last layer
         sliced_loaded_model = Sequential(loaded_model.layers[:-1])
```

```
    # set trainable=False for the layers from loaded_model
    for layer in sliced_loaded_model.layers:
        layer.trainable = True

    # add new layers
    sliced_loaded_model.add(Dense(32, activation='relu'))  # trainable=True is␣
↪default
    sliced_loaded_model.add(Dense(1))

    return sliced_loaded_model
```

[ ]:
```
#construt each model
img_slice = slice_model(img_best)

#combine the result from mlp and cnn as the new input
combinedInput = concatenate([mlp.output, img_best.output])

#add the final layers
x = Dense(32, activation = 'relu')(combinedInput)
x = Dense(4)(x)
x = Dense(1)(x)

multimodal = Model(inputs=[mlp.input, img_slice.input], outputs=x)
```

[ ]:

[ ]:

Explain multimodal

[ ]:
```
multimodal.summary()
```

[ ]:

[ ]:
```
img_modal = create_resnet(regress=False)

#add the final layers
img_x = Dense(32, activation = 'relu')(img_modal.output)
img_x = Dense(4)(img_x)
img_x = Dense(1)(img_x)

img_stream = Model(inputs=img_modal.input, outputs=img_x)
```

[ ]:
```
for i, layer in enumerate(img_stream.layers):
    layer._name = 'img_layer_' + str(i)
```

[ ]:
```
img_stream.summary()
```

[ ]:
```
img_stream.get_layer('img_layer_178')._name = 'dense_20'
img_stream.get_layer('img_layer_179')._name = 'dense_21'
img_stream.get_layer('img_layer_180')._name = 'dense_22'
```

[ ]:
```
def append_weights(trained_model, untrained_model):
    for trained_layer, untrained_layer in zip(trained_model.layers,␣
↪untrained_model.layers):
        if trained_layer.name == untrained_layer.name:
            if isinstance(trained_layer, tf.keras.layers.Conv2D) or␣
↪isinstance(trained_layer, tf.keras.layers.Dense):
                untrained_layer.set_weights(trained_layer.get_weights())
            elif isinstance(trained_layer, tf.keras.layers.BatchNormalization):
                untrained_layer.set_weights(trained_layer.get_weights())
                untrained_layer.momentum = trained_layer.momentum
                untrained_layer.epsilon = trained_layer.epsilon
```

[ ]:
```
append_weights(img_best, img_stream)
```

[ ]:
```
m_img_explainer = shap.DeepExplainer(model=img_stream, data=train_img[0:220])
```

[ ]:
```
img_stream.save(r"img_split.keras", save_format="tf")
```

[ ]:
```
shap.explainers._deep.deep_tf.op_handlers["AddV2"] = shap.explainers._deep.
↪deep_tf.passthrough #this solves the "shap_ADDV2" problem but another one will␣
↪appear
shap.explainers._deep.deep_tf.op_handlers["FusedBatchNormV3"] = shap.explainers.
↪_deep.deep_tf.passthrough #this solves the next problem which allows you to␣
↪run the DeepExplainer.
```

[ ]:
```
for i in range(0,100,5):
    print('first step')
    shap_values_i = m_img_explainer.shap_values(test_img[i:i+5])
    print('second step')
    shap.image_plot(shap_values_i,test_img[i:i+5],show=False)
    print('save')
    plt.savefig(r'{}.png'.format(i))
    plt.savefig(r'{}.pdf'.format(i))
```

[ ]:
```

```

csv modal

[ ]:
```
csv_modal = create_mlp_better(epc_train.shape[1], regress=False)

#add the final layers
```

```
csv_x = Dense(32, activation = 'relu')(csv_modal.output)
csv_x = Dense(4)(csv_x)
csv_x = Dense(1)(csv_x)

csv_stream = Model(inputs=csv_modal.input, outputs=csv_x)
```

[ ]:
```
for i, layer in enumerate(csv_stream.layers):
    layer._name = 'csv_layer_' + str(i)
```

[ ]:
```
csv_stream.summary()
```

[ ]:
```
csv_stream.get_layer('csv_layer_0')._name = 'dense_14_input'
csv_stream.get_layer('csv_layer_1')._name = 'csv_layer_0'
csv_stream.get_layer('csv_layer_2')._name = 'csv_layer_1'
csv_stream.get_layer('csv_layer_3')._name = 'csv_layer_2'
csv_stream.get_layer('csv_layer_4')._name = 'csv_layer_3'
csv_stream.get_layer('csv_layer_5')._name = 'dense_20'
csv_stream.get_layer('csv_layer_6')._name = 'dense_21'
csv_stream.get_layer('csv_layer_7')._name = 'dense_22'
```

[ ]:
```
append_weights(multimodal, csv_stream)
```

[ ]:
```
csv_explainer = shap.KernelExplainer(csv_stream.predict,epc_train)
```

[ ]:
```
epc_test1 = pd.read_csv(r'epc.csv', index_col=0)
```

[ ]:
```
epc_test1.info()
```

[ ]:
```
test_shap_values = csv_explainer.shap_values(epc_test1, nsamples = 100)
```

[ ]:
```
test_shap_values1 = pd.DataFrame(test_shap_values[0])
```

[ ]:
```
test_shap_values1.to_csv(r'I:\My␣
↪Drive\phd\year_2\redo_year3\outputs\epc\csv_shap_selected1.csv')
```

[ ]:
```
csv_shap_values = csv_explainer.shap_values(epc_test[0:800],nsamples=100)
```

[ ]:
```
features=epc_test.columns.values.tolist()
```

[ ]:
```
fig = shap.
↪summary_plot(csv_shap_values,epc_test,feature_names=features,show=False)
plt.savefig(r'csv_multi.png')
plt.savefig(r'csv_multi.pdf')
```

[ ]:
```
csv_shap_values[0][0]
```

[ ]:
```
test = pd.DataFrame(csv_shap_values[0])
```

[ ]:
```
test.to_csv(r'multi_csv_shap_values.csv')
```

[ ]:
```
shap.dependence_plot('processed_FLOOR_DESCRIPTION',␣
↪shap_values=csv_shap_values[0], features=epc_test[0:800], show=False,␣
↪interaction_index=None)
```

[ ]:
```
sample_ind = 20
```

[ ]:
```
shap.partial_dependence_plot(
    features[0], csv_stream.predict, epc_train, model_expected_value=True,
    feature_expected_value=True, ice=False,
    shap_values=csv_shap_values[0]
)
```

[ ]:
```

```

[ ]:
```
for f_name in features:
    fig2 = shap.dependence_plot(f_name, shap_values=csv_shap_values[0],␣
↪features=epc_test[0:800], show=False, interaction_index=None)
    plt.savefig(r'ddp_{}.png'.format(f_name))
    plt.savefig(r'ddp_{}.pdf'.format(f_name))
```

[ ]:
```
shap.summary_plot(shap_values=csv_shap_values[0], features=epc_test[0:800],␣
↪feature_names=features,show=True)
```

[ ]:
```
features[0]
```

[ ]:
```
range(len(epc_test[features[0]].unique()))
```

[ ]:
```
expected_value = csv_explainer.expected_value
if isinstance(expected_value, list):
    expected_value = expected_value[1]
```

[ ]:
```
shap.decision_plot(expected_value, csv_shap_values[0], features)
```

[ ]:
```
v_fig = shap.plots.violin(csv_shap_values[0],features,show=False)
plt.savefig(r'violin.png')
plt.savefig(r'violin.pdf')
```

[ ]:
```

```

Transfer learning using Autokeras best models

Import all packages required

[ ]:
```
import pandas as pd
import numpy as np
import glob
import cv2
import os
import sys
import PIL
from PIL import Image, ImageOps
import argparse
import locale
```

[ ]:
```
from sklearn.preprocessing import StandardScaler, OneHotEncoder,LabelEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.model_selection import train_test_split
```

[ ]:
```
print (PIL.__version__)
```

[ ]:
```
import tensorflow as tf
print('tensorflow: %s' % tf.__version__)
```

[ ]:
```
from tensorflow.keras.preprocessing.image import ImageDataGenerator,␣
↪img_to_array, load_img
import matplotlib.pyplot as plt
```

[ ]:
```
# import the necessary packages
from tensorflow.keras.models import Sequential
from tensorflow.keras.models import Model
from tensorflow.keras.layers import BatchNormalization
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Activation
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Input
from tensorflow.keras.layers import concatenate
from tensorflow.keras.optimizers.legacy import Adam
```

[ ]:
```
from sklearn.metrics import mean_absolute_percentage_error
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import r2_score
```

[ ]:
```python
import autokeras as ak
print('autokeras: %s' % ak.__version__)
```

[ ]:
```python
import shap
```

[ ]:
```python

```

Define the functios

functions for images

[ ]:
```python
#a function to calculate the size of the image
def area_cal(path):
    image = Image.open(path) #get the image
    w, h = image.size #get the width and height
    area = w * h #the cropped image is a rectangle so width X height for size
    return  (area)
```

[ ]:
```python
def crop_select(df, img_dir):
    final=[]
    for i in range(len(df)):
        #walk throgh all building reference number one by one
        building_ref = str(df.UPRN[i])
        print('now doing image number '+ str(i))

        crops = {'paths':[], 'size':[]}
        for m in range(4):  #assuming there won't be more than 5 housings in one␣
↪image
            img_filename = building_ref+'_'+str(m)+'.jpg'
            img_path = os.path.join(img_dir, img_filename)

            if os.path.exists(img_path)==True: #if is does exist
                area = area_cal(img_path)
                print('It does exists, now calculating the area')
                crops['size'].append(area)
                crops['paths'].append(img_path)

        if len(crops['paths']) > 1:
            print('now comparing sizes and selecting')
            sort = np.argmax(crops['size'])
            select = crops['paths'][sort]
            final.append(select)
            print('------------selection complete, move to next one----------')
        else:
            print('------------only one crop, move to next one-------------')
    return final
```

[ ]:
```python
def resize_img(inputpath, resize_width, resize_height):

    im = Image.open(inputpath)
    resizedim = im.resize((resize_width, resize_height),PIL.Image.Resampling.
 ↪NEAREST)
    rim = np.array(resizedim)

    return rim
```

[ ]:
```python

```

Load data

Load csv

[ ]:
```python
#b_gsv_epc = pd.read_csv(r'barnsley_epc.csv', index_col=0)
b_marvel_epc = pd.read_csv(r'I:\My␣
 ↪Drive\phd\year_3\1-transfer_learning\data\barnsley\barnsley_certificates.csv',␣
 ↪index_col=0)
#d_marvel_epc = pd.read_csv(r'I:\My␣
 ↪Drive\phd\year_3\1-transfer_learning\data\doncaster\doncaster_epc.csv',␣
 ↪index_col=0)
#m_marvel_epc = pd.read_csv(r'I:\My␣
 ↪Drive\phd\year_3\1-transfer_learning\data\Merthyr\EPC\Merthyr_epc.csv',␣
 ↪index_col=0)
```

[ ]:
```python

```

load image

[ ]:
```python
gsv = r'no_sky'
```

[ ]:
```python
#df1 = pd.read_csv(r'nosky_b_selected2.csv', index_col = 0)
b_img = pd.read_csv(r'barnsley_crop_path1.csv', index_col=0)
#d_img = pd.read_csv(r'doncaster_crop_path1.csv', index_col =0)
m_img = pd.read_csv(r'Merthyr_crop_path1.csv', index_col=0)
```

[ ]:
```python
def resize_img(path):
    resized_img = []
    for i in range(len(path)):
        img_path = path[i]
        img = Image.open(img_path)

        resize_width = 224
        resize_height = 224

        resizedim = img.resize((224, 224),PIL.Image.NEAREST)
        rim = np.array(resizedim)
```

```
        resized_img.append(rim)
    resized_img = np.asarray(resized_img)
    resized_img = resized_img / 255.0

    return resized_img
```

[ ]:
```
selected = b_img.merge(b_marvel_epc, left_on = 'uprn', right_on = 'UPRN')
```

[ ]:
```
selected.info()
```

[ ]:
```
def get_data_split(df, epc):
    print('--------merge------------')
    selected = df.merge(epc, left_on = 'uprn', right_on = 'UPRN')

    print('--------split---------')
    (Train, Test) = train_test_split(selected, test_size=0.2, random_state=42)
    print('Length of training data is: %' % len(Train))
    print('length of testing data is: %' % len(Test))
    train_img_path = Train.imgpath.values
    test_img_path = Test.imgpath.values

    print
    train_img = resize_img(train_img_path)
    test_img = resize_img(test_img_path)
    print ('Length of training image is: %' % len(train_img))
    print ('Length of training image is: %' % len(test_img))

    inputs = [11,12,32,43,44,50,53,56,62,65,71,85,92] #all variables energy␣
↪related only
    train_ = Train.iloc[:,inputs]
    test_ = Test.iloc[:, inputs]

    return train_img, test_img, train_, test_
```

[ ]:
```
btrain_img,btest_img,btrain_,btest_ = get_data_split(b_img, b_marvel_epc)
dtrain_img,dtest_img,dtrain_,dtest_ = get_data_split(d_img, d_marvel_epc)
mtrain_img,mtest_img,mtrain_,btest_ = get_data_split(m_img, m_marvel_epc)
```

[ ]:
```
len(b_img)
len(b_marvel_epc)
len(btrain_img)
len(btrain_)
```

[ ]:
```
len(d_img)
len(d_marvel_epc)
len(dtrain_img)
len(dtrain_)
```

[ ]:
```
len(m_img)
len(m_marvel_epc)
len(mtrain_img)
len(mtrain_)
```

[ ]:
```

```

[ ]:
```
(Train, Test) = train_test_split(selected, test_size=0.2, random_state=42)
```

[ ]:
```
Train.info()
```

[ ]:
```
Test.info()
```

[ ]:
```

```

Get image and tabular train and test from the split result

Image

[ ]:
```
train_img_path = Train.path.values
```

[ ]:
```
test_img_path = Test.path.values
```

[ ]:
```
def resize_img(path):
    resized_img = []
    for i in range(len(path)):
        img_path = path[i]
        img = Image.open(img_path)

        resize_width = 224
        resize_height = 224

        resizedim = img.resize((224, 224),PIL.Image.NEAREST)
        rim = np.array(resizedim)

        resized_img.append(rim)
        resized_img = np.asarray(resized_img)
        resized_img = resized_img / 255.0

    return resized_img
```

[ ]:
```
train_img = resize_img(train_img_path)
train_img = np.asarray(train_img)
train_img = train_img / 255.0
```

[ ]:
```
test_img = resize_img(test_img_path)
test_img = np.asarray(test_img)
```

```
test_img = test_img / 255.0
```

[ ]:
```
len(train_img)
```

[ ]:
```
len(test_img)
```

[ ]:
```

```

Process using the mapping csv

[ ]:
```
mapping_csv = pd.read_csv(r"encoded_barnsley1.csv", index_col=0)
```

[ ]:
```
mapping_dict_per_column = {}
for column in mapping_csv.columns:
    if column.startswith("processed_"):
        pass
    else:
        encoded_name = 'processed_'+column
        mapping_dict_per_column[column] = dict(zip(mapping_csv[column],␣
↪mapping_csv[encoded_name]))
```

[ ]:
```
def mapping_process(df, mapping_dict_per_column):
    for column in df.select_dtypes(include=["object"]).columns:
        if column in mapping_dict_per_column:
            encoded_name = 'processed_'+column
            df[encoded_name] = df[column].map(mapping_dict_per_column[column])
    for column in df.select_dtypes(include=['int64', 'float64']).columns:
        #if column in mapping_dict_per_column:
        encoded_name = 'processed_'+column
        df[encoded_name] = StandardScaler().fit_transform(X.loc[:,n].values.
↪reshape(-1,1)).astype(np.float32)

    return df
```

[ ]:
```
for column in train_.columns:
        if column in mapping_dict_per_column:
            encoded_name = 'processed_'+column
            train_pro=train_
            train_pro[encoded_name] = train_[column].
↪map(mapping_dict_per_column[column])
```

[ ]:
```
train_pro.info()
```

[ ]:
```
train_processed = mapping_process(train_, mapping_dict_per_column)
```

[ ]:
```
train_processed.info()
```

[ ]:
```
# Encode the categorical variables using the mapping dictionary
for column in train_.select_dtypes(include=["object"]).columns:
    if column in mapping_dict_per_column:
        encoded_name = 'processed_'+column
        train_[encoded_name] = train_[column].
 ↪map(mapping_dict_per_column[column])
```

[ ]:
```
# Encode the categorical variables using the mapping dictionary
for column in test_.select_dtypes(include=["object"]).columns:
    if column in mapping_dict_per_column:
        encoded_name = 'processed_'+column
        test_[encoded_name] = test_[column].map(mapping_dict_per_column[column])
```

[ ]:
```

```

select processed input

[ ]:
```
# for training data
processed = [13,14,15,16,17,18,19,20,21,22,23,24]
epc_train = train_.iloc[:, processed]
yepc_train = train_.processed_consumption

epc_test = test_.iloc[:, processed]
yepc_test = test_.processed_consumption
```

[ ]:
```
epc_train.info()
```

[ ]:
```

```

[ ]:
```
model_resnet = create_resnet(regress=True)
```

[ ]:
```
# Compile the model
opt = Adam(learning_rate=1e-3, decay=1e-3 / 200)
model_resnet.compile(loss="mse", optimizer=opt,␣
 ↪metrics=['mean_absolute_percentage_error'])
```

[ ]:
```
model_resnet.fit(train_img[0:3500], yepc_train[0:3500],
                 validation_data=(test_img[0:200],yepc_test[0:200]),
                 batch_size=32, epochs=100)
```

[ ]:
```
pred_resnet = model_resnet.predict(test_img)
ra_resnet = r2_score(yepc_test, pred_resnet)
print ('r2_score: %s' % ra_resnet)

mape_resnet = mean_absolute_percentage_error(yepc_test, pred_resnet)
print('mean_absolute_percentage_error: %s' % mape_resnet)
```

```
[ ]: try:
         model_resnet.save("resnet_barnsley", save_format="tf")
     except Exception:
         model_resnet.save("resnet_barnsley_h5.h5")
```

```
[ ]:
```

```
[ ]: explainer = shap.DeepExplainer(model=model_resnet, data=train_img[0:220])
```

```
[ ]: for i in range(0,100,5):
         print('first step')
         shap_values = explainer.shap_values(test_img[i:i+6])
         print('second step')
         shap.image_plot(shap_values,test_img[i:i+6],show=False)
         print('save')
         plt.savefig(r'G:\My␣
      ↪Drive\phd\year_2\redo_year3\outputs\explain\resnet\png\{}.png'.format(i))
         plt.savefig(r'G:\My␣
      ↪Drive\phd\year_2\redo_year3\outputs\explain\resnet\pdf\{}.pdf'.format(i))
```

```
[ ]:
```

```
[ ]:
```

```
[ ]: model_cnn = create_cnn(224, 224, 3, regress=True)
```

```
[ ]: # Compile the model
     opt = Adam(learning_rate=1e-3, decay=1e-3 / 200)
     model_cnn.compile(loss="mse", optimizer=opt,␣
      ↪metrics=['mean_absolute_percentage_error'])
```

```
[ ]: model_cnn.fit(train_img[0:3500], yepc_train[0:3500],
                   validation_data=(test_img[0:200],yepc_test[0:200]),
                   batch_size=32, epochs=100)
```

```
[ ]: pred_cnn = model_cnn.predict(test_img)
     ra_cnn = r2_score(yepc_test, pred_cnn)
     print ('r2_score: %s' % ra_cnn)

     mape_cnn = mean_absolute_percentage_error(yepc_test, pred_cnn)
     print('mean_absolute_percentage_error: %s' % mape_cnn)
```

```
[ ]: try:
         model_cnn.save("cnn_barnsley", save_format="tf")
     except Exception:
         model_cnn.save("cnn_barnsley_h5.h5")
```

```
[ ]: explainer = shap.DeepExplainer(model=model_cnn, data=train_img[0:220])
```

```
[ ]: for i in range(0,100,5):
         print('first step')
         shap_values = explainer.shap_values(train_img[i:i+6])
         print('second step')
         shap.image_plot(shap_values,train_img[i:i+6],show=False)
         print('save')
         plt.savefig(r'G:\My Drive\phd\year_2\redo_year3\outputs\explain\cnn\png\{}.
      ↪png'.format(i))
         plt.savefig(r'G:\My Drive\phd\year_2\redo_year3\outputs\explain\cnn\pdf\{}.
      ↪pdf'.format(i))
```

```
[ ]: shap.explainers._deep.deep_tf.op_handlers["AddV2"] = shap.explainers._deep.
      ↪deep_tf.passthrough #this solves the "shap_ADDV2" problem but another one will␣
      ↪appear
     shap.explainers._deep.deep_tf.op_handlers["FusedBatchNormV3"] = shap.explainers.
      ↪_deep.deep_tf.passthrough #this solves the next problem which allows you to␣
      ↪run the DeepExplainer.
```

```
[ ]:
```

```
[ ]: from tensorflow.keras.models import load_model
```

```
[ ]: csv_load = load_model(r'bestepc_barnsley')
```

```
[ ]: csv_load.fit(epc_train, yepc_train)
```

```
[ ]: tf.compat.v1.tables_initializer(
         name='init_all_tables')
```

```
[ ]: epc_test.info()
```

```
[ ]: y_predt_a = auto_bestcsvmodel.predict(epc_test)
     r2t_a = r2_score(yepc_test, y_predt_a)
     print ('r2_score: %s' % r2t_a)

     mapet_a = mean_absolute_percentage_error(yepc_test, y_predt_a)
     print('mean_absolute_percentage_error: %s' % mapet_a)
```

```
[ ]:
```

```
[ ]:
```

Transfer learning

[ ]:
```
loaded_model = tf.keras.models.load_model(r'G:\My␣
↪Drive\phd\year_2\redo_year3\code\multimodal_barnsley')
```

[ ]:
```
loaded_model.summary()
```

[ ]:
```
```

no transfer learning

[ ]:
```
bpred_multimodal = multimodal.predict({'dense_3_input':epc_test, 'img_layer_0':
↪test_img})
br2_multimodal = r2_score(yepc_test, bpred_multimodal)
print ('r2_score: %s' % br2_multimodal)

bmape_multimodal = mean_absolute_percentage_error(yepc_test, bpred_multimodal)
print('mean_absolute_percentage_error: %s' % bmape_multimodal)
```

[ ]:
```
```

with transfer learning

[ ]:
```
#freeze the layers
for i in range(len(loaded_model.layers[:-4])):
    loaded_model.layers[i].trainable = False
```

[ ]:
```
import tensorflow_addons as tfa
```

[ ]:
```
optimizers = [
    tf.keras.optimizers.Adam(learning_rate=1e-4),
    tf.keras.optimizers.Adam(learning_rate=1e-2)
]
optimizers_and_layers = [(optimizers[0], loaded_model.layers[:-4]),␣
↪(optimizers[1], loaded_model.layers[-4:])]
optimizer = tfa.optimizers.MultiOptimizer(optimizers_and_layers)
loaded_model.compile(optimizer=optimizer, loss="mse")
```

[ ]:
```
loaded_model.fit()
```

[ ]:
```
```