FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Overcoming Data Scarcity in Load Forecasting: A Transfer Learning Approach for Commercial Buildings

Felipe Dantas do Carmo

Master in Electrical and Computer Engineering

Supervisor: Tiago André Soares

July 5, 2023

Overcoming Data Scarcity in Load Forecasting: A Transfer Learning Approach for Commercial Buildings

Felipe Dantas do Carmo

Master in Electrical and Computer Engineering

July 5, 2023

Recognitions

This work was carried out with the support of INESC-TEC and supervised at this institution by Tiago André Soares and Wellington Fonseca.

Resumo

A previsão de carga é fundamental na gestão de energia de edifícios, uma vez que previsões precisas possibilitam o consumo eficiente de energia, a gestão do lado da demanda e contribuem para os esforços de descarbonização. Contudo, os modelos de aprendizado de máquina frequentemente enfrentam desafios relacionados à escassez de dados, em particular no contexto de edifícios com limitada disponibilidade informação. Esta tese explora a eficácia do uso da transferência de aprendizado na previsão de carga em edifícios, visando enfrentar a questão da escassez de dados e aprimorar a precisão das previsões, preservando a eficiência computacional.

Esta tese abrange uma revisão detalhada da literatura sobre abordagens tradicionais e de aprendizado de máquina para previsão de carga, concentrando-se especialmente em técnicas de aprendizado profundo e aplicações de transferência de aprendizado. O estudo de caso enfoca um conjunto de edifícios virtuais situados nas instalações do INESC TEC (Porto, Portugal). Um modelo pré-treinado é utilizado como ponto de partida para o desenvolvimento de novos modelos com camadas adicionais, os quais posteriormente passam por um processo de ajuste fino usando um conjunto de dados menor. O desempenho da abordagem de transferência de aprendizado é avaliado e comparado a modelos treinados com diferentes níveis de disponibilidade de dados.

A metodologia de pesquisa engloba a coleta e pré-processamento de dados, desenvolvimento e ajuste de modelos de aprendizado de máquina, bem como a avaliação de desempenho empregando diversas métricas e visualizações. Os resultados indicam que a transferência de aprendizado pode diminuir consideravelmente o tempo de treinamento, mantendo níveis de precisão similares aos de modelos treinados com abundância de dados. Ademais, a abordagem apresenta um desempenho superior em comparação com modelos treinados com escassez de dados. O estudo também ressalta a aplicabilidade da transferência de aprendizado em condições não ideais, como quando o alvo possuí distribuição probabilistica dos dados muito diferente da fonte.

A presente tese contribui para o desenvolvimento de modelos de previsão de carga mais eficientes e precisos na gestão de energia de edifícios. Concentrando-se na generalização e escalabilidade de aplicações baseadas em dados, este trabalho promove os benefícios mais amplos da descarbonização e estimula o progresso das operações sustentáveis de edifícios. Os resultados da pesquisa têm implicações para gestores de instalações, prestadores de serviços de energia e formuladores de políticas públicas, à medida que buscam otimizar o consumo de energia, reduzir as emissões de gases de efeito estufa e aprimorar a sustentabilidade geral do ambiente construído.

Abstract

Load forecasting is of paramount importance in building energy management, as accurate predictions facilitate efficient energy consumption, demand-side management, and contribute to decarbonization efforts. However, data-driven models often face challenges due to data scarcity, especially in the context of buildings with limited data availability. This thesis investigates the effectiveness of transfer learning for load forecasting in buildings, aiming to address the issue of data scarcity and enhance forecasting accuracy while maintaining computational efficiency.

The thesis presents a comprehensive literature review on traditional and data-driven approaches to load forecasting, with a particular focus on deep learning techniques and transfer learning applications. The case study involves a group of Virtual Building (VB)s located at the INESC TEC facilities (Porto, Portugal). A pre-trained model serves as the basis for developing new models with additional layers, which are subsequently fine-tuned using a smaller dataset. The performance of the transfer learning approach is assessed and compared to models trained with both abundant and scarce data availability.

The research methodology includes data collection and preprocessing, model development and fine-tuning, as well as performance evaluation using multiple metrics and visualizations. The results demonstrate that transfer learning can significantly reduce training time while maintaining accuracy levels comparable to those of models trained with abundant data. Furthermore, it exhibits superior performance when compared to models trained with scarce data. The study also highlights the applicability of transfer learning under non-ideal conditions, such as cases where the target has a highly different probability distribution of data in comparison to the source.

Moreover, this thesis contributes to the development of more efficient and accurate load forecasting models for building energy management. By focusing on the generalization and scalability of data-driven applications, this work promotes the broader benefits of decarbonization and fosters the advancement of sustainable building operations. The research findings have implications for facility managers, energy service providers, and policymakers, as they strive to optimize energy consumption, reduce greenhouse gas emissions, and improve the overall sustainability of the built environment.

Acknowledgements

First and foremost, I express my profound gratitude to my parents, Robson Figueiredo do Carmo and Carmen Lúcia Dantas do Carmo, my brother, Caio Túlio Dantas do Carmo, and the maternal figure, Andréa Teixeira, for their unwavering and unconditional support. Your love and encouragement have been fundamental in helping me achieve this milestone.

I am also grateful to my friends, both those who were physically close and those who supported me in other ways, for being my pillars during this journey. Your words and companionship have been essential in keeping me motivated and focused.

I extend my appreciation to my colleagues from the DECARBONIZE project. Your contributions and constructive criticism have been invaluable, especially to my supervisors, Dr. Tiago Soares and Eng. Wellington Fonseca, who guided me with wisdom and patience throughout this process.

To my partner, Letícia Larrat Correa, I express my eternal gratitude for the indispensable emotional support during this journey. Your presence and encouragement have helped me face challenges and uncertainties with resilience and determination.

Lastly, I am thankful to my university for providing the necessary theoretical framework and all the tools I needed to conduct this research. Without the academic support and infrastructure offered, this achievement would not have been possible.

To all of you, my sincerest thanks.

Felipe Dantas do Carmo

"We can't solve problems by using the same kind of thinking we used when we created them"

Albert Einstein

Contents

1 Introduction **1**
- 1.1 Background and motivation . . . 1
- 1.2 Objectives and research questions . . . 2
- 1.3 Related projects and publications . . . 4
- 1.4 Scope and Limitations . . . 4
- 1.5 Methodology overview . . . 5
- 1.6 Thesis structure . . . 6

2 Literature Review **7**
- 2.1 Theoretical Framework . . . 7
  - 2.1.1 Load forecasting . . . 7
  - 2.1.2 Transfer learning . . . 16
- 2.2 State-of-the-art . . . 19
  - 2.2.1 Overview of load forecasting in commercial buildings . . . 19
  - 2.2.2 Overview of Transfer learning in load forecasting . . . 21
- 2.3 Discussion on the Benefits and Limitations of Transfer Learning . . . 22

3 Transfer Learning Approach for Load Forecasting Models **24**
- 3.1 Data collection and preprocessing . . . 24
  - 3.1.1 Data source and cleaning . . . 24
  - 3.1.2 Data normalization and split . . . 25
- 3.2 Virtual building creation . . . 27
- 3.3 Load forecasting model development and training . . . 29
- 3.4 Transfer learning approach . . . 31
  - 3.4.1 Evaluation metrics . . . 33

4 Results and Analysis **35**
- 4.1 Data cleaning results . . . 35
- 4.2 Analysis of load forecasting results for each VB . . . 40
  - 4.2.1 Load forecasting models with full data availability . . . 40
  - 4.2.2 Load forecasting models with scarcity of data . . . 50
- 4.3 Comparison of transfer learning approach to baseline models . . . 60
- 4.4 Discussion of results . . . 71

5 Conclusion and Future Work **75**

A Appendix **78**

References **100**

List of Figures

6figure.caption.10

2.1 Architecture of an ANN. 8
2.2 Traditional machine learning versus transfer learning approach. 16

3.1 INESC TEC building. 25
3.2 Graphical representation of the data split. 26
3.3 Illustration of the division of the floors of INESC TEC in VBs. 27
3.4 Illustration of the model architecture. 30
3.5 Training and validation Mean absolute error (MAE) as function of the number of epochs. 31
3.6 Case Scenarios. 31
3.7 Illustration of the transfer learning model architecture. 32
3.8 Fluxogram describing the entire transfer learning process. 33

4.1 Floor A-1 cleaned load (last 5000 rows). 36
4.2 Floor A0 cleaned load (last 5000 rows). 36
4.3 Floor B-1 cleaned load (last 5000 rows). 37
4.4 Floor B0 cleaned load (last 5000 rows). 37
4.5 Floor B5 cleaned load (last 5000 rows). 38
4.6 Floor B2 cleaned load (last 5000 rows). 38
4.7 Before and after cleaning on B2 (last 5000 rows). 39
4.8 Cleaned time series of VB B3 (last 5000 rows). 40
4.9 Cleaned time series of virtual building B4 (last 5000 rows). 40
4.10 Actual load and predicted load for VB A1 (last 1500 rows). 42
4.11 Scatter plot with regression line between actual and predicted load for VB A1. . . 42
4.12 Actual load and predicted load for VB A2 (last 1500 rows). 43
4.13 Scatter plot with regression line between actual and predicted load for Virtual building A2. 43
4.14 Actual load and predicted load for virtual building A3 (last 1500 rows). 44
4.15 Scatter plot with regression line between actual and predicted load for virtual building A3. 44
4.16 Actual load and predicted load for VB A4 (last 1500 rows). 45
4.17 Scatter plot with regression line between actual and predicted load for virtual building A4. 45
4.18 Actual load and predicted load for VB B1 (last 1500 rows). 46
4.19 Scatter plot with regression line between actual and predicted load for virtual building B1. 46
4.20 Actual load and predicted load for VB B2 (last 1500 rows). 47

4.21 Scatter plot with regression line between actual and predicted load for virtual building B2. 47
4.22 Actual load and predicted load for VB B2 (Before the last 1500 rows). 48
4.23 Actual load and predicted load for VB B3 (last 1500 rows). 48
4.24 Scatter plot with regression line between actual and predicted load for VB B3. . . 49
4.25 Actual load and predicted load for VB B4 (last 1500 rows). 49
4.26 Scatter plot with regression line between actual and predicted load virtual building B4. 50
4.27 Actual load and predicted load for VB A1 in Case Scenario (CS) 2 (last 1500 rows). 51
4.28 Scatter plot with regression line between actual and predicted load for virtual building A1 in CS 2. 52
4.29 Actual load and predicted load for VB A2 in CS 2 (last 1500 rows). 52
4.30 Scatter plot with regression line between actual and predicted load for VB A2 in CS 2. 53
4.31 Actual load and predicted load for VB A3 in CS 2 (last 1500 rows). 53
4.32 Scatter plot with regression line between actual and predicted load for VB A3 in CS 2. 54
4.33 Actual load and predicted load for VB A4 in CS 2 (last 1500 rows). 54
4.34 Scatter plot with regression line between actual and predicted load for virtual building A4 in CS 2. 55
4.35 Actual load and predicted load for VB B1 in CS 2(last 1500 rows). 55
4.36 Scatter plot with regression line between actual and predicted load for virtual building B1 in CS 2. 56
4.37 Actual load and predicted load for VB B2 in CS 2 (last 1500 rows). 56
4.38 Scatter plot with regression line between actual and predicted load for VB B2 in CS 2. 57
4.39 Actual load and predicted load for VB B3 in CS 2(last 1500 rows). 57
4.40 Scatter plot with regression line between actual and predicted load for VB B3 in CS 2. 58
4.41 Actual load and predicted load for VB B4 (last 1500 rows). 58
4.42 Actual load and predicted load for VB B4 (first 1500 rows). 59
4.43 Scatter plot with regression line between actual and predicted load for VB B4 in CS 2. 59
4.44 Actual load and predicted load for VB A1 in CS 3 (last 1500 rows). 61
4.45 Scatter plot with regression line between actual and predicted load in CS 3 for VB A1. 62
4.46 Actual load and predicted load for VB A2 in CS 3 (last 1500 rows). 63
4.47 Scatter plot with regression line between actual and predicted load in CS 3 for VB A2. 63
4.48 Actual load and predicted load for VB A3 in CS 3 (last 1500 rows). 64
4.49 Scatter plot with regression line between actual and predicted load in CS 3 for VB A3. 65
4.50 Actual load and predicted load for VB A4 in CS 3 (last 1500 rows). 65
4.51 Scatter plot with regression line between actual and predicted load in CS 3 for VB A4. 66
4.52 Actual load and predicted load for VB B1 in CS 3 (last 1500 rows). 67
4.53 Scatter plot with regression line between actual and predicted load in CS 3 for VB B1. 67

4.54 Actual load and predicted load for VB B2 in CS 3 (last 1500 rows). 68
4.55 Scatter plot with regression line between actual and predicted load in CS 3 for VB B2. 68
4.56 Actual load and predicted load for VB B3 in CS 3 (last 1500 rows). 69
4.57 Scatter plot with regression line between actual and predicted load in CS 3 for VB B3. 70
4.58 Actual load and predicted load for VB B4 in CS 3 (last 1500 rows). 70
4.59 Scatter plot with regression line between actual and predicted load in CS 3 for VB B4. 71
4.60 RMSE on all CSs (%). 72
4.61 MAPE on all CSs (%). 72

List of Tables

2.1 Transfer learning classification. 18

3.1 Average monthly consumption and hourly peak/valley consumption for each building. 28
3.2 Pearsson correlation coefficient (PCC) between all buildings. 28
3.3 PCC between B2 and the other buildings. 29

4.1 Summary of the data cleaning results. 39
4.2 Results of the evaluation metrics for each VB in the first case scenario. 41
4.3 Evaluation metrics for the load forecasting in CS 2. 51
4.4 Comparison of errors between CS 2 and CS 1 for all VBs. 60
4.5 Evaluation metrics of the models for CS 3. 61
4.6 Comparison between CS 3 and baseline models using RMSE. 73
4.7 Comparison between CS 3 and baseline models using MAPE. 74

Acronyms

ADAGRAD Adaptive gradient algorithm. 11

ADAM Adaptive moment estimation. 11, 30

ANN Artificial Neural Network. 1, 7, 9, 18, 20

ARIMA Auto-regressive integrated moving average. 19, 20

BPNN Back propagation neural network. 22

CNN Convolutional neural network. 9, 10, 16, 18–21, 29

CS Case Scenario. viii–x, 31, 32, 40, 50–57, 59–74

DNN Deep neural network. 11, 12, 21

GCLSTM Graph convolutional long short therm memory. 20

GCN Graph convolutional network. 20

GRU Gated Recurrent Unit. 20

IQR Interquartile range. 24, 25

LSTM Long short term memory. 10, 18–20, 22, 29, 31, 32

MAE Mean absolute error. vii, 7, 13, 14, 30, 31, 33

MAPE Mean absolute percentage error. 14–16, 30, 33, 41–46, 48, 50–63, 65–71, 73, 74

MSE Mean squared error. 7, 30, 33

PCC Pearsson correlation coefficient. x, 15, 28, 71

ReLu Rectified linear unit. 7, 9, 12, 13, 29

RMSE Root mean squared error. 14, 16, 30, 33, 51–74

RMSPROP Root mean squared propagation. 11

RNN Recurrent Neural Network. 10, 11, 16, 19, 20

SGD Stochastic gradient descent. 11

SVM Support vector machine. 7, 19, 20

SVR Support vector regression. 20

Tanh Hiperbolic tangent. 7, 9, 12, 13

TCN Temporal Convolutional Network. 19, 20

VB Virtual Building. iii, vi, vii, x, 3–5, 24, 26–29, 33, 35, 36, 38–43, 45, 47, 49–51, 53, 55–57, 59–67, 69, 71–74

WT Wavelet transform. 20

Chapter 1

Introduction

1.1 Background and motivation

As the urgency to mitigate climate change intensifies, a global shift towards decarbonization has become imperative. This transition involves the adoption of sustainable practices across various sectors, including the construction and operation of commercial buildings, which contribute significantly to global greenhouse gas emissions [1].

Commercial buildings account for a considerable proportion of energy consumption globally. The International Energy Agency (IEA) estimates that buildings and building construction sectors combined are responsible for 36% of global energy consumption and nearly 40% of total direct and indirect CO_2 emissions [2]. These figures are projected to increase due to urbanization trends and growth in demand for cooling and heating services, highlighting the urgency to adopt energy-efficient practices in building management [3].

The advent of the data-driven era has led to the widespread utilization of Artificial Neural Network (ANN) models in the context of smart cities, enabling enhanced control, optimization, and flexibility [4]. These capabilities contribute to the reduction of energy consumption and the mitigation of the environmental impact arising from urban infrastructures [5].

Nonetheless, the implementation of these models is constrained by a crucial factor: data availability [6]. In situations where data is scarce, these models tend to generate outputs with increased error, leading to difficulties in controlling the system [6].

To address this challenge and ensure high replicability and scalability with minimal data requirements for model fitting, this dissertation presents, tests, and applies a technique known as transfer learning [7], assessing its advantages within the realm of load forecasting in buildings [8]. Transfer learning is a machine learning technique in which a model developed for one task is repurposed as the foundation for a model on a second related task [7]. The underlying principle is that features learned by the model for one task may be beneficial for learning the second task, ultimately saving time and resources in comparison to training a model from scratch [7]. Transfer learning is frequently employed in computer vision and natural language processing [7], wherein

the same basic features can be applied to various problems, such as object recognition in images or text comprehension.

Predicting building energy consumption is a complex task that relies on numerous factors, including weather, occupancy, and building characteristics [4]. Conventional load forecasting methods, such as time series models, face limitations in addressing these intricate dependencies [8]. Transfer learning has the potential to alleviate these constraints by capitalizing on pre-trained models that have already acquired useful features from related tasks [7]. The primary concept behind employing transfer learning for load forecasting in buildings involves leveraging a pre-trained model, previously trained on an extensive data set of building energy consumption data, as a starting point [9]. This pre-trained model can subsequently be fine-tuned using a smaller data set of energy consumption data specific to the building of interest [6]. The refined model can then be utilized to forecast energy consumption within the building [10]. It is worth noting that the efficacy of transfer learning hinges on the similarity between the source and target tasks. in case the tasks are overly dissimilar, transfer learning may prove ineffective [7]. Transfer learning presents a valuable approach for smart cities to augment forecasting, monitoring, and optimization of various systems and services [10]. Its objective is to reduce carbon emissions and promote decarbonization [11]. Moreover, by harnessing pre-trained models, transfer learning has the potential to curtail the financial and resource-intensive process of training new models from the ground up [6].

1.2 Objectives and research questions

The primary aim of this study is to investigate the effectiveness of transfer learning in the context of load forecasting in buildings, particularly under scenarios with limited data availability. This section outlines the specific objectives and research questions that guide the research.

The objectives of this study are as follows:

1. To explore the potential benefits of transfer learning for improving the accuracy of load forecasting in buildings with scarce data.
2. To identify the factors that contribute to the success of transfer learning in the context of building energy consumption forecasting.
3. To compare the performance of transfer learning-based models with traditional load forecasting methods, such as time series models.
4. To assess the relationship between the similarity of source and target tasks and the effectiveness of transfer learning for load forecasting in buildings.
5. To provide recommendations for the practical application of transfer learning in smart cities for the purposes of enhancing energy management and reducing environmental impact.

This study aims to formulate the necessary research questions to achieve the objectives proposed and these questions will be addressed by employing a systematic approach to reserach the

effectiveness of transfer learning in the context of load forecasting in buildings. The methodology for addressing each research question is outlined below:

1. **Research Question 1:** In what ways can transfer learning enhance the accuracy of load forecasting in buildings with scarce data?

   To address this question, a load forecasting model using a transfer learning approach will be developed. This approach involves using a pre-trained model as a starting point for training a new model on a smaller dataset specific to the building of interest. The model architecture and transfer learning methodology will be described in detail (Chapter 3), and the accuracy of the model will be evaluated (Chapter 4).

2. **Research Question 2:** Which factors contribute to the success of transfer learning in the context of building energy consumption forecasting?

   The study will analyze the performance of transfer learning models on a group of Virtual Buildings (VBs) with limited data availability. Factors that influence the success of transfer learning, such as the similarity between source and target tasks [12], data quality, and model architecture, will be identified and discussed.

3. **Research Question 3:** How do transfer learning-based models perform in comparison to traditional load forecasting methods, such as time series models?

   The performance of transfer learning models will be compared to that of traditional load forecasting methods, by evaluating their accuracy and training time. The results of this comparison will provide insights into the advantages and limitations of each approach.

4. **Research Question 4:** What is the effect of the similarity between source and target tasks on the success of transfer learning for load forecasting in buildings?

   The relationship between the similarity of source and target tasks and the effectiveness of transfer learning for load forecasting in buildings will be assessed. This will involve analyzing the performance of transfer learning models across different degrees of similarity between source and target tasks, and identifying any patterns or trends that emerge.

5. **Research Question 5:** What practical recommendations can be provided for the application of transfer learning in smart cities to strengthen energy management and reduce environmental impact?

   Based on the findings from the previous research questions, practical recommendations will be provided for the implementation of transfer learning in smart cities. These recommendations will focus on enhancing energy management and reducing environmental impact by leveraging the benefits of transfer learning, while addressing its limitations and challenges.

By employing the aforementioned methodologies to address each research question, this dissertation aims to contribute valuable insights into the application of transfer learning for load forecasting in buildings, ultimately moving towards more efficient energy management and reduced environmental impact in smart cities.

1.3 Related projects and publications

The research conducted within the scope of this dissertation is partially related to the objectives and outcomes of two distinct research projects, namely:

- **DECARBONIZE** - Development of strategies and policies based on energy and non-energy applications towards CARBON-neutral cities via digitalization for citizens and society (NO RTE-01-0145-FEDER-000065);
- **DECMERGE** – Decentralized decision-making for multi-energy distribution grid management (2021.01353.CEECIND).

The work carried out has culminated in the preparation of one scientific article intended for journal publication. The following reference should be consulted:

- Felipe Dantas do Carmo, Wellington Fonseca, Tiago Soares, "Overcoming Data Scarcity in Load Forecasting: A Transfer Learning Approach for Commercial Buildings", Energy and Buildings, under review.

1.4 Scope and Limitations

The scope of this thesis is to investigate the application of transfer learning for load forecasting models in the context of building energy management, particularly in scenarios with limited data availability. The study presents a case study of VBs, using a pre-trained model as a starting point to develop a new model with additional layers that are trained on a smaller dataset.

The limitations of this study include the following:

- The study focuses on load forecasting for commercial buildings and may not be directly applicable to other domains, such as residential or industrial buildings, since those domains may present characteristics that are not considered in a commercial building environment.
- The study uses a specific set of data for training and evaluation, and the results may not be generalizable to other datasets or regions with different characteristics.
- The study employs a particular transfer learning approach, and other approaches or architectures may yield different results.

- Although the computational cost of using transfer learning is relatively low in this application, studies employing more features and larger datasets may face difficulties regarding computational cost, which is not evaluated in this dissertation.
- The study does not explore the interpretability of the models developed using transfer learning, which may be a concern in certain applications where understanding the underlying relationships between variables is crucial.
- The study does not consider the potential impact of external factors such as weather conditions, building occupancy patterns, and region-specific characteristics, which may influence the effectiveness of transfer learning in load forecasting.

Despite these limitations, this dissertation provides valuable insights into the application of transfer learning for load forecasting models in building energy management. The results demonstrate the potential of transfer learning to reduce training time and improve accuracy in scenarios with limited data availability. This study contributes to the growing body of knowledge on the use of machine learning techniques for sustainable and efficient energy management in smart cities.

1.5 Methodology overview

First, a review of the literature was conducted to identify existing research on load forecasting and transfer learning. This involved a comprehensive search of academic databases and relevant publications. The aim of this step was to gain an understanding of the current state of the field and identify any research gaps that could be addressed through this study.

Next, a case study of a group of VBs was selected to apply the proposed transfer learning approach to load forecasting. The VBs were chosen to represent a diverse range of building types (within the scope of commercial buildings) and energy consumption profiles, allowing for a more comprehensive evaluation of the proposed approach.

After selecting the case study, the necessary data was collected and preprocessed for the load forecasting models. This involved cleaning and organizing the data, as well as transforming it into a suitable format for model training.

The load forecasting models were then developed and trained using a transfer learning approach. This involved initializing the models with pre-trained weights from a previously trained model and adding new layers that were trained on a smaller dataset. The performance of the transfer learning models was evaluated and compared to models trained with and without abundant data availability.

Finally, the results of the study were analyzed and discussed. The findings of the study were compared to existing research in the field, and the implications and limitations of the study were discussed. A conclusion was drawn regarding the effectiveness of the transfer learning approach for load forecasting in building energy management.

The methodology overview is summarized in Figure 1.1 fluxogram:

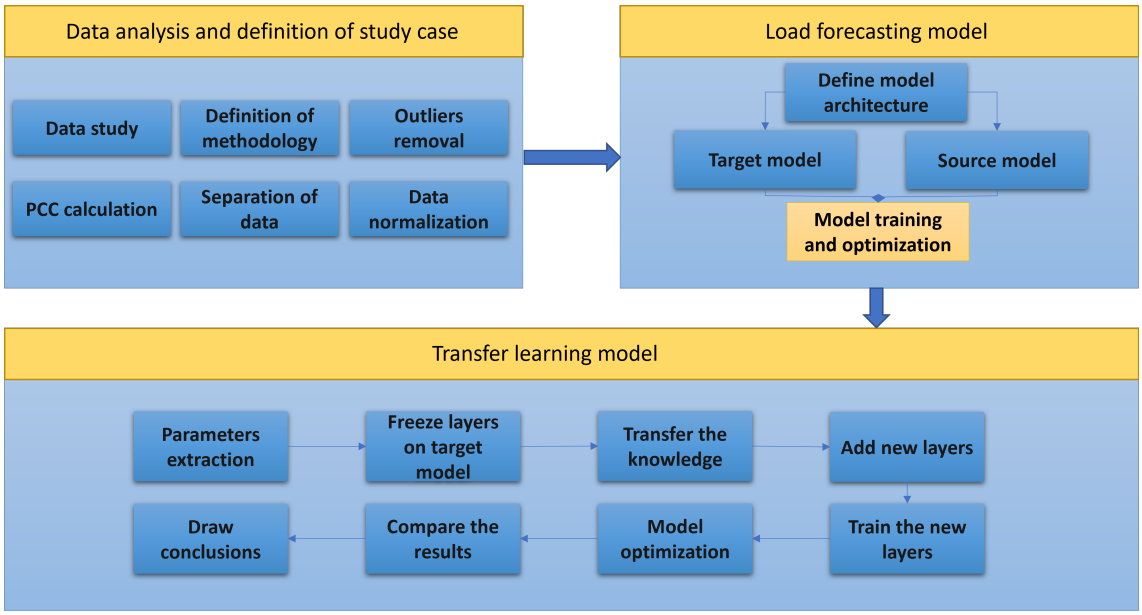

Figure 1.1: Fluxogram describing the steps in the implementation of the method[1].

1.6 Thesis structure

This thesis is organized as follows. Chapter 1 provides an introduction to the research problem, the motivation for the study, and the objectives and research questions. Chapter 2 presents a literature review of load forecasting models and transfer learning, covering both theoretical and practical aspects of these topics. Chapter 3 describes the process used to develop and evaluate the load forecasting models based on transfer learning, including data collection, pre-processing, model development, and performance evaluation.

In Chapter 4, the results of the experiments are presented and discussed. This chapter includes an analysis of the performance of the load forecasting models based on transfer learning and a comparison with models trained with abundant and scarce data availability. Finally, Chapter 5 provides a summary of the findings, conclusions, and recommendations for future research. The appendix includes technical details and additional information on the methodology, data, and models used in this study.

Chapter 2

Literature Review

2.1 Theoretical Framework

2.1.1 Load forecasting

In this section, the theoretical framework of load forecasting, a crucial component in building energy management systems, will be discussed. Load forecasting involves predicting the electrical load or energy consumption at a future time based on historical data and other relevant factors. Accurate load forecasting can lead to efficient energy management, reduced operational costs, and improved system reliability. Various methods have been proposed in the literature for load forecasting, such as statistical methods, machine learning techniques, and hybrid approaches [13].

One of the most widely used load forecasting methods is time series analysis, which uses historical data to model and forecast future load values. Time series can be modelled using machine learning, which includes techniques such as ANNs, Support vector machine (SVM)s, and decision trees. In the case of ANNs, the model can be represented as a composition of layers (Figure 2.1), each consisting of a set of neurons connected with a weight matrix. The output of each layer is computed using an activation function, such as the Rectified linear unit (ReLu) or the Hiperbolic tangent (Tanh). The model is trained by minimizing the difference between the predicted load values and the true load values, often measured by a loss function like Mean squared error (MSE) or Mean absolute error (MAE).

1. **ANNs** are computational models inspired by the biological ANNs found in the human brain [14]. ANNs consist of interconnected nodes or artificial neurons, which are organized in layers. These layers include an input layer, one or more hidden layers, and an output layer. Each connection between the nodes has an associated weight, which is adjusted during the learning process. The purpose of an ANN is to learn patterns or relationships in the input data and make predictions or decisions based on the learned knowledge.

   The core of an ANN is the artificial neuron or node, which receives input from other nodes in the previous layer and processes it to produce an output. The output is determined by

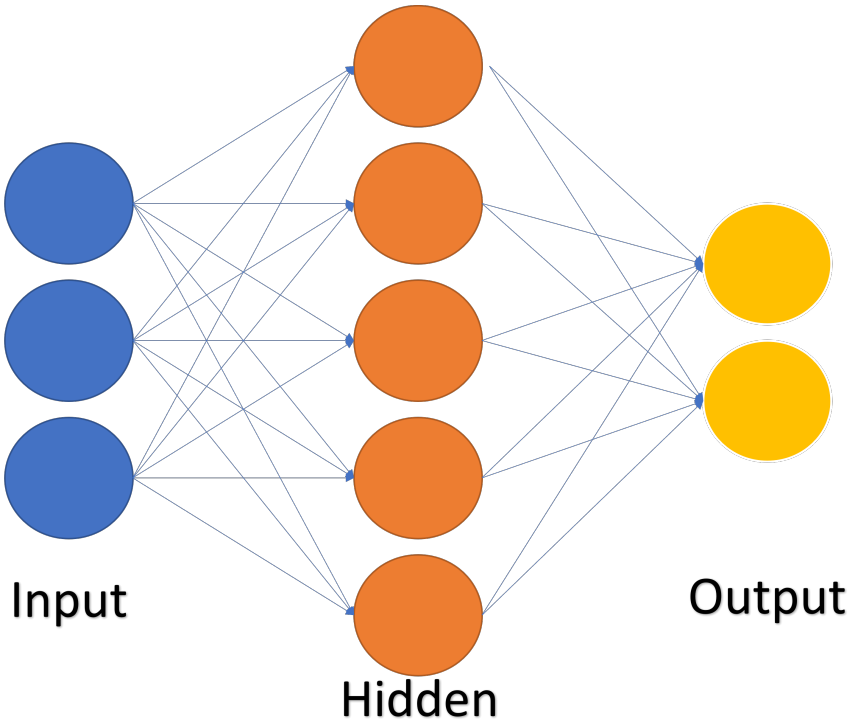

Figure 2.1: Architecture of an ANN.

the weighted sum of the inputs and a bias term, followed by the application of an activation function. The activation function is responsible for introducing non-linearities into the network, which enables ANNs to model complex relationships [15].

The weighted sum of the inputs and bias term is given by the following equation:

$$z = \sum_{i=1}^{n} w_i x_i + b \tag{2.1}$$

where, w_i represents the weight associated with the i-th input, x_i, b is the bias term, and n is the total number of inputs. The output of the artificial neuron, denoted by y, is computed by applying an activation function, $f(\cdot)$, to the weighted sum:

$$y = f(z) \tag{2.2}$$

Training an ANN involves adjusting the weights and biases to minimize the error between the predicted output and the actual output (i.e., the target values). This is typically achieved using a learning algorithm, such as gradient descent or a variant thereof, in combination with backpropagation, which is used to compute the gradient of the error with respect to the weights and biases [16].

The performance of an ANN is commonly evaluated using loss functions, such as the MSE for regression tasks or the Cross-Entropy Loss for classification tasks. These loss functions

quantify the discrepancy between the predicted output and the target values, guiding the learning process [17].

2. **A dense layer**, also known as a fully connected layer, is a type of layer commonly used in ANNs, where each neuron in the layer is connected to every neuron in the previous layer [17]. This layer is responsible for performing a linear transformation (Equation 2.3) followed by an activation function (Equation 2.4), where x is the input vector with n dimensions, W is the weight matrix of size $m \times n$, b is the bias vector of size m, and a is the output vector with m dimensions. The activation function g can be any function, such as ReLu, sigmoid, or Tanh, depending on the specific application and requirements.

$$a = Wx + b \tag{2.3}$$

$$y = g(a) \tag{2.4}$$

3. **Convolutional neural networks (CNNs)** are a specialized type of ANN architecture that has been highly successful in a wide range of applications, particularly in computer vision and image recognition tasks [18]. CNNs leverage the spatial structure of input data by employing convolutional layers, which perform local operations on the input, allowing the network to learn hierarchical and translation-invariant features.

   The main building block of a CNN is the convolutional layer. In this layer, a set of learnable filters or kernels is convolved with the input data, resulting in feature maps that capture spatial information from the input. Mathematically, the convolution operation can be represented as:

$$f(x,y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} g(i,j) \cdot h(x-i, y-j) \tag{2.5}$$

   where $f(x,y)$ is the output feature map, $g(i,j)$ is the input data, and $h(x-i, y-j)$ represents the filter or kernel.

   CNNs often consist of a series of convolutional layers, followed by non-linear activation functions (e.g., ReLu, Tanh), pooling layers, and fully connected layers [17]. Pooling layers are used to reduce the spatial dimensions of the feature maps, making the network more computationally efficient and invariant to small translations. Common pooling operations include max pooling and average pooling [17]. CNNs have been highly successful in various applications, including image classification [19], object detection [20], and semantic segmentation [21]. Their ability to learn hierarchical and translation-invariant features from input data has made them a powerful tool for tasks involving spatially structured data.

4. **Max pooling** is a downsampling operation commonly used in CNNs to reduce the spatial dimensions of feature maps, making the network more computationally efficient and invariant to small translations [18]. The max pooling layer operates by dividing the input feature map into non-overlapping rectangular regions and selecting the maximum value from each region as the output. Mathematically, the max pooling operation can be represented as:

$$y_{i,j} = \max_{m=0}^{M-1} \max_{n=0}^{N-1} x_{i \cdot M+m, j \cdot N+n} \tag{2.6}$$

where x is the input feature map, y is the output feature map, M and N are the dimensions of the pooling region, and i and j represent the spatial coordinates of the output feature map. Max pooling has been widely used in various CNN architectures, such as LeNet-5 [18] and AlexNet [19], for tasks like image classification and object recognition. By reducing the spatial dimensions, max pooling helps to control the number of parameters in the network, which can help prevent overfitting and improve generalization.

5. **Long short term memory (LSTM)** networks, first introduced by Hochreiter and Schmidhuber [22], are a type of Recurrent Neural Network (RNN) architecture designed to address the vanishing gradient problem encountered in traditional RNNs. LSTMs have been successful in various sequence-to-sequence learning tasks, such as natural language processing, time series prediction, and speech recognition.

   The key innovation of LSTMs lies in their memory cell, which consists of an input gate, a forget gate, an output gate, and a cell state. These gates and the cell state enable the LSTM to selectively store, update, and retrieve information over long sequences, allowing them to learn and model long-term dependencies in data. The equations governing the LSTM cell are as follows:

$$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \tag{2.7}$$
$$f_t = \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \tag{2.8}$$
$$g_t = \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \tag{2.9}$$
$$o_t = \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \tag{2.10}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{2.11}$$
$$h_t = o_t \odot \tanh(c_t) \tag{2.12}$$

   where i_t, f_t, g_t, and o_t represent the input, forget, cell update, and output gate activations, respectively. x_t is the input at time step t, h_t is the hidden state, c_t is the cell state, and $\odot$ denotes element-wise multiplication. LSTM networks have proven effective in various applications, such as machine translation [23], text generation [24], and speech recognition. Their ability to model long-term dependencies has made them a popular choice for tasks involving sequential data.

2.1.1.1 Optimizer

6. **Adaptive moment estimation (ADAM)** optimizer is a widely-used optimization algorithm for training ANNs and other machine learning models [25]. ADAM combines the advantages of two other optimization algorithms: Adaptive gradient algorithm (ADAGRAD) [26] and Root mean squared propagation (RMSPROP) [27]. It adapts the learning rate for each parameter by computing the first-order moment (mean) and the second-order moment (uncentered variance) of the gradients. ADAM updates the parameters using the following equations:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \tag{2.13}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \tag{2.14}$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{2.15}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{2.16}$$

$$\theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon} \tag{2.17}$$

In these equations, t is the current iteration, g_t represents the gradient of the objective function with respect to the parameters θ at iteration t, m_t and v_t are the first and second moment estimates, respectively, and $\hat{m}_t$ and $\hat{v}_t$ are the bias-corrected estimates. The parameters β_1 and β_2 are the exponential decay rates for the first and second moment estimates, respectively. α is the learning rate and ε is a small constant to prevent division by zero.

ADAM is known for its efficiency and robustness, as it requires minimal tuning of hyperparameters and works well with sparse gradients, making it suitable for various machine learning tasks [25].

Tieleman and Hinton [27] demonstrated the effectiveness of the ADAM optimizer through extensive experiments on various optimization tasks, including training Deep neural network (DNN) and training RNNs for sequence learning tasks. The results showed that ADAM converged faster and achieved better performance than other optimization methods, such as Stochastic gradient descent (SGD), AdaGrad, and RMSPROP, across different tasks.

The study also provided a thorough analysis of the algorithm's properties and established its convergence properties under specific conditions. The authors highlighted that the algorithm is computationally efficient, has little memory requirements, and is well-suited for problems with large data sets or parameter spaces.

2.1.1.2 Activation functions

7. **The ReLu** activation function is a popular choice in modern ANNs due to its simplicity and effectiveness in various deep learning tasks. The ReLu function is a non-linear activation function defined as follows:

$$f(x) = \max(0, x), \tag{2.18}$$

where x is the input to the function. The ReLu function can also be represented using a piecewise linear function:

$$f(x) = \begin{cases} x, \text{ if } x \geq 0 \\ 0, \text{ otherwise} \end{cases} . \tag{2.19}$$

The ReLu function is known for its ability to alleviate the vanishing gradient problem often encountered in training DNN. This is because its gradient remains constant for positive input values:

$$\frac{df(x)}{dx} = \begin{cases} 1, & \text{if } x \geq 0, \\ 0, & \text{otherwise.} \end{cases} \tag{2.20}$$

In practice, the ReLu function is computationally efficient and facilitates faster convergence during training compared to other activation functions, such as the sigmoid or Tanh functions.

8. **Tanh** activation function is another widely used activation function in ANNs. The Tanh function is similar to the sigmoid function but maps the input values to a range between -1 and 1. This can be advantageous in some cases, as it allows the output to have both positive and negative values, which may help with convergence during training. The equation for the Tanh activation function is as follows:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.21}$$

The derivative of the Tanh function is essential for back, as it is used to update the weights during the training process. The derivative can be expressed as:

$$\frac{d\tanh(x)}{dx} = 1 - \tanh^2(x) \tag{2.22}$$

Like the ReLu, the Tanh function is a popular choice for activation functions in ANNs due to its smooth, differentiable nature and ability to capture non-linear relationships.

9. **Sigmoid** activation function is a widely used activation function in Artificial Neural Networks (ANNs). The sigmoid function maps the input values to a range between 0 and 1. This is advantageous in cases where the output needs to be interpreted as a probability. The equation for the sigmoid activation function is as follows:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{2.23}$$

The derivative of the sigmoid function is crucial for backpropagation, as it is used to update the weights during the training process. The derivative can be expressed as:

$$\frac{d\sigma(x)}{dx} = \sigma(x)(1-\sigma(x)) \tag{2.24}$$

Like the ReLu and Tanh, the sigmoid function is a popular choice for activation functions in ANNs due to its smooth, differentiable nature and ability to capture non-linear relationships. However, it may suffer from the vanishing gradient problem for very large or very small input values.

2.1.1.3 Metrics

10. **Mean squared error (MSE)** is a commonly used loss function for regression tasks in machine learning and is particularly suited for problems where the target variable is continuous [28]. MSE measures the average squared difference between the predicted values and the true values, emphasizing larger errors over smaller ones. Given a dataset with N samples, the true target values y_i and the predicted values $\hat{y}_i$, the MSE can be calculated as follows:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2.25}$$

Minimizing the MSE aims to reduce the average squared error between the predicted values and the true values, which, in turn, leads to more accurate predictions. The choice of the MSE loss function is motivated by its simplicity, differentiability, and interpretability, as well as its relation to the maximum likelihood estimation when the output distribution is assumed to be Gaussian [29].

11. The **Mean absolute error (MAE)** is another widely used loss function for regression tasks in machine learning, particularly when the target variable is continuous [30]. In contrast to MSE, MAE measures the average absolute difference between the predicted values and the true values, making it less sensitive to outliers or large errors. Given a dataset with N samples, the true target values y_i and the predicted values $\hat{y}_i$, the MAE can be calculated as follows:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{2.26}$$

Minimizing the MAE aims to reduce the average absolute error between the predicted values and the true values, which can lead to more robust predictions in the presence of outliers. The choice of the MAE loss function is often motivated by its simplicity, interpretability, and robustness to outliers compared to the MSE loss function [30].

12. **Root mean squared error (RMSE)** is a widely-used metric for evaluating the performance of regression models and estimating the accuracy of predictions [31]. It measures the difference between the predicted values and the actual values of a dataset. The RMSE is particularly useful in assessing the model's overall performance, as it takes into account both the magnitude and direction of the errors. The lower the RMSE value, the better the model's performance in terms of prediction accuracy. The RMSE is calculated using the following formula:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2.27}$$

In this equation:

- RMSE represents the Root Mean Square Error value
- n is the total number of observations or data points in the dataset
- y_i denotes the actual value of the i-th observation
- $\hat{y}i$ denotes the predicted value of the i-th observation
- $\sum i = 1^n(y_i - \hat{y}_i)^2$ is the sum of the squared differences between the actual and predicted values for all data points

The RMSE value is computed by first calculating the squared differences between the predicted and actual values for each data point in the dataset. Next, the average of these squared differences is determined by dividing the sum of the squared differences by the total number of data points. Finally, the square root of the average squared difference is taken, yielding the RMSE value.

However, it is essential to compare the RMSE values of different models within the same context, as an absolute value may not provide sufficient information about the model's performance relative to other models [32].

13. **The Mean absolute percentage error (MAPE)** is a popular metric for evaluating the performance of forecasting models and estimating the accuracy of predictions [33]. It measures the average percentage error between the predicted values and the actual values of a dataset. The MAPE is particularly useful for comparing forecasting models, as it expresses the errors

in percentage terms, which allows for an intuitive interpretation of the results. The MAPE is calculated using the following formula:

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{2.28}$$

In this equation:

- MAPE represents the MAPE value
- n is the total number of observations or data points in the dataset
- y_i denotes the actual value of the i-th observation
- $\hat{y}i$ denotes the predicted value of the i-th observation
- $\sum i = 1^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|$ is the sum of the absolute percentage differences between the actual and predicted values for all data points

The MAPE value is computed by first calculating the absolute percentage differences between the predicted and actual values for each data point in the dataset. Next, the average of these absolute percentage differences is determined by dividing the sum of the absolute percentage differences by the total number of data points. Finally, the average absolute percentage difference is multiplied by 100% to express the MAPE value as a percentage. A lower MAPE indicates a better fit of the model to the data, suggesting that the model's predictions are more accurate. However, it is essential to compare the MAPE values of different models within the same context, as an absolute value may not provide sufficient information about the model's performance relative to other models [33]. Additionally, the MAPE can be sensitive to the presence of zero or very small values in the actual data, which may lead to large percentage errors and potentially distorted results.

14. **The Pearsson correlation coefficient (PCC)**, also known as Pearson's r, is a measure of the linear relationship between two variables. The PCC ranges from -1 to 1, where -1 represents a perfect negative linear relationship, 1 represents a perfect positive linear relationship, and 0 represents no linear relationship between the variables. Pearson's r is determined following Equation (2.29).

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2.29}$$

In the above equation, x_i and y_i are the individual sample points indexed with i; n is the total number of samples; $\bar{x}$ and $\bar{y}$ are the mean values of x and y respectively. The top of the equation, $\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$, is the covariance of x and y. The bottom of the equation, $\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}$, is the product of the standard deviations of x and y.

When using machine learning methods for load forecasting, it is essential to select relevant input features, such as historical load values, weather data, and calendar information. The forecasting performance can be evaluated using various performance metrics, such as RMSE and MAPE.

In recent years, deep learning techniques, such as CNNs and RNNs, have also been applied to load forecasting problems. These methods can automatically learn complex patterns and temporal dependencies from large-scale datasets, leading to improved forecasting accuracy in many cases.

2.1.2 Transfer learning

Transfer learning is a machine learning technique that enables the reuse of a pre-trained model on a new problem (Figure2.2), leveraging the knowledge acquired in the source domain to improve performance in the target domain [34]. This approach can be particularly beneficial in scenarios with limited labeled data or high computational costs associated with training a model from scratch.

In the context of deep learning, transfer learning typically involves fine-tuning a pre-trained ANN on the target task, retaining the initial layers that capture more general features and adapting the final layers that are more specific to the target task.

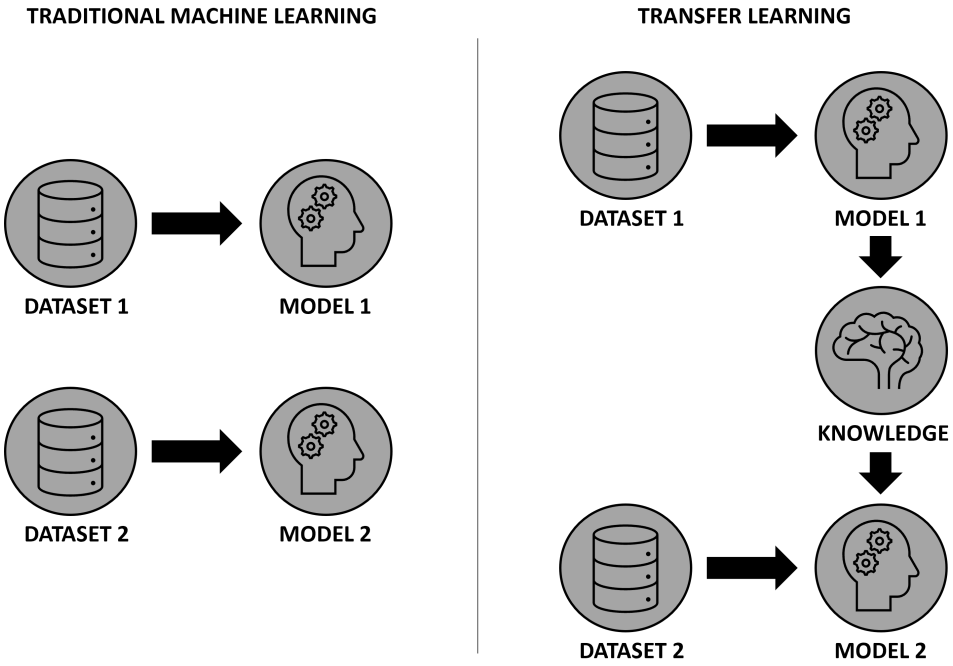

Figure 2.2: Traditional machine learning versus transfer learning approach.

Mathematically, transfer learning can be represented as a domain adaptation problem, where the source domain $\mathscr{D}_S$ and the target domain $\mathscr{D}_T$ have different probability distributions $P_S(X)$ and $P_T(X)$, respectively:

$$P_S(X) \neq P_T(X) \tag{2.30}$$

In transfer learning, the aim is to leverage the knowledge learned in the source domain $\mathscr{D}_S$ to improve the model's performance in the target domain $\mathscr{D}_T$ [35].

$$f_T(X) = \arg\min_{f \in \mathscr{F}} \mathbb{E}_{(X,Y)\sim P_T(X,Y)}[L(f(X),Y)] \tag{2.31}$$

1. The source domain ($\mathscr{D}_S$) consists of the dataset from which the initial model learns the patterns and features. This domain usually has an abundance of data to facilitate the model's training. In transfer learning, knowledge from the source domain is utilized to improve the model's performance in the target domain. The source domain is represented by the probability distribution $P_S(X,Y)$.

$$\mathscr{D}S = (X_i^S, Y_i^S)i = 1^{N_S}, \quad (X_i^S, Y_i^S) \sim P_S(X,Y) \tag{2.32}$$

2. The target domain ($\mathscr{D}_T$) is the domain in which the transferred knowledge is applied to improve the model's performance. Typically, the target domain has limited data availability or some similarities with the source domain, making transfer learning a suitable approach. The target domain is represented by the probability distribution $P_T(X,Y)$.

$$\mathscr{D}T = (X_i^T, Y_i^T)i = 1^{N_T}, \quad (X_i^T, Y_i^T) \sim P_T(X,Y) \tag{2.33}$$

3. Fine-tuning is a common technique used in transfer learning to adapt a pre-trained model to a new target task. Given a pre-trained model with a specific architecture, one or more layers of the model are adjusted by training on the target task data [36]. The fine-tuning process typically involves a smaller learning rate compared to the initial training of the model to ensure that the model does not diverge too far from the pre-trained weights. Let $W^{(0)}$ represent the pre-trained weights, and let α denote the learning rate. During the fine-tuning process, the weights $W^{(t)}$ at iteration t are updated as follows:

$$W^{(t+1)} = W^{(t)} - \alpha \nabla L(W^{(t)}), \tag{2.34}$$

where $\nabla L(W^{(t)})$ is the gradient of the loss function with respect to the weights at iteration t.

2.1.2.1 Transfer learning classification

Transfer learning can be classified into different categories based on various factors, such as the relationship between the source and target domains, and the type of learning involved according to the works of Pan and Yang [35] and Weiss et al. [7]. Here, we define four categories of transfer learning that can unite to form the classes in Table 2.1.

1. Inductive Transfer Learning: In inductive transfer learning, the source and target tasks are different, but the source and target domains may be the same or different. The goal is to improve the learning of the target predictive function $f_t(\cdot)$ in the target domain using the knowledge gained from the source domain. This is the most common form of transfer learning, which includes tasks like image classification, sentiment analysis, and natural language processing.

2. Transductive Transfer Learning: In transductive transfer learning, the source and target tasks are the same, but the source and target domains are different. The goal is to adapt the model learned from the source domain to the target domain without changing the task. This is especially useful when dealing with domain adaptation problems, such as adapting a model trained on one dataset to work well on another dataset from a different domain.

3. Homogeneous Transfer Learning: In homogeneous transfer learning, the feature spaces of the source and target domains are the same, i.e., $X_s = X_t$. The transfer learning process involves using the same type of features and representations for both the source and target domains. Examples of homogeneous transfer learning include fine-tuning a pre-trained model on a new dataset with the same input features.

4. Heterogeneous Transfer Learning: In heterogeneous transfer learning, the feature spaces of the source and target domains are different, i.e., $X_s \neq X_t$. This type of transfer learning involves dealing with different feature representations between the source and target domains. Techniques such as feature transformation, feature alignment, or feature space mapping are employed to bridge the gap between the different feature spaces. Examples of heterogeneous transfer learning include text-to-image retrieval, cross-language sentiment analysis, and cross-modal learning.

Table 2.1: Transfer learning classification.

	Domain	Task	Example
Homogeneous inductive learning	Source = Target	Source $\neq$ *Target*	Transfer learning is used to enhance building monthly electric load prediction leveraging information from similar buildings in different districts, that exhibits a different conditional probability.
Heterogeneous inductive learning	Source$\neq$ *Target*	Source $\neq$ *Target*	Transfer learning is used to fine-tune a pretrained ANN initially built to perform multi-class classification, to increase the accuracy of a prediction model for building temperature setback identification.
Homogeneous transductive learning	Source = Target	Source = Target	Transfer learning is used for improving the accuracy of home activity estimation by exploiting the data of a source house applied to a target house with no labelled data.
Heterogeneous transductive learning	Source $\neq$ *Target*	Source = Target	Transfer learning is used to predict building dynamics by extracting features from multiple households in an online fashion, without having access to labelled data.

Transfer learning has been widely applied in various deep learning architectures, such as CNNs and LSTMs, for tasks like image classification, natural language processing, and load forecasting

[11]. By leveraging the knowledge from the source domain, transfer learning can improve model performance, reduce training time, and enhance generalization in the target domain.

2.2 State-of-the-art

In this section, it is explored the state-of-the-art literature on the two most crucial topics addressed in this dissertation: (*i*) load forecasting models and (*ii*) transfer learning applied to load forecasting. More precisely, it is discussed the latest advancements, techniques, and methodologies employed in these fields, with a particular focus on their applications in commercial buildings.

2.2.1 Overview of load forecasting in commercial buildings

Load forecasting models are crucial for energy management, cost savings, and decarbonization efforts in commercial buildings. These models predict future electricity consumption and demand patterns, allowing facility managers and building owners to make informed decisions regarding energy procurement, storage, and usage optimization. The literature on load forecasting models can be broadly categorized into (*i*) traditional approaches and (*ii*) data-driven methods.

Traditional methods, such as time series analysis (e.g., Auto-regressive integrated moving average (ARIMA) models) and regression techniques (e.g., linear and multiple regression), depend on historical load data and potentially external factors, such as weather conditions, occupancy patterns, or economic indicators [37]. These methods have been widely used due to their simplicity and interpretability. However, they may face difficulties in capturing complex, non-linear patterns in load data. As the focus of this dissertation lies in data-driven models like deep learning approaches, traditional methods will not be extensively explored.

Data-driven approaches leverage machine learning algorithms to model the intricate relationships between input features and electricity consumption. ANNs [38], SVMs [39], and decision trees [40] are examples of widely used data-driven techniques. More recently, deep learning methods, including CNNs [41], RNNs [42], and LSTM networks [43], have demonstrated promising results in load forecasting tasks. In this dissertation, deep learning techniques are employed for load forecasting, and as such, they will be discussed in greater detail.

Hippert et al. [38] conducted a comprehensive review of ANNs and their application to electric load forecasting. They compared the performance of ANNs with traditional approaches such as time series models and regression techniques. The authors concluded that ANNs have the potential to provide better results than traditional methods, particularly for short-term load forecasting tasks. They highlighted that ANNs can capture complex, non-linear relationships in the data, which often leads to improved forecasting accuracy. However, they also noted the importance of selecting appropriate network architectures, input features, and training procedures to achieve optimal performance.

Lai et al. [41] presented a deep learning framework, called the Temporal Convolutional Network (TCN), for long-term and short-term time series prediction tasks, including load forecasting. The TCN model leverages causal CNNs for automatically learning temporal features in the time

series data. The authors compared their proposed TCN model with several state-of-the-art models, such as RNNs, LSTM networks, and Gated Recurrent Unit (GRU)s. The conclusions drawn from [41] are that the TCN model demonstrates competitive performance when compared to the other deep learning models, often outperforming them in terms of prediction accuracy and computational efficiency. Moreover, the TCN model is more interpretable and easier to parallelize, making it suitable for large-scale time series forecasting tasks. The results of the study support the potential of using deep learning techniques, specifically CNNs, for load forecasting applications.

Li et al. [42] proposed a hybrid approach for short-term load forecasting, which combines the strengths of the Wavelet transform (WT) technique and LSTM networks. The WT technique is used for decomposing the load time series into a set of sub-series with different frequency components, which helps in handling the non-stationary nature of the load data. The LSTM networks are then applied to model the decomposed sub-series and capture the underlying temporal dependencies. The conclusions drawn are that the proposed hybrid Wavelet-LSTM model demonstrates superior performance in short-term load forecasting when compared to other state-of-the-art techniques, such as ARIMA, Elman, Support vector regression (SVR). The study highlights the effectiveness of combining the WT technique with LSTM networks for enhancing the accuracy of load forecasting, especially in capturing the non-stationary and non-linear nature of the load time series data. This hybrid approach presents a promising direction for further research and development in load forecasting applications.

Marino et al. [43] presented a methodology for short-term load forecasting that leverages the advantages of graph-based techniques and LSTM networks. The proposed approach, referred to as Graph convolutional long short therm memory (GCLSTM), incorporates Graph convolutional network (GCN) to capture the spatial dependencies of power grids and LSTMs to model the temporal dependencies of load time series. The conclusions drawn from Marino et al. [43] highlight that the GCLSTM model demonstrates improved forecasting accuracy compared to other state-of-the-art methods, such as feedforward ANNs, graph CNNs, and vanilla LSTMs. The model successfully captures the complex spatial and temporal relationships in power grids, leading to better load forecasting performance. Additionally, the proposed approach is applicable to various power grid topologies, making it versatile and adaptable for different scenarios. The effectiveness of the GCLSTM model indicates that the combination of graph-based techniques and LSTM networks presents a promising direction for further research and development in load forecasting applications.

Zhang et al. [44] proposed a deep learning model integrating LSTMs and attention mechanisms for short-term load forecasting in commercial buildings. The main conclusions of the study are that the proposed model, which combines LSTMs with attention mechanisms, demonstrates better performance than traditional methods and other deep learning approaches, such as standalone LSTMs or CNNs. The attention mechanism helps the model to focus on the most relevant features and time steps in the input data, thereby improving the model's ability to capture complex, non-linear patterns in the load data. The authors also compared their model with other machine learning methods like SVM and found that the proposed model outperforms these traditional tech-

niques in terms of forecasting accuracy. The study emphasizes the importance of incorporating the attention mechanism in deep learning models for load forecasting, as it enhances the model's ability to capture and understand complex relationships in the data, leading to improved forecasting performance.

2.2.2 Overview of Transfer learning in load forecasting

Transfer learning has emerged as a potential solution to address data scarcity issues in load forecasting for commercial buildings. By leveraging knowledge from related tasks or domains, transfer learning can enhance the performance of load forecasting models, particularly when limited data is available for the target task [11]. The literature on transfer learning applied to forecasting is reviewed below.

Pan and Yang [35] presented a comprehensive survey on transfer learning, the authors discuss various transfer learning scenarios, methodologies, and algorithms proposed in the literature. They also provide an overview of the main research issues in transfer learning, including the relationships between tasks, domain adaptation, and the transfer of knowledge across different feature spaces. The authors conclude that transfer learning is a promising research direction in machine learning as it addresses the challenge of limited labeled data in many real-world applications. They highlight the different scenarios based on the relationship between source and target domains or tasks, such as inductive, transductive, and unsupervised transfer learning. Various methods and algorithms have been proposed for transfer learning, including instance-based, feature-representation-based, parameter-based, and relational-knowledge-based approaches. Despite progress in transfer learning research, several open issues and challenges remain [35]. Some of these challenges include how to transfer knowledge more effectively, how to measure the similarity between tasks or domains, and how to automatically discover related tasks or domains. The authors emphasize the importance of transfer learning in tackling real-world problems and suggest future research directions to address the open issues in this field.

Yosinski et al. [45] investigated the transferability of features learned by DNN and examined the factors that influence the performance of transfer learning. The authors conducted experiments using CNNs trained on the ImageNet dataset and evaluated the transferability of features by fine-tuning networks on new tasks. The authors found that the features learned by DNNs are indeed transferable across different tasks, and the transferability decreases as the dissimilarity between the source and target tasks increases. They observed that the lower layers of the network contain general-purpose features that can be effectively used for a wide range of tasks, while the higher layers capture more task-specific features that may not be as transferable. The study also demonstrated that the transferability of features can be improved by fine-tuning the network on the target task. This suggests that utilizing pre-trained models and fine-tuning them for new tasks can lead to better performance and reduced training time compared to training a model from scratch.

Cai et al. [46] proposed a two-layer transfer-learning-based STLF model to improve the accuracy of load in the target zone. In the inner layer, the latent parameter is introduced to represent the latent factors that result in differences in electricity consumption between different zones. An

iterative algorithm is developed in the outer layer to solve the weights. The authors reported that the proposed architecture always resulted in an improvement in forecasting accuracy compared to classic STLF algorithms.

Zhuang et al. [47] offered a thorough survey on transfer learning, with a focus on deep transfer learning methods and their applications. Their work serves as a valuable reference for researchers and practitioners working on transfer learning and its applications in various domains, providing insights into state-of-the-art techniques, challenges, and future research directions.

Li et al. [48] used the dataset from the open source building genome project [49] with around 400 non-residential buildings to develop a transfer learning-based ANN model. This model was developed to make the building energy forecast one-hour ahead with the task of improving forecasting accuracy for a target building with limited data available. The authors developed a three-layer Back propagation neural network (BPNN) model to evaluate the performance of the model through different source and target data samples and different source-target building pairs. They concluded that the less available data are, the more accuracy improvement transfer learn can achieve.

Pinto et al. [11] conducted a survey that specifically focused on the application of transfer learning for smart buildings. This study identified four primary application areas for transfer learning: (*i*) building load prediction, (*ii*) occupancy detection and activity recognition, (*iii*) building dynamics modeling, and (*iv*) energy systems control. Among these applications, building load forecast was the most widely adopted. However, the study also noted that few of the examined studies had been deployed in real-world settings, and it presented opportunities for future research. In addition to discussing applications, the survey provided a comprehensive theoretical description of transfer learning.

Ahn and Kim [50] applied transfer learning to the task of predicting building power consumption using a simulated dataset. The researchers developed a transfer learning LSTM model that was trained on just 24 hours of data and tasked with predicting the subsequent 24-hour period. The study showcased the ability of transfer learning-based models to improve prediction accuracy in comparison to LSTM models without transfer learning. Additionally, the researchers emphasized that higher accuracy was achieved when the climate zones for the source and target datasets were identical.

2.3 Discussion on the Benefits and Limitations of Transfer Learning

Transfer learning has emerged as an influential paradigm in machine learning, with applications ranging from natural language processing to computer vision and predictive modeling. This section presents a detailed discussion about the potential advantages and potential challenges associated with this technique.

Advantages:

- *Reduced training time:* One of the major benefits of transfer learning is the ability to leverage pre-trained models, which have already learned relevant features from large-scale

datasets. This ability greatly minimizes the need for extensive computational resources and significantly reduces the time required for training, thereby facilitating faster model deployment [35, 36].

- *Less data requirement:* Transfer learning is particularly beneficial when there is a scarcity of labeled data in the target domain. It offers an effective approach to compensate for this deficiency by utilizing the knowledge from the source domain, thereby enhancing model performance even with limited target data [35, 47].

- *Improved generalization:* Transfer learning encourages the model to develop more generalized and robust representations, by introducing it to diverse and extensive knowledge from the source domain. This characteristic makes it suitable for tasks with a high degree of variability and can lead to improved performance on unseen data [45, 35].

Limitations:

- *Negative transfer:* While transfer learning offers numerous advantages, it also carries the risk of negative transfer, where the performance of the model on the target task deteriorates rather than improving. This situation typically arises when the source and target domains are not adequately related or exhibit distinct characteristics, which can lead to the transfer of misleading or irrelevant knowledge [35, 45].

- *Model complexity:* The implementation of transfer learning often involves complex architectural modifications or additional fine-tuning steps. These intricacies can increase the model's complexity and the computational cost, possibly making it challenging to adapt and interpret [36, 51].

- *Domain adaptation challenges:* Adapting the pre-trained model to the target domain can be a challenging process, especially when there exists a significant disparity or distribution shift between the source and target domains. Addressing this requires effective strategies to align the source and target distributions, which may not always be straightforward to achieve [47, 35].

Chapter 3

Transfer Learning Approach for Load Forecasting Models

3.1 Data collection and preprocessing

3.1.1 Data source and cleaning

The data for this study was obtained from the INESC TEC building database, which contains energy consumption data from 156 meters across two buildings (Building A and Building B), as presented in Figure 3.1. Specifically, this study used processed data that contained the energy consumption for each floor in 15-minute intervals. The data was divided by floor to facilitate the creation of VBs, which represented each floor of the two buildings. It is important to note that some floors were not used in the study because the pattern they presented were highly different from most of the floors, meaning they represented a totally different statistical distribution. This approach enabled the creation of load forecasting models that were specific to each floor, allowing for the implementation of transfer learning.

The selected data covered a period of under two years, from March 2021 to October 2022, and was stored in CSV files. The data was chosen for its completeness and reliability, and any missing or corrupted data was removed during the data cleaning process.

An important observation to be made is that the selected data does not encompass the entirety of the information contained in the INESC TEC database. Instead, it covers a period deemed unaffected by the recent pandemic episode. This pandemic has dramatically altered the load consumption behavior in the building. Therefore, training the model with data from this period could potentially result in a skewed understanding of consumption patterns and, consequently, higher errors in the load forecasting.

The data cleaning process involved the use of several techniques to ensure that the data was suitable for analysis. One of these techniques was the Interquartile range (IQR) method, which is a statistical method used to identify outliers in a data set. The code calculated the first and third quartiles of the load consumption time series and then computed the IQR by subtracting the first quartile from the third quartile. The lower and upper bounds were defined by subtracting and

Figure 3.1: INESC TEC building.

adding 1.5 times the IQR, respectively. Any values of load consumption outside of these bounds were considered outliers and were removed from the data set.

In addition to the IQR method, a manual cleaning was conducted to identify and remove any obvious errors in the data collection that were not detected by the IQR algorithm. This process involved the graphical and analytical observation of the time series. One of the cleaning methods used in this part was to remove any data that does not have its immediate predecessor. For instance, consider a situation where data for 19:15 is present in the database, but data for 19:00 is absent. In such a case, the data for 19:15 must be removed. The rationale for this is rooted in the data collection methodology employed in the database, which involves the accumulation of energy consumption data. Consequently, the absence of data for 19:00 leads to the combined sum of data for both 19:00 and 19:15 being present at 19:15, which constitutes a clear error. This error may not be easily detectable through outlier removal methods. To mitigate any inaccuracies, this cleaning step was executed prior to all other data cleaning processes.

Overall, the data used in this study were carefully selected and preprocessed to ensure that it was of high quality and suitable for the load forecasting models used in the study.

3.1.2 Data normalization and split

Data normalization is a widely used technique in machine learning to standardize input data and improve model performance [17]. In this study, the data was normalized using the Min-Max normalization method, which scales the data to a range between 0 and 1.

- **MinMax normalization** is a scaling technique used to normalize the range of input features. It scales the values of the features to be within a specified range, usually [0, 1]. The MinMax normalization can be calculated as:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{3.1}$$

where x_{norm} is the normalized value, x is the original value, x_{min} is the minimum value of the feature, and x_{max} is the maximum value of the feature. To revert the normalized values back to the original scale, one can use the following inverse transformation formula:

$$x = x_{norm}(x_{max} - x_{min}) + x_{min} \tag{3.2}$$

The normalization process was applied to the features of the load forecasting models, which included the load consumption time series (the model's target), day of the week, week in the year, hour, minute, holiday (a Boolean flag), the load consumption delayed one day, and load consumption delayed seven days. By normalizing the input features, the models could more effectively learn the patterns and relationships between the features.

Next, the data was split into training, validation, and testing sets for each VB. The last two months of data were reserved for the test set to evaluate the performance of the predictions, while the remaining data was divided into 80% for training and 20% for validation as represented in Figure 3.2.

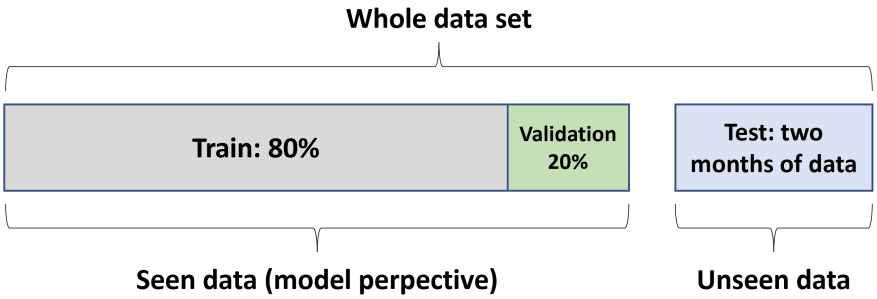

Figure 3.2: Graphical representation of the data split.

The training set was used to train the load forecasting models, while the validation set was used to fine-tune the hyperparameters and prevent overfitting. Finally, the testing set was used to assess the model's performance on new and unseen data, which is crucial for ensuring that the models generalize well. By applying data normalization and splitting the data into three sets, the load forecasting models in this study were optimized to accurately predict energy consumption for each VB.

3.2 Virtual building creation

As discussed, the data from the different floors in the INESC TEC building were divided into what was referred to as VBs. Each VB has its own unique characteristics, which can be compared to a company that rents a floor from a building and operates independently from the other floors. This means that each VB has a different load consumption behavior than the other floors, but it is still located in the same geographical region with similar weather conditions, time zones, holidays, and other factors that can influence the energy consumption pattern.

Using this methodology, the two buildings composing INESC TEC were divided into eight VBs as described in Figure 3.3 and a load forecasting model will be developed for each VB. Additionally, the transfer learning technique will be applied and evaluated to improve the accuracy of the models.

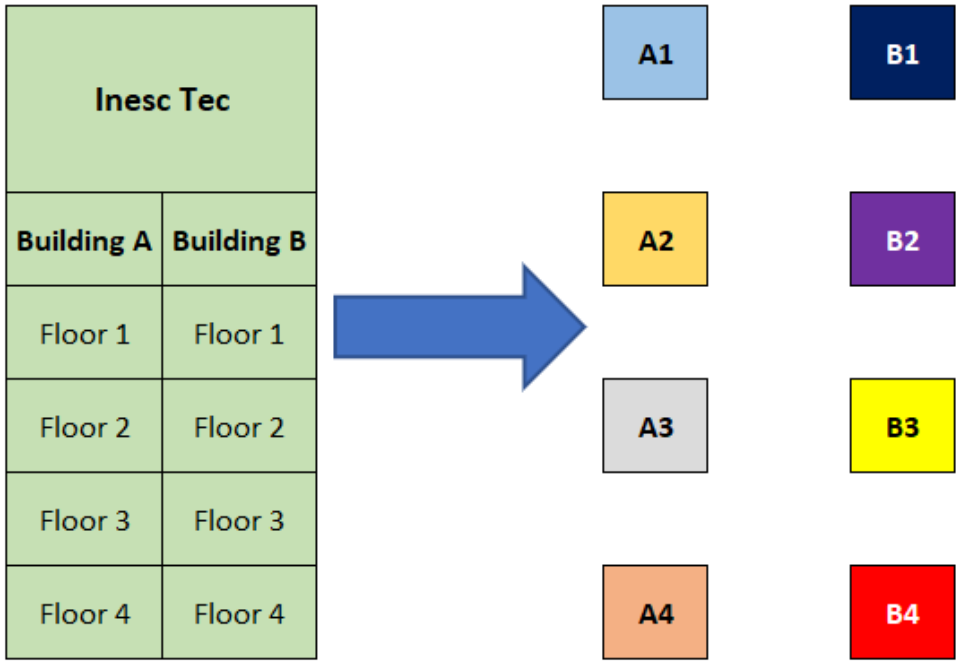

Figure 3.3: Illustration of the division of the floors of INESC TEC in VBs.

To further characterize the VBs, the average monthly energy consumption (in kWh) was calculated, providing insight into the load profiles and intensities of energy consumption across each VB. In addition, to evaluate the consumption during peak and valley hours, the average values were calculated and presented in Table 3.1.

Table 3.1: Average monthly consumption and hourly peak/valley consumption for each building.

VB	Average consumption(kWh)	Peak(Wh)	Valley(Wh)
A1	1194.10	6558.91	344.73
A2	817.08	4074.91	325.45
A3	1398.86	6765.82	290.91
A4	1246.35	6628.00	389.45
B1	1764.73	6383.64	622.91
B2	814.16	3574.18	361.09
B3	604.23	2159.27	380.73
B4	417.61	1891.27	385.82

Another factor to be considered include the similarities between the load consumption time series of the different VBs. It is expected that the higher the similarity between the buildings, the greater the efficiency of the transfer learning technique. The similarity was calculated using the PCC and shown in Table 3.2.

Table 3.2: PCC between all buildings.

PCC	A1	A2	A3	A4	B1	B2	B3	B4
A1	X	0.8070	0.7898	0.8189	0.6099	0.7819	0.4352	0.4255
A2	X	X	0.7687	0.7686	0.5797	0.7757	0.4013	0.4152
A3	X	X	X	0.7859	0.6914	0.7890	0.4536	0.3947
A4	X	X	X	X	0.5655	0.7700	0.4826	0.4375
B1	X	X	X	X	X	0.6576	0.3040	0.2645
B2	X	X	X	X	X	X	0.4169	0.4544
B3	X	X	X	X	X	X	X	0.2358
B4	X	X	X	X	X	X	X	X

Through the PCC analysis, it was possible to identify high similarity between Buildings A1, A2, A3, A4, B1, and B2. However, VBs B3 and B4 showed low PCC values (under 0.5) between each other and with other buildings, indicating that they have more diverse patterns compared to the others. This information is crucial for selecting the appropriate transfer learning approach to be used in the load forecasting models.

To implement transfer learning, it is important to choose a building as the base model from which the weights will be transferred. It is crucial that the base model has good load forecasting performance so that the knowledge transferred from the source to the target results in an improvement. In this study, after data analysis, it was evident that one of the VBs presented a clearer and more recognizable pattern of energy consumption, with less noise or corrupted data, and a higher PCC with other buildings (Table 3.3). Therefore, the VB **B2** was selected as the base model in the preliminary analysis.

Table 3.3: PCC between B2 and the other buildings.

PCC	B2
A1	0.7819
A2	0.7757
A3	0.7890
A4	0.7700
B1	0.6576
B3	0.4169
B4	0.4544

3.3 Load forecasting model development and training

In this section, the development of the load forecasting models for all VBs in this study will be described, using the same model architecture for each one of them. The primary focus of this section is to create a highly accurate model for the base VB (B2), while the models for the other buildings will serve for comparison purposes. The load forecasting models were developed using deep learning techniques, specifically a combination of CNNs and LSTM networks. The models were implemented using the Keras library in Python [52], which is built on top of TensorFlow library [53].

The first step in the training process was to preprocess the data. The data was first scaled using the MinMaxScaler from the scikit-learn library (subsection 3.1.2). The data was then split into training and testing sets (according to Figure 3.2). The training set consisted of seven features of chronological and historical data, while the testing set consisted of data that was not seen by the model during training.

The architecture of the sequential model (illustrated in Figure 3.4) comprises an initial 1D convolutional layer designed to extract features from the input sequence. This layer is configured with 64 filters, a kernel size of 3, and the ReLu activation function. A subsequent max pooling layer, with a pool size of 2, serves to reduce the output size of the convolutional layer. The output of the max pooling layer is then channeled into an LSTM layer, containing 32 hidden neurons and utilizing the tanh activation function, to model the sequence of features extracted by the convolutional layer. An additional LSTM layer, featuring the same architecture as its predecessor but with 16 hidden neurons, follows. To prevent overfitting, dropout layers with a rate of 0.2 are incorporated. Finally, a dense output layer is employed to generate the ultimate prediction.

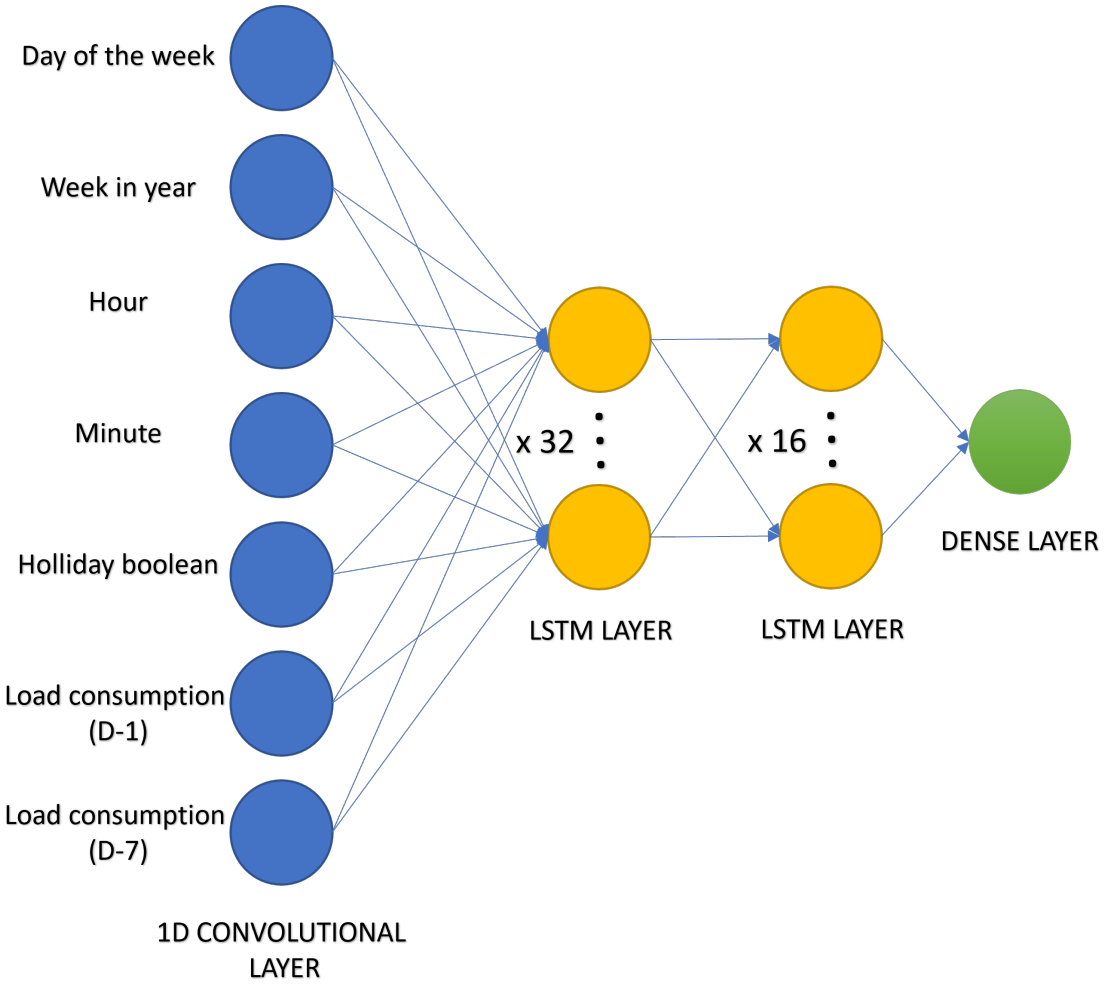

Figure 3.4: Illustration of the model architecture.

The models were trained using the MSE loss function and the ADAM optimizer for up to 1000 epochs using a batch size of 32. Early stopping was used to prevent overfitting and improve the generalization performance of the model. The patience of the early stopping callback was defined at 15 epochs, this value was reached through a process of running the training and evaluating the graphic of training and validation error as presented in Figure 3.5.

The best performing model was selected based on a model checkpoint module provided by the Keras library, which saves the weights of the epoch with lower validation error. Once the best model was selected, its weights are saved to be used later for transfer learning, and predictions are made using the test set that was separated from the core data. The predictions were evaluated using a range of metrics, MSE, RMSE, MAE, and MAPE and training time. The predicted values are plotted against the true values for manual observation.

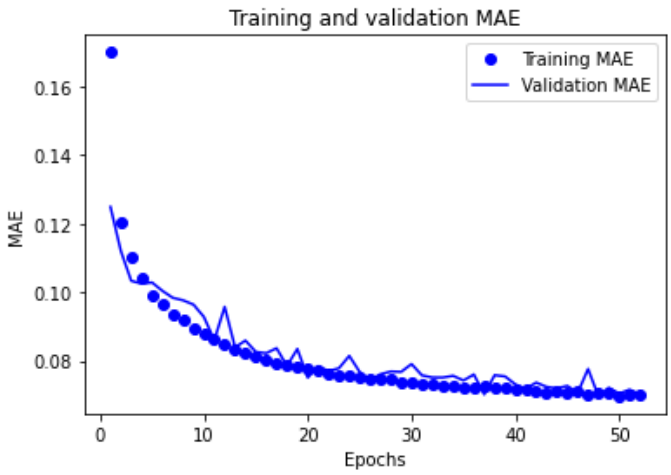

Figure 3.5: Training and validation MAE as function of the number of epochs.

3.4 Transfer learning approach

Transfer learning is a powerful machine learning technique that enables us to use pre-trained models to perform a new task. It allows us to leverage the knowledge learned from previous tasks and apply it to new ones. In this section, it is discussed how it has been used transfer learning to enhance the performance of our load forecasting models.

To implement transfer learning, it has been removed data from the inputs, leaving only a month of data for training and validation of the new models. The previous load forecasting model was then retrained with this reduced data. The outcome of this new model was used as a baseline to evaluate the benefits and performance of transfer learning. We have three Case Scenario (CS)s: the first one involves models trained with an abundance of data, the second one involves models trained in a data-scarce scenario. The next step was to create the third, which is a hybrid scenario in which data is scarce, but we have access to the weights of the pre-trained model from the first case. The three CSs are presented and summarized in Figure 3.6.

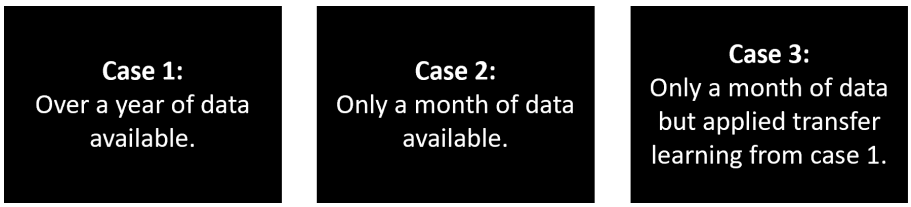

Figure 3.6: Case Scenarios.

To initialize the model, we constructed an architecture identical to the one described in the previous section. These layers were loaded with the weights of the pre-trained model and frozen so that new training sessions wouldn't change the loaded weights. We then added a few new layers to the model that will be trained on the specific dataset of building energy consumption data (Figure 3.7). Added two LSTM layers and a dense output layer. The first LSTM layer with

32 hidden neurons, modeled the sequence of features extracted by the convolutional layer, and the second LSTM layer, with 16 hidden neurons, performed further sequence modeling. Dropout layers were added after each LSTM layer to prevent overfitting. Finally, a dense output layer was added to produce the final prediction.

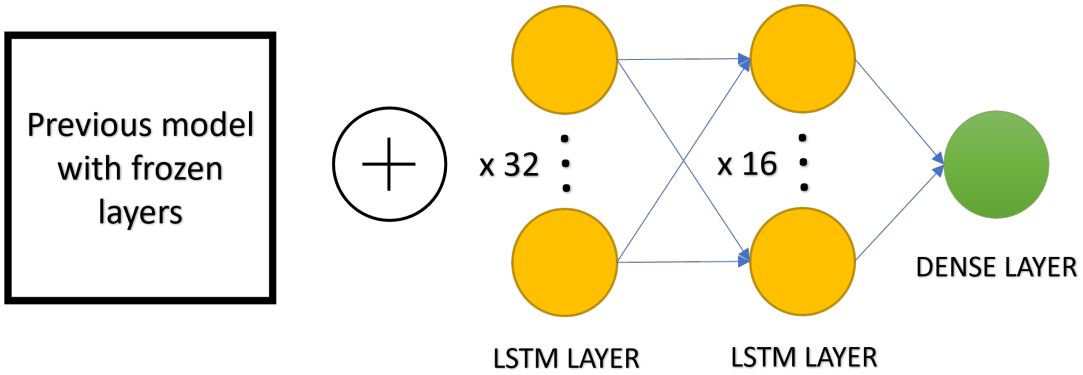

Figure 3.7: Illustration of the transfer learning model architecture.

The transfer learning approach allowed to evaluate the performance of the load forecasting model in comparison to the other CSs. By using the pre-trained model as a starting point, one can expect to reduce the training time in comparison to the first case and improve accuracy in comparison to the second case.

In summary, the process consisted in a data science project in the beginning, where the data is selected and cleaned, followed by the methodology definition, which is facilitated by the previous process. After methodology is defined, the data is separated, in order to create several different models to use as sources and targets. The source models goes through a stage of several trial and error in order to optimize it's performance. Once the results are deemed adequate, a process of knowledge extraction is made, where the weights and hyper-parameters of the source are loaded into the target models. Those layers loaded with the knowledge from the source are freeze, so that further training won't change the results acquired, and new layers are added to be trained with the small target data set, in a process called fine tuning. The architecture and several parameters of the new layers are put through a trial and error process to optimize the model and, as soon as the results go in line with what was expected and further enhancement seem to make little to no effect, the discussion and conclusions are drawn. A workflow of the process is depicted in Figure 3.8.

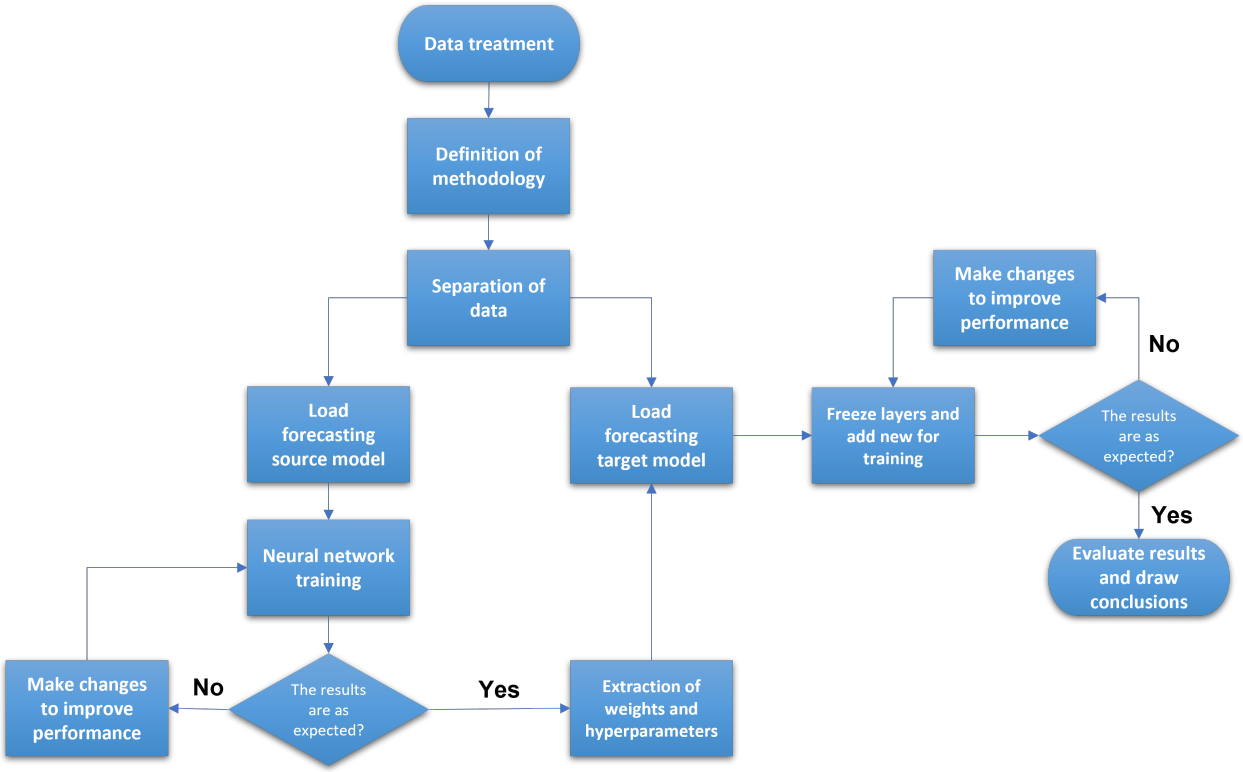

Figure 3.8: Fluxogram describing the entire transfer learning process.

3.4.1 Evaluation metrics

To evaluate the performance of the transfer learning approach, various metrics were used. The following metrics were considered:

- **MAE:** It is the average absolute difference between the actual and predicted values. It gives a measure of how well the model performs on average.
- **MSE:** It is the average squared difference between the actual and predicted values. It is a measure of the model's ability to predict values accurately.
- **RMSE:** It is the square root of the MSE. It gives a measure of the model's ability to predict values accurately while taking into account the scale of the data.
- **MAPE:** It is the average absolute percentage difference between the actual and predicted values. It gives a measure of how well the model performs in percentage terms.

The evaluation metrics were computed for each VB, and the outcomes were stored in a Dataframe[1]. This Dataframe contains the metrics for every VB, encompassing MAE, MSE, RMSE, MAPE, and

[1] A dataframe is a two-dimensional, size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). Essentially, it's a data structure that represents data in a table format, similar to an SQL table or an Excel spreadsheet.

training time. These metrics served to compare the performance of the transfer learning method with the conventional approach.

Scatter plots featuring regression lines were employed to visually represent the model's performance. The scatter plots depict predicted values on the y-axis and true values on the x-axis, while the regression line illustrates the relationship between predicted and actual values. The closer the regression line approaches a forty-five-degree angle, the stronger the correspondence between actual and forecasted values. Additionally, line plots were used to display the model's performance over time. These plots exhibit predicted and true values, enabling a comparison of trends between them.

Both the evaluation metrics and visualizations contributed to the assessment of the transfer learning approach's performance. The results were compared with those of the traditional approach to determine the efficiency of transfer learning in load forecasting.

Chapter 4

Results and Analysis

This chapter presents the results obtained from the load forecasting models developed using the transfer learning approach explained in Chapter 3. This chapter is divided into four sections, each of which focuses on different aspects of the results and analysis. Section 4.1 presents the data cleaning results, where it is provided insights into the preprocessing steps taken to prepare the data for the models.

Section 4.2 provides an in-depth analysis of the load forecasting results for each VB. This section includes a discussion of the performance of the models developed using the transfer learning approach and the baseline models developed without transfer learning. Section 4.3 compares the transfer learning approach to the baseline models in terms of their performance and efficiency. Finally, Section 4.4 summarizes the key findings of the study and draws conclusions about the effectiveness of the transfer learning approach for load forecasting in building energy management.

Moreover, the overall objective of this chapter is to provide a detailed analysis of the results obtained from the load forecasting models developed using transfer learning. By examining the performance of the models across multiple VBs, one can identify the strengths and limitations of the transfer learning approach and provide insights into its potential for improving the efficiency and accuracy of load forecasting in building energy management. The following sections provide a comprehensive overview of the results obtained from studying and analyzing the performance of the models developed using transfer learning.

4.1 Data cleaning results

The data cleaning process applied in this study aimed to ensure that the data used for load forecasting models were of high quality and suitable for analysis. In this section, one can present the results of the data cleaning process described in Chapter 3. The first part consists of selecting which floors were gonna take part in the study and eventually become VBs. The floors excluded were the ones that represented either the reception of the building, a laboratory or just a floor with a really different statistical distribution of data. The floors removed from the study were:

- **Building A floor -1:** The decision to remove this floor was due to the low visibility of a pattern that could benefit from the transferred knowledge from the base VB. This conclusion is based on the graphical analysis of the load in Figure 4.1, where each point presents the consumption (in watts) with a frequency of fifteen minutes.

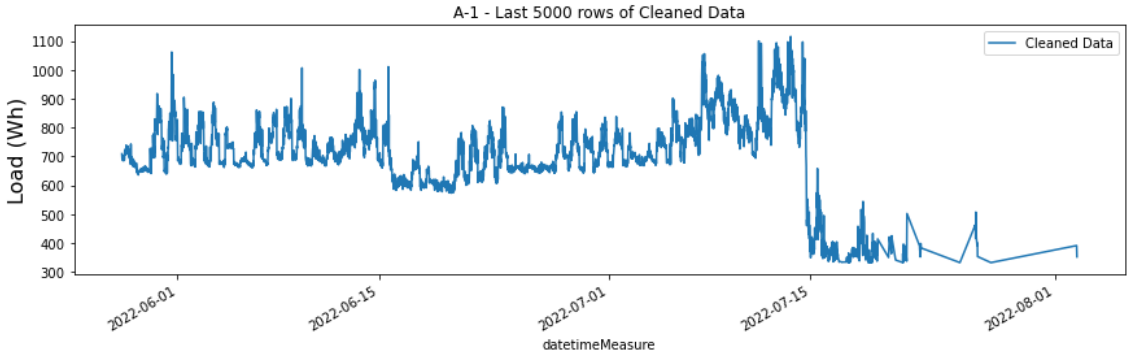

Figure 4.1: Floor A-1 cleaned load (last 5000 rows).

- **Building A floor 0:** This floor is the reception of Building A and represents a consumption pattern that is different from the base VB. While it was not suitable for the purposes of this study, it could potentially be used in a future study to test transfer learning between reception floors of different buildings. The consumption pattern of this floor is presented in Figure 4.2.

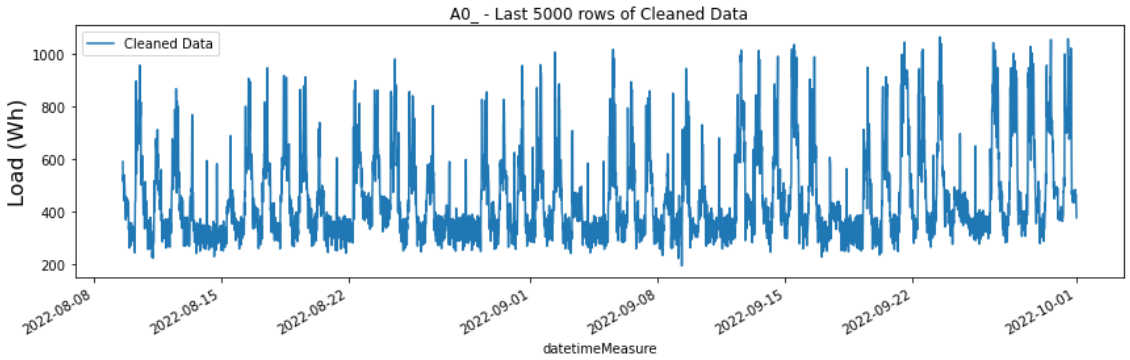

Figure 4.2: Floor A0 cleaned load (last 5000 rows).

- **Building B floor -1:** The aforementioned floor in Building B is INESC TEC's Smart Grids and Electric Vehicles laboratory. Even though it presented a clear consumption pattern, it was removed from the study due to the difference in the nature of the work carried out in that laboratory. The load for this floor can be observed in Figure 4.3.

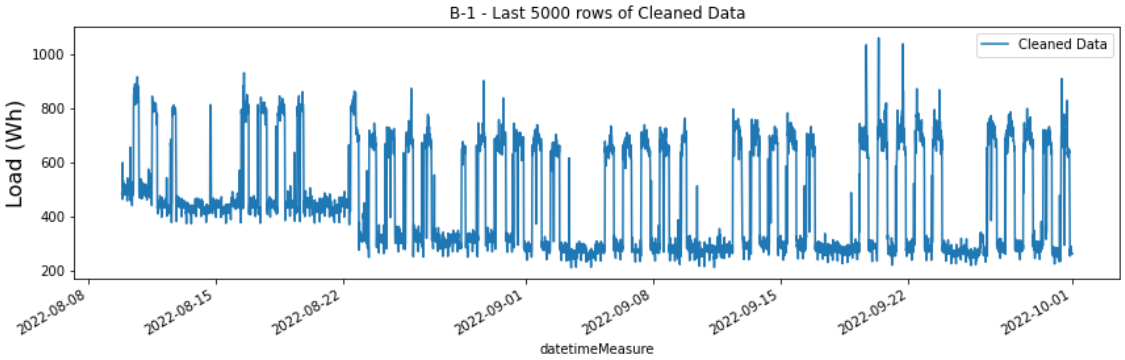

Figure 4.3: Floor B-1 cleaned load (last 5000 rows).

- **Building B floor 0:** Similar to building A floor 0, building B floor 0 also corresponds to the reception area and presents a different consumption pattern from the base building. The load consumption for this floor is shown in Figure 4.4. It is evident that during the first half of the analyzed period, the consumption behaves in an unpredictable and non-recurring manner, almost like noise, which further justifies the removal of this floor from the study.

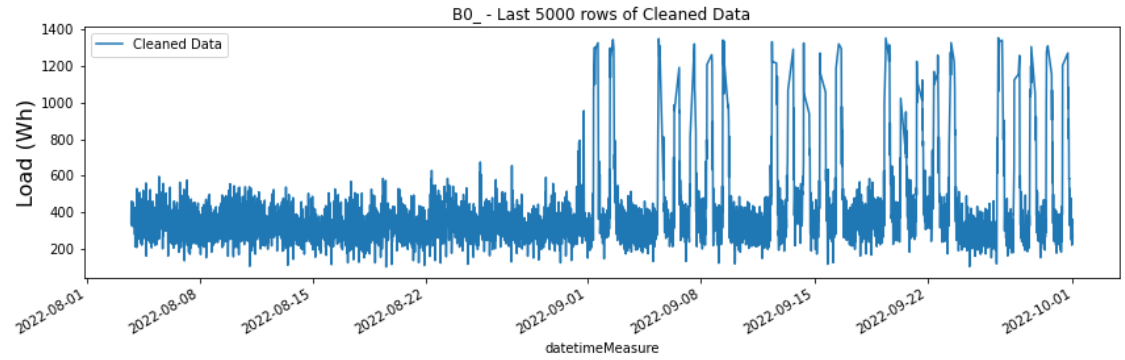

Figure 4.4: Floor B0 cleaned load (last 5000 rows).

- **Building B floor 5:** The analysis of the load consumption on Building B floor 5, as presented in Figure 4.5, indicates that this time series does not exhibit any discernible pattern that could contribute to the objectives of this paper. As expected, this is due to the fact that this floor is used as a rooftop in the building, which explains not only the low consumption but also the completely different statistical distribution of data compared to the other floors. Thus, the decision to remove this floor from the data set was necessary for the accuracy of the load forecasting models developed in this study.

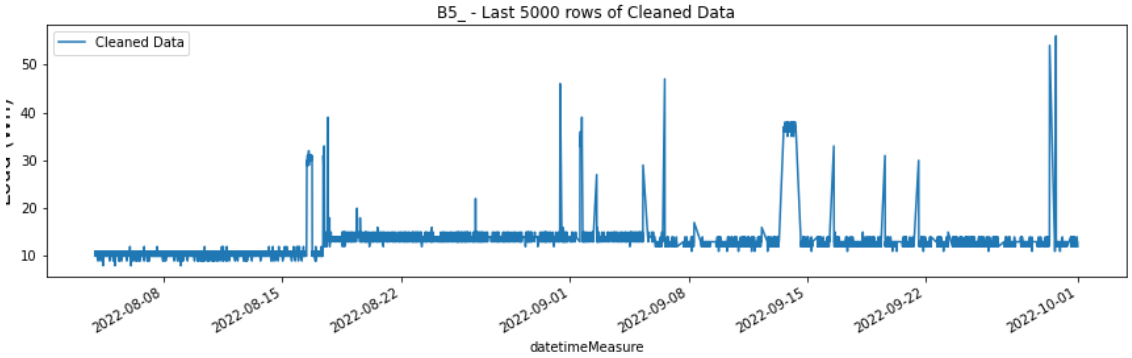

Figure 4.5: Floor B5 cleaned load (last 5000 rows).

The removal of all the aforementioned floors is justifiable, as they do not provide any significant pattern that could contribute to the development of the load forecasting models. The base VB, which corresponds to building B floor 2, has a high correlation with the day of the week and the hour of the day, which are the main features that this work attempts to extract and transfer. This is evident from the data distribution shown in Figure 4.6. Thus, it can be concluded that the floors removed do not provide any useful information for the transfer learning approach used in this study.

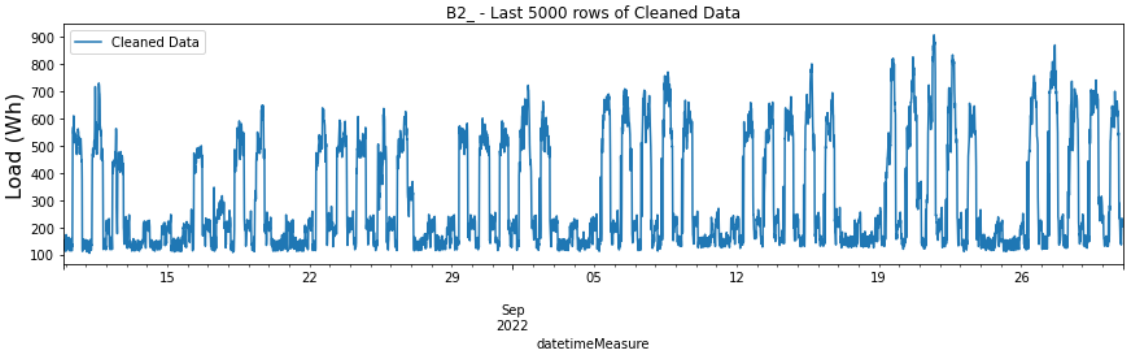

Figure 4.6: Floor B2 cleaned load (last 5000 rows).

In the data cleaning process, as described in Section 3.1, several rows were identified as outliers and were removed for the remaining floors. Table 4.1 provides the number of removed rows for each floor.

Table 4.1: Summary of the data cleaning results.

VB	Nº rows	Nº removed	Nº final	% Removed
A1	53175	4453	48722	8.37
A2	54290	5072	49218	9.34
A3	54289	4536	49753	8.36
A4	54291	4921	49370	9.06
B1	54317	4574	49743	8.42
B2	54318	4604	49714	8.48
B3	53789	10099	43690	18.78
B4	54315	11542	42773	21.25

By analyzing the graphical comparison of the time series before and after cleaning, it is clear that the process was successful. Figure 4.7 illustrates this comparison for the Base VB. Without data cleaning, it was impossible to visualize the data due to clear errors in data collection.

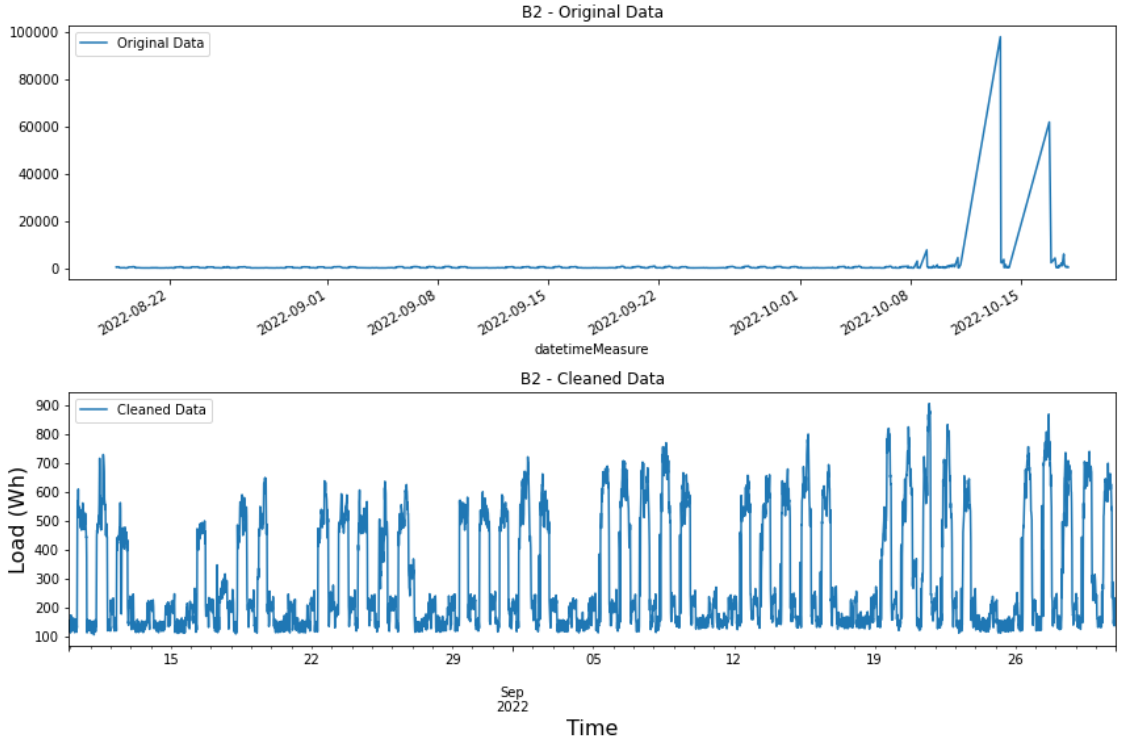

Figure 4.7: Before and after cleaning on B2 (last 5000 rows).

Despite presenting clearly different consumption patterns and a high presence of outliers, buildings B3 and B4 were kept in the study to demonstrate the results of transfer learning even in non-ideal circumstances. Both buildings represent the VBs with the least amount of correlation among all buildings, as shown in Table 3.2. Figure 4.8 presents the time series of VB B3, while Figure 4.9 shows the same for B4. It is clear that these VBs exhibit significantly different patterns when compared to the base VB in Figure 4.6, which will significantly impact the results discussed in the following sections.

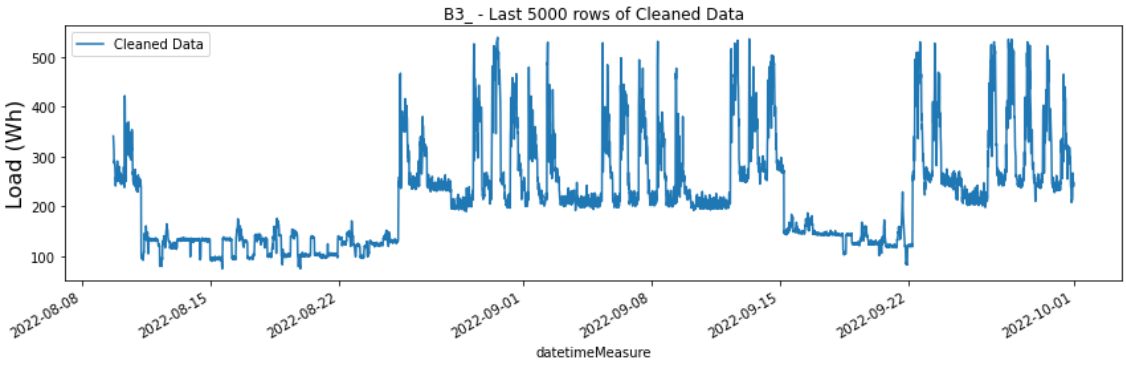

Figure 4.8: Cleaned time series of VB B3 (last 5000 rows).

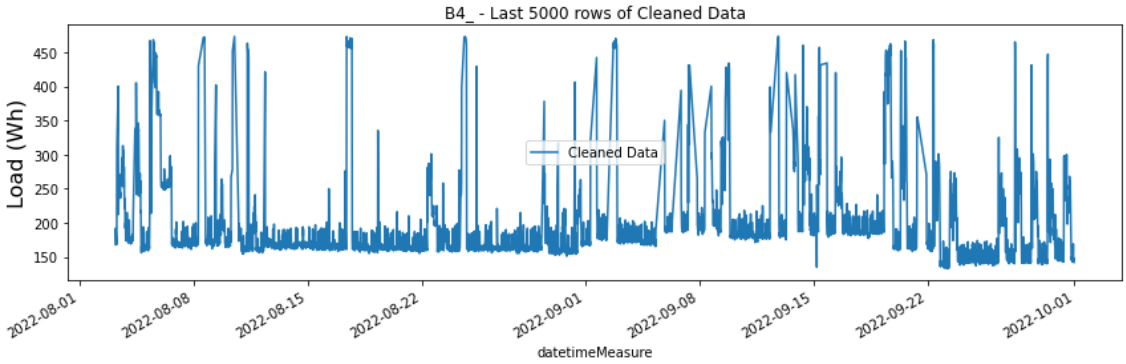

Figure 4.9: Cleaned time series of virtual building B4 (last 5000 rows).

4.2 Analysis of load forecasting results for each VB

In this section, it is presented a detailed analysis of the load forecasting results obtained for each VB. One can assess the performance of the proposed transfer learning model by comparing its predictions with the actual load values for each building. In addition, it is provided insights into the accuracy and reliability of the model for different scenarios. By analyzing the results of the experiments, it is intended to identify the strengths and weaknesses of the proposed model and to shed light on the factors that may influence its performance. This analysis allows us to draw conclusions on the suitability of the transfer learning approach for load forecasting in VBs and to identify potential avenues for future research.

4.2.1 Load forecasting models with full data availability

To start the analysis under the first CS, as explained in Section 3.4 and showcased in Figure 3.6, the evaluation metrics results for each VB are presented in Table 4.2, as discussed in Chapter 3. While these metrics alone may not be enough to understand and evaluate the models, they serve as a basis for comparing the transfer learning approach with the baseline models in the subsequent sections. The metrics represent the error between the actual and predicted values for the test set,

which was not provided to the model during training and spans a duration of two months. Other than the errors in mae, mse, rmse and mape, the table also presents the time it took for the model to compile, the angular coefficient and the R^2 of the regression line between the predicted and the validation data set.

Table 4.2: Results of the evaluation metrics for each VB in the first case scenario.

VB	**mae**	**mse**	**rmse**	**mape (%)**	**time (s)**	**m**	**R^2**
A1	141.27	37405.30	193.40	53.62	95.79	1.013751	0.834947
A2	93.74	19764.76	140.59	25.87	90.16	0.739909	0.74964
A3	225.09	87791.56	296.30	87.03	52.08	0.943177	0.829429
A4	93.55	17618.90	132.74	24.92	80.04	1.020041	0.879153
B1	190.07	54619.77	233.71	52.01	57.13	0.899292	0.796311
B2	60.85	8267.05	90.92	25.02	91.88	0.835454	0.855772
B3	51.85	5553.23	74.52	29.90	112.94	0.505781	0.506303
B4	43.44	3860.55	62.13	20.57	53.26	0.542767	0.452881

In order to get a comprehensive understanding of the model's performance, it is necessary to evaluate the results for each VB individually. For the sake of clarity and to ensure that the graphics fit properly within the size of the thesis layout, the following visual representations will only display a segment of the test set and corresponding predictions. This test set consists of 5376 rows of data with a frequency of 15 minutes, spanning a period of two months. However, for ease of presentation and analysis, only the final 1500 rows of data will be displayed.

- **Building A floor 1:** Despite the high error rate identified in Table 4.2, it was anticipated that the model would perform well for this VB, given that the data presented a discernible pattern that could be comprehended by the model. Despite a MAPE of 53.62%, a graphical analysis revealed that the predictions closely matched the actual values, affirming that the model had indeed captured the patterns, as evidenced by the visual inspection of Figure 4.10 and Figure 4.11. The regression model presented a slope of 1.01 and a R^2 of 0.83, both of them being close to one, which indicates the predictions are in line with the test set. All things considered, this model was deemed to be good and reliable despite its high MAPE.

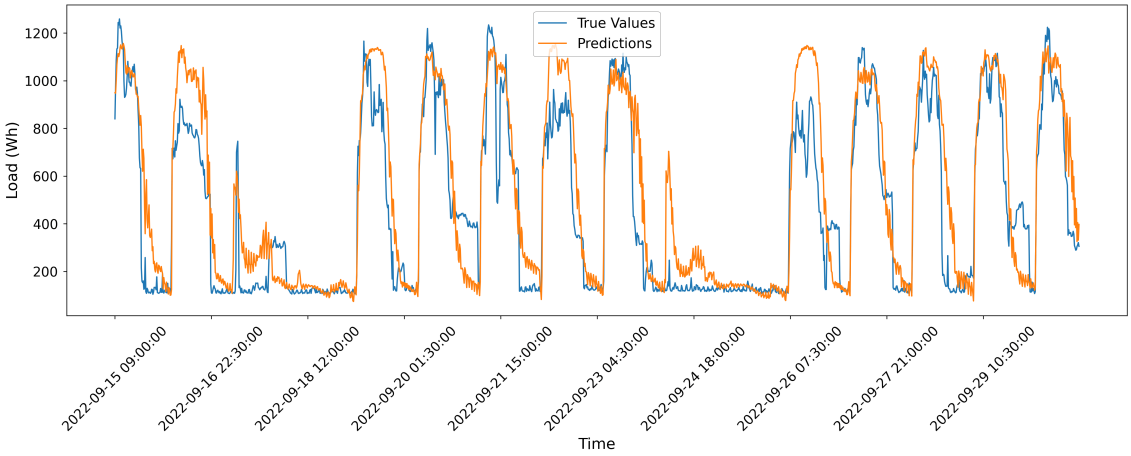

Figure 4.10: Actual load and predicted load for VB A1 (last 1500 rows).

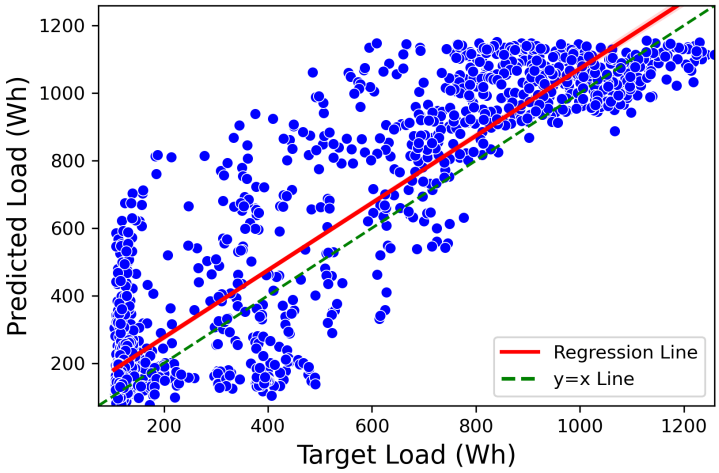

Figure 4.11: Scatter plot with regression line between actual and predicted load for VB A1.

- **Building A floor 2:** The second VB in Building A, in contrast to the first one, demonstrates a MAPE of 25.87% as indicated in Table 4.2, representing one of the lowest errors in the entire set. This smaller error was expected due to the statistical distribution of the data from the provided training set, which presents a much clearer pattern, facilitating the comprehension from the model. Further analysis of the predicted load, as shown in Figure 4.12, and the regression, as demonstrated in Figure 4.13, with 0.73 angular coefficient and R^2 of 0.74, supports the low error rate and provides additional evidence of the model's accuracy.

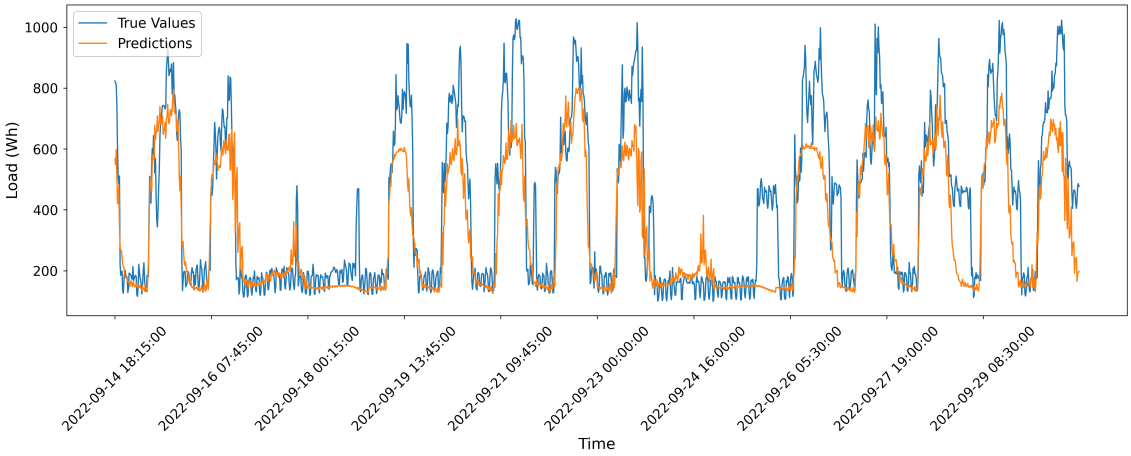

Figure 4.12: Actual load and predicted load for VB A2 (last 1500 rows).

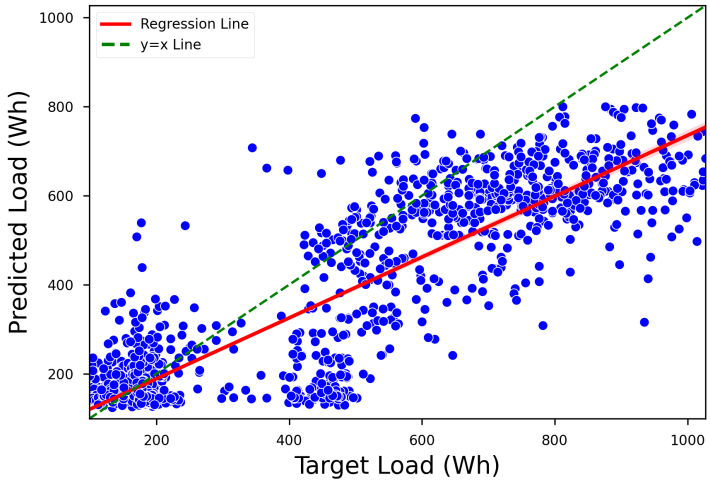

Figure 4.13: Scatter plot with regression line between actual and predicted load for Virtual building A2.

- **Building A floor 3:** For VB A3, similar to A2, it exhibits the highest MAPE among all VBs (87%), but this high MAPE is not reflect in the graphics. Upon evaluating Figure 4.14, it becomes clear that the model provides predictions that accurately follow the actual values. The regression also does not explain the high MAPE presented in Table 4.2, since the slope of the regression (m=0.94 and R^2=0.83) is close to one, as seen in Figure 4.15.

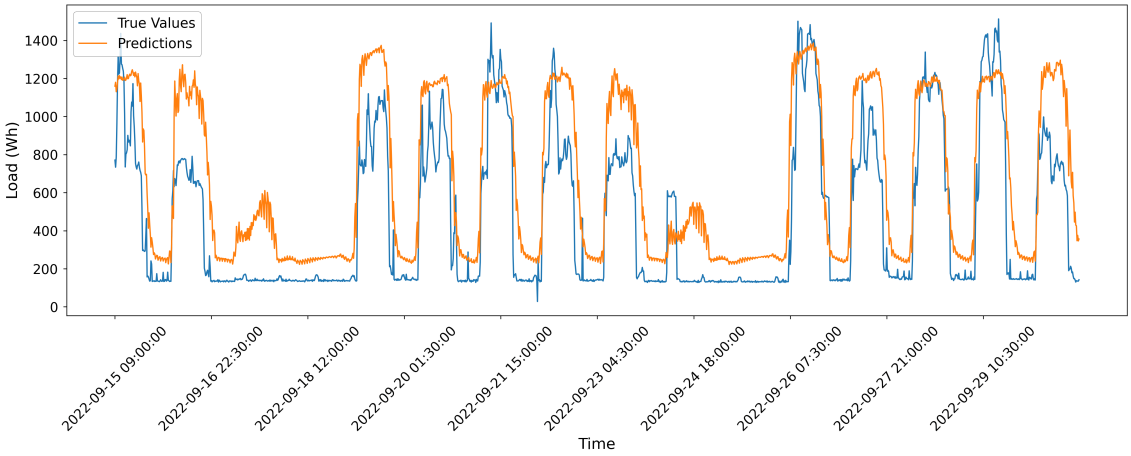

Figure 4.14: Actual load and predicted load for virtual building A3 (last 1500 rows).

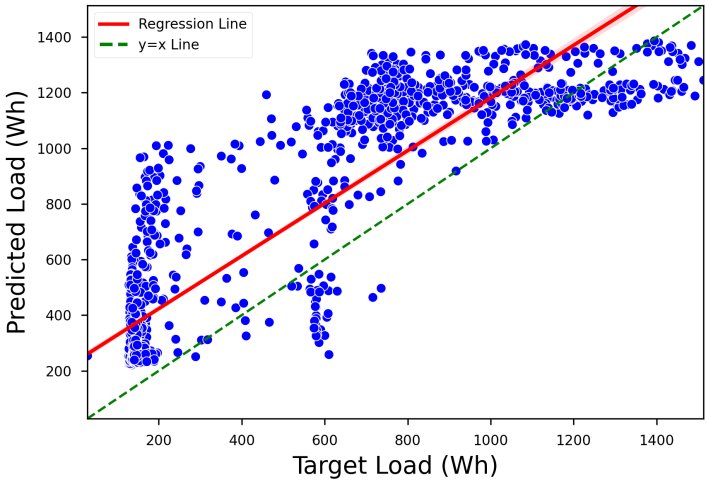

Figure 4.15: Scatter plot with regression line between actual and predicted load for virtual building A3.

- **Building A floor 4:** Building A4 is characterized by a MAPE of 24.92%, which represents the third smallest value in Table 4.2. The adequacy of the model's predictions can be seen in Figure 4.16, where the predicted values closely follow the actual values. Additionally, the regression analysis of the model, depicted in Figure 4.17, also displays one of the best results.

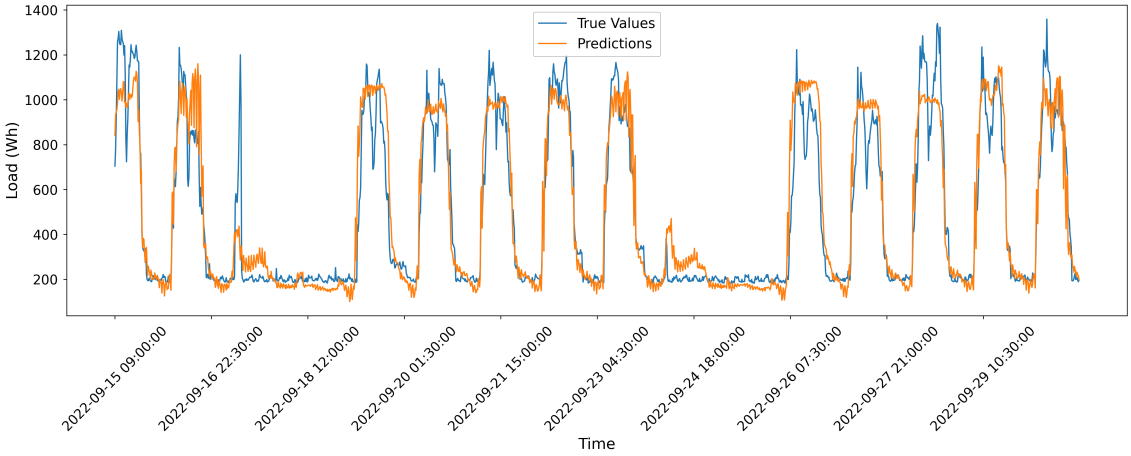

Figure 4.16: Actual load and predicted load for VB A4 (last 1500 rows).

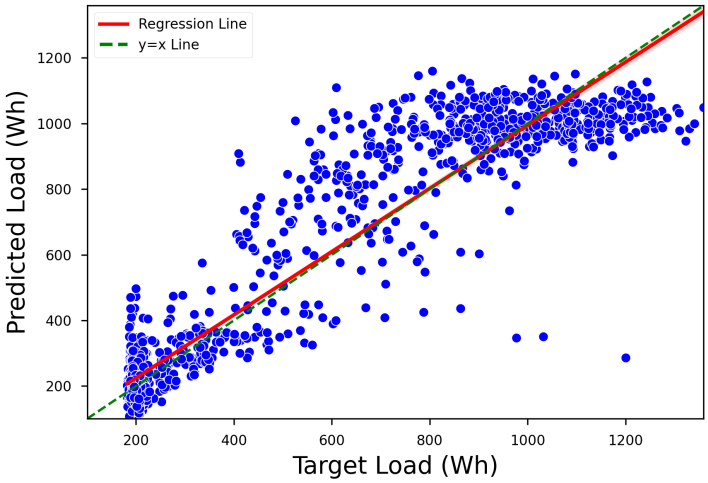

Figure 4.17: Scatter plot with regression line between actual and predicted load for virtual building A4.

- **Building B floor 1:** Building B floor 1 exhibits a MAPE of 52.01%, which is significantly higher than the best-performing models. However, this result was somewhat anticipated due to the high measurement errors in most building B floors. These errors had a substantial impact on the statistical distribution of the input time series. Despite the model's trend to overestimate the load during most of the period, Figure 4.18 reveals that it can recognize weekly patterns and performs better during peak hours than during valley hours. The regression line in Figure 4.19 indicates that the model performs relatively well, as the slope and the R^2 are close to one (m=0.89 and R^2=0.79). However, it also suggests that the model has a slight bias towards the overestimation of the load, as the dots tending to be over the regression line indicates.

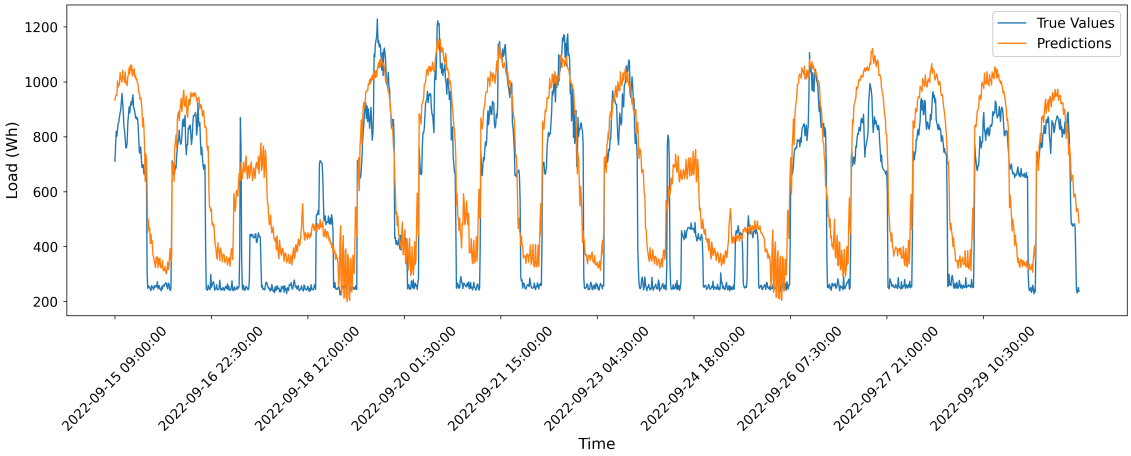

Figure 4.18: Actual load and predicted load for VB B1 (last 1500 rows).

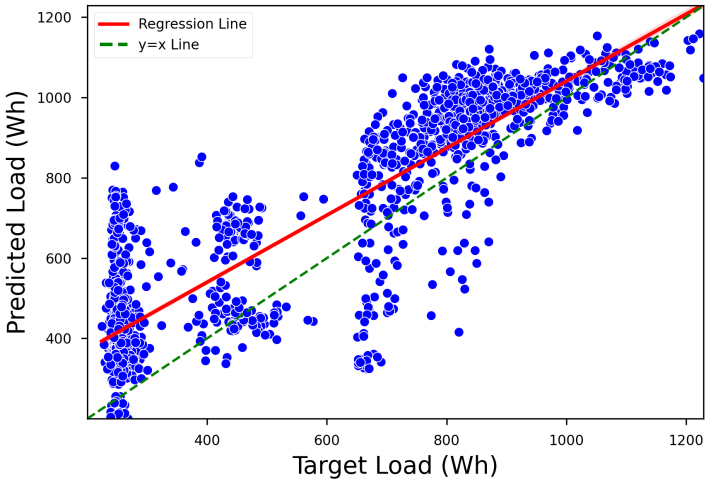

Figure 4.19: Scatter plot with regression line between actual and predicted load for virtual building B1.

- **Building B floor 2:** The analysis of B2 requires special attention as it serves as the base model, presenting a relatively small MAPE of 25.02%. This low error is expected since, unlike other floors in building B with higher measurement errors, B2 exhibits a clear and consistent statistical distribution pattern, as shown in Figure 4.6. This figure also reveals a higher load measurement in September compared to the previous month, contributing to the underestimation of the model in Figure 4.20, which displays the last two weeks of September. The regression line in Figure 4.21 shows a slope very close to one, indicating a good model performance. However, it also reveals the model's tendency to underestimate the load in certain periods.

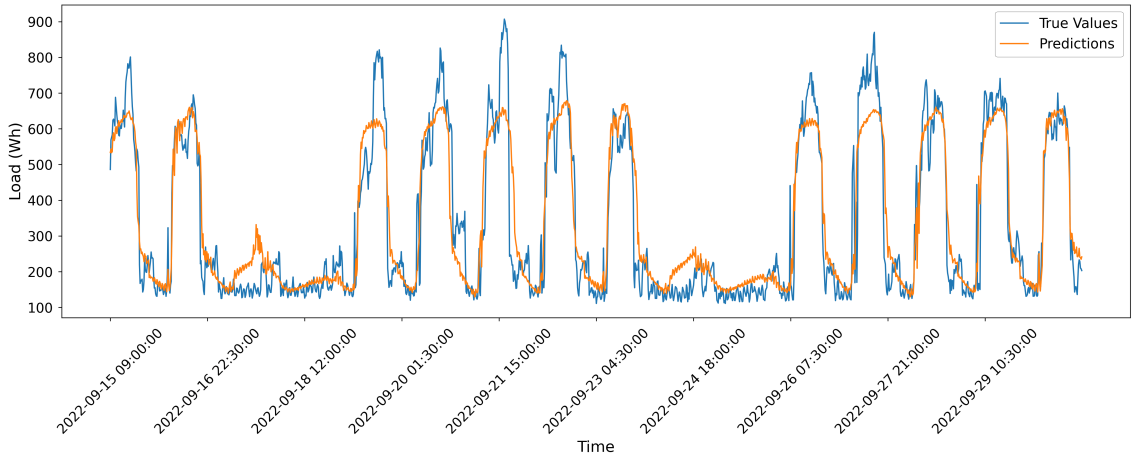

Figure 4.20: Actual load and predicted load for VB B2 (last 1500 rows).

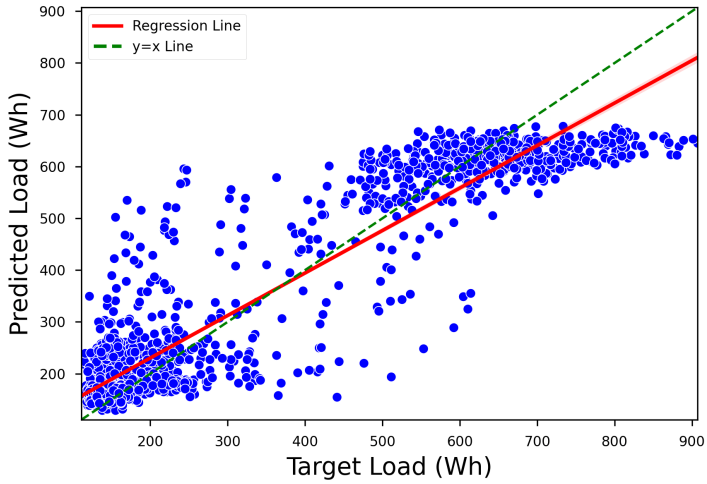

Figure 4.21: Scatter plot with regression line between actual and predicted load for virtual building B2.

When selecting the 1500 rows before the last 1500 rows, the graph in Figure 4.22 illustrates that during this period, the model exhibits an even better performance by avoiding the unusually high consumption at the end of September as shown in Figure 4.6. This reinforces the choice of B2 as the base model.

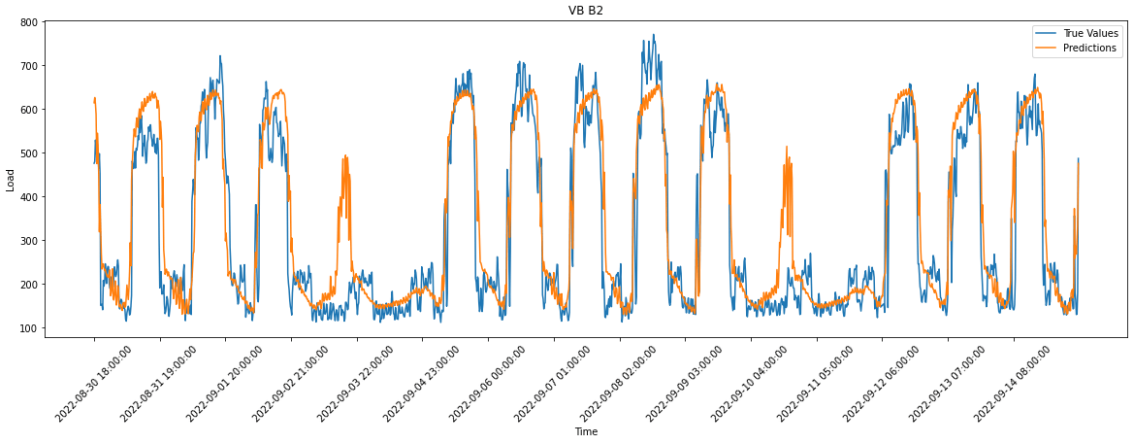

Figure 4.22: Actual load and predicted load for VB B2 (Before the last 1500 rows).

- **Building B floor 3:** Building B3 was a unique case from the beginning, as it had a significantly higher percentage of removed rows during the data cleaning process, amounting to 18.78%, which was evidence of the high measurement errors in the floor. Despite this, B3 did not have one of the highest MAPE values, instead having a value of 29.9%. However, Figure 4.23 depicts a low degree of accuracy in the model, visually appearing lower than other models with higher MAPE values. The statistical distribution of the input data shown in Figure 4.8 displays an unusual pattern that does not correspond to the expected load consumption during the week, which would justify the removal of B3 from the study. Nevertheless, it was decided to keep B3 in the analysis to test transfer learning even in non-ideal scenarios. The regression line in Figure 4.24 showed a modest fit of the model, with the angular coefficient of 0.5 and R^2 of 0.5, but also presented extreme outliers.

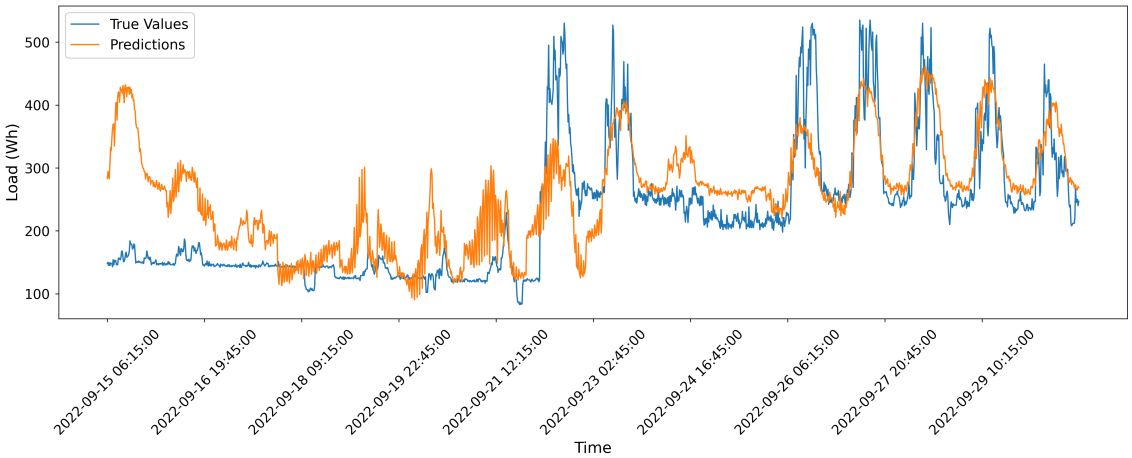

Figure 4.23: Actual load and predicted load for VB B3 (last 1500 rows).

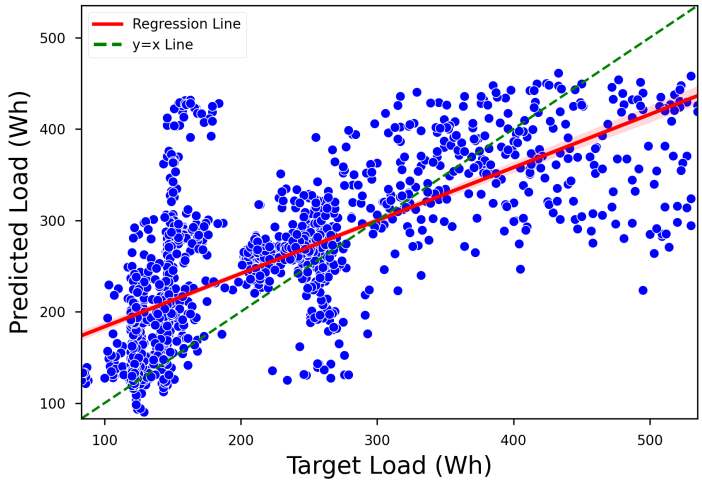

Figure 4.24: Scatter plot with regression line between actual and predicted load for VB B3.

- **Building B floor 4:** The analysis for B4 is similar to the analysis for B3. B4 had the highest percentage of removed rows in Table 4.1, indicating a higher error in the input data due to measurement errors on this floor. However, B4 had an unexpectedly low error among all models, around 20.57%. The graphical analysis of the predicted load in Figure 4.25 shows a strange weekly pattern of consumption, with the model struggling to match it with low success. The regression line in Figure 4.25 shows a good fit but a high bias, similar to B3.

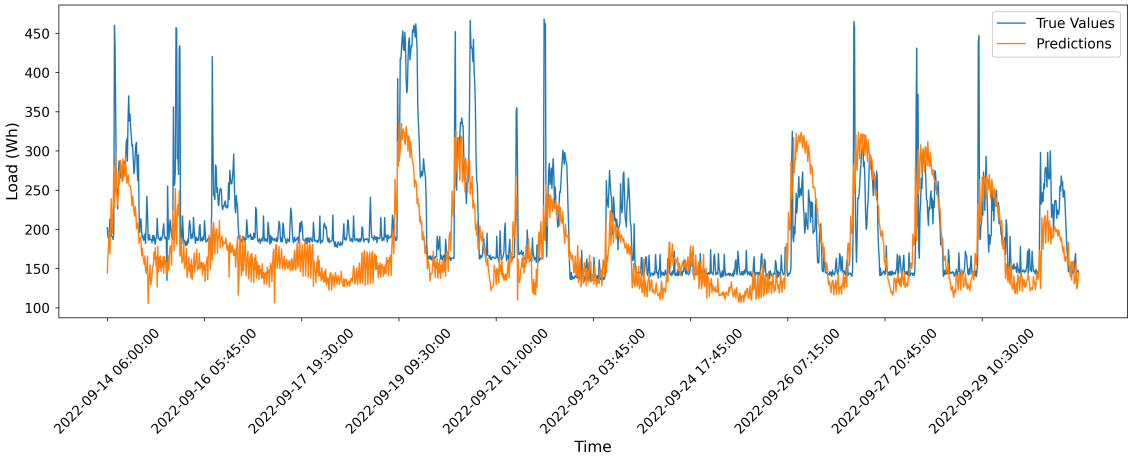

Figure 4.25: Actual load and predicted load for VB B4 (last 1500 rows).

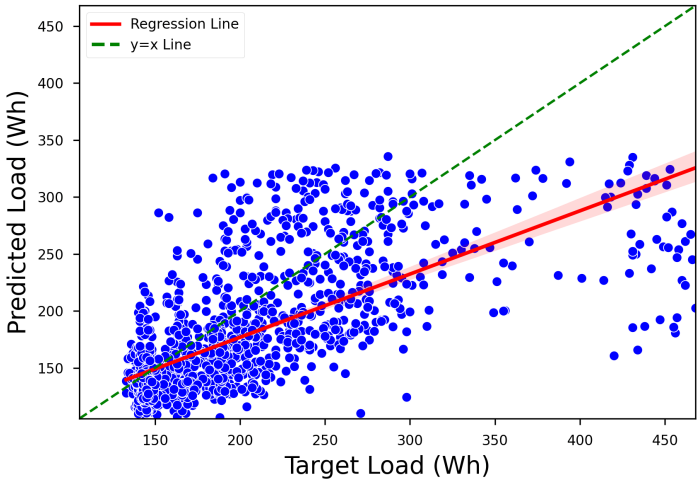

Figure 4.26: Scatter plot with regression line between actual and predicted load virtual building B4.

Note that, as shown in Table 4.2, the VBs that performed best according to MAPE are B4, A4, B2, and A2. While MAPE is a useful metric for evaluating model performance, it is important to use a variety of evaluation metrics and tools, as some may have different characteristics that can lead to incorrect conclusions. This is particularly evident in some of the VBs in this study, such as B4, which has the smallest MAPE but a graphical evaluation reveals that its performance is worse than other VBs with higher MAPE. Therefore, a comprehensive analysis of multiple metrics is crucial for an accurate evaluation of model performance.

Upon closer inspection of the results, it can be observed that the models tend to overestimate the load during the night, while underestimating it during the day. This behavior may be attributed to the fact that the training set contains more data for the daytime period, thus causing the model to bias towards these patterns. It is recommended to conduct further analysis and adjustments to improve the model's performance during nighttime hours. However, despite this limitation, the model shows promise and is deemed sufficient for the proposed case study on the implementation and analysis of transfer learning. While further refinement of the model could potentially enhance the results, it is essential to consider the scope and focus of this work. The primary objective of this research is not to develop the absolute best load forecasting model, but to effectively apply transfer learning in the context of building load forecasting. Consequently, the resources required for additional optimization may not yield proportional improvements in the context of transfer learning application, thus not justifying the associated costs.

4.2.2 Load forecasting models with scarcity of data

This section focuses on the models developed under the second CS presented in Section 3.4 and illustrated in Figure 3.6. This scenario is characterized by the limited availability of data and is included for comparison purposes, as both CSs 2 and 3 have the same data availability. This will

allow us to assess the performance of the transfer learning approach compared to that of a deep learning approach that does not involve knowledge transfer.

The discussion begins by presenting the results of the evaluation metrics for CS 2 in Table 4.3. It is clear that almost all errors have increased, as the available data for this CS is limited to only 2688 rows (four weeks of data). Additionally, the training time is significantly reduced, although the model tends to take more epochs to complete training, as the amount of data to learn from is significantly less than in CS 1. However, as stated previously, a simple analysis of the errors presented by the model is insufficient to evaluate the model accurately. Therefore, this study provides a case-to-case analysis, equal to the one in the previous section, to further understand the performance of the models.

Table 4.3: Evaluation metrics for the load forecasting in CS 2.

VB	mae	mse	rmse	mape (%)	time (s)	m	R^2
A1	277.32	148086.06	384.82	61.57	10.48	-0.01697	0.019916
A2	194.16	92704.02	304.47	49.38	19.71	-0.01106	0.010452
A3	264.74	176568.40	420.20	61.82	11.20	0.011686	0.011166
A4	206.15	130563.83	361.34	34.59	20.53	-0.00466	0.027029
B1	238.54	110190.49	331.95	39.98	15.63	0.074444	0.467513
B2	144.18	47455.83	217.84	41.65	12.97	-0.01305	0.020612
B3	87.04	12478.05	111.71	54.28	12.35	0.016114	0.042767
B4	35.39	3749.88	61.24	15.13	12.97	0.01142	0.002411

- **Building A floor 1:** In VB A1, the expected increase in errors due to scarce data is clear, as presented in Table 4.3. The MAPE grows by almost 15% and the RMSE increases by almost 99%. Figure 4.27 clearly shows that, for the last 1500 rows of the test set, the model is unable to accurately predict the load consumption pattern, in contrast to its counterpart from CS 1. The increase in RMSE more clearly shows the decrease in accuracy than the increase in MAPE. Additionally, the regression in Figure 4.28 has a slope far from the forty-five degrees reference, indicating that the model cannot accurately capture the pattern.

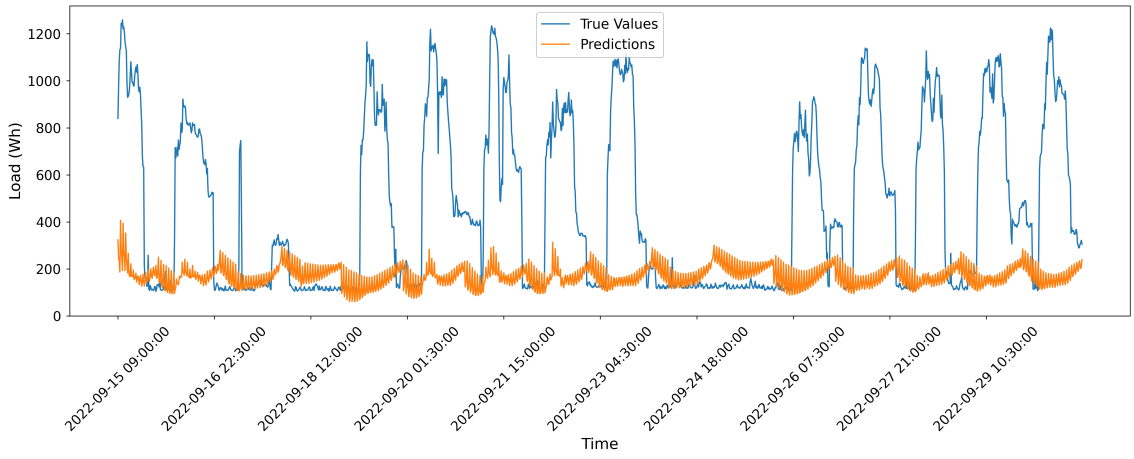

Figure 4.27: Actual load and predicted load for VB A1 in CS 2 (last 1500 rows).

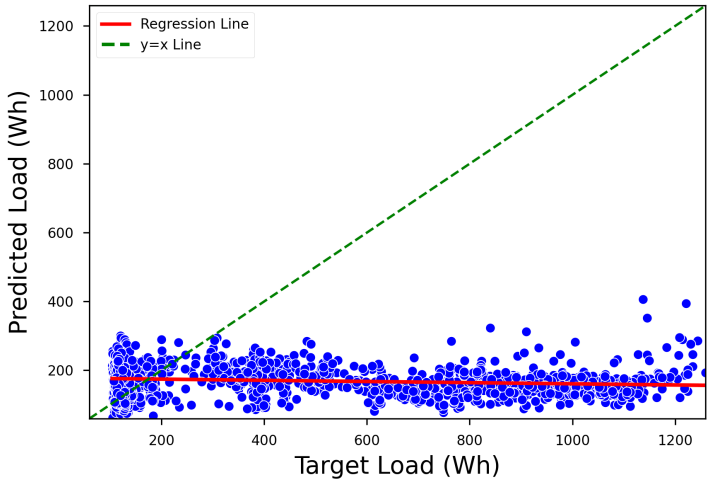

Figure 4.28: Scatter plot with regression line between actual and predicted load for virtual building A1 in CS 2.

- **Building A floor 2:** For VB A2, it is observed that the MAPE has increased by 90.8% in comparison to CS 1, while the RMSE has grown by a staggering 116%. The growth in error is clear when analyzing Figure 4.29, which shows an almost continuous line that fails to capture any pattern of consumption. Furthermore, Figure 4.30 shows a slope with a negative inclination and high bias, confirming the visual analysis of the line plot.

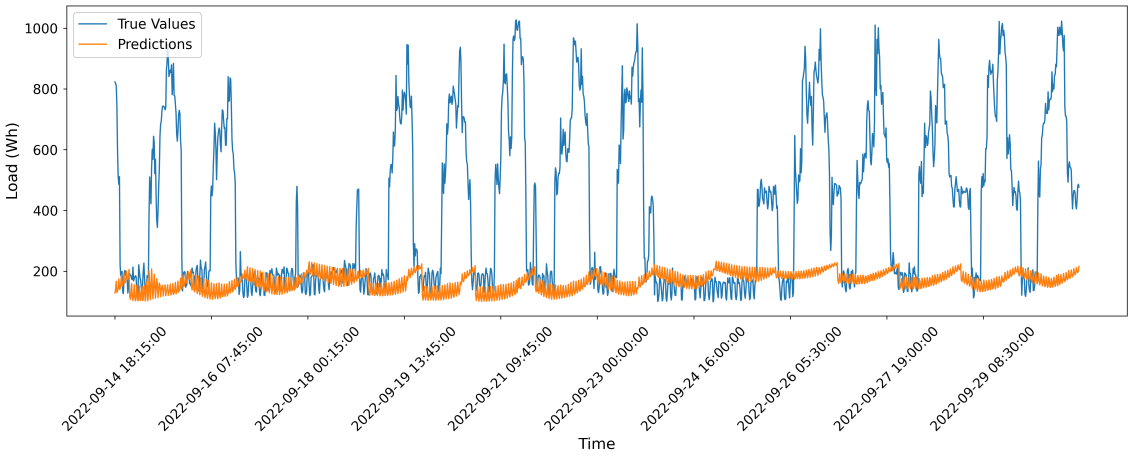

Figure 4.29: Actual load and predicted load for VB A2 in CS 2 (last 1500 rows).

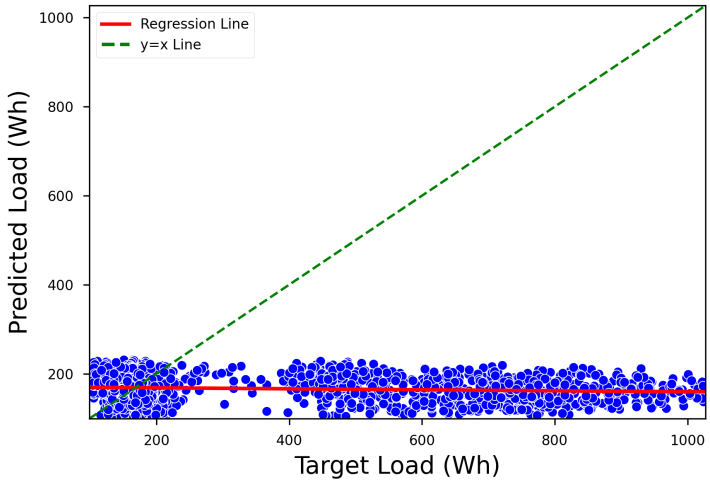

Figure 4.30: Scatter plot with regression line between actual and predicted load for VB A2 in CS 2.

- **Building A floor 3:** When analyzing the metrics for VB A3, it is important to note that the MAPE for CS 2 shows an unexpected reduction of 28.96% compared to CS 1. However, the RMSE presents a significant increase of 41.82%, which is expected since CS 2 has fewer data available, making it a worse scenario. When analyzing Figure 4.31 in comparison to Figure 4.14, it is clear that the model's error is much higher in CS 2, as it is unable to identify the pattern or magnitude of energy consumption. The regression line in Figure 4.32 also shows evidence of the model's poor performance, with a negative slope and a distribution of dots that do not fit the expected pattern. For this VB, the MAPE shows a different scenario than the RMSE, but the graphics presented clearly show that the RMSE metric is more reliable, emphasizing the need for several evaluation metrics.

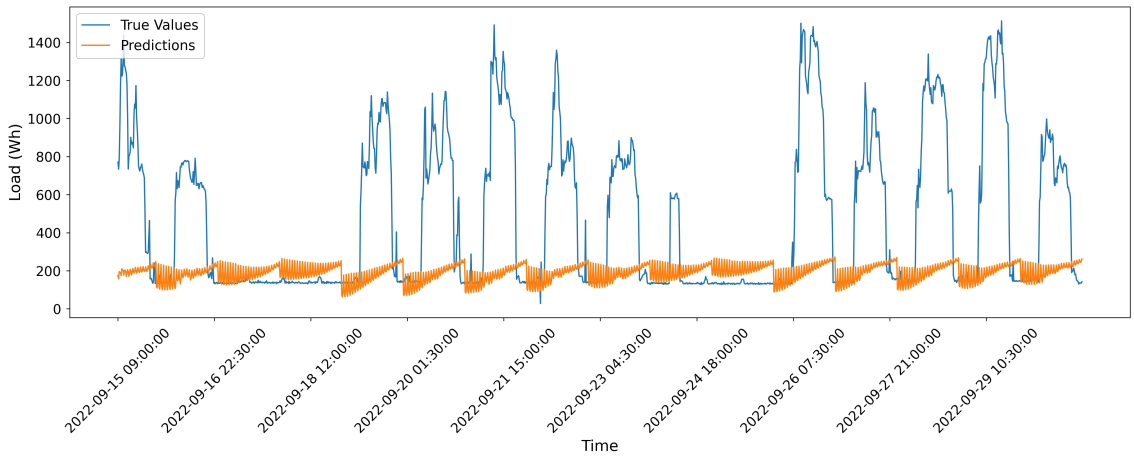

Figure 4.31: Actual load and predicted load for VB A3 in CS 2 (last 1500 rows).

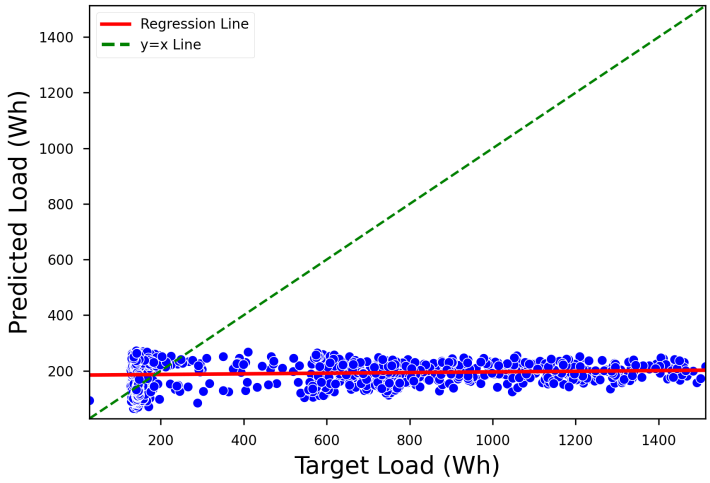

Figure 4.32: Scatter plot with regression line between actual and predicted load for VB A3 in CS 2.

- **Building A floor 4:** For A4, all error metrics show an increase in comparison to CS 1. The MAPE shows an increase of 38.77%, while the RMSE shows a much higher increase in error, of about 172.22%. Looking at Figure 4.16, it is clear that the model misses the pattern and magnitude of energy consumption, without a clear pattern of consumption. The visual error increase is much more noticeable with the RMSE than with the MAPE. The regression line in Figure 4.34 also shows a negative slope and a distribution that is different than expected, reinforcing the impression given by the line plot.

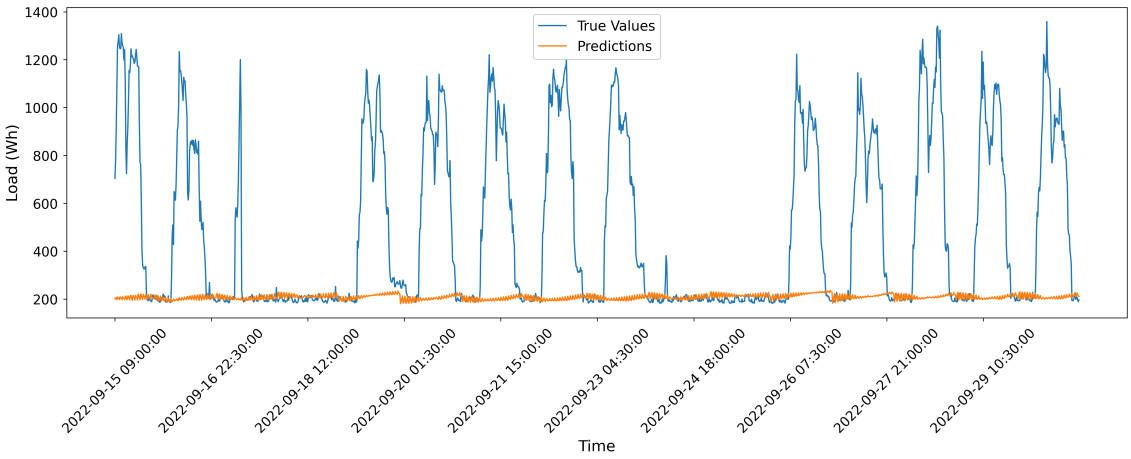

Figure 4.33: Actual load and predicted load for VB A4 in CS 2 (last 1500 rows).

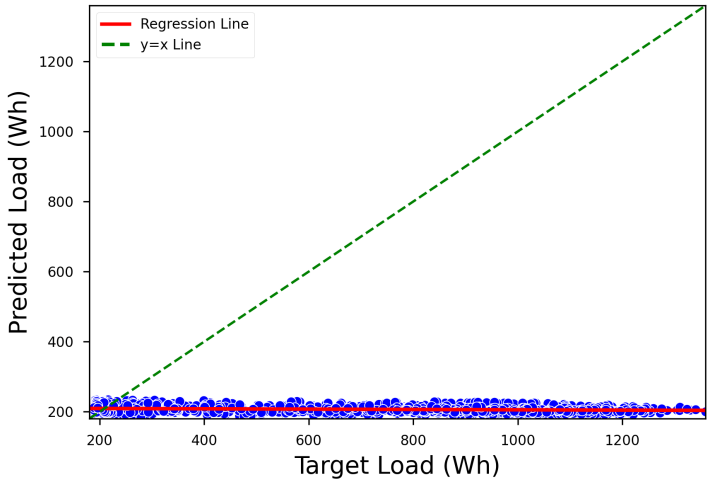

Figure 4.34: Scatter plot with regression line between actual and predicted load for virtual building A4 in CS 2.

- **Building B floor 1:** The results for VB B1 are similar to those observed for A3, with a decrease of 23.13% in MAPE and an increase of 42.05% in RMSE when compared to CS 1. The plot in Figure 4.35 shows a significant deviation from the expected pattern, thus reinforcing the scenario presented by the RMSE. Moreover, the regression analysis depicted in Figure 4.36 also confirms this trend, with a regression line that is far from desirable angular coefficient and a distribution of dots that does not follow the expected pattern for a good model.

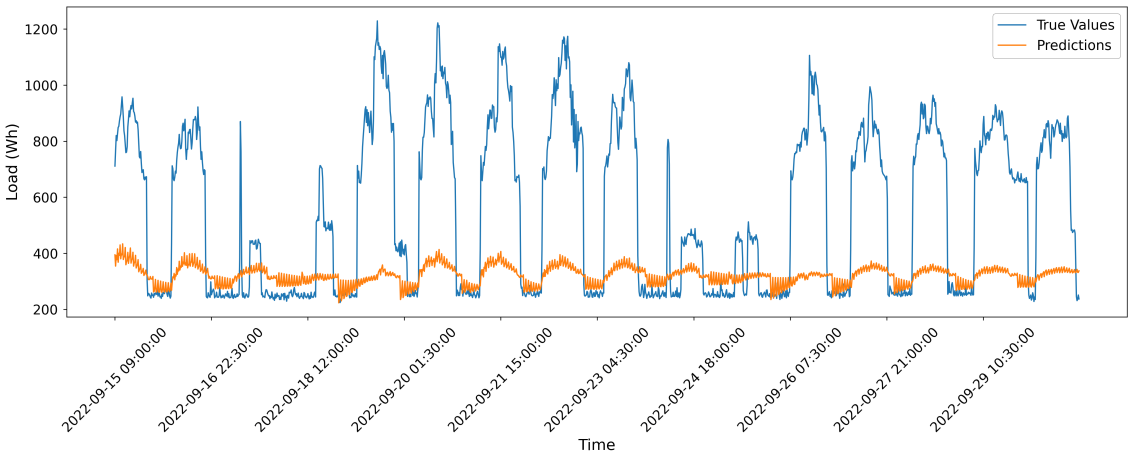

Figure 4.35: Actual load and predicted load for VB B1 in CS 2(last 1500 rows).

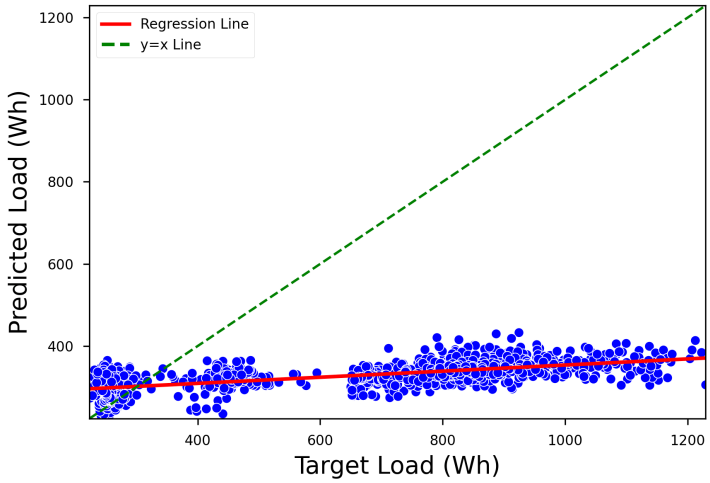

Figure 4.36: Scatter plot with regression line between actual and predicted load for virtual building B1 in CS 2.

- **Building B floor 2:** For the base VB (B2), the results align with the expectation of increased errors across all metrics in comparison to CS 1. Specifically, the MAPE shows a 66.45% increase and the RMSE shows a 139.59% increase. Figure 4.37 also indicates the poor performance of the model, with an inability to capture the pattern of consumption due to the low availability of data. Similarly, Figure 4.38 corroborates the same outcome, thus highlighting the difficulty faced in accurately modelling the load consumption in this case.

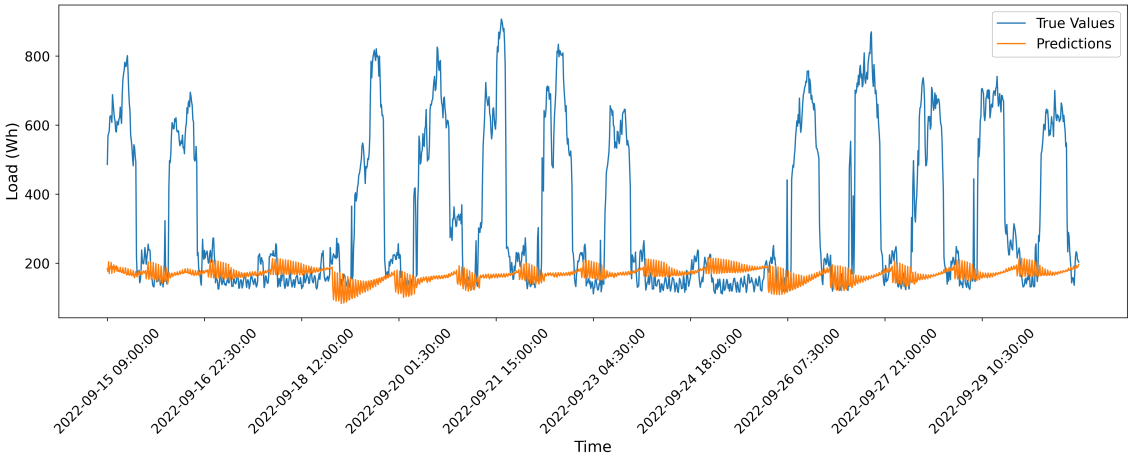

Figure 4.37: Actual load and predicted load for VB B2 in CS 2 (last 1500 rows).

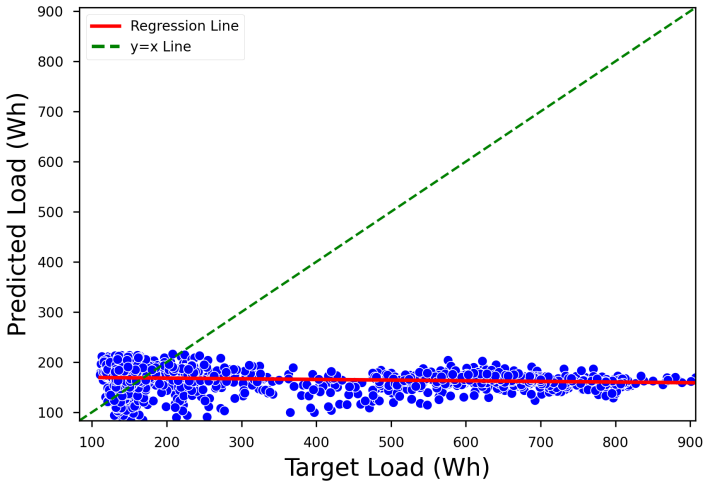

Figure 4.38: Scatter plot with regression line between actual and predicted load for VB B2 in CS 2.

- **Building B floor 3:** For VB B3, as previously discussed in subsection 4.2.1, it is expected to have worse performance than other models due to its high measurement errors. As anticipated, the model for this CS shows an increase in error across all metrics. However, there is a particularity in this case, as the MAPE shows a higher increase, about 81.55%, compared to an increase of 49.9% in the RMSE. This is the first time in this study that the MAPE shows a higher increase in error than the RMSE. Figure 4.39 illustrates the increased error compared to Figure 4.23, and the regression line in Figure 4.40 further confirms the poor performance of the model. In this scenario, it is difficult to determine which metric is more significant in telling the story of the comparison between the two CSs.

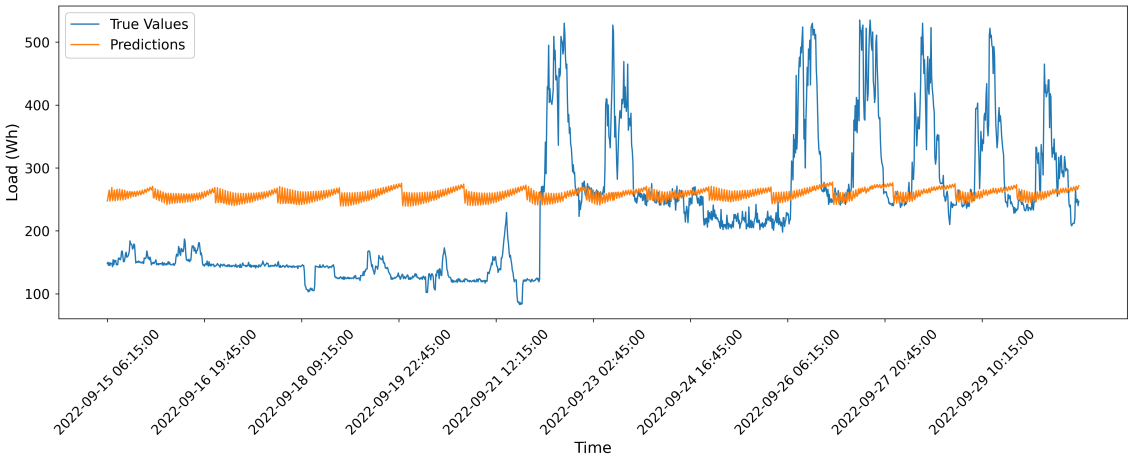

Figure 4.39: Actual load and predicted load for VB B3 in CS 2(last 1500 rows).

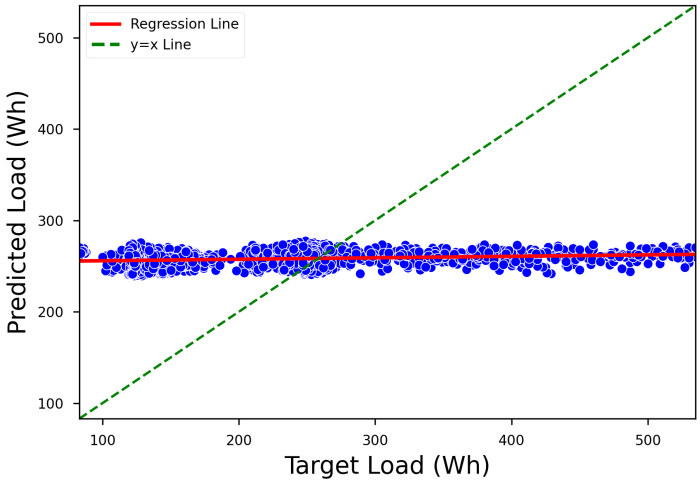

Figure 4.40: Scatter plot with regression line between actual and predicted load for VB B3 in CS 2.

- **Building B floor 4:** B4 is a specific case, since both the MAPE and RMSE presents a decrease in error in comparison to the case of high availability of data. MAPE decreased in 26.47% while RMSE descreased 1.44%. This is a non-expected result since the expected outcome of more data availability is an increase in performance. Figure 4.41 shows that the predictions are far from the true values, not corresponding to the 15.13% MAPE presented in Table 4.3, although this figure does not show the whole validation set, meaning that the model could have performed better for the other frame.

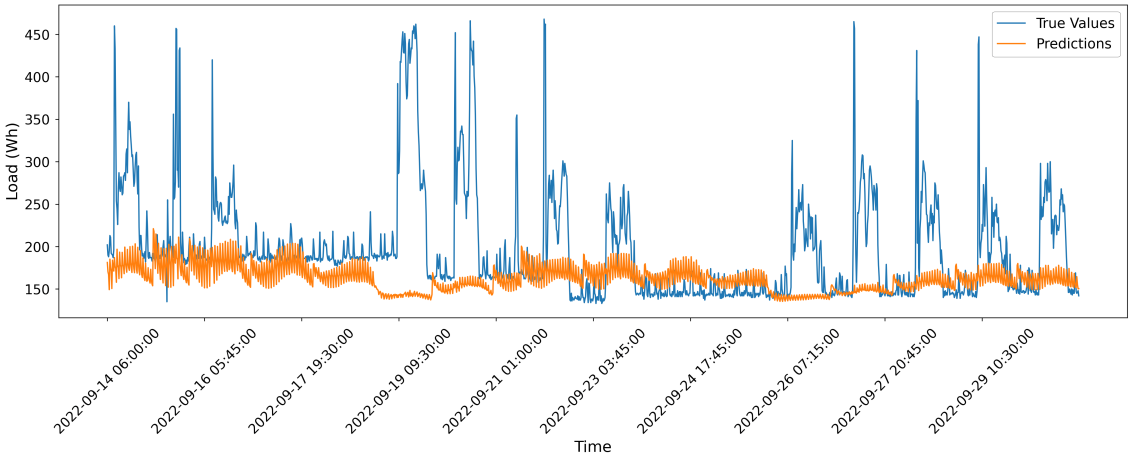

Figure 4.41: Actual load and predicted load for VB B4 (last 1500 rows).

To study that, instead of plotting the last 1500 rows of the test set (Figure 4.41), Figure 4.42 depicts the first 1500 rows, showing a little improvement in performance compared to Figure 4.41. Yet, this still does not explain the far above average error for this model. The

regression line in figure 4.43 shows a slope much closer to forty-five degrees than the other in this CS, but still does not corroborate the 15% error presented by the MAPE.

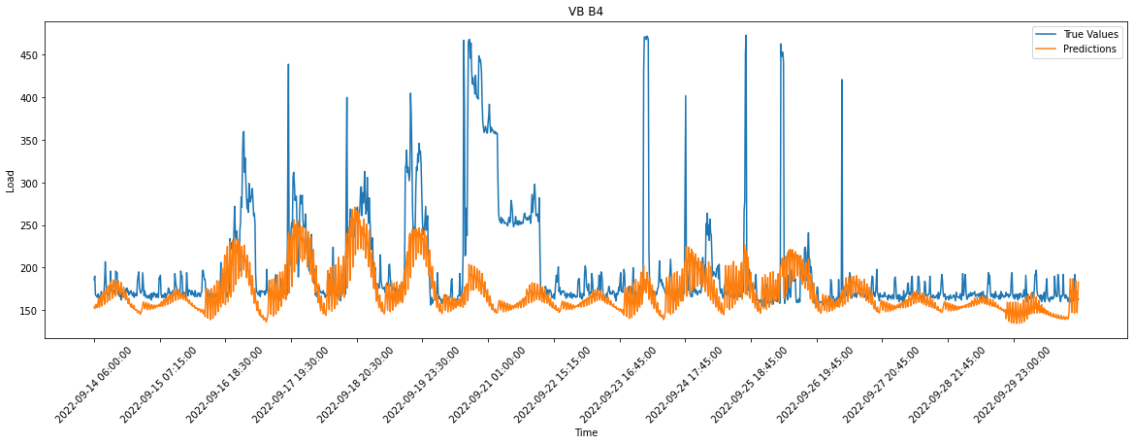

Figure 4.42: Actual load and predicted load for VB B4 (first 1500 rows).

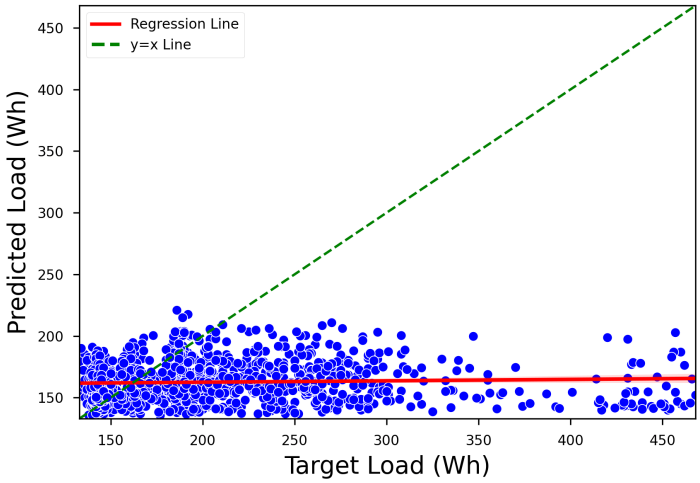

Figure 4.43: Scatter plot with regression line between actual and predicted load for VB B4 in CS 2.

In this case, the results align with the expected outcome, as there was an increase in error due to low data availability in almost all VBs, as Table 4.4 indicates, which presents an opportunity for the application of transfer learning. However, some of the VBs showed an increase in MAPE, such as A3 and B1, while one VB showed an increase in all error metrics. For A3 and B3, the increase in MAPE did not reflect in other evaluation tools used in this study, indicating that this metric is not able to properly assess the model performance. In the case of B4, both MAPE and RMSE increased, but this result is not concerning as this building has a complex consumption pattern due to measurement errors. In this way, the performance of the model is impacted by the randomness of the statistical distribution. While removing B4 from the study could have been justified, it

was kept to test transfer learning in data-challenging contexts. Another point to be noticed is the significant reduction of training time in all models in CS 2, going from an average of 79.16 to 14.48 seconds, meaning they need less computational effort due to limited data. Additionally, it is evident that CS 1 tends to perform better than CS 2, the mean angular coefficient of CS 1 is of 0.81 compared to 0.00849, while the mean R^2 for CS 1 is 0.73 against 0.0752 on CS 2, indicating that even if CS 2 performs better than CS 1 according to some metrics in specific cases, the results are not reliable when analysing graphically.

Table 4.4: Comparison of errors between CS 2 and CS 1 for all VBs.

VB	**Case Scenario 1**		**Case Scenario 2**		**Case 2/Case 1**	
	RMSE	**MAPE (%)**	**RMSE**	**MAPE (%)**	**RMSE**	**MAPE (%)**
A1	193.40	53.62	384.82	61.57	98.97%	14.82%
A2	140.59	25.87	304.47	49.38	116.57%	90.85%
A3	296.30	87.03	420.20	61.82	41.82%	-28.96%
A4	132.74	24.92	361.34	34.59	172.22%	38.77%
B1	233.71	52.01	331.95	39.98	42.04%	-23.13%
B2	90.92	25.02	217.84	41.65	139.59%	66.45%
B3	74.52	29.90	111.71	54.28	49.90%	81.55%
B4	62.13	20.57	61.24	15.13	-1.44%	-26.47%

4.3 Comparison of transfer learning approach to baseline models

In the previous section (Section 4.2), it has been presented the results of the experiments on applying transfer learning to the task of energy consumption forecasting in VBs. The performance of the models in two different CSs was evaluated, one with high availability of data and one with low availability of data. In this section, a third CS (Figure 3.6) is developed, comparing the results of the transfer learning approach with the results of the baseline models. Note that the baseline models were developed using traditional deep learning models without transfer learning, as described in chapter 3.

The comparison will be based on the evaluation metrics presented in the previous chapter, such as MAPE, RMSE, the regression angular coefficient, R^2 and training time. Additionally, it is analyzed the graphics generated by the models, which provide a visual representation of their performance. Our goal is to determine whether the transfer learning approach offers significant results over the baseline models, particularly in the case of low data availability. The analysis will be performed separately for each VB since they have varying levels of correlation with the base model B2 (as shown in Table 3.3) and are expected to have different outcomes. To start with, Table 4.5 shows the results of the metrics used to evaluate the model's performance for this CS.

Table 4.5: Evaluation metrics of the models for CS 3.

VB	mae	mse	rmse	mape (%)	time (s)	m	R^2
A1	179.14	53336.18	230.95	57.77	16.85	0.86413977	0.688583574
A2	98.60	21811.66	147.69	34.32	12.89	0.821721401	0.728215736
A3	163.88	53081.38	230.39	52.14	31.97	0.851125453	0.688551049
A4	115.06	30405.07	174.37	27.31	15.18	0.876891476	0.748942704
B1	145.41	33269.52	182.40	36.28	17.37	0.870978007	0.74941258
B2	62.14	9115.53	95.48	24.88	29.18	0.769839011	0.794294547
B3	108.32	17508.88	132.32	70.93	14.43	0.261461595	0.176170862
B4	35.10	3059.17	55.31	16.27	16.95	0.484999986	0.510738059

- **Building A floor 1:** The evaluation of the transfer learning approach for VB A1 is presented in Table 4.5. The MAPE for this model is found to be 57.77%, which is 6.17% lower than that of CS 2, and 7.74% greater than that of CS 1. This outcome aligns with the expected scenario, where the transfer learning model is expected to perform better than the one with scarce data (CS 2), and approach the performance of the model trained with full data availability (CS 1). The RMSE, when compared across the scenarios, shows an even better performance for the transfer learning approach, with a 39.99% improvement against CS 2, and a higher proximity from CS 1, with an RMSE that is 19.41% higher. The line plot in Figure 4.44, in comparison to Figure 4.27, shows a significant improvement in the model's capability to capture the consumption pattern, with a plot comparable to that of Figure 4.10, where both show a good forecasting performance.

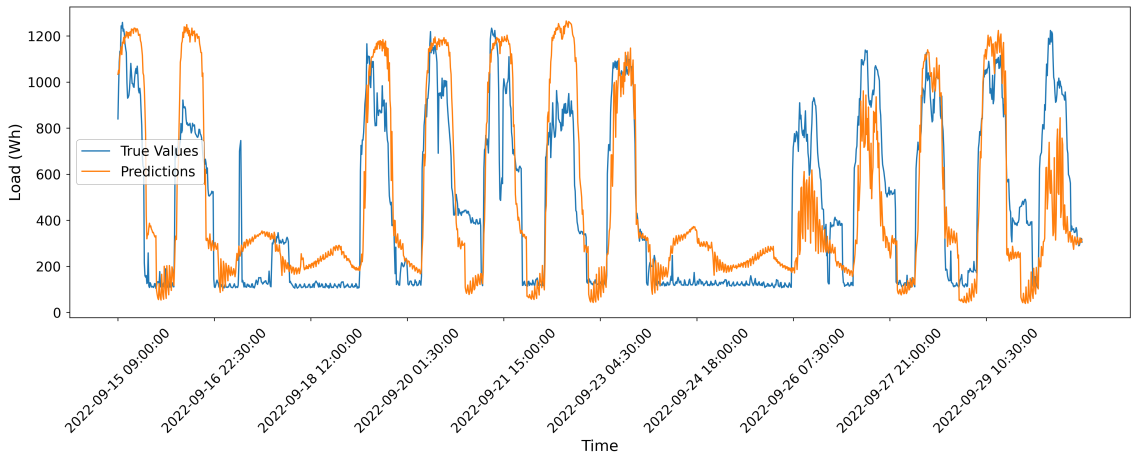

Figure 4.44: Actual load and predicted load for VB A1 in CS 3 (last 1500 rows).

Upon examination of the regression plot presented in Figure 4.45, it can be observed that the slope of the regression closely approaches forty-five degrees, indicating an improved understanding of the pattern of consumption in comparison to the regression plot in Figure 4.28, with the angular coefficient and R^2 much closer to one. However, a significant amount of bias is also evident from the distribution of dots, which is higher than that observed in

Figure 4.11. This difference in bias might account for the variation in accuracy between the two models.

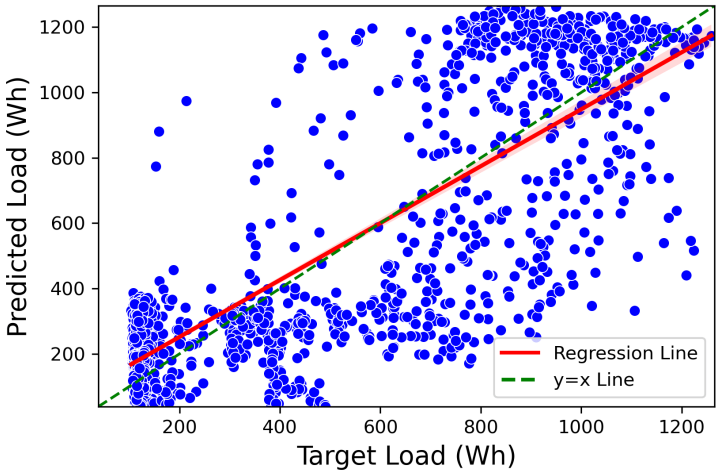

Figure 4.45: Scatter plot with regression line between actual and predicted load in CS 3 for VB A1.

Overall, the transfer learning approach for this VB presents remarkable outcomes, as it surpasses CS 2 and approaches CS 1, taking only 17.6% of the training time required for the latter (training time in CS 1 is 95.79 seconds against 16.85 seconds in CS 3), leading to reduced computational effort. These results were expected, given that the VB exhibits a high correlation with the base model, approximately 78.2%, which is the second-highest correlation among all models (Table 3.3).

- **Building A floor 2:** For VB A2, similar to A1, it is expected to achieve good results when applying transfer learning mostly because of the high correlation with the base building, as shown in Table 3.3. As presented in Table 4.5, this model showed a MAPE of 34.32%, indicating a 30.49% reduction when compared to CS 2 (table 4.3) and 32.66% higher than CS 1 (table 4.2). Upon investigating the RMSE, the results improved significantly, with a reduction of 51.49% against CS 2 and only 5.05% more error than CS 1. In Figure 4.46, a clear consumption pattern is visible, with a good match with the actual energy consumption. It is very different from what was seen in its counterpart in CS 2 (Figure 4.29) and much closer to the line plot in CS 1 (Figure 4.12).

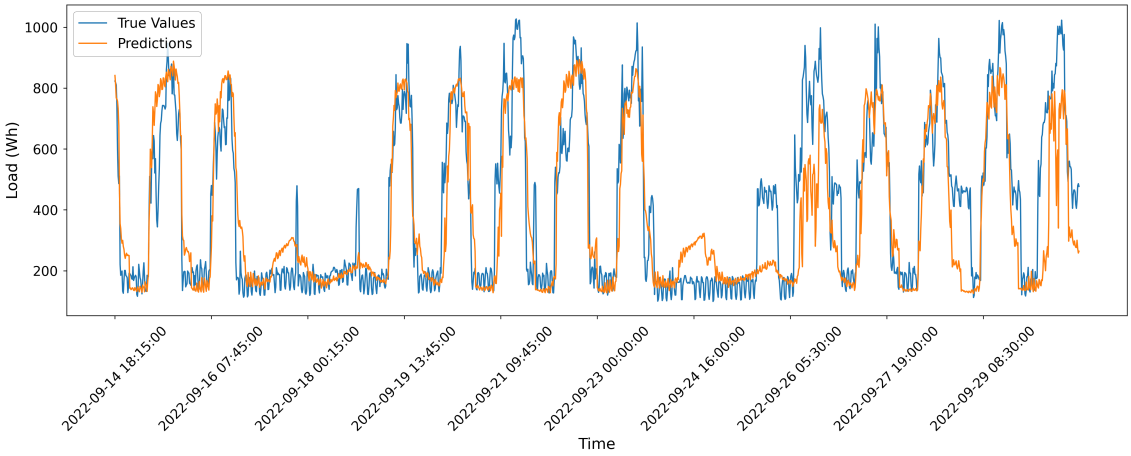

Figure 4.46: Actual load and predicted load for VB A2 in CS 3 (last 1500 rows).

Upon scrutinizing the scatter plot in Figure 4.47, a slope that is much closer to the forty-five degrees angle is noticeable, with angular coefficient of 0.82 and R^2 of 0.72, providing further evidence of the model's precision. In comparison to CS 1 (Figure 4.13), a slight increase in the dispersion of the dots is observed, indicating a greater bias that could explain the difference in accuracy between the two models. Once again, CS 3 demonstrated excellent results, surpassing CS 2 in all aspects, while requiring little computational effort compared to CS 1. The graphical analysis also validates the evaluated metrics, especially the RMSE.

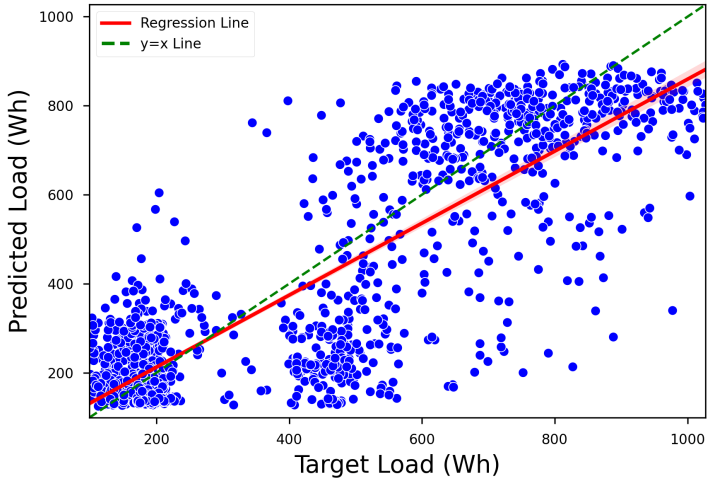

Figure 4.47: Scatter plot with regression line between actual and predicted load in CS 3 for VB A2.

- **Building A floor 3:** Great results are expected from this VB, as it has the highest correlation with the base VB (Table 3.3). As discussed in Section 4.2.2, for this VB, CS 1 presented a higher MAPE than CS 2, and CS 3 showed a 15.66% reduction in MAPE compared to CS

2 and a 40.09% reduction from CS 1. Regarding the RMSE, the context returned to the expected scenario, with CS 1 presenting a lower error than CS 2. However, it was unexpected that CS 3 remained the best scenario, with a 45.17% reduction in error compared to CS 2 and a 22.24% reduction compared to CS 1. Figure 4.48 demonstrates a great understanding of the pattern, and when compared with Figure 4.14, it is clear that although both predictions appear to be quite accurate, the one from CS 3 shows fewer data noisy in general.

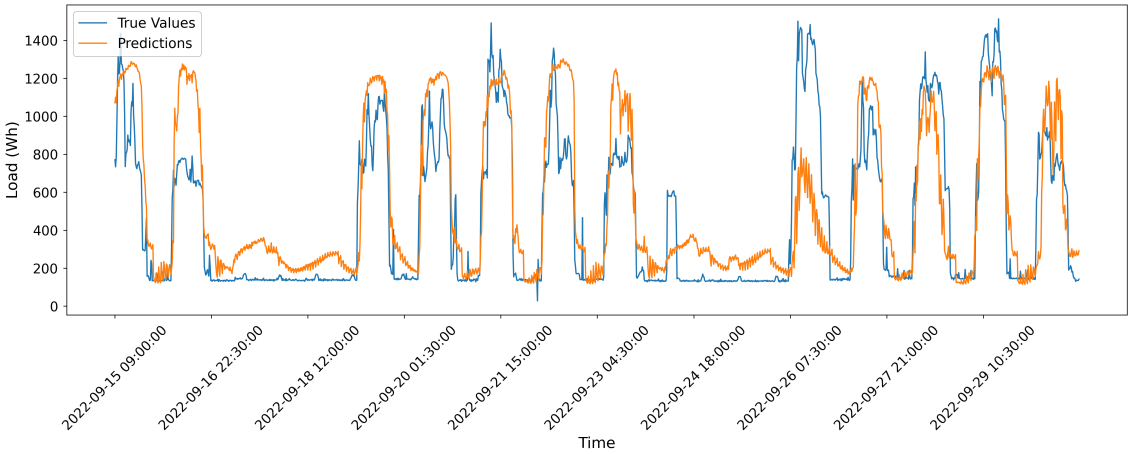

Figure 4.48: Actual load and predicted load for VB A3 in CS 3 (last 1500 rows).

The scatter plots for both CS 1 (Figure 4.14) and CS 3 (Figure 4.48) show a regression line with a slope close to the forty-five degrees mark, but the latter presents a higher dispersion of the dots. Overall, the transfer learning model for VB A3 performs extremely well, with significantly better results than CS 2 and slightly better than CS 1 (althought with worst angular coefficient and R^2 in the regression), while taking only 61% of the training time required for CS 1.

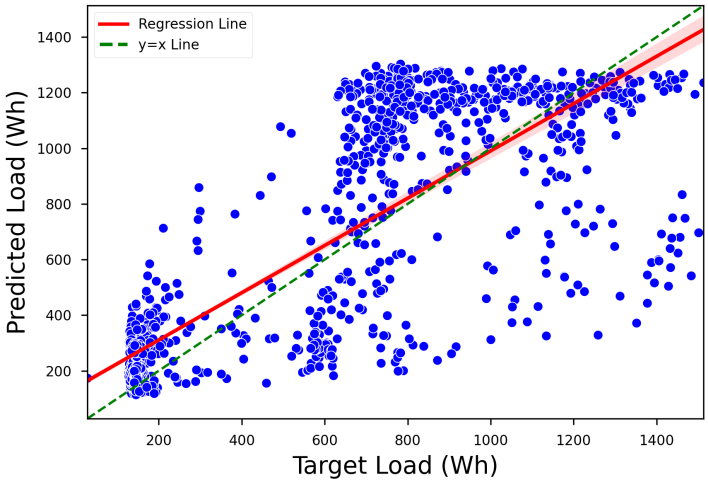

Figure 4.49: Scatter plot with regression line between actual and predicted load in CS 3 for VB A3.

- **Building A floor 4:** The results for VB A4 show an increase in error in all metrics when compared to CS 1 (Table 4.2) and a decrease in MAPE and RMSE when compared to CS 2 (Table 4.3). Specifically, the MAPE presents a reduction of 21.03% while the RMSE decreases by 51.74% against CS 2. On the other hand, compared to CS 1, the MAPE increases by 9.59% and the RMSE increases by 31.37%. Regarding the line plot in Figure 4.50, it presents a clear pattern of consumption and performs extremely well compared to CS 2, and slightly worse than CS 1, showing less noisy data but making more mistakes in the magnitude of consumption.

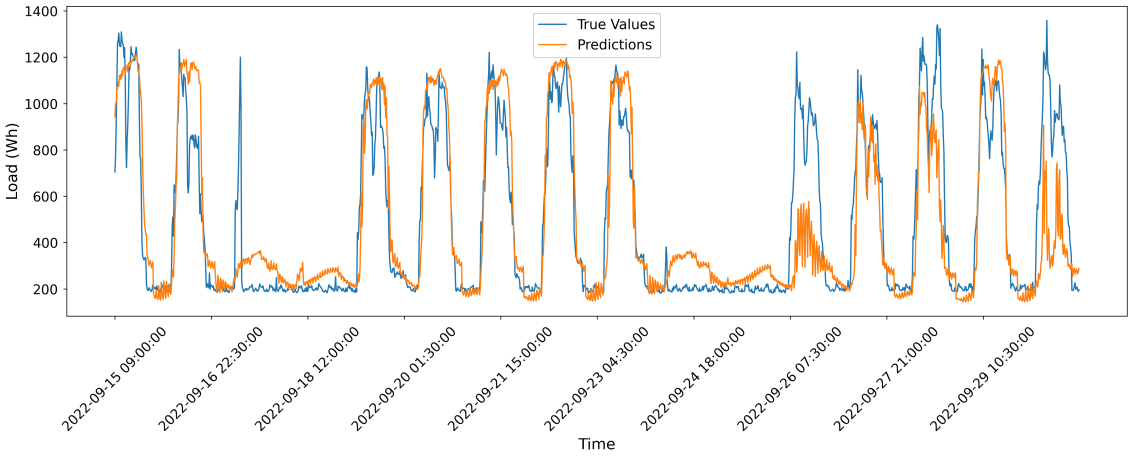

Figure 4.50: Actual load and predicted load for VB A4 in CS 3 (last 1500 rows).

Regarding the regression line, both CS 1 (Figure 4.17) and CS 3 (Figure 4.51) have a good slope, but CS 1 has considerably less bias and the dots are way less dispersed, indicating a

higher accuracy of the model. This is also evidenced by the angular coefficient and the R^2 of the regression line, which are closer to one in CS 1. Overall, CS 3 still shows a significant improvement compared to CS 2 and achieves relatively good performance given the low data availability and a training time of about 19% of the training time for CS 1 (Table 4.5 and Table 4.2).

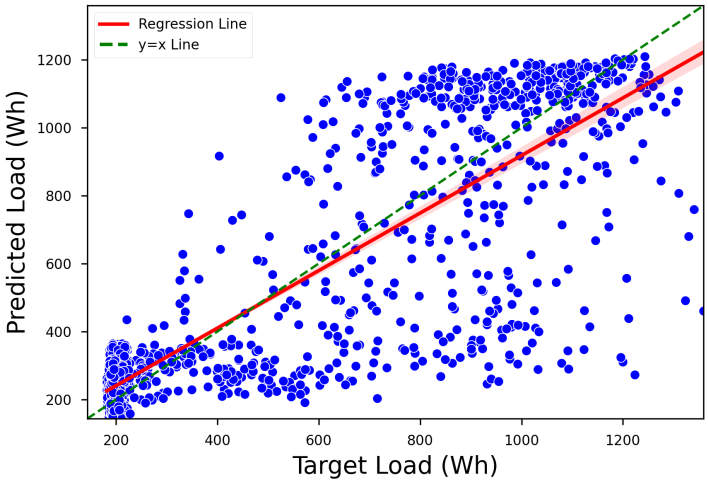

Figure 4.51: Scatter plot with regression line between actual and predicted load in CS 3 for VB A4.

- **Building B floor 1:** For VB B1, similar to A3, the transfer learning model presented CS 3 as the best performing scenario. The model showed a MAPE reduction of 9.25% in comparison to CS 2 and a reduction of 30.24% compared to CS 1. Regarding the RMSE, it shows a decrease of 45.05% in comparison to CS 2 and 21.95% compared to CS 1 (Table 4.5). The line plot in Figure 4.52 presents a much better understanding of the pattern than in CS 2 and shows less noisy data than CS 1, performing better in the magnitude of the peaks and valleys.

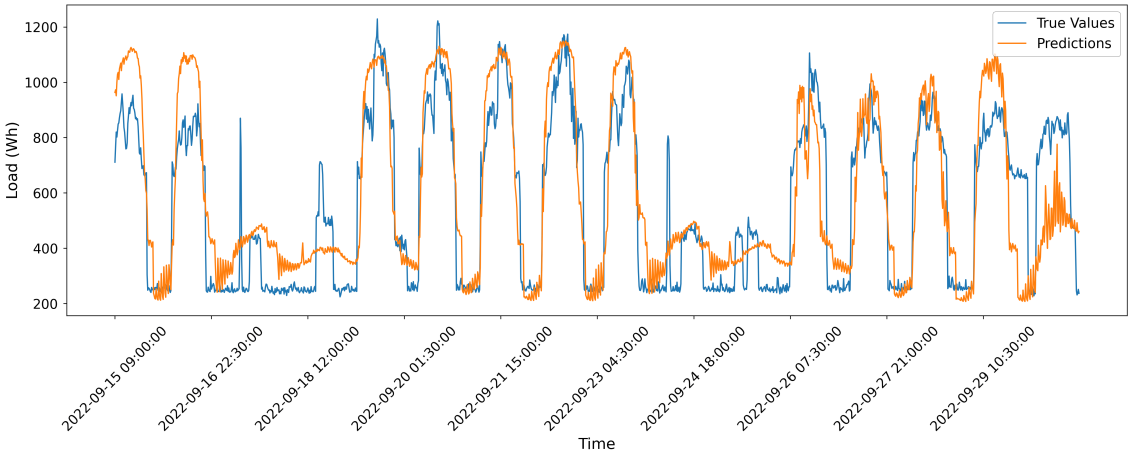

Figure 4.52: Actual load and predicted load for VB B1 in CS 3 (last 1500 rows).

The regression lines in Figure 4.53 are quite similar to CS 1, making it hard to visually spot the differences, with angular coefficient and R^2 very close. Once again, the transfer learning model showed excellent results, performing above expectations with limited resources.

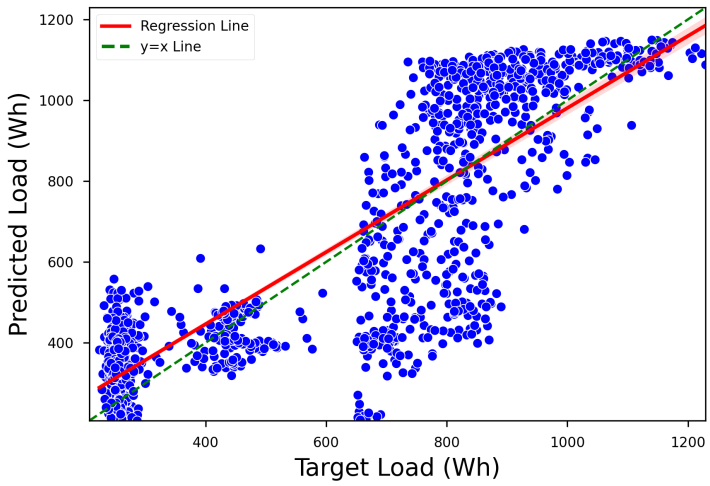

Figure 4.53: Scatter plot with regression line between actual and predicted load in CS 3 for VB B1.

- **Building B floor 2:** Upon analyzing Building B floor 2, it was observed that this VB was included in the study to demonstrate the best possible scenario for the application of transfer learning. This is the scenario where the target and source are as similar as possible. The weights and parameters from the pre-trained model in CS 1 were transferred to the model in CS 3, with the expected outcome being that they would have very similar results, and of course, a significant improvement from CS 3 in comparison to CS 2. The results showed a 40.26% improvement in MAPE and a 56.17% improvement in RMSE.

In terms of the comparison between CSs 3 and 1, there was a small improvement of 0.57% in MAPE in favor of CS 3, while RMSE showed 5% more error in CS 3. The graphical analysis confirmed the evaluation of metrics, where a significant improvement was observed in Figure 4.54 compared to Figure 4.37 and it was very similar to Figure 4.20. The CS 1 shows better accuracy in the magnitude of consumption during peak hours but also predicted some spikes during weekends that were not present in CS 3.

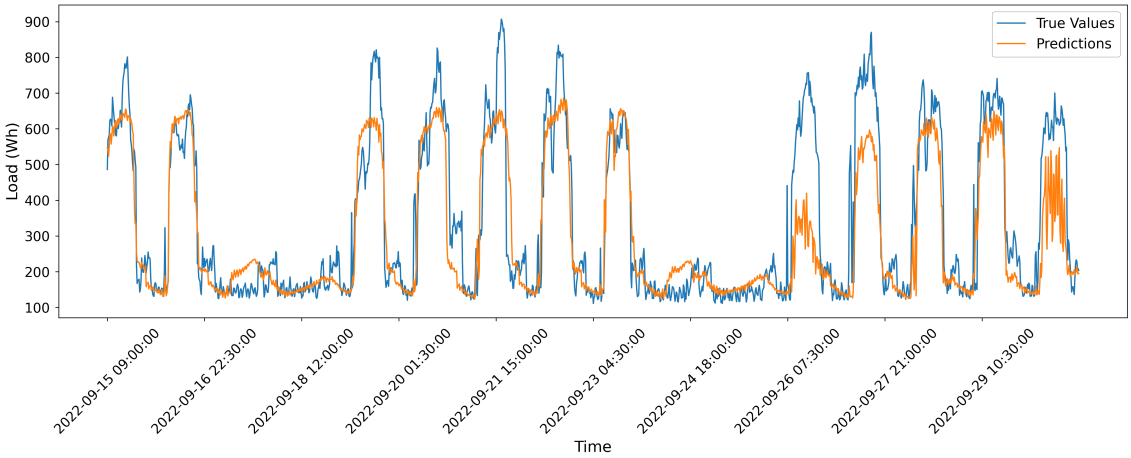

Figure 4.54: Actual load and predicted load for VB B2 in CS 3 (last 1500 rows).

Upon inspecting the scatter plot in Figure 4.55, it was observed that the slope is excellent and very similar to CS 1 (Figure 4.21). However, there is a higher density of dots at the end of the graphic for CS 1, indicating less bias. Overall, the results for this CS line up well with expectations, as CS 3 shows significant improvement from CS 2 and is very close to CS 1, all while requiring much less computational effort.

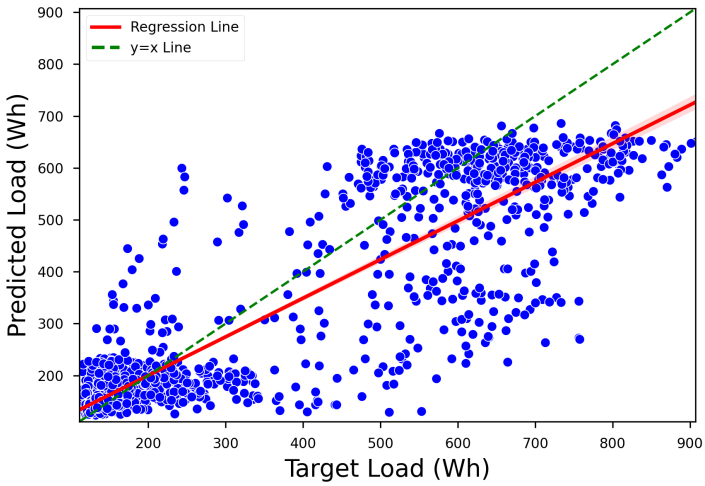

Figure 4.55: Scatter plot with regression line between actual and predicted load in CS 3 for VB B2.

- **Building B floor 3:** The analysis of VB B3 reveals that this VB presented the most challenging task in the study, given its low correlation with the base VB, only 41.69% (Table 3.3). This low correlation is reflected in the results, showing a clear case of negative transfer [11], where the transferred knowledge results in a downgrade of the model. When comparing to CS 2, the model showed a 30.68% higher MAPE and 18.46% more RMSE. Compared to CS 1, the MAPE increased by 137.25% and the RMSE by 77.56%. Graphical analysis shows that the model understands some pattern, which does not happen in CS 2, but this pattern recognized by the model does not fit well with the unusual behavior of the load in the first half of the load curve of VB B3 (Figure 4.8), as seen in Figure 4.56.

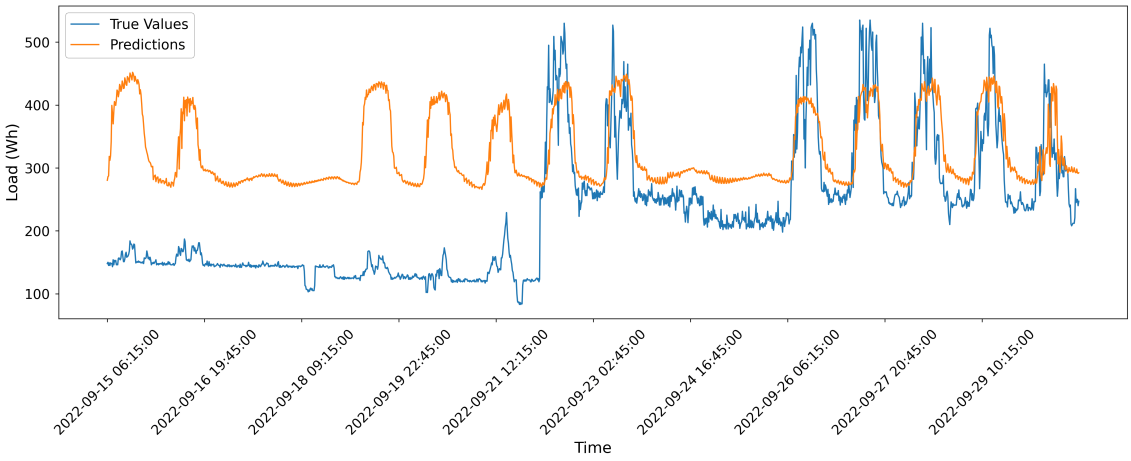

Figure 4.56: Actual load and predicted load for VB B3 in CS 3 (last 1500 rows).

The scatter plot for CS 3 in Figure 4.57 indicates an improvement compared to CS 2 (Figure 4.40), with the slope of the regression much closer to forty-five degrees and the angullar coefficient and R^2 much higher than CS 2. For this VB, the analysis indicates that it performs much worse than CS 1. The graphical analysis indicates a slight improvement in comparison to CS 2, but this improvement is not corroborated by the evaluation metrics (MAPE and RMSE). The pattern displayed by the forecasting curve for this scenario suggests that if the load had a more reliable pattern, not degraded by measurement errors, an improvement from this model to the CS 2 model would be expected.

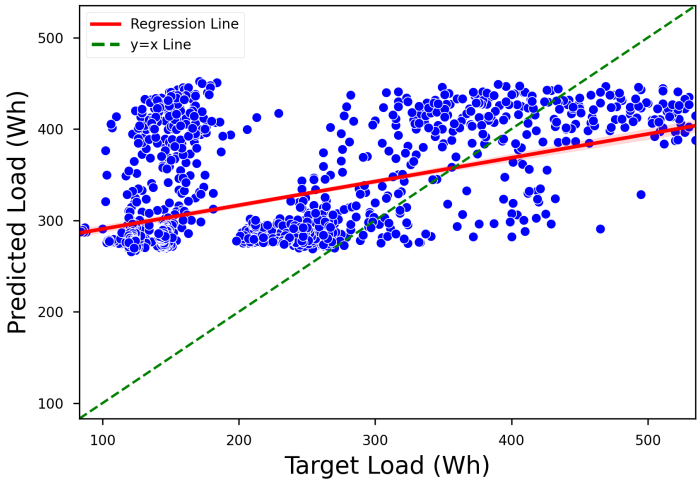

Figure 4.57: Scatter plot with regression line between actual and predicted load in CS 3 for VB B3.

- **Building B floor 4:** The evaluation of floor 4 in Building B yields some of the most unanticipated findings. In comparison to CS 1, there is a 20.93% reduction in and a 10.98% reduction in RMSE, thus surpassing CS 1 based on these performance indicators. When juxtaposed with CS 2, the MAPE is 7.54% higher. However, the RMSE reveals a 9.68% decrease in error. These metrics alone would suggest that CS 3 is the most effective scenario, followed by CS 2, with CS 1 being the least desirable. Upon examining the line plot of CS 3 in Figure 4.58, it becomes evident that the model recognizes a pattern similar to that in CS 1 (Figure 4.25), albeit with reduced data noise. Furthermore, the line plot demonstrates a marked improvement over CS 2 (Figure 4.41), which fails to exhibit any discernible pattern.

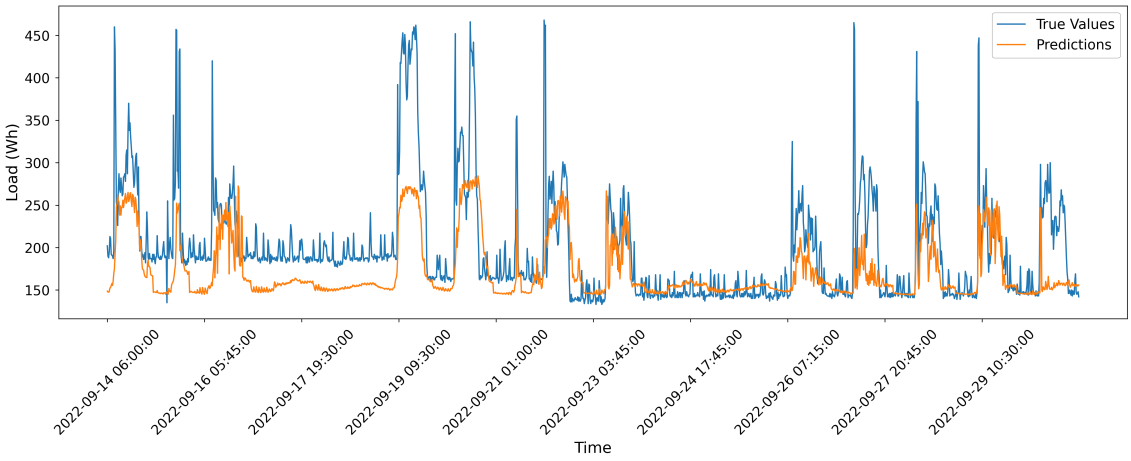

Figure 4.58: Actual load and predicted load for VB B4 in CS 3 (last 1500 rows).

The scatter plot depicted in Figure 4.59 exhibits performance strikingly similar to that of CS

1 (Figure 4.26), rendering it visually challenging to discern any differences in effectiveness between them. In contrast to CS 2, the regression line displays a significantly improved slope, and the data points are less dispersed. Upon scrutinizing all metrics and visual representations for this scenario, it becomes apparent that, in this instance, applying transfer learning not only enhances performance compared to CS 2 but also outperforms CS 1 (in MAPE, RMSE and R^2). This is despite the low PCC between the base VB and this VB (Table 3.3).

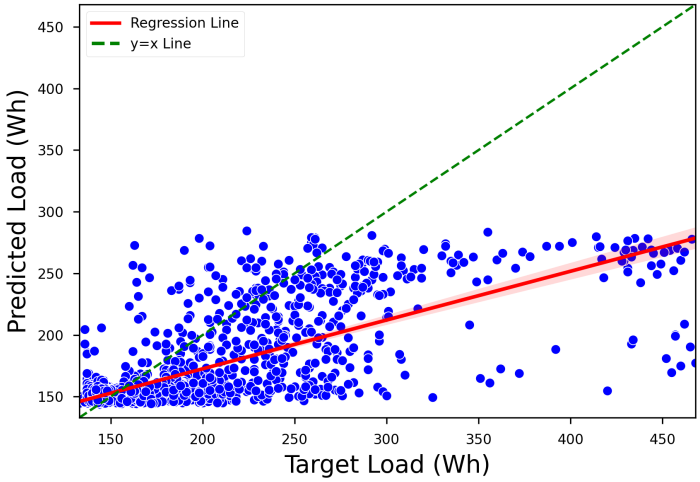

Figure 4.59: Scatter plot with regression line between actual and predicted load in CS 3 for VB B4.

4.4 Discussion of results

This section presents the findings and conclusions derived from the application of transfer learning for load forecasting in buildings. The study's primary objective was to investigate the effectiveness of employing transfer learning techniques to improve the accuracy and reliability of load forecasting models. The results and analysis were conducted based on multiple scenarios, comparing the performance of different models and techniques.

The results demonstrated that transfer learning could significantly enhance load forecasting performance under several building contexts. In most of the cases, due to the high correlation between source and target (Table 3.3), the transfer learning approach (CS 3) outperformed the baseline model with a scarcity of data (CS 2), even surpassing, for some cases, the baseline model with a richness of data (CS 1). The improvement was evaluated across different performance metrics, including MAPE, RMSE, and the manual observation of graphics such as line and scatter plots.

The findings also revealed that the performance of the transfer learning models varied depending on the specific scenarios and the degree of similarity between the base VB and the other VBs.

In some cases, transfer learning models exhibited remarkable improvement over the traditional baseline models, while in other instances, the performance gains were more modest or even non existent, configuring a case of negative transfer. Figure 4.60 and Figure 4.61 illustrates this results.

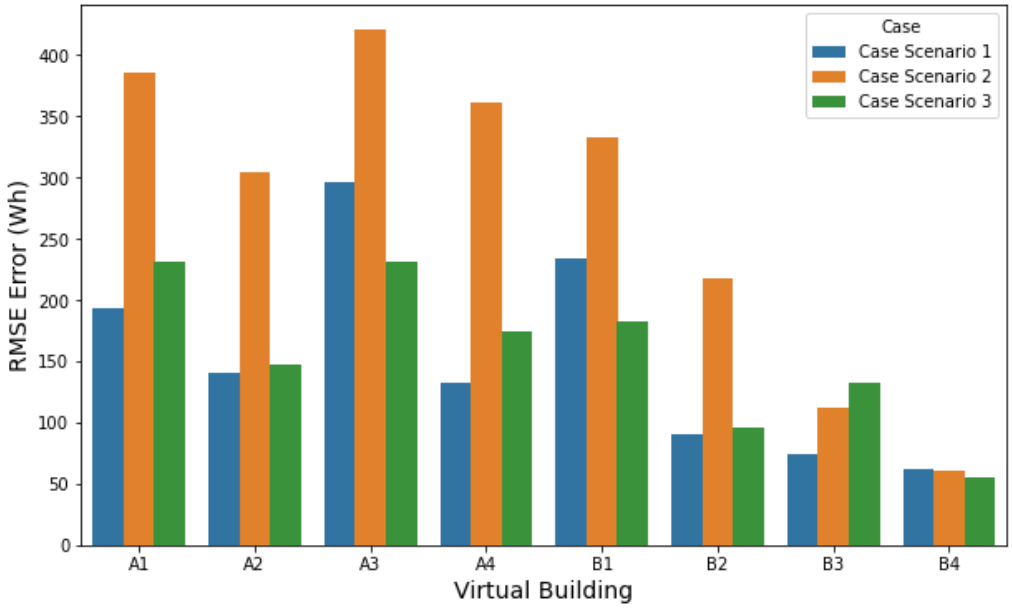

Figure 4.60: RMSE on all CSs (%).

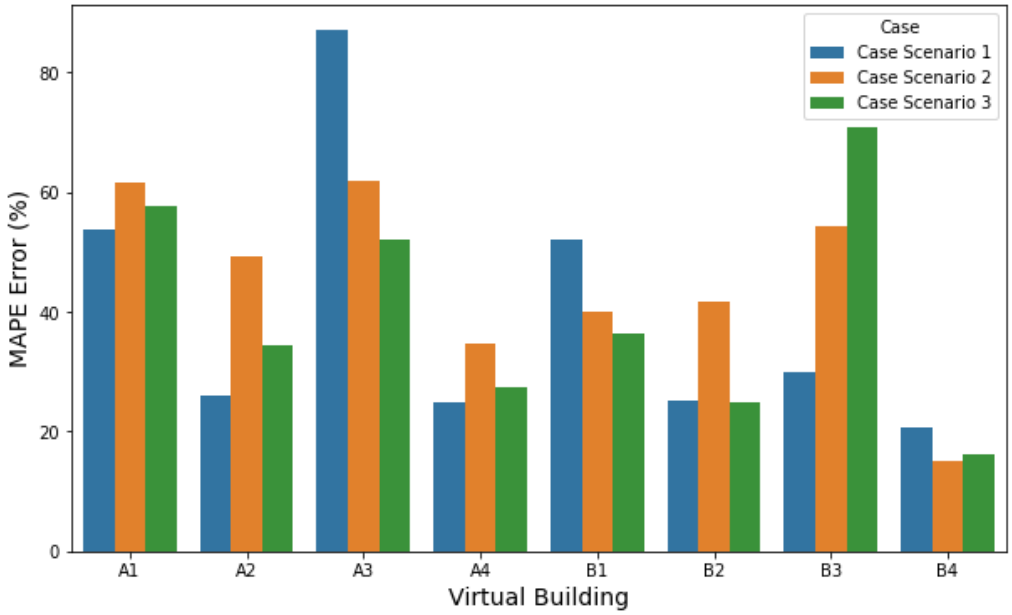

Figure 4.61: MAPE on all CSs (%).

Considering the RMSE metric, seven out of the eight VBs exhibited improved performance, with an average error reduction of 42.76% compared to CS 2. The only VB that demonstrated negative transfer was B3, with an increase of the RMSE of about 18.46%. B3 and B4 represent

specific cases where the application of transfer learning may not be ideal. The target building B3 has a load curve that is substantially different from the source building B2 for most of the domain (Figure 4.8 and Figure 4.6), which implies that any knowledge transferred from the base case may lead to prediction errors.

A similar issue arises for B4, which has an unreliable load curve (Figure 4.9) due to measurement errors. Consequently, even the model for CS 1 encounters prediction challenges. However, for B4, the most accurate model in terms of RMSE employed the transfer learning approach. In comparison to CS 1, only three buildings demonstrated superior performance: B4 (as previously mentioned), A3, and B1, with error reductions of 10.98%, 22.24%, and 21.95%, respectively. These results are summarized in Table 4.6.

Table 4.6: Comparison between CS 3 and baseline models using RMSE.

VB	**CS 3/CS 2**	**CS 3/CS 1**
A1	39.99%	-19.41%
A2	51.49%	-5.05%
A3	45.17%	22.24%
A4	51.74%	-31.37%
B1	45.05%	21.95%
B2	56.17%	-5.01%
B3	-18.46%	-77.56%
B4	9.68%	10.98%

In what concerns to the MAPE metric, the results are more modest, with an average error reduction of 20.48% compared to CS 2 (accounting only for positive transfer cases). B3 was the sole building exhibiting negative transfer, with the transfer learning application yielding 30.68% more error than CS 2. Despite the error observed in Building 4 (B4) for Case Study (CS) 3 being higher than that of CS 2, this should not be construed as an occurrence of negative transfer. This is due to the fact that B4's performance exceeded that of the scenario from which it assimilated knowledge, specifically, CS 1. Indeed, there was a reduction of 20.93% in MAPE, as can be substantiated in Table 4.7.

Despite the models for B3 and B4 are not ideal, the transfer learning results would likely be positive if the test data utilized for metric calculation were replaced with actual building consumption data. The data for these buildings (Figure 4.8 and Figure 4.9) exhibit abnormal behavior for load consumption in a commercial building, which could explain the observed discrepancies in performance.

Table 4.7: Comparison between CS 3 and baseline models using MAPE.

VB	CS 3/CS 2	CS 3/CS 1
A1	6.17%	-7.74%
A2	30.49%	-32.66%
A3	15.66%	40.09%
A4	21.03%	-9.59%
B1	9.25%	30.24%
B2	40.26%	0.57%
B3	-30.68%	-137.25%
B4	-7.54%	20.93%

According to the graphical analysis conducted in section 4.3, the transfer learning approach resulted in improvements compared to CS 2 for all VBs, except for B3 and B4. However, even in these cases, the knowledge transfer provided a pattern for the model that, if not for the measurement errors distorting the load curve, would likely have led to performance enhancement. This statement is clear when comparing the slope of the scatter plot from CS 3 (Figure 4.57) to CS 2 (Figure 4.40), where the first one presents a slope much closer to forty-five degrees. the mean angular coefficient for CS 2 was of 0.00849, while for CS 3 was of 0.72, stunningly better. Also, the R^2 averaged 0.0752 on CS 2 against 0.63 on CS 3. Those metrics indicates that despite the error results, the model's pattern understanding was much better in CS 3 compared to CS 2. It is important to note that the graphical analysis aligned more closely with the results obtained from the RMSE metric than those from the MAPE metric.

The analysis conducted in this study indicates that transfer learning can be an effective approach for enhancing load forecasting performance in buildings. By leveraging the knowledge acquired from a base building, and supplying only a month of data, the transfer learning models can adapt and generalize to new, yet related, VB contexts, leading to improved forecasting accuracy and reliability.

However, the extent of improvement in forecasting performance depends on several factors: (*i*) the degree of similarity between the base and VBs, (*ii*) the quality and quantity of available data both in source and target models, (*iii*) the choice of transfer learning techniques, (*iv*) geographical and weather conditions. Therefore, it is essential that practitioners carefully consider these factors when implementing transfer learning for building load forecasting.

This study has demonstrated the potential of transfer learning as a promising technique for improving load forecasting in buildings. The findings contribute to the growing body of knowledge on the application of machine learning in energy forecasting and support the adoption of transfer learning in building energy management systems to optimize energy consumption and reduce operating costs.

Chapter 5

Conclusion and Future Work

This thesis presents a comprehensive literature review on data-driven methods for load forecasting, with particular focus on transfer learning applied to commercial buildings. To evaluate the importance of the proposed methodology, three case scenarios were assessed. The first is a scenario of full data availability, with more then two years of collected data at disposal. The second is a case scenario simulating a situation with low data availability, containing only a month of measured data. Lastly, the third is a hybrid scenario, where low quantity of data is available, but the models have access to the weights of a pre-trained neural network model with high data availability to gather knowledge from.

The results effectively demonstrate the advantages of transfer learning approach for load forecasting within the context of virtual buildings employed in this study. By using pre-trained models and fine-tuning them with limited data, mean forecasting accuracy improvement was observed in 42.76% in RMSE and 20.48% in MAPE compared to the CS with scarce data. This achievement bears significance, as it addresses one of the foremost challenges encountered by data-driven models, i.e., data scarcity. Moreover, the enhanced generalization and scalability of data-driven models can facilitate building manager in implementing techniques for reducing energy consumption, thereby contributing to decarbonization.

Another contribution of transfer learning methods is the potential for reducing computational time from load prediction. It was observed a 72.62% reduction in computational time whilst using transfer learning when compared to the traditional approach on CS 1. Such accomplishments may be of interest for building operators to update existing models upon the acquisition of new data by transferring knowledge between models. The proposed methodology can be particularly beneficial for buildings with newly installed energy metering and desire to implement their own load forecasting model but lack data for traditional architectures. Furthermore, the results presented in this thesis also indicate that transfer learning can contribute to improve patterns recognition within dataset, even under usually difficult scenarios.

Finally, the implementation of transfer learning techniques applied to data-driven models for load forecasting in commercial buildings provide insights and lay further groundwork for research and development in this field of study. The results observed hold significant implications for smart

cities, as they underscore the potential of transfer learning in improving energy management and reducing environmental impact of the building sector.

Nevertheless, to successfully employ the proposed model presented in this thesis, it is of utmost importance to consider its limitations. The implementation of models considered electricity consumption in commercial buildings. Therefore, it may not be as effective when employed to different categories of buildings such as residential and industrial constructions. Furthermore, in this study it was employed the transfer learning technique of homogeneous inductive learning, which was the best fit for the scope of this project. There are other frameworks, as discussed in chapter 2 that may address potential reader's needs more properly. Although it was observed improvement in pattern recognition, the employed method did not explore model interpretability, which is an aspect of increasing interest but without a solid literature available yet.

Considering the limitations and opportunities identified during the development of this thesis, it is presented below potential directions for future work that can be explored:

1. **Diverse Building Types and Domains:**investigate the applicability of transfer learning for load forecasting in diverse building types and domains, such as residential or industrial environments;

2. **Multiple Datasets and Contexts:** assess the generalizability of the transfer learning approach to multiple contexts, weathers and geographical locations to determine its broader utility in different scenarios;

3. **Alternative Transfer Learning Approaches:** explore alternative transfer learning approaches and compare their performance in load forecasting for building energy management;

4. **Computational Effort Implications:** evaluate the computational effort implications of transfer learning in more complex environments, involving larger datasets or more features, to better understand its practical limitations and possible benefits;

5. **Model Interpretability:** examine the interpretability of models developed using transfer learning and explore techniques to improve their explainability in real-world applications;

6. **Additional Data Sources:** Develop methods to enhance the transfer learning approach by incorporating additional data sources, such as building automation systems or IoT devices, to provide a richer context for forecasting and energy management.

7. **Transfer learning integrated in a smart management environment:** Directly integrate the transfer learning forecast enhanced model to a management system to evaluate the enhancement in control of the energy system provided by the transfer learning approach.

Pursuing these directions in future work will help address the limitations of the current study and expand our understanding of the potential applications and benefits of transfer learning in the domain of building energy management and load forecasting.

Funding

This research received partial support from the Norte Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, through the European Regional Development Fund (ERDF), within the DECARBONIZE project under agreement NORTE-01-0145-FEDER-000065 and by the Scientific Employment Stimulus Programme from the Fundação para a Ciência e a Tecnologia (FCT) under the agreement 2021.01353.CEECIND.

Appendix A

Appendix

Highlights

Overcoming Data Scarcity in Load Forecasting: A Transfer Learning Approach for Commercial Buildings

Felipe Dantas do Carmo, Wellington Fonseca, Tiago André Soares

- Development of a case study involving virtual buildings in the INESC TEC facilities;
- Development of multiple load forecasting models using recurrent and LSTM layers;
- Development of transfer learning based load forecasting models with data scarcity;

Overcoming Data Scarcity in Load Forecasting: A Transfer Learning Approach for Commercial Buildings

Felipe Dantas do Carmo[a,1], Wellington Fonseca[a], Tiago André Soares[a,1]

[a] *Center for Power and Energy Systems, Institute for Systems and Computer Engineering, Technology and Science (INESC TEC), R. Dr. Roberto Frias, Porto, 4200-465, Portugal*
[b] *Faculty of Engineering of University of Porto (FEUP), R. Dr. Roberto Frias, Porto, 4200-465, Portugal*

Abstract

Load forecasting is an asset for sustainable building energy management, as accurate predictions enable efficient energy consumption and contribute to decarbonisation efforts. However, data-driven models are often limited by dataset length and quality. This study investigates the effectiveness of transfer learning (TL) for load forecasting in buildings, with the aim of addressing data scarcity issues and improving forecasting accuracy. The case study consists in a group of eight virtual buildings (VB) located in Porto, Portugal. One VB serves as pre-trained base model to transfer knowledge to the remaining VBs, which are analysed in varying degrees of data availability. Our findings indicate that TL can significantly reduce training time, for up to 87%, while maintaining accuracy levels comparable to those of models trained with full dataset, and exhibiting superior performance when compared to models trained with scarce data, with average RMSE reduction of 42.76%. The study also demonstrates the application of TL under unstable data patterns.

Keywords: Transfer Learning, Load forecasting, Machine Learning, Neural Networks, Data scarcity
PACS: 0000, 1111
2000 MSC: 0000, 1111

Nomenclature

ADAM	Adaptive moment estimation
ANN	Artificial Neural Network
BPNN	Back propagation neural network
CNN	Convolutional neural network
CS	Case Scenario
DNN	Deep neural network
IQR	Interquartile range
LSTM	Long short term memory
MAE	Mean absolute error
MAPE	Mean absolute percentage error
MSE	Mean squared error
$P_S(X)$	Source probability distribution
$P_T(X)$	Target probability distribution
PCC	Pearsson correlation coefficient
ReLu	Rectified linear unit
RMSE	Root mean squared error
RNN	Recurrent Neural Network
seq2seq	sequence-to-sequence
STLF	Short-term load forecasting
$\mathcal{D}_S$	Source domain
$\mathcal{D}_T$	Source domain
Tanh	Hiperbolic tangent
TL	Transfer Learning
VB	Virtual Building

---

1. Introduction

1.1. Background and motivation

The urgency of mitigating greenhouse gas emissions is escalating the significance of energy-efficient buildings worldwide. Considering that approxi-

mately one-third of these emissions come from buildings, energy efficiency in this sector has become an indispensable focal point (Pinto et al., 2022). Load forecasting , a prominent factor in these efforts, is very important in facilitating energy management strategies that are both effective and sustainable.

The advent of data-driven methods, particularly those employing deep learning techniques, has become the foundation of load forecasting (Tan et al., 2018). Data-driven models hasve the capability to understand complex, non-linear patterns within load data, thus potentially improving forecasting accuracy. However, such models need great amounts of data for effective training. The latter is a prerequisite that is often unfulfilled in the context of individual buildings.

Despite the remarkable advances in data-driven for load forecasting, the bottleneck that data scarcity represents remains a great barrier for the widespread adoption of smart buildings (Alanne and Sierla, 2022).

In this light, transfer learning (TL) emerges as a promising solution. TL consists in a machine learning technique that harnesses the knowledge from one domain to enhance learning in another (Figure 1), TL offers a potential pathway to overcome data scarcity (Pan and Yang, 2010). Within the scope of load forecasting, TL can employ pre-trained models on buildings to which available dataset require improvement to develop reliable forecasting models (Ahn and Kim, 2022).

This paper concentrate its efforts into the application and efficacy of TL for load forecasting in commercial buildings. The study employs a pre-trained model as its foundation and fine-tunes it on a smaller dataset. In this sense, the research aims to evaluate whether TL can retain high forecasting accuracy meanwhile mitigating the requirement for copious training data.

1.2. Literature review

It is essential to comprehend the research gaps in the fields of load forecasting and TL, as well as to grasp the context and background of the study. The overview is primarily focused on the concept of TL, its applications, advantages, and limitations, with a special emphasis on its use in load forecasting.

Cai et al. (2020) introduced a two-layer transfer learning-based short-term load forecasting (STLF) model designed to enhance load prediction accuracy in the target zone. This innovative model utilizes a latent parameter in the inner layer to encapsulate the latent factors influencing electricity consumption variances across different zones. Concurrently, the outer layer utilizes

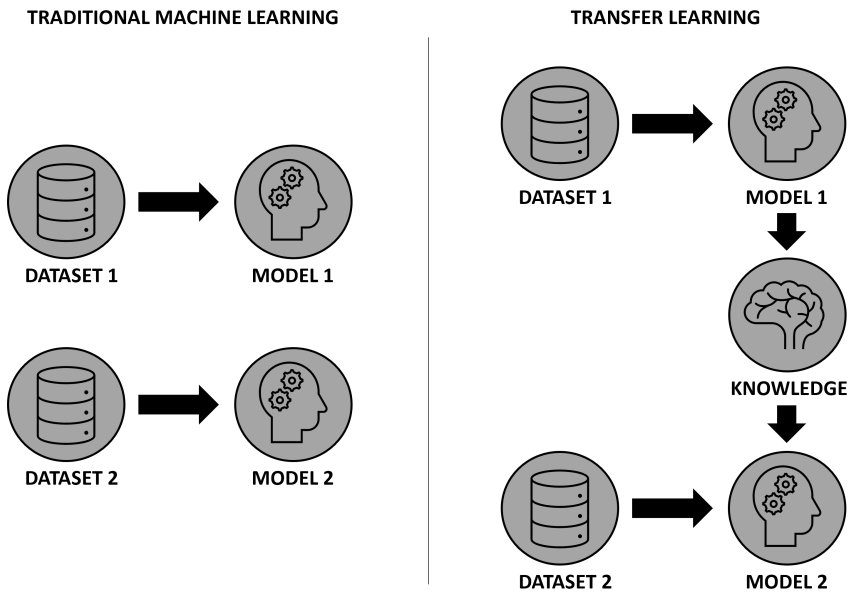

Figure 1: Transfer learning graphical description.

an iterative algorithm to optimize the weights. The results underscored the efficacy of this model as it consistently improved forecasting accuracy in comparison to conventional STLF algorithms.

Gao et al. (2020) tackled the challenge of accurate energy consumption prediction in buildings with limited historical data by proposing two deep learning models: a sequence-to-sequence (seq2seq) model and a two-dimensional convolutional neural network (2D CNN) with an attention layer. Incorporated into a TL framework, these models aimed to enhance prediction accuracy for a target building with scarce data. The case study involving three commercial buildings showed that the proposed models outperformed a long short-term memory (LSTM) network, improving forecast accuracy in mean absolute percentage error (MAPE) by 19.69% and 20.54% on average, respectively.

In their research, Li et al. (2021) utilized the open source Building Genome Project's dataset, which encompasses around 400 non-residential buildings, to establish a TL-based Artificial Neural Network (ANN) model for one-hour ahead building energy prediction. The main objective was to enhance prediction accuracy for a target building with limited available data. To evalu-

ate the model's performance, a three-layer Backpropagation Neural Network (BPNN) model was developed, tested across various source and target data samples and different source-target building pairings. The study concluded that the scarcity of available data inversely corresponds to the accuracy improvement achievable through TL.

In their work, Pinto et al. (2022) presented a detailed survey concentrating on the utilization of TL within the realm of smart buildings. Their research distinguished four central domains for the application of this technique: the prediction of building load, the detection of occupancy and recognition of activity, the modeling of building dynamics, and the control of energy systems. Out of these, building load prediction emerged as the most prevalent application. Nevertheless, the authors highlighted that a limited number of the reviewed studies have seen real-world deployment, thereby presenting potential opportunity for future research.

Ahn and Kim (2022) utilized TL in the context of building power consumption prediction, leveraging a simulated dataset. They engineered a Long Short-Term Memory (LSTM) model that incorporates TL, where the model was trained utilizing 24 hours of data to predict the next 24 hours. The findings from their research underscored the potential of transfer learning-based models to augment prediction precision, particularly when compared to regular LSTM models.

Current literature has largely focused on non-residential buildings (Cai et al., 2020; Li et al., 2021). This study narrows the scope to commercial buildings, providing valuable insights into the unique challenges and opportunities within this specific building type. Studies such as those by Gao et al. (2020) and Li et al. (2021) have demonstrated the efficacy of TL in enhancing prediction accuracy in situations with limited data availability. Therefore, this study further explores this aspect, investigating scenarios with limited data availability, and aims to provide additional evidence supporting the effectiveness of TL under such conditions. Finally, besides evaluating the accuracy of the TL models, this research also examines computational time, which is often neglected in the current literature, thereby contributing to the understanding of the further benefits of TL in building load forecasting.

In this light, the present paper seeks to provide the following contributions to the current literature on TL techniques focused in commercial building load forecasting models:

- Designed and implement a transfer learning approach for load forecast-

ing in the context of commercial buildings.

- Presented the effectiveness of transfer learning for improving forecasting accuracy, particularly when data is scarce.
- Demonstrated the ability of transfer learning to achieve comparable performance to models trained on abundant data.
- Identified a positive correlation between Pearson correlation coefficient (PCC) and the success of transfer learning, suggesting that similarities in load patterns play a crucial role in the transferability of learning.
- Highlighted the resilience of the transfer learning model under non-ideal conditions.
- Demonstrated a significant reduction in model training time through the implementation of transfer learning.

1.3. Paper structure

The rest of this paper will be structured as follows. In Section 2, the base load forecasting model and its structure will be presented. Section 3 will introduce the applied TL method and the connection between the base model and the target model. The results will be displayed in the first subsection of Section 4 and a discussion will be held in the second subsection of the same section. To finalize, Section 5 will present the conclusions.

2. Case Study

This study explores the application and effectiveness of TL techniques for load forecasting in commercial buildings by conducting a case study. Hereby we use virtual buildings (VB) located at the INESC TEC, in Porto, Portugal. The VB were selected for their diverse characteristics and varying energy consumption patterns, providing a comprehensive set of data for the analysis.

INESC TEC is originally divided in two building blocks, labelled A and B. In this study, we are considering only floors with office analogue activities, being floors 1 to 4. Thus, there are a total of eight VBs analysed in this study. One of them was selected as a base VB from which the knowledge will be transferred to the remaining VBs. Figure 2 illustrates that division.

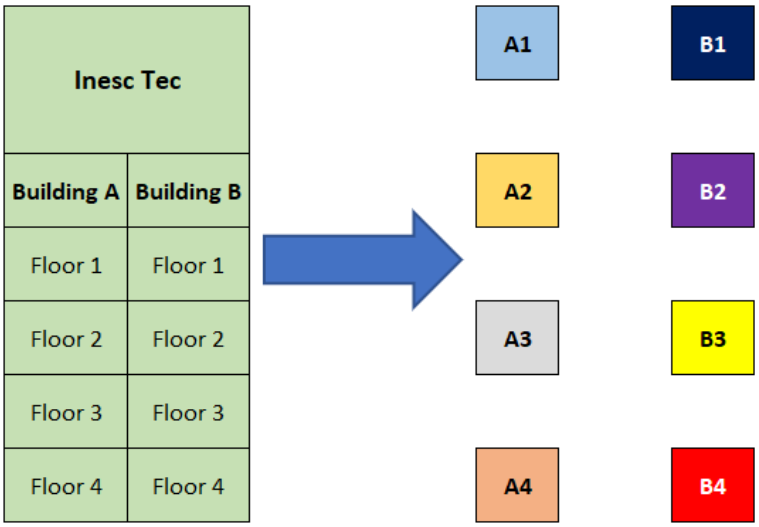

Figure 2: Division of INESC TEC floors in virtual buildings.

The case study involves training deep learning models using historical load data from the VBs. A robust model, pre-trained on a large dataset from a building with abundant data, serves as the base for the development of new models. For comparison, each VB will have a model trained, with the same architecture, using full and scarce data, to provide ground for comparison with the TL model. The TL models will be developed utilizing the same architecture as the baseline models. That is, the networks layers will be loaded with the hyperparameters of the pre-trained base model. In order to retain the knowledge previously acquired, these layers are frozen to avoid loosing previous information with further training. Finally, new additional layers are added to the model for fine tuning with a small dataset from the target VB. A fluxogram describing the process is presented in Figure 3.

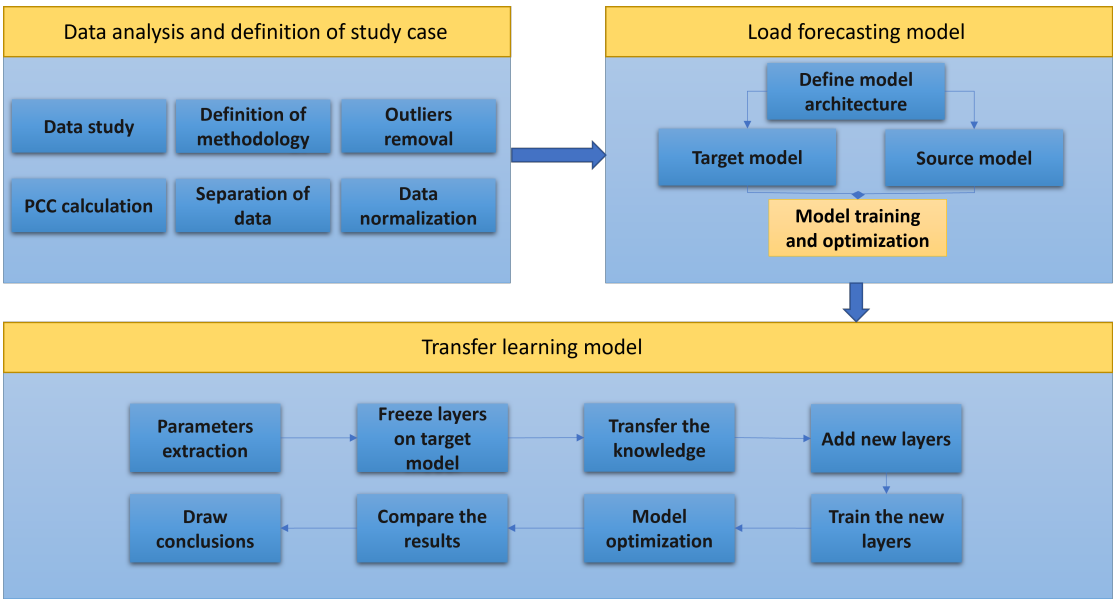

Figure 3: Fluxogram describing the transfer learning models development.

In the case study, we aim to answer the following key questions:

- Can transfer learning provide high forecasting accuracy when training data is scarce?
- How does the performance of models trained using transfer learning compare with those trained without it?
- What is the impact of the source-target building pair on the performance of the transfer learning model?
- How does the similarity between load consumption time series of different virtual buildings influence the effectiveness of the transfer learning technique?

Subsequently, the performance of these new models is evaluated and compared with models trained on large and small datasets. The evaluation includes metrics such as Root Mean Square Error (RMSE), and training time. It also includes a visual representation of the model's performance through scatter plots and line plots.

The findings from this case study contribute to our understanding of the potential of TL in improving load forecasting, especially in scenarios where data availability is limited. They also shed light on the conditions under which TL can be most effectively applied.

3. Baseline Models

The baseline models were implemented in compliance to good practices in ML. That is, the original INESC TEC database was cleaned, processed, and structured to minimize quality issues and bias from the dataset. The inputs of the model consisted in day of the week, week in year, hour, minute, a boolean flag stating if that day is a holiday or not, the load consumption delayed by a day and the load consumption delayed by a week. The features of the model were normalized using the Min-Max normalization method and divided in according to Figure 4, being 80% for training and 20% for validation. Also, the last two months of data were removed for later testing and calculation of evaluation metrics. Therefore, the last two months represent data that was not used to calibrate model hyperparameter nor used for training early stopping.

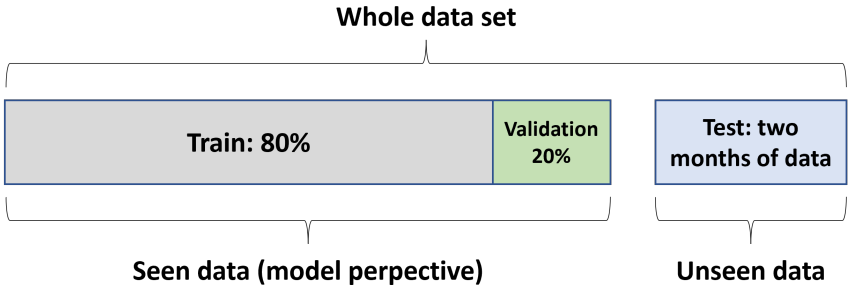

Figure 4: Data split graphical representation.

The model was implemented in Python and its architecture consists in a 1D Convolutional Layer with 7 inputs (one for each feature), A 32 memory cell LSTM layer, a 16 memory cell LSTM layer, and a dense layer for output. Figure 5 illustrates the model architecture employed in this study.

The architecture commences with a one-dimensional convolutional layer, equipped with 64 filters, a kernel size of 3, and the Rectified Linear Unit

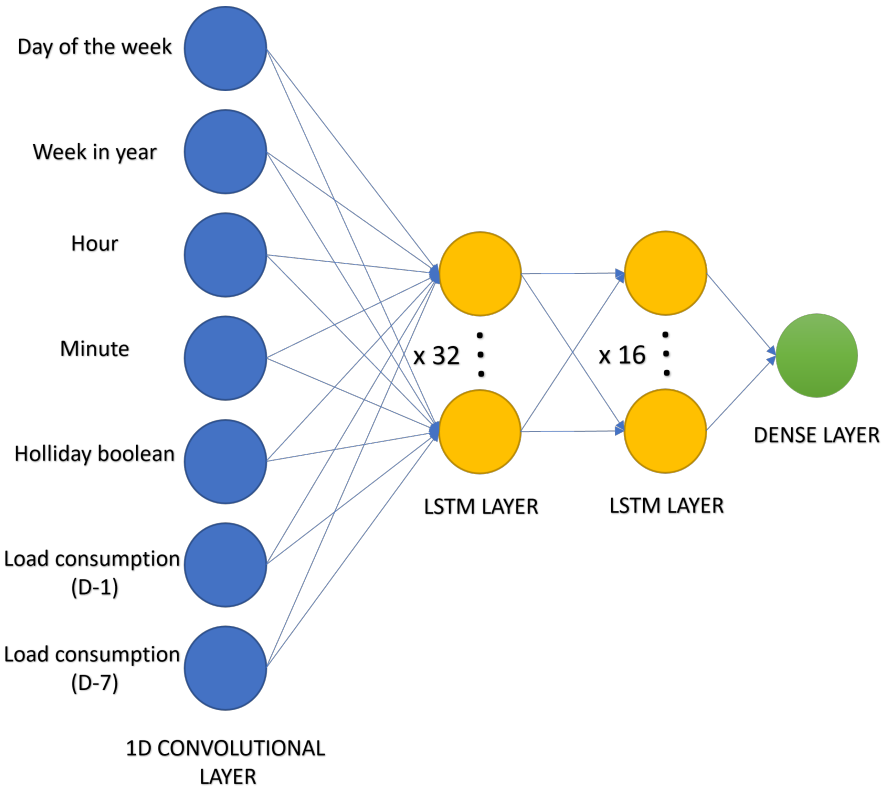

Figure 5: Graphical description of the model architecture and features.

(ReLU) activation function. This layer is purposed for the extraction of salient features from the input sequence.

Subsequently, a max pooling layer with a pool size of 2 is utilized, aiming to diminish the output dimensions of the convolutional layer. This output is then fed into a Long Short-Term Memory (LSTM) layer, which consists of 32 hidden neurons and employs the hyperbolic tangent (tanh) activation function. This layer aids in modeling the sequence of features extracted previously.

Following, a second LSTM layer mirrors the architecture of the first but incorporates only 16 hidden neurons. To avoid overfitting, dropout layers with a drop rate of 0.2 are interspersed within the architecture. The architecture culminates in a dense output layer, which is responsible for generating the final prediction.

The aforementioned architecture was employed to construct the baseline models for all VBs, accommodating both abundant and scarce data scenarios.

4. Transfer learning model

The TL method utilized in this study is the homogeneous inductive learning (Table 1), where the target uses information extracted by the source to enhance load prediction, and, later, is fine-tuned with limited target data, to enable the model to adapt to the target task (Pinto et al., 2022).

Table 1: Transfer learning classification.

Type	Domain	Task	Example
Homogeneous inductive learning	Source = Target	Source ≠ Target	Transfer learning is used to enhance building monthly electric load prediction leveraging information from similar buildings in different districts, that exhibits a different conditional probability.
Heterogeneous inductive learning	Source≠ Target	Source ≠ Target	Transfer learning is used to fine-tune a pretrained neural network initially built to perform multi-class classification, to increase the accuracy of a prediction model for building temperature setback identification.
Homogeneous transductive learning	Source = Target	Source = Target	Transfer learning is used for improving the accuracy of home activity estimation by exploiting the data of a source house applied to a target house with no labelled data.
Heterogeneous transductive learning	Source ≠ Target	Source = Target	Transfer learning is used to predict building dynamics by extracting features from multiple households in an online fashion, without having access to labelled data.

The development of the TL model consists in applying the architecture defined in Figure 5, and subsequently loading the hyperparameters of the

pre-trained model B2, which was trained using abundant data (two years of data before cleaning). These layers were subsequently frozen, with new layers being appended to each model in accordance with Figure 6.

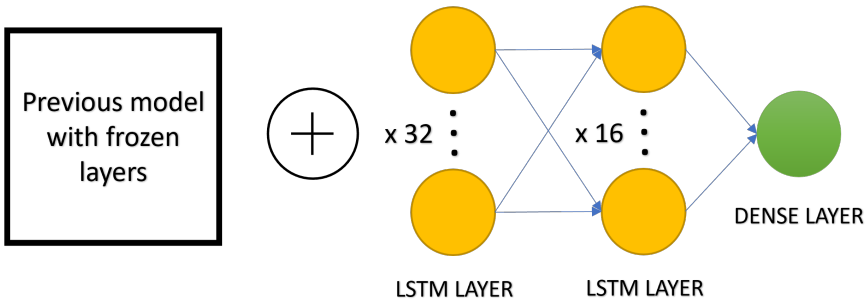

Figure 6: Graphical description of the transfer learning models.

5. Results and Discussion

5.1. Results

The effectiveness of the TL approach was evaluated based on Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE). These metrics were computed over a period of two months of data that were not involved in the training process. The baseline for this investigation was the model trained with abundant data from Virtual Building B2, from which knowledge was transferred to other models. Figure 7 showcases the comparison between predicted and actual values for the base model's test set, consisting in a period of two months. The model, from which knowledge will be transferred, demonstrates a MAPE of 25.02% and a RMSE of 90.92 Wh.

For discussion purposes, the implemented models will be categorized into three distinct case scenarios, as described in Figure 8. The first scenario (CS1) encompasses models trained with the full dataset. The second scenario (CS2) includes models trained with partial dataset, for a one month period. The third and final scenario (CS3) comprises models trained with limited data, similar to the CS2, but with the application of TL.

5.2. Root Mean Square Error (RMSE)

An analysis of the RMSE metric revealed that, with the exception of VB B3, the TL model led to improvements in all VBs when compared to

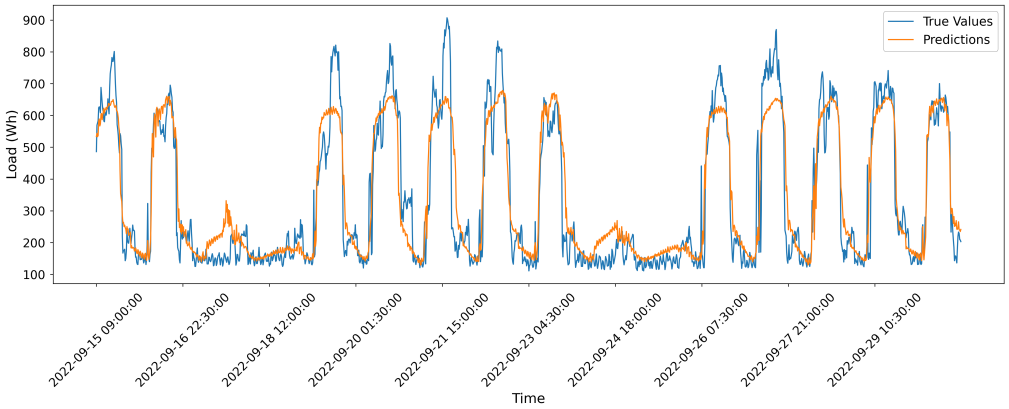

Figure 7: Line plot of the predicted load for the baseline model B2.

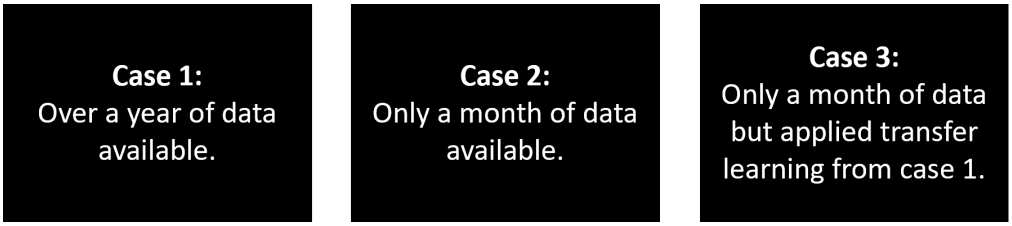

Figure 8: Case scenarios.

the model trained with scarce data. The average improvement across all buildings was approximately 42.76% and the results are present in Table 2 and demonstrated in Figure 9. When the TL model was compared with the model trained on abundant data, it was found to deliver comparable performance. Interestingly, the TL model even surpassed the performance of the model trained with abundant data in Virtual Buildings A3, B1, and B4.

Table 2: RMSE for all case scenarios and comparison.

VB	CS 1	CS 2	CS3	CS 3/CS 2	CS 3/CS 1
A1	193.40	384.82	230.95	39.99%	-19.41%
A2	140.59	304.47	147.69	51.49%	-5.05%
A3	296.30	420.20	230.39	45.17%	22.24%
A4	132.74	361.34	174.37	51.74%	-31.37%
B1	233.71	331.95	182.40	45.05%	21.95%
B2	90.92	217.84	95.48	56.17%	-5.01%
B3	74.52	111.71	132.32	-18.46%	-77.56%
B4	62.13	61.24	55.31	9.68%	10.98%

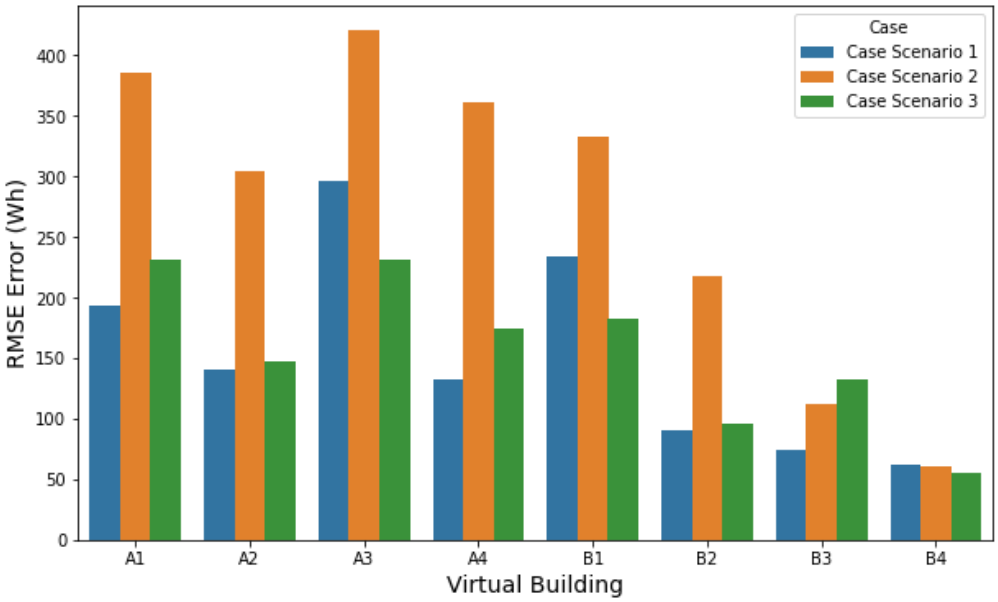

Figure 9: RMSE results for all case scenarios.

5.3. Mean Absolute Percentage Error (MAPE)

The MAPE metric offered further insights into the performance of the TL model. The model succeeded in enhancing the accuracy of load forecasting for all virtual buildings except B3 and B4, with an average improvement of approximately 20.48%. The results are depicted in Table 3 and illustrated in Figure 10 In comparison to the model trained with abundant data (CS 1),

the TL model yielded similar, and in some cases superior, results. The latter model demonstrated better performance for Virtual Buildings A3, B1, B2, and B4.

Table 3: MAPE for all case scenarios and comparison.

VB	**CS 1**	**CS 2**	**CS 3**	**CS 3/CS 2**	**CS 3/CS 1**
A1	53.62	61.57	57.77	6.17%	-7.74%
A2	25.87	49.38	34.32	30.49%	-32.66%
A3	87.03	61.82	52.14	15.66%	40.09%
A4	24.92	34.59	27.31	21.03%	-9.59%
B1	52.01	39.98	36.28	9.25%	30.24%
B2	25.02	41.65	24.88	40.26%	0.57%
B3	29.90	54.28	70.93	-30.68%	-137.25%
B4	20.57	15.13	16.27	-7.54%	20.93%

Another metric evaluated in this study was the training time, specifically comparing the time required to train the models in CS1 versus CS3. The findings from this comparison are summarized in Table 4.

Table 4: Training time (in seconds) comparison between case scenario 1 and 3.

CS 1 training time (s)	**CS 3 training time (s)**	**Reduction**
95.79	16.85	82.41%
90.16	12.89	85.71%
52.08	31.97	38.61%
80.04	15.18	81.03%
57.13	17.37	69.59%
91.88	29.18	68.24%
112.94	14.43	87.23%
53.26	16.95	68.17%

5.4. Discussion

The results of this study provide compelling evidence of the successful application of TL in the context of load forecasting for building energy management. Despite the challenges posed by data scarcity in individual buildings, TL has demonstrated its capacity to leverage knowledge from a pre-trained model to improve forecasting accuracy in the target buildings. This was

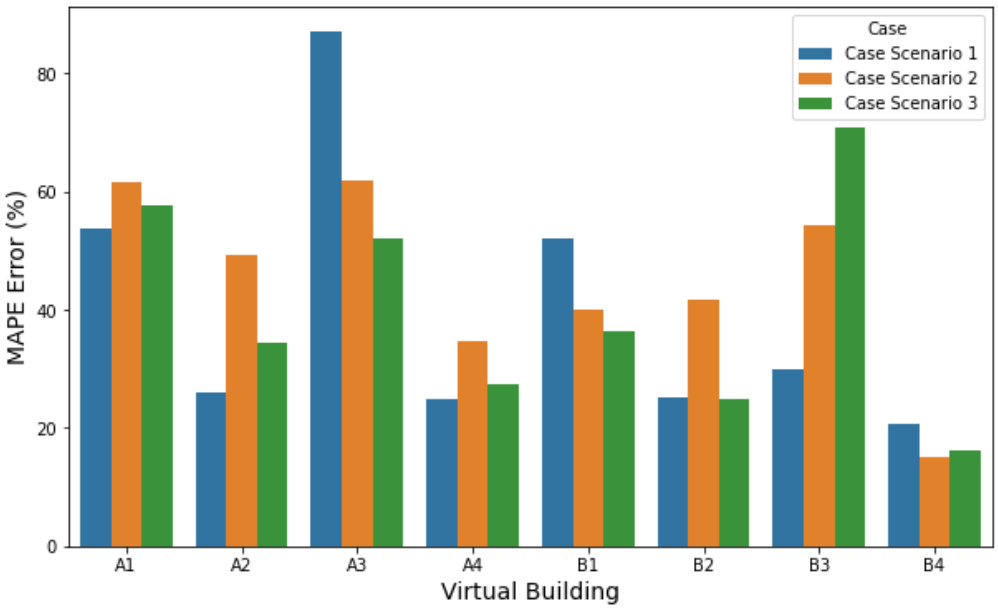

Figure 10: MAPE results for all case scenarios.

accomplished without necessitating an abundance of training data, thereby overcoming one of the most significant hurdles in the deployment of data-driven forecasting models. Our findings underscore the potential of TL to address issues of data scarcity, and its consequent implications for the broader adoption of smart building technologies.

In the growing field of study of TL, a significant finding of this study is the relationship between the Pearson Correlation Coefficient (PCC) and the effectiveness of the TL technique. The results suggest a direct correlation: the higher the PCC between the base case and the target building, the more successful the TL application tends to be, this is depicted in Figure 11. This observation aligns with the core principle of TL, which leverages commonalities and patterns shared between different datasets. Therefore, the higher the correlation between the base case and the target, the more effective the transfer of knowledge becomes. It underlines the importance of selecting a relevant and representative base model when employing TL methods.

Particular attention is drawn to the results corresponding to VBs B3 and B4. These buildings demonstrate the least successful implementation of TL. The reasons are twofold: firstly, these buildings exhibit a low PCC with the

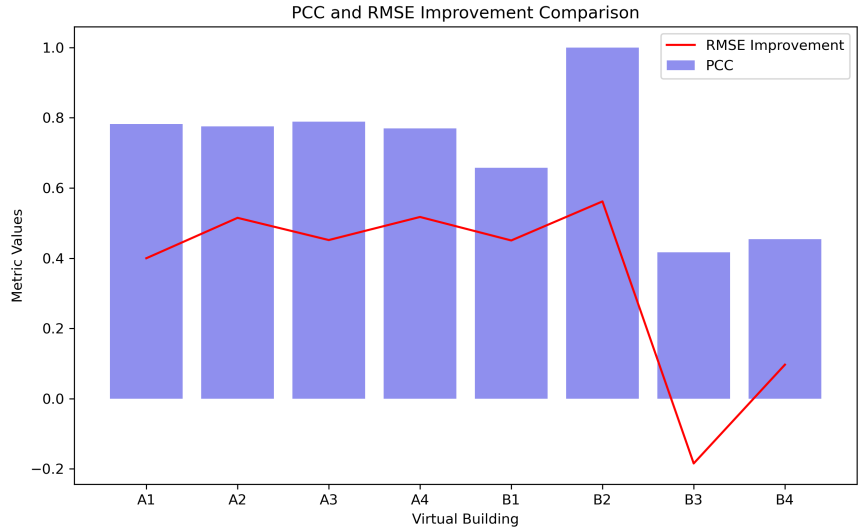

Figure 11: PCC and RMSE (comparison from CS 3 to CS 2) for all virtual buildings.

base case (0.41 and 0.45 respectively), which, based on the aforementioned discussion, inherently limits the effectiveness of the TL technique. Secondly, the original measurements in these buildings were abundant in errors and missing values, resulting in distorted load patterns. This distortion complicates the task for the predictive model, as it must attempt to predict patterns based on flawed data. In Figure 12, we present the load consumption for the testing set of VB B3, compared to the corresponding predictions. Upon examination, we can discern a certain weekly pattern in the load consumption during the second half of the analyzed time series. However, this pattern is absent in the first half. This inconsistency in the data significantly jeopardizes the model's ability to generate accurate predictions, since the model learned the pattern from the base model and the presented load for B3 has a far distinct pattern comparing to the one present in Figure 7. This serves to underscore the challenges associated with forecasting in environments where data exhibits irregular or unpredictable patterns.

Interestingly, in the case of building B4, the model trained with scarce data outperforms the model trained with abundant data (Table 3 and 2). This phenomenon might seem counter-intuitive at first, given the general

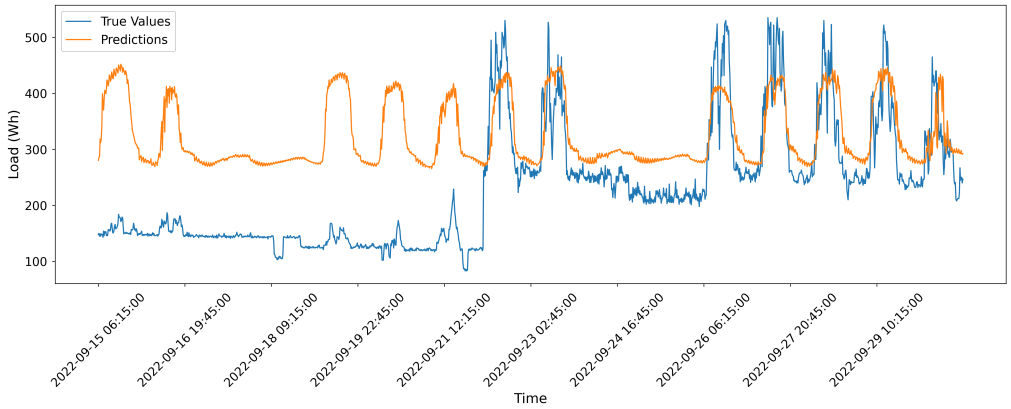

Figure 12: Line plot of the actual and predicted load for Virtual Building B3 in case scenario 3.

principle that models trained with more data tend to be more accurate. However, it underlines the notion that the quality of the data is more crucial than the quantity, especially in the context of deep learning models. When the data is implemented with measurement errors, a larger dataset may worsen the model's performance by further reinforcing incorrect patterns. Consequently, the results for B4 underscore the importance of data quality and integrity in the training of predictive models.

When it comes to training time, the results are rather straightforward: a significant reduction in training time was observed across all models, averaging a decrease of 72.62%. In this particular study, the longest training time recorded was 112.94 seconds for the larger models. However, in a more expansive study involving more features and data, this duration could significantly increase, underscoring the potential of TL as a method to reduce training time while preserving high levels of accuracy.

6. Conclusion

This study delved into the application of TL for load forecasting in virtual buildings, with a particular emphasis on addressing the constraints of data scarcity. The findings firmly establish the efficacy of TL in augmenting forecasting accuracy, especially under conditions of limited data availability.

Consistently, the TL approach surpassed the performance of models that were trained exclusively on scarce data, with an average enhancement of

42.76% in RMSE across the virtual buildings. Moreover, this approach exhibited comparable performance to models trained with full dataset, even exceeding their performance in certain instances.

The research further discovered a positive correlation between the Pearson correlation coefficient (PCC) and successful application of TL. Buildings that demonstrated a higher PCC with the base case often yielded superior results, suggesting that the resemblance in load patterns is a crucial factor in the effective transfer of learning.

Nonetheless, the results derived from the virtual buildings B3 and B4 deviated noticeably, largely due to their low correlation with the base case and distortions in load patterns due to measurement errors. Nevertheless, the TL model outperformed the model trained on scarce data for building B4, indicating its robustness in less than ideal situations.

Importantly, this investigation also revealed a significant reduction in model training times when employing TL, with an average reduction of 72.62% compared to the models trained with large amount of data. This highlights another significant advantage of TL, particularly in larger-scale projects where training times could be considerably longer.

In conclusion, this study contributes to the ongoing development of precise and efficient load forecasting models, thus supporting building energy management and wider decarbonisation efforts. It underscores the potential of TL as a promising strategy to overcome data scarcity, thereby facilitating wider adoption and generalization of smart building applications.

Acknowledgement

This research received partial support from the Norte Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, through the European Regional Development Fund (ERDF), within the DECARBONIZE project under agreement NORTE-01-0145-FEDER-000065 and by the Scientific Employment Stimulus Programme from the Fundação para a Ciência e a Tecnologia (FCT) under the agreement 2021.01353.CEECIND.

References

Ahn, Y., Kim, B.S., 2022. Prediction of building power consumption using transfer learning-based reference building and simulation dataset. Energy and Buildings 258, 111717.

URL: https://doi.org/10.1016/j.enbuild.2021.111717, doi:10.1016/j.enbuild.2021.111717.

Alanne, K., Sierla, S., 2022. An overview of machine learning applications for smart buildings. Sustainable Cities and Society 76, 103445. URL: https://doi.org/10.1016/j.scs.2021.103445, doi:10.1016/j.scs.2021.103445.

Cai, L., Member, S., Gu, J., Jin, Z., 2020. Two-Layer Transfer-Learning-Based Architecture for Short-Term Load Forecasting 16, 1722–1732.

Gao, Y., Ruan, Y., Fang, C., Yin, S., 2020. Deep learning and transfer learning models of energy consumption forecasting for a building with poor information data. Energy and Buildings 223, 110156. URL: https://doi.org/10.1016/j.enbuild.2020.110156, doi:10.1016/j.enbuild.2020.110156.

Li, A., Xiao, F., Fan, C., Hu, M., 2021. Development of an ANN-based building energy model for information-poor buildings using transfer learning. Building Simulation 14, 89–101. doi:10.1007/s12273-020-0711-5.

Pan, S.J., Yang, Q.g., 2010. A Survey on Transfer Learning. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING 22, 1345–1359. URL: https://ieeexplore.ieee.org/document/5288526, doi:10.1007/978-981-15-5971-6_83.

Pinto, G., Wang, Z., Roy, A., Hong, T., Capozzoli, A., 2022. Transfer learning for smart buildings: A critical review of algorithms, applications, and future perspectives. Advances in Applied Energy 5, 100084. URL: https://doi.org/10.1016/j.adapen.2022.100084, doi:10.1016/j.adapen.2022.100084.

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., Liu, C., 2018. A survey on deep transfer learning. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 11141 LNCS, 270–279. doi:10.1007/978-3-030-01424-7_27.

References

[1] IPCC. Global warming of 1.5°c, 2018. URL https://www.ipcc.ch/sr15/.

[2] International Energy Agency. *Energy Technology Perspectives 2020*. International Energy Agency, 2020. URL https://www.iea.org/reports/energy-technology-perspectives-2020.

[3] O. Lucon, D. Ürge Vorsatz, A. Zain Ahmed, H. Akbari, P. Bertoldi, L. F. Cabeza, et al. Buildings. In *Climate Change 2014: Mitigation of Climate Change*, 2014. URL https://www.ipcc.ch/report/ar5/wg3/buildings/.

[4] Kari Alanne and Seppo Sierla. An overview of machine learning applications for smart buildings. *Sustainable Cities and Society*, 76(July 2021):103445, 2022. ISSN 22106707. doi: 10.1016/j.scs.2021.103445. URL https://doi.org/10.1016/j.scs.2021.103445.

[5] Aya Nabil Sayed, Yassine Himeur, and Faycal Bensaali. Deep and transfer learning for building occupancy detection: A review and comparative analysis. *Engineering Applications of Artificial Intelligence*, 115(April):105254, 2022. ISSN 09521976. doi: 10.1016/j.engappai.2022.105254. URL https://doi.org/10.1016/j.engappai.2022.105254.

[6] 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. URL https://doi.org/10.1016/j.enbuild.2020.110156.

[7] Karl Weiss, Taghi M. Khoshgoftaar, and Ding Ding Wang. *A survey of transfer learning*, volume 3. Springer International Publishing, 2016. ISBN 4053701600. doi: 10.1186/s40537-016-0043-6.

[8] Muhammad Qamar Raza and Abbas Khosravi. A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. *Renewable and Sustainable Energy Reviews*, 50:1352–1372, 2015. ISSN 18790690. doi: 10.1016/j.rser.2015.04.065. URL http://dx.doi.org/10.1016/j.rser.2015.04.065.

[9] Han Li, Zhe Wang, Tianzhen Hong, and Mary Ann Piette. Energy flexibility of residential buildings: A systematic review of characterization and quantification methods and applications. *Advances in Applied Energy*, 3(July):100054, 2021. ISSN 26667924. doi: 10.1016/j.adapen.2021.100054. URL https://doi.org/10.1016/j.adapen.2021.100054.

[10] Seung Min Jung, Sungwoo Park, Seung Won Jung, and Eenjun Hwang. Monthly electric load forecasting using transfer learning for smart cities. *Sustainability (Switzerland)*, 12(16), 2020. ISSN 20711050. doi: 10.3390/SU12166364.

[11] Giuseppe Pinto, Zhe Wang, Abhishek Roy, Tianzhen Hong, and Alfonso Capozzoli. Transfer learning for smart buildings: A critical review of algorithms, applications, and future perspectives. *Advances in Applied Energy*, 5(November 2021):100084, 2022. ISSN 26667924. doi: 10.1016/j.adapen.2022.100084. URL https://doi.org/10.1016/j.adapen.2022.100084.

[12] Bens Pardamean, Hery Harjono Muljo, Tjeng Wawan Cenggoro, Bloomest Jansen Chandra, and Reza Rahutomo. Using transfer learning for smart building management system. *Journal of Big Data*, 6(1), 2019. ISSN 21961115. doi: 10.1186/s40537-019-0272-6. URL https://doi.org/10.1186/s40537-019-0272-6.

[13] Tao Hong and Shu Fan. Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting*, 32(3):914–938, 2016. ISSN 01692070. doi: 10.1016/j.ijforecast.2015.11.011. URL http://dx.doi.org/10.1016/j.ijforecast.2015.11.011.

[14] Simon S Haykin. *Neural networks and learning machines*. Pearson, 2009.

[15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, 2015.

[16] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[17] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[18] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[20] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[21] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9 (8):1735–1780, 1997.

[23] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[24] Ilya Sutskever, James Martens, and Geoffrey E Hinton. Generating text with recurrent neural networks. 2011.

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[26] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. In *Journal of Machine Learning Research*, volume 12, pages 2121–2159, 2011.

[27] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, 2012.

[28] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[29] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[30] Cort J Willmott and Kenji Matsuura. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, 30(1):79–82, 2005.

[31] Rob J. Hyndman and Anne B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, 2006. ISSN 01692070. doi: 10.1016/j.ijforecast.2006.03.001.

[32] T. Chai and R. R. Draxler. Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, 7 (3):1247–1250, 2014. ISSN 19919603. doi: 10.5194/gmd-7-1247-2014.

[33] Eliana Vivas, Héctor Allende-Cid, and Rodrigo Salas. A systematic review of statistical and machine learning methods for electrical power forecasting with reported mape score. *Entropy*, 22(12):1–24, 2020. ISSN 10994300. doi: 10.3390/e22121412.

[34] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2009.

[35] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. doi: 10.1109/TKDE.2009.191.

[36] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

[37] Hesham K Alfares and Mohammad Nazeeruddin. Electric load forecasting: literature survey and classification of methods. *International Journal of Systems Science*, 33(1):23–34, 2002.

[38] Henrique S Hippert, Carlos E Pedreira, and Reinaldo C Souza. Neural networks for short-term load forecasting: A review and evaluation. *IEEE Transactions on Power Systems*, 16 (1):44–55, 2001.

[39] Chien-Hwa Chen. A support vector machine-aided method for short-term load forecasting. *Electric Power Systems Research*, 81(2):544–552, 2011.

[40] M R AlRashidi and M E El-Hawary. A survey of particle swarm optimization applications in electric power systems. *IEEE Transactions on Evolutionary Computation*, 14(4):913–918, 2010.

[41] Wei-Cheng Lai, Chen-Yi Fan, Yi-Hsiang Huang, and Pai-Hsi Chang. Recurrent convolutional neural networks for electricity load forecasting. *IEEE Transactions on Smart Grid*, 8 (4):1625–1634, 2017.

[42] Xiaochen Li, Xiaoling Wang, Huanjia Yang, and Qingchen Zhang. Short-term load forecasting with deep residual networks and multi-objective evolutionary algorithm. *Applied Energy*, 195:222–233, 2017.

[43] Davide L Marino, Stavros Ntalampiras, and Aurelio Uncini. Building energy load forecasting using deep learning and clustering algorithms. *Applied Energy*, 212:372–381, 2018.

[44] Yuxiang Zhang, Gang Liu, and Xuesong Zhou. Short-term load forecasting in commercial buildings using deep learning with attention mechanism. *Applied Energy*, 287:116653, 2021.

[45] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

[46] Long Cai, Student Member, Jie Gu, and Zhijian Jin. Two-Layer Transfer-Learning-Based Architecture for Short-Term Load Forecasting. 16(3):1722–1732, 2020.

[47] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, 109(1):43–76, 2021. ISSN 15582256. doi: 10.1109/JPROC.2020.3004555.

[48] Ao Li, Fu Xiao, Cheng Fan, and Maomao Hu. Development of an ANN-based building energy model for information-poor buildings using transfer learning. *Building Simulation*, 14(1):89–101, 2021. ISSN 19968744. doi: 10.1007/s12273-020-0711-5.

[49] Clayton Miller and Forrest Meggers. The Building Data Genome Project: An open, public data set from non-residential building electrical meters. *Energy Procedia*, 122:439–444, 2017. ISSN 18766102. doi: 10.1016/j.egypro.2017.07.400. URL https://doi.org/10.1016/j.egypro.2017.07.400.

[50] Yusun Ahn and Byungseon Sean Kim. Prediction of building power consumption using transfer learning-based reference building and simulation dataset. *Energy and Buildings*, 258:111717, 2022. ISSN 03787788. doi: 10.1016/j.enbuild.2021.111717. URL https://doi.org/10.1016/j.enbuild.2021.111717.

[51] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11141 LNCS: 270–279, 2018. ISSN 16113349. doi: 10.1007/978-3-030-01424-7{_}27.

[52] François Chollet et al. Keras. https://keras.io, 2015.

[53] TensorFlow. Tensorflow: Large-scale machine learning on heterogeneous systems. https://www.tensorflow.org, 2015.