Multivariate Deep Transfer Learning for Robust Building Electric Load Forecasting

Master's Thesis

for acquiring the degree of Master of Science (M.Sc.)

in Information Systems
at the School of Business and Economics
of Humboldt-Universität zu Berlin

submitted by

Benedikt Rein
Student Number: 565136

First Examiner: Prof. Dr. Stefan Lessmann

Second Examiner: Prof. Dr. Benjamin Fabian

Berlin, July 30, 2024

Abstract

This master's thesis proposes to use global or multi-target multivariate and channel-dependent deep learning solutions for building electric load forecasting to capture complex and non-linear correlations in favour of extensively modelling entity-specific covariates to achieve state-of-the-art predictive performance. New buildings or newly installed measuring infrastructure struggle with the cold-start problem. We evaluate different transfer learning approaches and give an outlook on how the best predictive performance can be achieved over the initial building lifetime.

Transformer, iTransformer, TSMixer and N-HiTS models are selected to evaluate predictive performance and transfer learning robustness when applied to three building electric load datasets. All datasets are used as source and target data. Jumpstart and asymptotic performance are adapted as time-series transfer learning metrics for the building electric load forecasting task in a sparse data setting. ARIMA is used as a statistical baseline and TimeGPT as a foundation model baseline.

Especially iTransformer and TSMixer can improve their results through transfer learning. However, this is still strongly dependent on the source and target data. TimeGPT shows impressive predictive accuracy on two datasets but fails on one. ARIMA is not in a comparable range without using meaningful covariates.

Overall, we can show that deep-learning models can capture meaningful information from multivariate, global datasets with only a short horizon of data present. For new buildings, predictions can be made by zero-shot transfer learning but should be replaced by fine-tuned models when multiple weeks of data are present to increase performance. Choosing the right model and similar source data is still crucial for accuracy and robustness. A case study shows the potential to save a maximum of USD 35 per household by increasing forecasting accuracy in the first year by transfer learning compared to deep learning solutions without transfer learning.

Contents

List of Abbreviations

CD channel-dependent

CI channel-independent

CNN convolutional neural network

DL deep learning

DNN deep neural networks

DTW Dynamic Time Warping

ELD Electronc Load Diagram 2011-2014

FFNN feed forward neural network

GP2 Building Data Genome Project 2

GRN gated residual network

GWh Gigawatt hours

KNN k-nearest neighbour

LLM large language model

LSTM long-short-term-memory

MAE mean average error

MAPE mean absolute percentage error

MLP multi layer perceptron

MSE mean squared error

MW Megawatt

MWh Megawatt nhours

NN neural network

RevIn reverse instance normalisation

RNN recurrent neural network

SOTA state-of-the-art

SVR Support Vector Regression

TL transfer-learning

TSFM time-series forecasting mining

WD Wasserstein distance

List of Figures

List of Tables

1 Introduction

Motivation and Relevance

Buildings consume approximately one-third of global energy (Pinto et al., 2022). As the demand for electrical energy changes, the proportion of energy generated from renewable sources, such as photovoltaics and wind turbines, leads to highly variable energy supply and demand patterns. Short- and long-term storage aids to save energy but complicates the setting even further. Efficient management of the energy infrastructure and reduction in fossil fuel usage necessitate robust and accurate electric load forecasting. This is becoming increasingly crucial as smart meters become widespread because of mandates such as the EU Directive on Energy Efficiency ("Directive (EU) 2019/944", 2022). Smart meters can transmit electricity, gas or water consumption data to generate substantial amounts of time-series data, a trend that coincides with the pressing challenges in the energy sector. These challenges include managing the intermittent supply of renewable energy, accommodating fluctuating demands across different seasons, times, and locations, and maintaining grid stability in the face of individualised consumption (Tian et al., 2019).

Accurate and readily available forecasts can aid individual households, infrastructure providers, and energy companies plan their energy usage and production more effectively. Households can optimise the timing of charging electric vehicles, operating energy-intensive appliances such as washing machines, and utilising or storing electricity generated by photovoltaic panels. Similarly, energy providers can respond better to fluctuating demand and make more informed decisions about managing storage and using additional fossil-fuel-powered plants. Studies have shown that improving the accuracy of load forecasting models can have significant economic impacts. For instance, enhancing mean absolute percentage error (MAPE) by just 1% can yield annual savings of approximately $1.6 million for a utility with a throughput of 10 GW annually (Hobbs et al., 1999). Lin et al. (2022) demonstrated that employing energy information systems can reduce the median building energy consumption by up to 3%.

During the transition to an era of data abundance, traditional statistical models are becoming less favourable owing to their complexity and the computational effort required to manage them, especially when dealing with thousands of households. The large number of similar and correlated time-series that can be efficiently combined into multivariate datasets represents a challenge for statistical approaches but provides a strong advantage for deep learning (DL) techniques. These models can handle hundreds or even thousands of series simultaneously, offering superior computational efficiency and potentially enhanced forecasting accuracy (Gasparin et al., 2019; Li, Li, et al., 2021). Despite the widespread exploration of transfer-learning (TL) for building electric load forecasting, most studies have focused on univariate or local time-series (Pinto et al., 2022). Often, they include only basic covariates, such as temperature and time of day, neglecting the potential benefits of a global, multi-target approach.

Problem Statement

Building electric load forecasting is inherently complex due to multiple influencing factors. These include individual behaviours, weather conditions, events, socioeconomic circumstances (Chen et al., 2020), building characteristics (Miller et al., 2020b), and seasonal variations. Recent developments, such as the increased adoption of electric vehicles and decentralised solar energy production, have further individualised energy consumption patterns (Forootani et al., 2024). While extensive covariates can theoretically help model these factors, applying such models to multiple thousand time-series datasets poses significant feasibility challenges. Although a valuable asset for creating well-fitted energy forecasting models, the wealth of data does not omit the "cold start" problem in newly deployed infrastructure, where sufficient historical data is lacking. Conversely, older infrastructure may struggle due to sparse data availability (Fan et al., 2020).

The scarcity of historical entity-specific data impedes the generation of meaningful predictions, thereby highlighting the utility of TL. This approach, which has seen significant advances and widespread adoption in recent years (Pinto et al., 2022), leverages knowledge from related tasks to improve prediction accuracy with minimal data. TL has the potential to serve as an effective method to enhance the initial predictive performance for new infrastructure and sustain improved performance over extended periods. Recent research by Brinkmeyer et al. (2022) suggests that multivariate forecasting not only enhances predictive performance but also ensures shorter and more predictable inference times, which is supported by Gunduz et al. (2023b).

Research Objective

This thesis investigates the applicability and effectiveness of TL for multi-target building energy load forecasting. Specifically, we examine channel-dependent DL methods that are recognised for their TL capabilities, making them suitable candidates for pre-trained foundational models (Chen et al., 2023; Liang et al., 2024; Liu et al., 2023). Our study evaluates the TL potential by comparing their predictive performance against traditional methods across different training and TL strategies.

In addition, the recently proposed *TimeGPT* model is used as a baseline to understand its effectiveness in this domain. With increasing advocacy in the literature for DL solutions in forecasting (Benidis et al., 2022; Liu et al., 2022; Yuan et al., 2023), our research aims to assess the practicality of these advanced models in real-world applications. Due to the parameter size of foundation models, smaller, more task-specific models focusing only on a single domain or frequency are an enticing alternative. The need for foundation models is critically examined versus domain-specific pre-trained models and more traditional statistical approaches.

We aim to answer the following questions:

- How can many electric load time-series efficiently be forecasted without extracting metadata

and modelling covariates?

- How to implement accurate and robust building electric load forecasting from the start without a significant amount of data?

Adopting multi-target or global multivariate and channel-dependent forecasting methods is proposed. These methods are designed to capture complex and nonlinear correlations, offering a potential improvement over heavily modified, entity-specific covariate approaches to achieve state-of-the-art (SOTA) predictive performance in building load forecasting. The key contributions to the topic of building electric load forecasting are as follows:

Key Contributions

- Identify useful settings for TL for building electric load forecasting
- Adapt time-series TL metrics to a sparse data setting
- Evaluate cross-dataset TL
- Evaluate potential savings in a case study
- Compare domain-specific pre-trained model to foundation model

Overview and Structure

The paper is organised as follows: Section 2 introduces the theoretical background of multivariate forecasting and TL, while Section 3 reviews related work in the field, establishing the context for our study. Section 4 outlines our methodology, including the data preparation, model configuration, selected datasets, evaluation criteria, and details of the experimental setup and case study. Section 5 presents our findings, followed by a discussion in Section 6, which explores the implications, limitations, and potential future research avenues. We conclude with a summary of our contributions to the field of electric load forecasting.

2 Theoretical Background

This section introduces the theoretical background of time-series forecasting, focusing on DL. Intricacies of multivariate forecasting and novel normalisation techniques are explained. A definition of TL is provided and key concepts are introduced. The section concludes with a short outlook on different pre-trained foundation models as a final stage of TL.

2.1 Forecasting

Within the realm of time-series analysis, typical tasks are classification, anomaly detection, and forecasting (Pinto et al., 2022), which will be the focus of this research. Forecasting considers past inputs and predicts future values. Forecasting is generally classified into different duration categories. Each category has unique challenges owing to specific human-, physics, or nature-made seasonalities, complex nonlinearities and uncertainties. The short-, mid- and long-term forecasting tasks will be presented in related works in a more building-load-specific setting. Ye and Dai (2021) give a clear distinction between uni- and multivariate as well as local and global forecasting, which is as follows:

> "In univariate forecasting methods, the future direction of a time-series is detected only by studying its past values, while multivariate methods usually model the dependency structure between the time-series. Global methods are applicable to predict the demand for numerous similar time-series, where model parameters are collectively assessed based on all the time-series, whereas in local methods, parameters are individually assessed for each time-series."

Multivariate Forecasting

Multivariate forecasting involves predicting future values based on multiple interrelated time-series data and leveraging the dependencies and interactions between multiple variables to improve predictive accuracy (Han et al., 2023). This approach is particularly beneficial in complex systems.

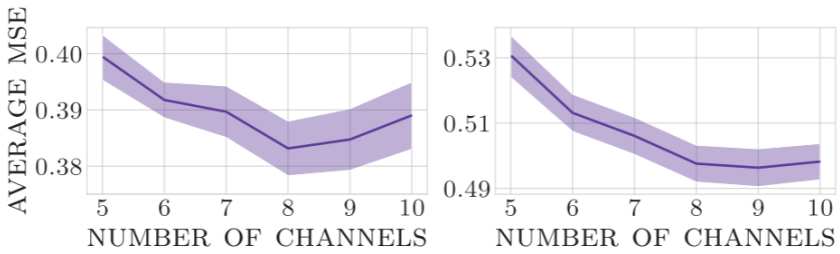

Figure 2.1: Influence of the no. of channels on predictive performance (Brinkmeyer et al., 2022)

This figure shows how the predictive performance changes with the number of channels used in the model. Left shows the results for $t_0 + 10$ and right for $t_0 + 80$.

In recent years, DL has significantly advanced multivariate forecasting. DL models, such as recurrent neural networks (RNNs), long-short-term-memorys (LSTMs), feed forward neural networks (FFNNs), and convolutional neural networks (CNNs), have demonstrated remarkable capabilities for capturing intricate patterns and temporal dependencies in large-scale multivariate data (Chan et al., 2019; Chen et al., 2023; Han et al., 2023). Recently, combinations of different architecture blocks and transformer architectures have been applied in the forecasting domain (Lim

et al., 2019; Liu et al., 2023; Wen et al., 2022). These models can learn and extract relevant features from raw data without extensive feature engineering, making them highly effective for multivariate forecasting. This can potentially give them an edge compared to statistical models, as correlations in multivariate datasets have been shown to improve the predictive performance (Brinkmeyer et al., 2022; Gunduz et al., 2023a).

Channel Dependency and Normalisation

Multivariate data can be forecasted by fitting one univariate model for all input series and calculating the loss for each series individually. This solution prevents models from learning correlations and interactions between series, practically omitting one of the biggest advantages of multivariate forecasting. Another approach is to simultaneously fit one model to all series while incorporating interactions between series and optimising the model to achieve the lowest loss on all series collectively. Brinkmeyer et al. (2022) demonstrate that a specific channel number can improve predictive performance, as shown in Fig.2.1. Han et al. (2023) explored those training strategies as channel-dependent (CD) and channel-independent (CI) training strategies. "[In Fig. 2.2, (a) shows] the CD strategy, where all channels are taken as input and forecasted future values depend on the history of all the channels. (b) shows the CI strategy, which treats the multivariate series as multiple univariate series and trains a unified model on these series. The prediction of each channel depends solely on its historical values, and the relationship between different channels is ignored." (Han et al., 2023). PatchTST was introduced as a CI model but is outperformed by the iTransformer benchmark, even though it employs similar, but also patched, cross-variate attention (Liu et al., 2023). They cocnluded this might prevent the model from capturing correlations between different time-patches. Han et al. (2023) found CI models to be more robust to statistical differences between the training and test set. CD models are more deceptive to statistical shifts in the input data, making them less accurate on test data, while Han et al. (2023) concluded, that more research into robust data normalisation is needed.

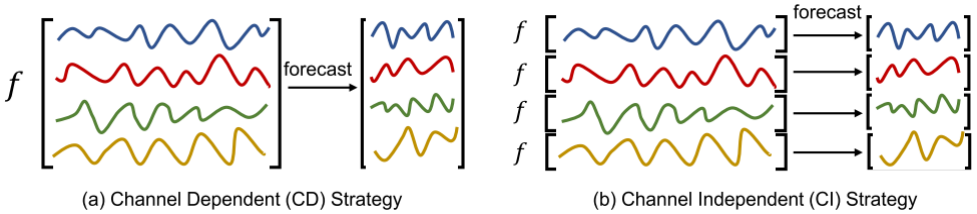

Figure 2.2: Concept of channel-dependent and channel-independent strategies (Han et al., 2023)

Stationarity is a well-established concept for statistical time-series analysis and has been widely ignored until recently in DL-based forecasting because research focused mostly on different architectures and was able to achieve SOTA results, depending on the model and dataset used. Achieving stationarity is the final task of data manipulation before fitting statistical models. The best practice in DL is to normalise the full data before using it as input for a model. This makes

the training more robust and less deceptive to outliers. However, this overlooks the key aim of stationarity, removing trends, drifts, and seasonal patterns. Recently, this challenge was approached by series stationarization (Liu et al., 2022), where each input chunk for the neural network (NN) is again normalised before the training. This was extended by reverse instance normalisation (RevIn) (Kim et al., 2022), which additionally computes trainable β and γ parameters, removing the difference in amplitude and time shift between the training and test data. This novel normalisation technique has already been widely implemented in standard forecasting libraries, such as Darts (Herzen et al., 2022) and PyTorch-Lightning (Falcon & The PyTorch Lightning team, 2019), and has been used for pre-trained models (Liang et al., 2024). Within a few years, normalisation has reached a point where it is an inherent part of many model architectures.

2.2 Transfer Learning

This section presents a theoretical background for TL in general and highlights specific aspects within the forecasting realm with a focus on DL.

TL is a powerful machine learning technique that involves taking knowledge gained from solving one problem and applying it to a different but related problem as visualised in Fig.2.3 (Himeur et al., 2022). This approach is particularly valuable for scenarios where data scarcity poses a challenge to training models from scratch or when there is a need to reduce the time and computational resources required for model training. By leveraging pre-trained models and insights obtained from extensive datasets in one domain, TL enables researchers and practitioners to jump-start the learning process in a new domain with limited data. This not only accelerates model development and deployment but may also enhance model performance, making it a useful tool in the rapidly evolving fields of machine learning and artificial intelligence. Language models and computer vision have most likely benefited the most from TL to date.

According to a frequently used definition of Lu et al. (2015), TL is the process of fitting a target predictive function on a task T1 on a source domain D1. This function can be transferred to either the same task T1 on a different target domain D2 or potentially a different task T2 within domain D1.

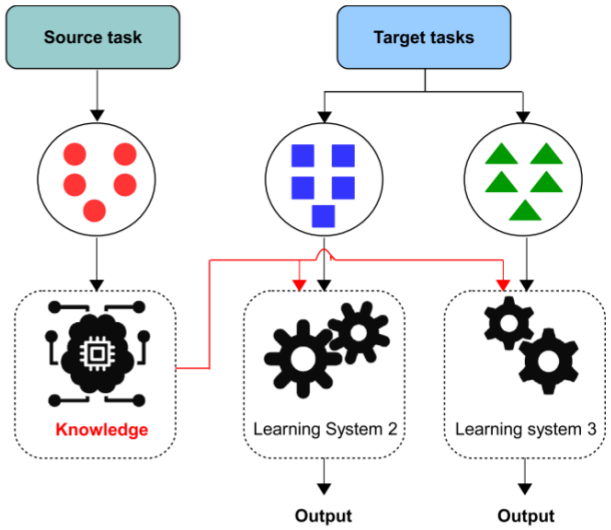

Figure 2.3: High-level concept of TL (Himeur et al., 2022)

Generally, there are three TL **strategies** based on the domains and tasks at hand: inductive, transductive, and unsupervised TL (Fan et al., 2020; Gunduz et al., 2023b). In *inductive TL*, both the source and target domains contain labelled data, yet the source and target tasks are different. It involves learning a general rule or model from the labelled training data in the source domain, and using this model to perform related but different tasks in the target domain. (Fan et al., 2020). *Transductive TL* is used when the source and target tasks are the same but in different domains. Typically, the source domain contains sufficient labelled data, whereas the target domain lacks labelled data. The aim is to predict outcomes specific to the target domain without the intention of generalising the model further (Fan et al., 2020).

The settings for *unsupervised TL* are similar to those for inductive learning, where the source and target domains are the same for different but related tasks. However, in unsupervised TL, neither domain has labelled data (Fan et al., 2020).

Each of these learning paradigms has unique applications and challenges in the field of TL. They are chosen based on the nature of the data and the overlap between the source and target tasks (Fan et al., 2020). For our goal of building electric load forecasting, we predict previously unseen data; therefore, our TL approach is classified as inductive. According to Pinto et al. (2022), this inductive approach can be implemented in various ways.

Knowledge transfer **implementation** can be done in different settings. Four methods have been proposed in the literature. The most widely used solution is *parameter-based* TL (Weber et al., 2021). This is implemented by training a model on a source task and domain and using the same model parameters and weights for fine-tuning or inference on a different target task or domain. If fine-tuning is applied, then either all or some model layers are fine-tuned (Pinto et al., 2022).

Multivariate Time-Series TL

Many building blocks for more complex models have been evaluated for their time-series TL capabilities. Examples include RNNs, FFNNs, and CNNs (Pinto et al., 2022). Ye and Dai (2021) evaluate different CNN architectures on global time-series TL but do not specify improvements made due to TL.

Many researchers have claimed that their model has TL capability (Chen et al., 2023; Liu et al., 2023). Liu et al. (2023) decided to apply attention across variates instead of the time steps and show the impressive zero-shot capability of iTransformer on unseen variates (Liu et al., 2023). PatchTST applies a similar architecture but uses multiple tokens per time-series while applying a CI training strategy (Nie et al., 2022).

NBEATS, which consists of multiple blocks of fully connected layers and residual connections, was evaluated for its knowledge transfer capabilities in a zero-shot setting (Kamalov et al., 2024). TSMixer, which can be classified as an multi layer perceptron (MLP) model (Chen et al., 2023), has been extended to a pre-trained model (Ekambaram et al., 2024) and is mentioned in the survey by Liang et al. (2024) exploring pre-trained models. For many of the recent models Liang et al. (2024) find that “another critical design is the normalization layer, where RevIn techniques, standardizing data through instance-specific mean and variance then reverting it at the output layer [...]“, are implemented.

2.2.1 Pre-Trained Models

In the following section, we give an overview of the state of pre-trained or foundation models. Leading to an introduction to which models are selected for this thesis, where they can be located in the realm of pre-trained models and why they are deemed good candidates for the experiments of this thesis. Pre-trained models, foundation models or universal forecasting models all "envision[s] a single Large Time Series Model capable of addressing diverse downstream forecasting tasks" (Woo et al., 2024). While Ma et al. (2023) focus their review on the pre-training approaches and architectures, completely omitting the buzzword "foundation model", Woo et al. (2024) focus more on the universal approach of a foundation model and the applicability on all downstream tasks. Liang et al. (2024) add the aspect of the time-series type like standard, spatial, or other types, such as event-based models, to the discussion. All discuss different architectural approaches, building blocks and training strategies.

Within the standard time-series category, many models are transformer-based (Liang et al., 2024). Some utilise pre-trained large language model (LLM), while others implement a more time-series-specific architecture. When no attention is used the RNN, MLP, and CNN architectures are the most common, Diffusion-based models were also implemented (Liang et al., 2024). According to Ma et al. (2023) and Woo et al. (2024), transformer-based architectures show tremendous potential

for use as pre-trained models. “A common practice in [time-series forecasting mining (TSFM)s] segments time-series into patches, which can effectively encapsulate local dynamics within input tokens“ (Liang et al., 2024). Most pre-trained models are designed to handle different frequencies, making additional encoding necessary and increasing the computational load and data needed for training (Woo et al., 2024).

Most pre-trained models are single-modality because they can only take temporal numeric data as input and not, for example, additional text or prompts (Liang et al., 2024). In particular, domain-specific models are trained in a fully supervised setting where sufficient labelled data are present, and models can specifically be fitted for the downstream task. In forecasting, values are generally the labels, making unlabelled data a rare issue (Liang et al., 2024). With sufficient homogeneity between the source and target domain, pre-trained models can be trained domain-specific for real-world application (Liang et al., 2024)

In the following, we will give an overview of where the models selected for this thesis can be located in the field of pre-trained models. The selected models are iTransformer, Transformer, N-BEATS, N-HiTS, and TSMixer and TimeGPT.

Transformer and iTransformer have a very similar architecture. The key difference is how the attention mechanism is implemented. Transformer applies attention across all time-steps, practically estimating which time-steps are the most influential for each other. The initial encoding merges all variates per time-step, according to Liu et al. (2023), this potentially removes useful information from each variate. iTransformer applies attention across all variates, practically encoding each variable and its specifics and estimating correlations between variables (Liu et al., 2023). The pre-trained model MOIRAI proposed by Woo et al. (2024) merges both approaches by flattening all series, practically enabling cross-variate and cross-time-step attention. This enables the model to extract meaningful dependencies between all input series. This 2-D attention was also implemented by Wang (2023), which is the PyTorch implementation of the iTransformer used in this research. The closed-source model TimeGPT also utilises attention mechanisms, although it is less clear about the specific implementation (Garza & Mergenthaler-Canseco, 2023).

The MLP architecture of TSMixer is used as a building block for the pre-trained models TimeHetNet proposed by Brinkmeyer et al. (2022) and Tiny Time Mixers according to Woo et al. (2024). They are used for heterogeneous and multivariate TL (Liang et al., 2024). The MLP architecture enables the model to extract information across variates and time-steps, according to Chen et al. (2023) this approach is computationally more efficient than using attention, especially for long inputs.

The iTransformer was selected because Liu et al. (2023) were the first to propose cross-variate attention coupled with a CD training strategy, which is now a crucial idea for training foundation models. Additionally, the authors suggest its potential as a pre-trained model and show impressive

results on unseen variates. A short introduction to each model's architecture is presented in section Model Description.

N-BEATS (Oreshkin et al., 2020) was selected for its supposed meta-learning capabilities (Kamalov et al., 2024) but consequently dropped for evaluation as it failed to capture the underlying information of the data in multiple cases, skewing the evaluation metrics, and preventing us from gaining meaningful insights. N-HiTS (Challu et al., 2022) with its very similar architecture is also selected because of its supposed meta-learning capabilities and the implementation as a CD model. With the extensions made to the model ("N-HiTS", n.d.), it proved more capable of extracting knowledge from small or irregular datasets and is kept for the experiments.

3 Related Work

In this section, we present the topics of building electric load forecasting, domain-specific trends in multivariate forecasting, and TL. We explore multivariate DL approaches for load forecasting and highlight the challenges which motivated us to explore TL. We provide an overview of the training approaches and their chances in load forecasting, and argue for our selection.

3.1 Building Load Forecasting Tasks

Many researchers have explored the possibility of saving energy or money by increasing the accuracy of energy-consumption forecasting. Xu et al. (2018) show the potential in a residential setting. More than 20 years ago, Hobbs et al. (1998) evaluated the potential of an NN in multiple industrial settings.

Although no consistent distinctions are made, three overall categories or tasks are given. *Short-term forecasting*, which looks at seconds up to multiple hours to days ahead, is especially important for managing the electricity grid by, e.g. activating storage or managing wind farms. Although individual loads do not have a significant influence, aggregating multiple thousand loads can be a sensible solution to accurately predict the overall load. *Mid-term forecasting*, which predicts multiple days up to multiple weeks, can be important for maintenance planning and managing long-term storage solutions. *Long-term forecasting*, which looks at multiple weeks up to potentially multiple years, can have a significant impact on the general planning and organisation of electricity infrastructure (Al-Hamadi & Soliman, 2005; Chan et al., 2019; Fu et al., 2022).

Buildings can be classified into different categories; private households can be subdivided by metadata, such as square meters and insulation. Industrial, public buildings and offices are other categories. All categories contained in *Building Data Genome Dataset 2* are shown in Fig. 3.1. Using building metadata can be challenging, as many categories contain more than 50% of missing values. This is one additional argument to only focus on the target series and omitting covariates.

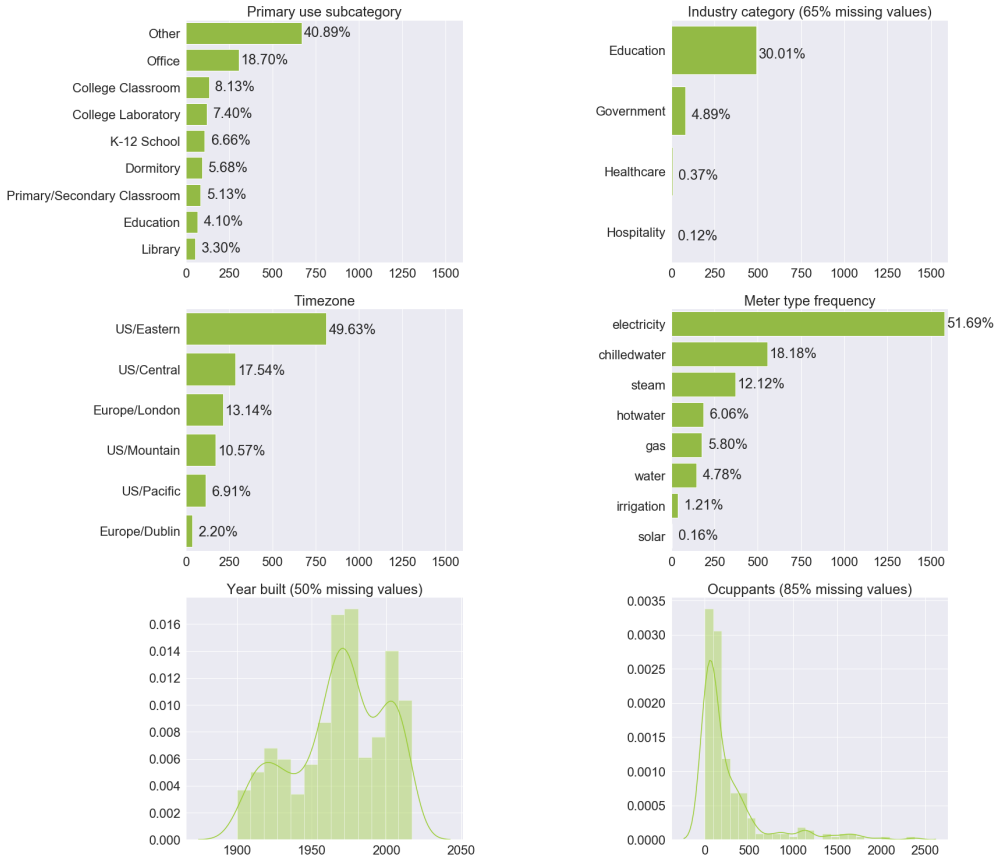

Figure 3.1: Building metadata for GP2 dataset (Miller et al., 2020b)

Within the realm of building electric load forecasting, typical domains include appliance, household, or building-level load forecasting. Other domains can be grid- or country-level forecasts (Kamalov et al., 2024). A typical statistical solution for building load forecasting is ARIMA, which combines multiple statistical approaches (Jiang et al., 2022; Lee & Rhee, 2021). Typical unsupervised methods used are Support Vector Regression (SVR) (Grolinger et al., 2016) and k-nearest neighbour (KNN) (Luo et al., 2022).

Challenges

Despite their significant potential, several challenges must be addressed for real-world application. Modelling highly complex relationships requires training data that spans at least one complete seasonal cycle, with multiple years of data potentially enhancing the accuracy.

Increased installation of smart meters across Europe due to new regulations ("Directive (EU) 2019/944", 2022) underscores the need for innovative forecasting solutions. Managing thousands of statistical models and preprocessing rich covariate data for forecasting can become labour-intensive tasks. Newly installed measurement infrastructures often lack historical data, making effective TL approaches essential. Pre-trained models can facilitate predictions from the first day of operation and potentially enhance the predictive performance over subsequent months or years.

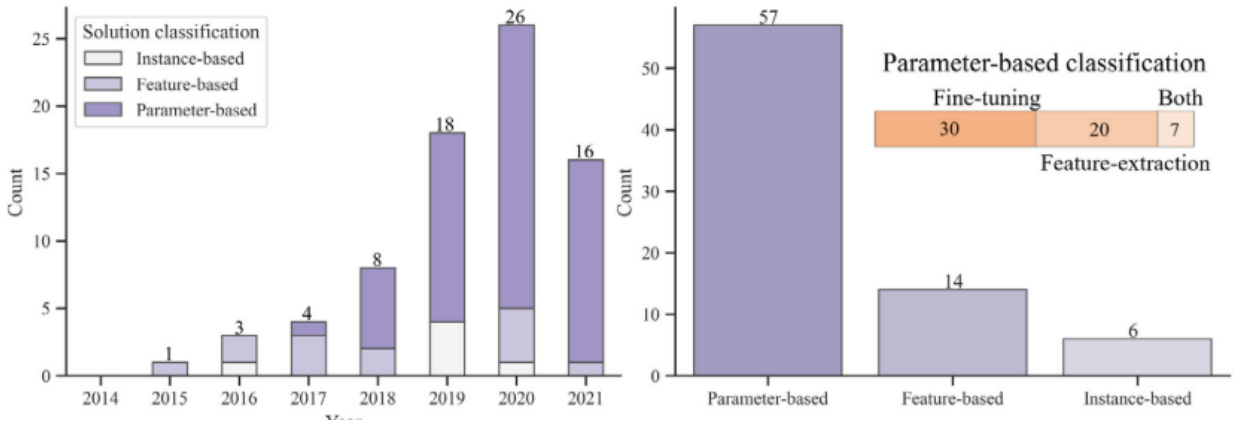

Figure 3.2: TL solutions: (left) over the years; (right) by solution type (Pinto et al., 2022)

3.2 Transfer-Learning Strategies

In load forecasting, inductive TL can be applied by training a model on multiple households from one city and using this model to predict household loads in a different city. The goal is to effectively generalise from the source domain to any new data in the target domain (Fan et al., 2020). The most reliable improvement from TL was demonstrated by the short but complete data input (Fan et al., 2020).

Gunduz et al. (2023b) evaluated TL methods for electricity price forecasting. They identified four TL modes using an MLP architecture. These are integrated, pre-train-fine-tuned, multi-task-learned, and pre-trained-only models, as shown in Fig. 3.3. Overall, the fine-tuned model outperformed all other implementations, but sometimes not on all evaluation metrics (Gunduz et al., 2023b). For this reason, we selected the fine-tuning approach. Because of the recent relevance of zero-shot capability, especially with pre-trained models, this approach is also selected.

The pre-trained fine-tuning approach was extensively evaluated by Fan et al. (2020). They split the dataset using its unique IDs, using 80% of the time-series for training and 20% for fine-tuning. This intra-dataset TL was implemented in our research as an initial and successful sanity check but is only presented in the accompanying repository.

3.3 Transfer-Learning in Building Load Forecasting

Univariate or local TL has been the go-to load-forecasting solution for a long time. Training a single model from one or multiple source domains and fine-tuning the model to a specific time-series. Lee and Rhee (2021) evaluated models like LSTM, MLP, and Seq2Seq on their TL capabilities, but chose an individualised approach using a one-for-all model. A ResNet model was used to predict residential electric load and showed that it performed better with TL (Zhang et al., 2022). For day-ahead electricity price forecasting, the positive influence of covariates was demonstrated, and improvements through TL were proven in multiple settings (Gunduz et al., 2023b).

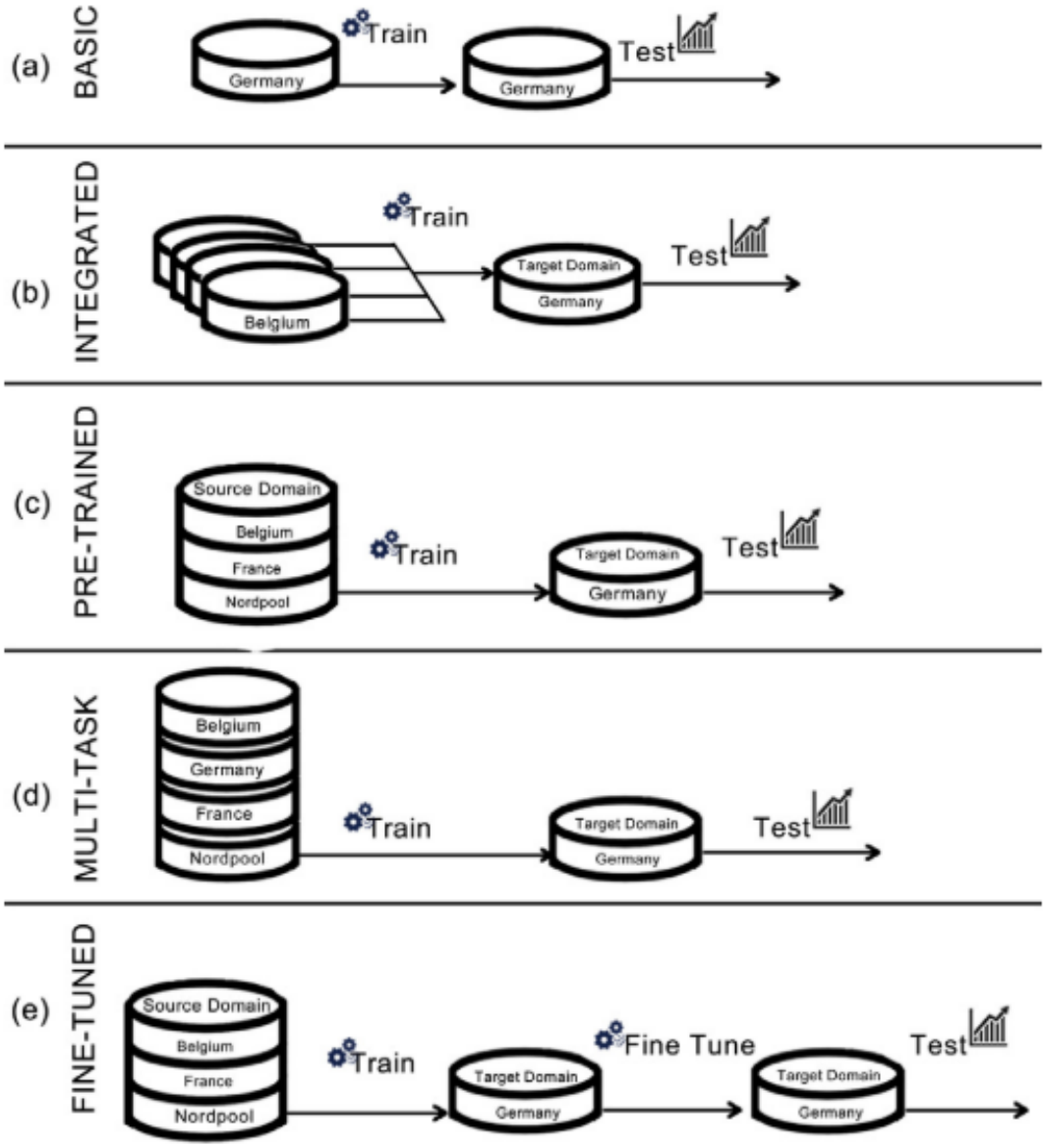

Figure 3.3: Different TL strategies (Gunduz et al., 2023b)

Statistical methods, but also CI DL models, prohibit the use of multi-target correlations for forecasting because of their architecture. All RNNs, such as gated residual network (GRN) or their widely used LSTM subtype, require iterative processing of data, preventing parallelisation and efficient implementation of multivariate DL models (L'heureux et al., 2022).

Although SOTA DL models are available as multivariate implementations, load forecasting continues to focus on local TL (Kamalov et al., 2024). While multivariate TL and even pre-trained (foundation) models (Liang et al., 2024; Ma et al., 2023) have recently become a highly discussed topic in DL research, implementations focusing on the electricity forecasting domain are still sparse.

Pinto et al. (2022) present an overview of TL implementations in smart buildings. They found 25 papers focusing on TL for building-specific electric load forecasting. Most studies overlapped with the relevant literature found in our exploratory literature analysis. Regression is the most frequent task before classification. Most studies evaluate parameter-based TL strategies, this conclusion was also made by Weber et al. (2021) and visualised more specifically in Fig. 3.2. Typical architectures include MLP, LSTM (Lee & Rhee, 2021), GRN, CNN (Voß et al., 2018), and combinations of these, as building blocks (Wei et al., 2024). Recent research has incorporated attention-based NN into load-forecasting models (Li, Xiao, et al., 2021).

No evaluation using big and multi-target building load datasets can be found, even though many researchers use covariates (Kamalov et al., 2024; Laitsos et al., 2023). Wang et al. (2023) are unclear about their inference setting. The Electronc Load Diagram 2011-2014 (ELD) dataset is a widely used benchmark dataset for multivariate and CD training but is not specifically used for building-specific TL. Li, Xiao, et al. (2021) compared a univariate and a multivariate model, although only focussing on covariates such as time-of-day and weather data and ignoring target correlations.

Source and target domain similarities are key topics, and measures such as Wasserstein distance (WD) (Wei et al., 2024) or Dynamic Time Warping (DTW) (Ye & Dai, 2021) are used to determine suitable source data.

3.4 Metrics

Forecasting tasks are typically evaluated using held-out test datasets. Common metrics include the mean average error (MAE) and mean squared error (MSE) losses. In electric-load-specific research, these values are sometimes presented in actual Megawatt (MW), Megawatt nhours (MWh), or similar units, and MAPE is often used for better comparability. More theoretical research has focused on the normalised values of actual data to enhance comparability.

Pinto et al. (2022) examined potential use cases for TL in the smart building domain. They conducted

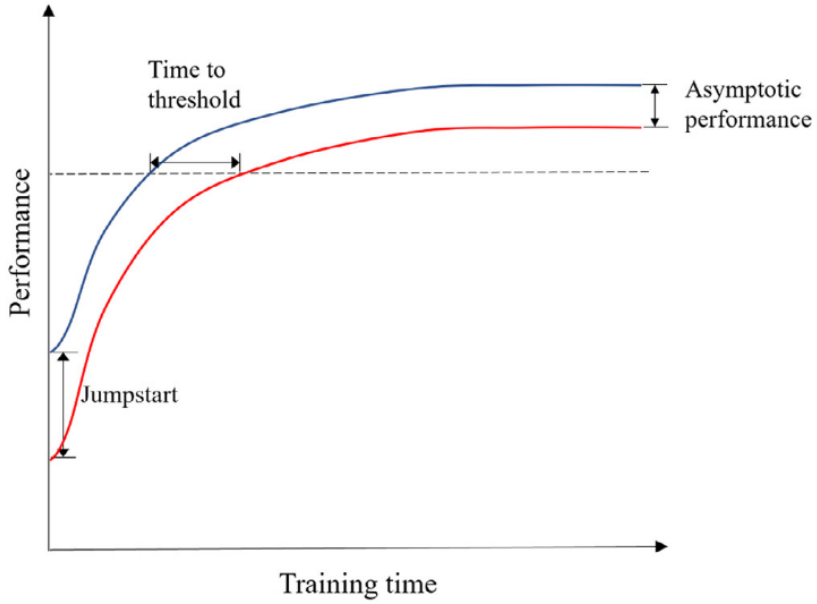

Figure 3.4: Widely used TL metrics (Pinto et al., 2022)

an extensive evaluation of research in this field and identified three useful metrics for evaluating TL: “Jumpstart”, “Time to Threshold” and “Asymptotic Performance”. The metrics are shown in Fig. 3.4.

"Jumpstart“ assesses the initial performance boost on a new task using transferred knowledge. “Time to Threshold” measures the speed at which a performance benchmark is reached with TL, as opposed to learning from scratch. “Asymptotic Performance” gauges the ultimate performance level on a new task with the aid of TL. These metrics quantify the early gains, learning efficiency, and potential long-term benefits (Pinto et al., 2022). The formulae are specified in the Methodology - Evaluation Metrics section 4.2.1.

Challenges

Typical challenges include avoiding negative TL, as shown through sparse data input by Fan et al. (2020). This is often caused by a high statistical deviation between the source and target data, which is the second major challenge for TL. This disparity between datasets makes source data selection an important task for a successful TL. This has been widely discussed in research but exceeds the scope of this thesis.

Pre-Trained Models

Of the many pre-trained models published in 2023 and 2024, only one explicitly focuses on load forecasting. Wang et al. (2023) published DiffLoad to forecast future loads with the ability to quantify uncertainty. Model-intrinsic uncertainties and dataset noise were quantified. They evaluated three

multivariate datasets, one of which was also used in our research; however, the local or global training setting is unclear.

Global CD TL applied to the domain of electric load forecasting has only been researched by Wang et al. (2023). While multivariate TL for time-series forecasting has recently received growing attention, its specific application to the electricity domain lacks attention. With the novel foundation models introduced recently, the usability and cost of inference are important factors to be determined.

Conclusion

In this section, we reviewed significant research concerning building electric load forecasting, particularly emphasising the potential of multivariate forecasting and TL to address the associated challenges. We discussed the pressing global need for effective load forecasting because of its direct impact on CO2 emissions and energy efficiency. Different forecasting horizons present unique challenges and are crucial for various operational and planning purposes. We explored DL and TL approaches, which offer robust alternatives to traditional statistical methods due to their ability to model complex nonlinearities and handle large multivariate datasets. By adopting proven TL settings and exploring innovative models, we aim to advance the capabilities of load forecasting systems, making them more accurate and economically efficient.

4 Methodology

In this section, the research design and reasoning for the experiments is presented. An overview of the datasets and our processing is provided. The forecasting task and evaluation metrics are defined. We explain the baseline and benchmark models used, why they are selected, define the different TL setups and our case study. The key part of this research is to train all models on a source dataset and evaluate them using two different target datasets using zero-shot inference and inference after fine-tuning. The results are compared with those of the same model trained only on the target dataset.

4.1 Datasets

Three datasets are employed. ELD as a homogeneous and well-established benchmark dataset (Trindade, 2015). A dataset collected in southern Germany referred to as “Bavaria”, has fewer and more diverse series from room, appliance, and machine levels while incorporating solar load and public buildings (“Data Package Household Data. Version 2020-04-15.” 2020). Lastly, the “Building Genome Project 2” dataset, published by Miller et al. (2020b), consists of more unique series and contains different types of buildings. This dataset was used by Wang et al. (2023) to train the pre-trained model DiffLoad. All data is available in hourly format or resampled. A summary of key dataset metrics can be found in Tab. 1.

Dataset	Timesteps	Series	Mean Test	SD Test
ELD	8,761	348	0.09	1.06
Bavaria	14,142	59	2.59	0.63
GP2	14,621	1,454	−0.09	1.18

Table 1: Dataset metrics after cleaning.

The first dataset **ELD** (Trindade, 2015) consists of 371 households' electricity load time-series over multiple years. They are resampled at an hourly frequency. The data is cleaned, and series with too many missing values are dropped. We end up with 348 series. The training, testing, and validation horizons are chosen according to recent publications (Chen et al., 2023; Lim et al., 2019; Liu et al., 2023).

The second dataset **Bavaria** ("Data Package Household Data. Version 2020-04-15." 2020) contains multiple industrial buildings, residential buildings, public buildings, and sub-meters, which are machine- or household-equipment-specific. A dataset that was resampled to an hourly frequency is provided. First, columns containing more than 70 % of missing values are dropped. Second, the start and end points of all series are removed if they contain more than 20% of missing values. All the remaining missing values are set to zero. The dataset still contains solar-based energy production, because the overall consumption does not include the energy consumed from solar generation, this variable is retained. No prevalent train-validation-test-split can be found in research, therefore we select a generic split using 70% of the data for training, 10% for validating and 20% of the data for evaluation.

Third, we chose a readily available dataset, **Building Data Genome Project 2 (GP2)** (Miller et al., 2020b). It contains data from 3,053 energy meters in 1,636 buildings. It contains hourly measurements from 2016 and 2017. The measured data is electricity, heating and cooling water, steam, and irrigation meters. In this study, we only examine the electrical data. The cleaned electric load data can be found in the accompanying repository (Miller et al., 2020a). Miller et al. (2020b) explore the diversity of the dataset extensively. Different time zones are present, and different industry categories and multiple primary use and sub-use categories are classified in the metadata, as visualised in 3.1. The authors implemented breakout detection and showed that irregularities, outliers, and use changes can be observed in some IDs, particularly at later timestamps in the dataset. We find a significant increase in outliers after approximately 12,000 time steps. Variable 707 causes a significant statistical deviation between the train and target split, and is therefore dropped. Because no prevalent train-validation-test-split can be found in research, we select the same split as for the Bavaria dataset.

Processing

Z-score **normalisation** is implemented for each unique series. The training split is normalised to a mean of zero and a standard deviation of one, and the resulting normalisation values are used to normalise the validation and test splits. The statistical differences between the training and test splits are presented in Tab. 1. Particularly for the Bavaria dataset, a strong deviation can be observed.

One goal of this research is omitting extensive **feature engineering**. Only for the ARIMA baseline 24, 48, 72, and 96-hour lags are created. The lags are time-shifted variates, which give information on what the target values was the previous days at the same time. The time of day is encoded using sine and cosine functions.

The data is transformed into a **sliding window dataloader**, as described by Gunduz et al. (2023b), to be used for training and inference. For the iTransformer, TimeGPT, and ARIMA models, a custom function is implemented. For the models implemented in the Darts library, NHiTS, TSMixer and Transformer, a proprietary class is provided. Input and target sequences with a length of 96 are created according to previous benchmarks (Liu et al., 2023; Pinto et al., 2022). The features for each channel are defined as $t_0, t_1, \ldots, t_{95}$ and the targets are $t_{96}, t_{97}, \ldots, t_{191}$. To train the DL models, several overlapping input and output windows are created. For inference, TimeGPT and ARIMA do not have overlapping horizons. The case study does not use any overlapping horizons because we are working with absolute values.

A **short training horizon** of four weeks is selected for baseline comparisons and a short fine-tuning horizon. Models need to engage with multiple seasonal cycles, literature suggests that even a minimum of 100 data points can be sufficient (Smith et al., 2017). In our setting, sub-daily, daily and weekly cycles can be captured while monthly and yearly cycles are not.

Four weeks is a brief timespan compared to the time required to operationalise new buildings. This horizon balances the need for actionable data with sufficient horizons shown in previous research and the ramp-up period of new buildings, during which usage patterns evolve. Fan et al. (2020) demonstrated that even two months of data are sufficient for extracting underlying structures, showing slight improvements in transfer settings, as shown in Fig. 4.1.

4.2 Forecasting Task

The forecasting task is defined according to established benchmarks, taking 96 h as the input and forecasting the following 96 h (Lim et al., 2019; Liu et al., 2023; Nie et al., 2022). This is performed for all target series simultaneously, except for the ARIMA model, which needs to be executed iteratively.

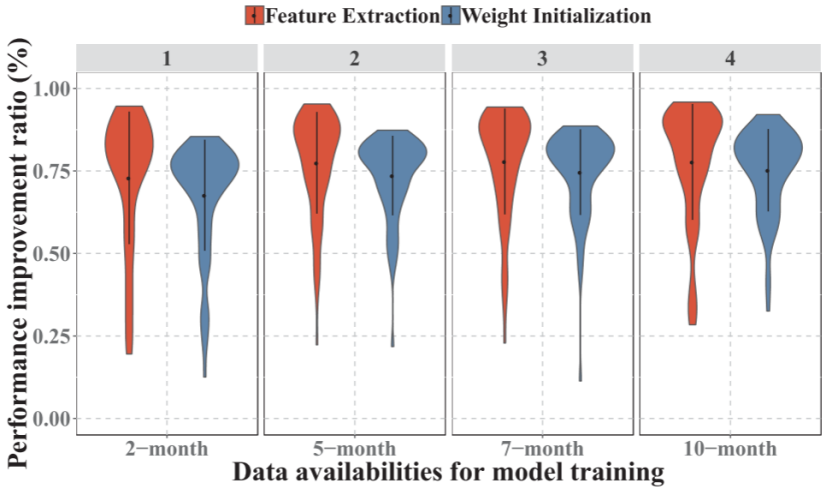

Figure 4.1: Improvement observed for different data input for fine-tuning, separated by horizon length (Fan et al., 2020)

4.2.1 Evaluation Metrics

According to previous research, MSE and MAE are provided as metrics. Because the MSE is used for model optimisation, it is also used as the key metric to evaluate all models. As section Metrics in Related Works explains, TL performance will also be assessed based on the jumpstart and asymptotic performance metrics. These metrics are calculated from the MSE losses calculated over the test set. The metrics are defined in Tab.2. Granular tracking of the training metrics was not feasible due to the number of experiments.

Metric	Definition
Jumpstart Zero-Shot	$\text{Jumpstart}_{\text{zero-shot},m} = \frac{\text{MSE}_{\text{short training},m}}{\text{MSE}_{\text{zero-shot},m}} - 1$
Jumpstart Fine-Tuning	$\text{Jumpstart}_{\text{fine-tuning},m} = \frac{\text{MSE}_{\text{short training},m}}{\text{MSE}_{\text{short fine-tuning},m}} - 1$
Asymptotic Performance	$\text{Asymptotic performance}_{m} = \frac{\text{MSE}_{\text{full training},m}}{\text{MSE}_{\text{full fine-tuning},m}} - 1$

Table 2: Definitions of Jumpstart and Asymptotic Performance Metrics measured for each source-target dataset combination and model m.

4.3 Model Description

ARIMA is used as a statistical baseline because it is one of the most widely used models that typically performs quite well within the scope of statistical models. The univariate models are fitted on the last

2,000 time steps of the training set and used to predict the test set without re-fitting. The proprietary pre-trained TimeGPT model is used as a pre-trained foundation model baseline. **TimeGPT** has a transformer-based encoder-decoder architecture and is trained using masking with multiple billion data points. It is only accessible via the proprietary API. All three test splits of the multivariate datasets are split into 96-hour input and 96-hour target subsets and predicted iteratively using the long-forecast version of TimeGPT (Garza & Mergenthaler-Canseco, 2023). Nixtla, the company behind TimeGPT, does not specify if the long-forecasting version has a different architecture or is only trained differently, for our horizon of 96 time steps this architecture is suggested.

iTransformer encodes each series and uses attention across those encodings, extracting the information using feed-forward neural networks. Transformer uses the same architecture but encodes each time step instead of each variable.

N-HiTS and **N-BEATs** use similar architecture. N-BEATS is notable for its purely feed-forward architecture that avoids recurrent layers, making it distinct from many traditional DL models for time-series. N-BEATS uses a series of fully connected layers (blocks) that output a forecast directly, along with backcasts used to refine the input for subsequent blocks (Oreshkin et al., 2020). Each block in the model learns a different representation of the data, focusing on different temporal patterns such as trends or seasonality. N-HiTS "attempts to provide better performance at lower computational cost by introducing multi-rate sampling of the inputs and multi-scale interpolation of the outputs" (Herzen et al., 2022).

TSMixer with its stacked MLP architecture is computationally more efficient than Transformers which scales quadratically to the input length (Liang et al., 2024). It is designed to extract features along the time and feature axis and includes gating mechanisms to remove noise. It utilises patching like PatchTST, but merges this with a CD training strategy which utilises interactions between variates (Chen et al., 2023).

Hyperparameter Selection

The parameters are selected according to the values provided in the initial iTransformer paper, for some parameters, a range is given; therefore, we choose values in the middle of the range provided by Liu et al. (2023). All other model parameters are selected accordingly to use a similar number of trainable parameters between the models. All hidden dimensions have a width of 256, and the depth for all blocks is set to two, as most experiments in iTransformer research are found in the accompanying repository (Liu et al., 2023). The dropout values are not provided, we select the same dropout to the DiffLoad model, which is set to 0.25 (Wang et al., 2023). Depending on the channel number, the models range between 500,000 for the Bavaria dataset and a maximum of 250 million trainable parameters for the GP2 dataset. Because only iTransformer can be reshaped to the necessary channel number, the number of trainable parameters is changed between training and

fine-tuning or inference. The overall goal is to keep the number of trainable parameters within a similar range for each dataset.

ARIMAs parameters are selected by the AutoArima function provided in the Python implementation, with the opportunity to try three optimisation settings (Smith et al., 2017).

The API of TimeGPT does not offer specific parameter tuning, the only parameter is "finetuning-steps", a specific loss function and the model to be used ("Foundational Time Series Model (Beta)", 2023).

Training Parameters

No specific optimisation is implemented. All the models are trained using MSE loss. The learning rate is set to 0.0005 for all models, in line with the original iTransformer paper, and within a similar range as generic implementations provided by Darts, which are usually 0.001. As in the original paper, we use a batch size for training of 32. Values in the Darts documentation and other research range between 16 and 64.

4.4 Experiments

4.4.1 Training Procedure

Prior to our main experiments, we validated our approach using the iTransformer on the ETL dataset, replicating the experiments from the original paper (Liu et al., 2023). We achieved comparable results across multiple forecasting horizons, thus confirming the reliability of the implementation used for this thesis. We also conducted intra-dataset TL validations across all three datasets, employing the iTransformer model configured in a 5-fold cross-validation scheme, utilising 80% of the data for training and 20% for testing to ensure model robustness. The results vary significantly depending on the split but are in a range comparable to the overall results. While not presented in this thesis, the code for the experiments can be found in the accompanying repository. Based on the findings of Ye and Dai (2021), forecasting research does not sufficiently focus on the cross-dataset TL. We propose our main experiment to explore various training strategies to assess the efficacy of TL models under different conditions across the datasets. The small training subsets are the previously mentioned short training horizon of four weeks. Each strategy was applied to every model as follows:

Baseline strategies: only training on the target dataset

- **Direct Training:** Models are trained directly on the entire target training dataset.
- **Subset Training:** Training occurs on a smaller subset of the target training dataset.

TL strategies: all models are pre-trained on the train split of the source dataset:

- **Zero-Shot Transfer:** Models are directly evaluated on the target test dataset split.
- **Partial Fine-Tuning:** Models are fine-tuned for 10 epochs on a small subset of the target training dataset and evaluated on the target test set.
- **Full Fine-Tuning:** Models are fine-tuned for 5 epochs on the entire target training dataset and evaluated on the target test set.

For each target dataset, the other two datasets are used as source datasets; therefore, the TL experiments are executed six times. All baseline models are trained for 15 epochs, which is in a range comparable to previous research, with 50% more epochs to ensure convergence is reached (Liu et al., 2023). The best model is selected based on the lowest MSE of the validation split. All fine-tuning is executed without a validation split based on the assumed data scarcity. Small fine-tuning is done for ten epochs to provide the models with sufficient time for convergence, which, according to some initial experiments, takes longer with less data. Five epochs are selected for full fine-tuning, which is one-third of the initial training.

4.4.2 Case Study

To strengthen the practical relevance of our research, we compare the performance of the pre-trained models against the non-TL training strategy. We use the GP2 dataset, which includes a variety of building types and the most series, to simulate the first twelve months of forecasting for the new measuring infrastructure. For simplicity, each month is represented as four weeks. In each data split, the first three days are used exclusively as input data, with the subsequent values used for forecasting. Once a data split is forecasted, it becomes part of the training data for future predictions.

Initial experiments indicated that retraining only on the next four-week horizon caused significant fluctuations. Therefore, we opted for a more robust approach: training or retraining on all available past data.

As a **baseline**, the first month is predicted using data from the previous three days which is shifted by three days. For subsequent months, an iTransformer model is instantiated and trained on all past data using the same parameters as in the previously explained experiments.

For the **first TL benchmark**, the initial month is predicted using a model pre-trained on the ELD dataset, performing zero-shot inference with the same three-day input and output structure. For the next eleven splits, the pre-trained model is fine-tuned on all available past data and trained for three epochs with a learning rate of 0.0005.

The **second benchmark** follows a similar approach but involves re-instantiating the pre-trained model for each month. This model is then trained on all past data for five epochs.

We evaluate the performance using the sum of the MAE of the Gigawatt hours (GWh) predictions compared to the actual values. We select the better-performing TL approach and compare it with the baseline method.

4.4.3 Implementation

All experiments are implemented in Python and run on a university-provided Linux server using a Tesla V100 GPU with 32GB of VRAM. The iTransformer model is taken from GitHub (Wang, 2023), provided by Phil Wang. The ARIMA model is a widely used Python implementation (Smith et al., 2017). TimeGPT is used via the proprietary API, and all other models are trained using the Darts library (Herzen et al., 2022). Fitting the ARIMA models was the most time-intensive, running over three days on a CPU. Depending on the dataset size, fitting the DL models took a couple of minutes to an hour. The inference time was not measured.

We can reshape the pre-trained iTransformer model channel number and normalisation parameters according to the target series for TL. The Darts library does not provide any information on how to reshape the model according to the needs of the target series, although a multivariate TL application is presented (Herzen et al., 2020). Exploratory trials for reshaping are difficult using the high-level Darts library, which only offers the fit method and remains unsuccessful. We decided to extend the source dataset to the number of series of the target dataset, duplicating multiple series but making it possible to perform multivariate and CD inferences on the target set. If the target dataset requires fewer channels, the model is trained using a subset of the source series.

Initially, we set the RevIn parameters β and γ of the iTransformer to zero and one after source training. Because of the sub-par performance on the Bavaria dataset compared to the Darts models, we decided to re-use the pre-trained parameters by only reshaping according to the target dataset channel number. Although not random, no specific selection was performed.

5 Results

5.1 Baseline Results

Before looking at the results of the DL models in regular and TL settings, we define some baselines on which predictive results can be expected for all three datasets. ARIMA is consistently outperformed by TimeGPT, with a magnitude between one and two. The MSE of the ARIMA model is approximately 0.3 higher than the MAE for the ELD and Bavaria datasets, leading to the assumption of a similar influence of outliers, which skews the MSE more than the MAE. For

the GP2 dataset, we can see a significantly higher MSE, leading us to assume that ARIMA is less capable of modelling outlier values for this dataset or is too sensitive. TimeGPT also seems to struggle more with outlier values on the GP2 dataset, whereas almost no deviation between MSE and MAE can be found for Bavaria. TimeGPT outperforms all known benchmarks on GP2 and ELD while underperforming on the Bavaria dataset and is only better than iTransformer, as shown in Tab. A.12.

Table 3: Normalised ARIMA results.

Dataset	MSE	MAE
ELD	1.012	0.709
Bavaria	1.265	0.916
GP2	1.695	0.941

Table 4: Normalised TimeGPT results.

Dataset	MSE	MAE
ELD	0.067	0.037
Bavaria	0.023	0.022
GP2	0.083	0.037

5.2 Overall Results

For a first overview, we show average results for all source-target scenarios in Tab. 5. Although not ideal because of different ranges for dataset-specific losses, it is still useful to gain some initial insights. iTransformer outperforms all benchmark models by a significant margin when looking at the *Overall Mean*. For *Short TL* and *Short Training*, each model has similar values; however, NHiTS deteriorates from TL. The same can be found when looking at *Full TL* and *Full Training*; overall, no improvements can be made. This shows that TL is not a silver bullet for improving the predictive performance without rigorous selection and training. iTransformer is the only model that improvements in at least two out of three settings, *Jumpstart-TL* and *Asymptotic Performance* improvements are between 0.67% and 1.97%, respectively.

Table 5: Mean MSE for all source-target combinations per Model-Learning Scenario and overall TL metrics per model. Positive values are an improvement.

Learning Scenario	NHiTS	Transformer	TSMixer	iTransformer
Zero-Shot	4.428	3.371	3.470	1.162
Short-TL	0.904	0.662	0.421	0.279
Full-TL	0.562	0.654	0.303	0.219
Short training	0.459	0.666	0.402	0.284
Full training	0.400	0.641	0.269	0.220
Overall Mean [MSE]	1.351	1.199	0.973	0.433
Mean Jumpstart Zero-Shot [%]	−89.634	−80.253	−88.427	−75.542
Mean Jumpstart TL [%]	−49.199	0.514	−4.542	1.974
Mean Asymptotic Perf. [%]	−28.873	−2.018	−11.015	0.671

Jumpstart Zero-Shot shows that no improvements can be made compared to any baseline. The metric is in a comparable range for all models, showing the overall reliability of the experiment,

the metric, and that the models manage to capture the underlying information of unseen data. The iTransformer performs best for this metric. Even though NHiTS and Transformer have comparable values with 89.6% and 80.3% higher MSE, respectively, their baseline performance is already significantly worse.

From initial experiments, we observed NHiTS and Transformer to be less reliable than the other models, sometimes diverging to local minima and sometimes even failing to capture the underlying information within the data. Furthermore, selecting the best-performing model on the source validation split significantly decreases the TSMixer's performance. For consistency, we selected the best model based on the validation MSE as the first indicator for TL performance.

Table 6: Results for ELD test set [*MSE*].

Source	Learning scenario	NHiTS	Transformer	TSMixer	iTransformer	TimeGPT	ARIMA
Bavaria	Zero-Shot	15.634	10.373	9.903	2.736	0.067	
Bavaria	Short TL	2.587	0.961	0.481	0.337		
Bavaria	Full TL	0.951	0.978	0.308	0.211		
GP2	Zero-Shot	0.693	0.957	0.983	0.450	0.067	
GP2	Short TL	0.763	0.952	0.362	0.221		
GP2	Full TL	0.470	0.947	0.218	0.180		
ELD	Short Training	0.408	0.954	0.352	0.265		
ELD	Full Training	0.345	0.947	0.212	0.182		
Overall Mean	All Experiments	2.731	2.134	1.602	0.573	0.067	1.012

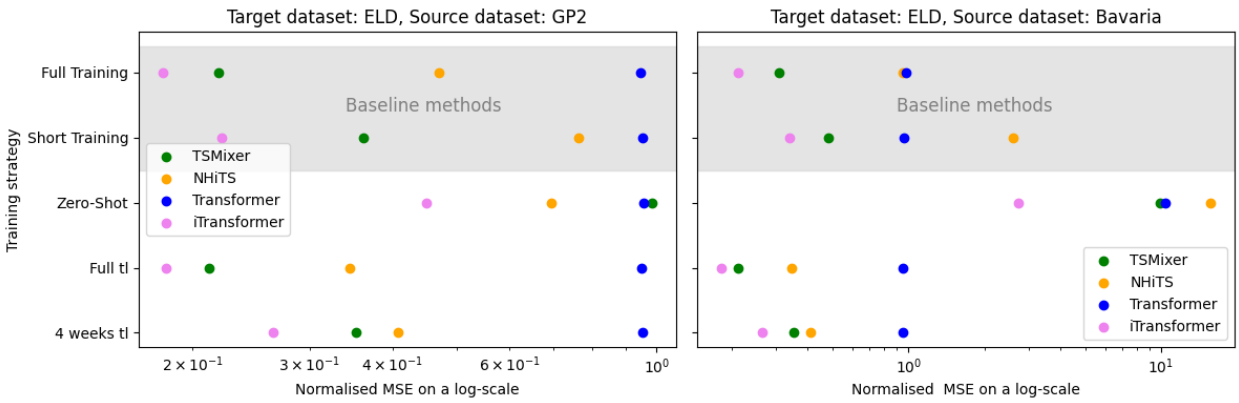

Figure 5.1: Results on ELD test sets with different source sets.

This section presents the results on the **target dataset ELD** while ignoring the Transformer results, as it exhibited unusual behaviour, sometimes managed to converge, but overall gives similar results for almost all *Learning Scenarios*. Evaluating the experiments using *ELD* as the target, as presented in Tab. 6, the *Zero-Shot* setup is consistently outperformed by the *Short TL*, which in turn is always outperformed by the *Full TL* setup. The iTransformer achieves the best results by a significant margin and is outperformed only by TimeGPT by over 50%. TSMixer is the third-best model. All

models outperform ARIMA, except for the zero-shot setting when using Bavaria as the source dataset. GP2 enables all models to improve their predictions in *Zero-Shot* and *Short TL* settings compared to using Bavaria as a source. Only the iTransformer can improve from its baselines using TL when comparing the short baseline with *Short TL* and the full baseline with the *Full TL*. Fig. 5.1 visualises the results while excluding TimeGPT, as it skews the scale and makes it difficult to distinguish other results.

Table 7: Results for GP2 test set [*MSE*].

Source	Learning scenario	NHiTS	Transformer	TSMixer	iTransformer	TimeGPT	ARIMA
Bavaria	Zero-Shot	9.312	7.905	8.562	2.493	0.083	
Bavaria	Short TL	1.035	1.071	0.862	0.604		
Bavaria	Full TL	0.975	1.025	0.710	0.468		
ELD	Zero-Shot	0.927	0.987	1.370	0.866	0.083	
ELD	Short TL	1.035	0.987	0.818	0.508		
ELD	Full TL	0.976	0.976	0.578	0.451		
GP2	Short Training	0.969	1.042	0.852	0.587		
GP2	Full Training	0.855	0.976	0.596	0.478		
Overall Mean	All Experiments	2.011	1.871	1.794	0.807	0.083	1.695

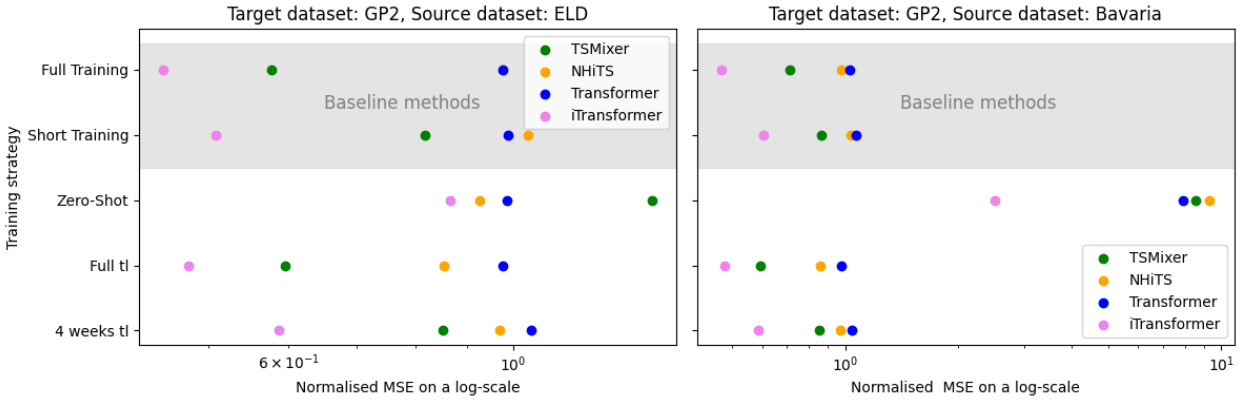

Figure 5.2: Results on GP2 test sets with different source sets.

The results for **target dataset GP2** are presented in Tab. 7. Using Bavaria as a source dataset prevents all models from making accurate zero-shot predictions, as observed for ELD. TimeGPT outperforms all the predictions by at least a factor of five. iTransformer is again the second-best model, and TSMixer is the third-best. When using ELD as a source dataset, both TSMixer and iTransformer improve their results compared to the respective baselines. The results are shown in Fig. 5.2, excluding TimeGPT, which is difficult to display because of its significantly better performance.

Target dataset Bavaria results are scaled to be in an easy-to-read format [see Tab. 8]. This is the only target dataset where TimeGPT is outperformed by many models (besides the iTransformer)

by a significant margin. Depending on the model, ELD or GP2, as a source dataset, offers more accurate predictions for the test set. iTransformer *Zero-Shot* predictions are far from the results exhibited for the two prior target datasets. An in-depth discussion of the potential influence of RevIn normalisation is presented in the following chapter. Transformer and TSMixer improve in some settings because of TL. Interestingly, iTransformer performs the best after *Full TL*, making the significant difference between *Zero-Shot* and *Full TL* an interesting topic for further evaluation. The results are shown in Fig. 5.3. Another finding that underlines the unusual nature of the Bavaria dataset is that the MSE in many cases is lower than the MAE, caused by values smaller than one decreasing by being squared, whereas the MAE does not change [see Tab. A.12].

Table 8: Results for Bavaria dataset (all values scaled by 10^3) [*MSE*].

Source	Learning scenario	NHiTS	Transformer	TSMixer	iTransformer	TimeGPT	ARIMA
ELD	Zero-Shot	1.543	1.627	1.782	195.043	22.917	
ELD	Short TL	0.521	1.654	1.054	0.837		
ELD	Full TL	0.326	0.314	0.356	0.255		
GP2	Zero-Shot	1.811	1.673	1.722	232.959	22.917	
GP2	Short TL	0.468	0.489	0.509	0.423		
GP2	Full TL	0.318	0.314	0.269	0.309		
Bavaria	Short Training	0.341	0.495	0.502	1.315		
Bavaria	Full Training	0.278	0.315	0.287	0.239		
Overall Mean	All Experiments	0.701	0.860	0.810	53.923	22.917	1,265.209

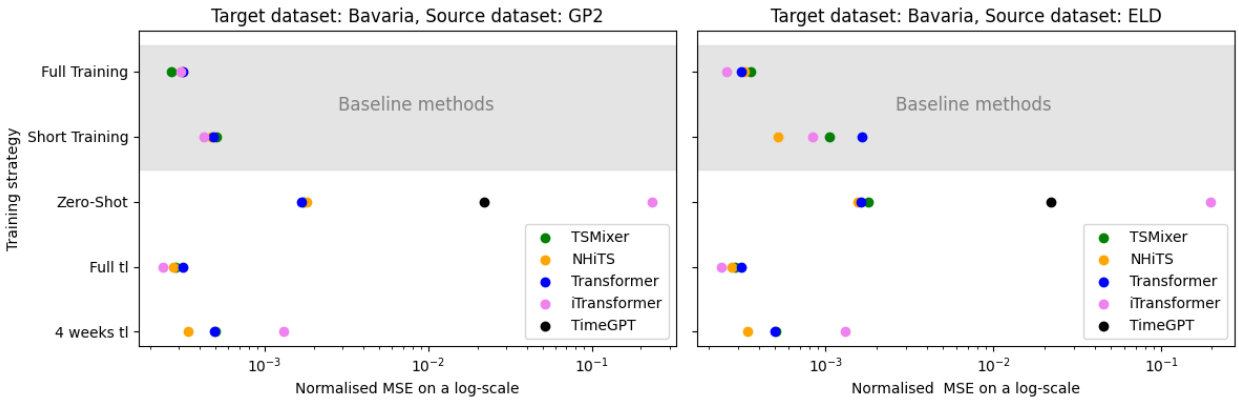

Figure 5.3: Results on Bavaria test sets with different source sets

5.3 Transfer-Learning Metrics

In the following section, we discuss the time-series specific TL metrics more closely. Because of the rather unreliable results from using Bavaria as a source or target, we focus specifically on the transfer between the GP2 and ELD datasets. An evaluation without the Bavaria dataset potentially provides clear insights. Additionally, we select TSMixer and iTransformer as the most robust and best-performing models overall, therefore, we focus on their specific results.

In Tab. 9 we list the specific metrics for each target dataset; the second dataset is therefore used as the source. For *Jumpstart Zero-Shot*, we can see a significant improvement when Bavaria is excluded. For the iTransformer, the values are only worse between 41% and 32%, which are impressive results when reflecting that the data has never been seen before during training. Slightly worse results are obtained for TSMixer. For *Jumpstart Short TL*, in three out of four cases, improvements can be made between 4.2% and 19.5%. The only case with a slight deterioration for TSMixer is 2.7%.

For *Asymptotic Performance*, improvements can be made in many cases. Only the TSMixer evaluated on the ELD dataset deteriorates by 3.2%.

Table 9: TL metrics for ELD and GP2, ignoring Bavaria. Positive values are an improvement.

Target	TL Metric	TSMixer	iTransformer
GP2	Jumpstart Zero-Shot	−37.8%	−32.3%
GP2	Jumpstart Short TL	4.2%	15.5%
GP2	Asymptotic Perf.	3.2%	5.8%
ELD	Jumpstart Zero-Shot	−64.2%	−41.2%
ELD	Jumpstart Short TL	−2.7%	19.5%
ELD	Asymptotic Perf.	−3.2%	0.8%

5.4 Case Study

The case study results are shown in Fig. 5.4. For every split, the sum of the MAE is visualised. Zero-shot inference brings a significant boost compared to a base forecast relying only on the last three days of data. With more time elapsed and more available data, the predictive accuracy improved for all the models. The baseline is outperformed by both TL implementations. The multi-model approach is approximately 3% more accurate than the single-model approach, and will be used for further evaluation. The summed MAE of the baseline is 124.0 GWh, for multi-model TL 95.9 GWh and for single-model TL 98.9 GWh.

For all splits, the summed MAE of the best TL approach is 29.3% lower than that of the non-TL approach, this equals 28.1 GWh. Excluding the first month, the improvement is still approximately 10.5% or 8.7 GWh. Excluding the first month, the TL approach enables a predictive improvement equal to 0.5% of the overall electricity consumed during the forecasting horizon. Including the first month, we achieved an improvement of 1.6% compared with the baseline for all electricity consumed. The actual electricity consumed during the forecasting horizon equals 1722 GWh. The values are obtained by dividing the sum of all MAE values by the sum of all the actuals; therefore, they can be described as MAPE.

The previously mentioned annual savings of USD 1.6 Million are estimated for 10GW, which equals 87,600 GWh annually (Hobbs et al., 1999). Our 1,722 GWh is approximately 1.97% of this. From

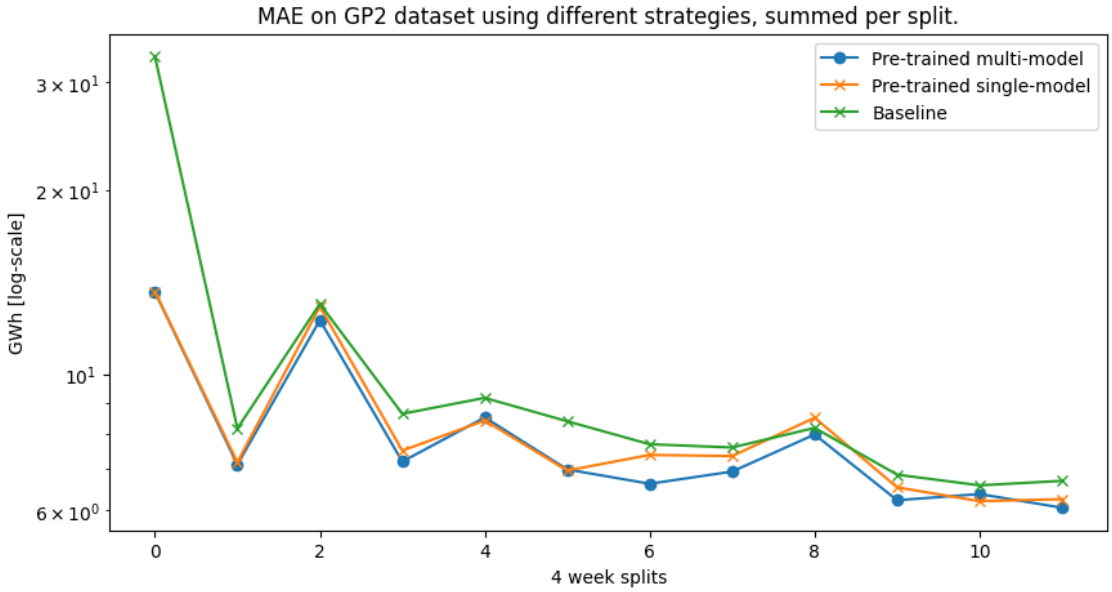

Figure 5.4: Case-study results on a log-scale, summed for each split.

this, we can estimate a saving of approximately USD 31,340 for each per cent point in MAPE improvement. The most optimistic case, a 1.6% improvement compared to the baseline, leads to savings of USD 51,340. Divided by the 1,454 buildings contained in the dataset, this equates to approximately USD 35 savings per measured building per year. Even though the Hobbs et al. (1999) publication is highly recognised, and the values mentioned are conservative estimates, the estimates are based on a time without significant renewable production and electricity storage.

6 Discussion

This section discusses the experimental results and research objectives. Pre-trained task-specific models are compared to foundation models in the building electricity domain, and additional insights derived from the experiments are provided.

6.1 Research Objective

Multiple DL models have been evaluated for answering our research objectives, most settings proved to be more accurate and robust than the application of statistical models without covariates. In our experiments, global multivariate forecasting adeptly substitutes computationally inefficient statistical models without modelling covariates. SOTA performance was achieved by using the iTransformer model. It was not possible to reproduce similar results using the Transformer and N-HiTS models.

The case study showed how the initial year of hourly electric load can be effectively forecasted for more than a thousand buildings. The approach of zero-shot inference in the first month followed by iterative training on all past data each month reliably outperformed a baseline without TL, and

proved to be a viable solution for forecasting the load of buildings without past data on a large scale. The key aspect of achieving reliable and robust forecasts is a carefully chosen source dataset combined with an SOTA model.

We showed that the multivariate CD DL models iTransformer and TSMixer can be reliably trained with four weeks of data, without using entity-specific covariates. The iTransformer showed impressive results when excluding zero-shot inference on the Bavaria dataset. As Transformer and NHiTS performed comparably well on the smaller Bavaria dataset, we are unsure if appropriate parameters were selected for those models using the same parameters on the three datasets with different channel numbers. However, the iTransformer outperformed all other models with the same hidden dimensions, suggesting a higher capability to model highly complex data in our setting.

For the sub-par zero-shot performance of the iTransformer and TimeGPT on the Bavaria test set, there is a plethora of potential failure points. We were unable to determine the most likely cause. Re-using learned RevIn parameters β and γ was tried, but was not influential. We assume the very small overall MSE as a potential reason for this result. Another reason for the strong deviation of the results on the Bavaria dataset could be the strong statistical drift between the training and test sets.

When selecting sensible source datasets, we did not find a significant risk of deteriorating forecasts when employing TL. *Jumpstart Zero-Shot* showed, that iTransformer is only around 32% to 41% worse on unseen data compared to being trained on four weeks' worth of data. TSMixer, as the second-best model, is already significantly worse, with a deterioration of approximately 37% to 64%. The iTransformer improves reliably on all other TL metrics, proving the robustness of RevIn normalisation on novel data, showing the potential of attention for forecasting applied across variates and the possibility of fitting a wide range of datasets, even if the initial performance is sub-par.

N-BEATS did not show robust results, did not learn reliably from small datasets, and was excluded from the experiments. Interestingly, N-HiTS showed a more robust and accurate behaviour, even though its architecture was derived from N-BEATS. The Transformer model also exhibited irregular behaviour when used for inference on the two larger datasets, ELD and GP2. However, when evaluated on the Bavaria dataset, it did show a similar ratio between the different training strategies as the other models. Even though similar hyperparameters to the iTransformer were used, this leads to the assumption, that the Transformers parameter count overall was too small for the two big datasets or not enough data was present to learn reliably. This again underlines the potential of cross-variate attention. Of all the chosen models implemented in the Darts library, TSMixer showed the best performance, albeit still struggling in some settings and not reliably outperforming N-HiTS and Transformer.

Computational efficiency is not a significant reason for TL compared with domains such as computer vision. However, gains in predictive accuracy may be sufficient in specific settings to use time-series

TL in real-world settings. We demonstrated that fine-tuning on all past data results in significantly more robustness than fine-tuning on, for example, the past month.

The option to alter the channel number after training brought significant computational benefits. It enables the option to save pre-trained models that can be universally used for every target dataset and potentially select the best-suited models within seconds. Existing models, as implemented in the Darts library, cannot easily be adapted for universal knowledge transfer to any dataset and need to fit specifically.

6.2 Pre-Trained Models

TimeGPT showed that pre-trained foundation models can yield impressive results. However, an explicit evaluation on unseen data is required. The iTransformer and TSMixer have the potential for TL, which is one of the reasons for their use as building blocks in foundation models. We showed that domain- and task-specific deep neural networks (DNN) can be efficiently and accurately used for novel data, and can be fine-tuned efficiently on more than a thousand series within minutes. The iTransformer is still limited to the specific training task of taking predefined time steps as the input and forecasting them. While this is a decisive limit for a foundation model, a domain-specific pre-trained model can be trained on one specific task and adapted to different entities for the same task. Forecasting less than 20% of the ELD test set using TimeGPT costs around USD 3,000 in March 2024. Inference for the case study using the initial TimeGPT pricing would cost more than the most optimistic potential savings we have found using TL. Nixtla announced a price adjustment, the final pricing is however still unclear. With topics such as data ownership, security, and privacy, sending crucial business or customer data to an external provider is typically unpopular. Selecting locally run, owned, and domain-specific models is feasible and viable. For pre-trained domain-specific models, we conclude that channel flexibility is significantly more important than flexibility in terms of input and output horizons. Predicting the next 24 hours on many buildings is a typical task. Grid operators have to manage a different number of buildings, but many of them need to optimise the same 24-hour forecasting task. We assume, that many domains have specific tasks that need to be optimised.

6.3 Limitations

We first discuss the specific limitations of this thesis and its experiments, and second, the overall limitations found in the research field that should be noted. A noteworthy limitation of this study was the non-specific selection of parameters. While we tried to maintain the total number of parameters within a consistent range across models, this approach did not ensure equivalent computational costs or processing times. Owing to the very different architectures, the overall number of parameters still differed significantly. Furthermore, it was assumed that the TimeGPT model was pre-trained on the available datasets. This assumption is based on the availability of

these datasets online, the outstanding accuracy of TimeGPT and humorously echoes the satirical notion that "pretraining on the test set is all you need" (Schaeffer, 2023). Future research should address this limitation by incorporating additional tests with synthetic or new data to more robustly validate the models. This study ignored the topic of different time-series frequencies. We assume that task- and domain-specific models do not require this capacity, making them smaller, faster, and easier to train. Additionally, it adds another layer of complexity and is beyond the scope of a master's thesis. Next, it is difficult to determine the accuracy of the calculations from Hobbs et al. (1999) because the electric grid has changed tremendously in the last 25 years. Lastly, the inability to adjust the Darts models to the target channel number added some irregularity to the experiments.

An overall limitation concluded from our experiments is that the inability to adjust most model parameters after training is a significant constraint for task-specific multivariate models. The number of channels can only be adjusted for the iTransformer. In contrast, TimeGPT can be used without considering the parameters at all; it adjusts the input and output lengths as well as the channel number.

6.3.1 Future Topics

For further research, a benchmark between CI and CD training strategies implemented for the same architecture would be insightful, especially when merged with a more thorough evaluation of the value-added of RevIn. We find that channel-flexible architectures such as iTransformer and TimeGPT bring meaningful advantages that should be explored further. This is already a relevant topic in pre-trained models. Evaluating these pre-trained foundation models using widely available benchmark datasets is not a sensible approach for future research. Therefore, novel or synthetic data must be used. New research evaluating the saving potential of more accurate forecasting applied to the current state of grid infrastructure would be useful, the highly noted research of Hobbs et al. (1999) is likely outdated. Lastly, adding building-specific covariates to the experiments would give a clear insight into how much value could be added from extensive covariate modelling.

6.4 Conclusion

Global TL for electric-load forecasting is highly effective in scenarios with limited data. CD models can adeptly handle statistical shifts between datasets within the same domain. Our findings demonstrate significant improvements for *Jumpstart TL* of 15% and 19% using the iTransformer model and also show enhancements with TSMixer when a suitable source dataset is employed. Additionally, iTransformers *Asymptotic Performance* is improved using the GP2 and ELD datasets.

In our case study, we observed that the iTransformer, used for *Jumpstart Zero-Shot*, significantly outperformed a simple baseline. This pattern is consistent with experiments in which sensible source data was utilised. The potential cost savings are substantial, with estimates of approximately

USD 35 per building in the first year of operation using TL.

We challenge the common belief that DL models require large amounts of target data to be effective. In our results, all zero-shot settings outperformed statistical baselines. Models trained on only four weeks of data performed 30% to 40% worse than those trained on the target data. We show that task- and domain-specific pre-trained models are a sensible solution for large-scale building electric load forecasting needs. Forecasting multivariate datasets using pre-trained models such as TimeGPT can boost the predictive performance in real-world applications. However, the cost is considerable, with initial pricing estimates for zero-shot predictions in this study reaching approximately USD 50,000.

References

Al-Hamadi, H., & Soliman, S. (2005). Long-term/mid-term electric load forecasting based on short-term correlation and annual growth. *Electric Power Systems Research*, *74*(3), 353–361. https://doi.org/https://doi.org/10.1016/j.epsr.2004.10.015

Benidis, K., Rangapuram, S. S., Flunkert, V., Wang, Y., Maddix, D., Turkmen, C., Gasthaus, J., Bohlke-Schneider, M., Salinas, D., Stella, L., Aubet, F. X., Callot, L., & Januschowski, T. (2022). Deep learning for time series forecasting: Tutorial and literature survey. *ACM Computing Surveys*, *55*. https://doi.org/10.1145/3533382

Brinkmeyer, L., Drumond, R. R., Burchert, J., & Schmidt-Thieme, L. (2022). Few-shot forecasting of time-series with heterogeneous channels. http://arxiv.org/abs/2204.03456

Challu, C., Olivares, K. G., Oreshkin, B. N., Garza, F., Mergenthaler-Canseco, M., & Dubrawski, A. (2022). N-hits: Neural hierarchical interpolation for time series forecasting. https://doi.org/https://doi.org/10.48550/arXiv.2201.12886

Chan, S., Oktavianti, I., & Puspita, V. (2019). A deep learning cnn and ai-tuned svm for electricity consumption forecasting: Multivariate time series data. *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 0488–0494. https://doi.org/10.1109/IEMCON.2019.8936260

Chen, S.-A., Li, C.-L., Yoder, N., Arik, S. O., & Pfister, T. (2023). Tsmixer: An all-mlp architecture for time series forecasting. http://arxiv.org/abs/2303.06053

Chen, Y., Yang, W., & Zhang, B. (2020). Using mobility for electrical load forecasting during the covid-19 pandemic. http://arxiv.org/abs/2006.08826

Data package household data. version 2020-04-15. (2020). https://data.open-power-system-data.org/household_data/2020-04-15/

Directive (eu) 2019/944. (2022). *European Union*. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:02019L0944-20220623

Ekambaram, V., Jati, A., Nguyen, N. H., Dayama, P., Reddy, C., Gifford, W. M., & Kalagnanam, J. (2024). Tiny time mixers (ttms): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series. https://doi.org/https://doi.org/10.48550/arXiv.2401.03955

Falcon, W., & The PyTorch Lightning team. (2019, March). *PyTorch Lightning* (Version 1.4). https://doi.org/10.5281/zenodo.3828935

Fan, C., Sun, Y., Xiao, F., Ma, J., Lee, D., Wang, J., & Tseng, Y. C. (2020). Statistical investigations of transfer learning-based methodology for short-term building energy predictions. *Applied Energy*, *262*. https://doi.org/10.1016/j.apenergy.2020.114499

Forootani, A., Rastegar, M., & Zareipour, H. (2024). Transfer learning-based framework enhanced by deep generative model for cold-start forecasting of residential ev charging behavior. *IEEE Transactions on Intelligent Vehicles*, *9*, 190–198. https://doi.org/10.1109/TIV.2023.3328458

Foundational time series model (beta). (2023). Retrieved March 27, 2024, from https://docs.nixtla.io/reference/forecast_forecast_post

Fu, Y., Wu, D., & Boulet, B. (2022). On the benefits of transfer learning and reinforcement learning for electric short-term load forecasting. *IEEE Computer Society*, 659. https://doi.org/http://dx.doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics55523.2022.00020

Garza, A., & Mergenthaler-Canseco, M. (2023). Timegpt-1. http://arxiv.org/abs/2310.03589

Gasparin, A., Lukovic, S., & Alippi, C. (2019). Deep learning for time series forecasting: The electric load case. *CAAI Trans. Intell. Technol.*, *7*, 1–25. https://doi.org/10.1049/cit2.12060

Grolinger, K., Capretz, M. M., & Seewald, L. (2016). Energy consumption prediction with big data: Balancing prediction accuracy and computational resources. *2016 IEEE International Congress on Big Data (BigData Congress)*, 157–164. https://doi.org/10.1109/BigDataCongress.2016.27

Gunduz, S., Ugurlu, U., & Oksuz, I. (2023a). Transfer learning for electricity price forecasting. *Sustainable Energy, Grids and Networks*, *34*, 100996. https://doi.org/https://doi.org/10.1016/j.segan.2023.100996

Gunduz, S., Ugurlu, U., & Oksuz, I. (2023b). Transfer learning for electricity price forecasting. *Sustainable Energy, Grids and Networks*, *34*. https://doi.org/10.1016/j.segan.2023.100996

Han, L., Ye, H.-J., & Zhan, D.-C. (2023). The capacity and robustness trade-off: Revisiting the channel independent strategy for multivariate time series forecasting. http://arxiv.org/abs/2304.05206

Herzen, J., Lässig, F., Piazzetta, S. G., Neuer, T., Tafti, L., Raille, G., Pottelbergh, T. V., Pasieka, M., Skrodzki, A., Huguenin, N., Dumonal, M., Kościsz, J., Bader, D., Gusset, F., Benheddi, M., Williamson, C., Kosinski, M., Petrik, M., & Grosch, G. (2022). Darts: User-friendly modern machine learning for time series. *Journal of Machine Learning Research*, *23*(124), 1–6. http://jmlr.org/papers/v23/21-1177.html

Herzen, J., Ravasi, F., Raille, G., & Grosch, G. (2020). Transfer learning for time series forecasting with darts. https://unit8co.github.io/darts/examples/14-transfer-learning.html#Part-3:-Training-an-N-BEATS-model-on-m4-dataset-and-use-it-to-forecast-air-dataset

Himeur, Y., Elnour, M., Fadli, F., Meskin, N., Petri, I., Rezgui, Y., Bensaali, F., & Amira, A. (2022, October). Next-generation energy systems for sustainable smart cities: Roles of transfer learning. https://doi.org/10.1016/j.scs.2022.104059

Hobbs, B., Helman, U., Jitprapaikulsarn, S., Konda, S., & Maratukulam, D. (1998). Artificial neural networks for short-term energy forecasting: Accuracy and economic value. *Neurocomputing*, *23*, 71–84. https://doi.org/10.1016/S0925-2312(98)00072-1

Hobbs, B., Jitprapaikulsarn, S., Konda, S., Chankong, V., Loparo, K., & Maratukulam, D. (1999). Analysis of the value for unit commitment of improved load forecasts. *IEEE Transactions on Power Systems*, *14*, 1342–1348. https://doi.org/10.1109/59.801894

Jiang, Y., Dai, Y., Si, R., Chen, J., Gao, T., & Zhang, J. (2022). Short-term state electricity load forecasting based on transfer-informer. *2022 IEEE 2nd International Conference on Digital*

Twins and Parallel Intelligence, DTPI 2022. https://doi.org/10.1109/DTPI55838.2022.9998911

Kamalov, F., Sulieman, H., Moussa, S., Reyes, J. A., & Safaraliev, M. (2024). Powering electricity forecasting with transfer learning. *Energies*, *17*. https://doi.org/10.3390/en17030626

Kim, T., Kim, J., Tae, Y., Park, C., Choi, J.-H., & Choo, J. (2022). Reversible instance normalization for accurate time-series forecasting against distribution shift. *International Conference on Learning Representations*. https://openreview.net/forum?id=cGDAkQo1C0p

Laitsos, V., Vontzos, G., & Bargiotas, D. (2023). Investigation of transfer learning for electricity load forecasting. *14th International Conference on Information, Intelligence, Systems and Applications, IISA 2023*. https://doi.org/10.1109/IISA59645.2023.10345954

Lee, E., & Rhee, W. (2021). Individualized short-term electric load forecasting with deep neural network based transfer learning and meta learning. *IEEE Access*, *9*, 15413–15425. https://doi.org/10.1109/ACCESS.2021.3053317

L'heureux, A., Grolinger, K., & Capretz, M. A. (2022). Transformer-based model for electrical load forecasting. *Energies*, *15*. https://doi.org/10.3390/en15144993

Li, A., Xiao, F., Zhang, C., & Fan, C. (2021). Attention-based interpretable neural network for building cooling load prediction. *Applied Energy*, *299*, 117238. https://doi.org/https://doi.org/10.1016/j.apenergy.2021.117238

Li, Z., Li, Y., Liu, Y., Wang, P., Lu, R., & Gooi, H. (2021). Deep learning based densely connected network for load forecasting. *IEEE Transactions on Power Systems*, *36*, 2829–2840. https://doi.org/10.1109/TPWRS.2020.3048359

Liang, Y., Wen, H., Nie, Y., Jiang, Y., Jin, M., Song, D., Pan, S., & Wen, Q. (2024). Foundation models for time series analysis: A tutorial and survey. https://doi.org/https://doi.org/10.48550/arXiv.2403.14735

Lim, B., Arik, S. O., Loeff, N., & Pfister, T. (2019). Temporal fusion transformers for interpretable multi-horizon time series forecasting. http://arxiv.org/abs/1912.09363

Lin, G., Kramer, H., Nibler, V., Crowe, E., & Granderson, J. (2022). Building analytics tool deployment at scale: Benefits, costs, and deployment practices. *Energies*. https://doi.org/10.3390/en15134858

Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., & Long, M. (2023). Itransformer: Inverted transformers are effective for time series forecasting. http://arxiv.org/abs/2310.06625

Liu, Y., Wu, H., Wang, J., & Long, M. (2022). Non-stationary transformers: Exploring the stationarity in time series forecasting. http://arxiv.org/abs/2205.14415

Lu, J., Behbood, V., Hao, P., Zuo, H., Xue, S., & Zhang, G. (2015). Transfer learning using computational intelligence: A survey. *Knowledge-Based Systems*, *80*. https://doi.org/10.1016/j.knosys.2015.01.010

Luo, X., Zhang, D., & Zhu, X. (2022). Combining transfer learning and constrained long short-term memory for power generation forecasting of newly-constructed photovoltaic plants. *Renewable Energy*, *185*, 1062–1077. https://doi.org/10.1016/j.renene.2021.12.104

Ma, Q., Liu, Z., Zheng, Z., Huang, Z., Zhu, S., Yu, Z., & Kwok, J. T. (2023). A survey on time-series pre-trained models. http://arxiv.org/abs/2305.10716

Miller, C., Kathirgamanathan, A., Picchetti, B., Arjunan, P., Park, J. Y., Nagy, Z., Raftery, P., Hobson, B. W., Shi, Z., & Meggers, F. (2020a). The building data genome 2 (bdg2) data-set. https://github.com/buds-lab/building-data-genome-project-2/tree/master

Miller, C., Kathirgamanathan, A., Picchetti, B., Arjunan, P., Park, J. Y., Nagy, Z., Raftery, P., Hobson, B. W., Shi, Z., & Meggers, F. (2020b). The building data genome project 2, energy meter data from the ASHRAE great energy predictor III competition. *Scientific Data*, *7*, 368. https://doi.org/https://doi.org/10.48550/arXiv.2006.02273

N-hits. (n.d.). https://unit8co.github.io/darts/generated_api/darts.models.forecasting.nhits.html

Nie, Y., Nguyen, N. H., Sinthong, P., & Kalagnanam, J. (2022). A time series is worth 64 words: Long-term forecasting with transformers. http://arxiv.org/abs/2211.14730

Oreshkin, B. N., Carpov, D., Chapados, N., & Bengio, Y. (2020). N-beats: Neural basis expansion analysis for interpretable time series forecasting. https://doi.org/https://doi.org/10.48550/arXiv.1905.10437

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*. https://doi.org/10.1016/j.adapen.2022.100084

Schaeffer, R. (2023). Pretraining on the test set is all you need. https://doi.org/https://doi.org/10.48550/arXiv.2309.08632

Smith, T. G., et al. (2017). pmdarima: Arima estimators for Python. http://www.alkaline-ml.com/pmdarima

Tian, Y., Sehovac, L., & Grolinger, K. (2019). Similarity-based chained transfer learning for energy forecasting with big data. *IEEE Access*, *7*, 139895–139908. https://doi.org/10.1109/ACCESS.2019.2943752

Trindade, A. (2015). ElectricityLoadDiagrams-2011-2014. https://doi.org/https://doi.org/10.24432/C58C86

Voß, M., Bender-Saebelkampf, C., & Albayrak, S. (2018). Residential short-term load forecasting using convolutional neural networks. *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 1–6. https://doi.org/10.1109/SmartGridComm.2018.8587494

Wang, P. (2023). Itransformer. https://github.com/lucidrains/iTransformer

Wang, Z., Wen, Q., Zhang, C., Sun, L., & Wang, Y. (2023). Diffload: Uncertainty quantification in load forecasting with diffusion model. https://doi.org/10.48550/arXiv.2306.01001

Weber, M., Auch, M., Doblander, C., Mandl, P., & Jacobsen, H.-A. (2021). Transfer learning with time series data: A systematic mapping study. *IEEE Access*, *9*, 165409–165432. https://doi.org/10.1109/ACCESS.2021.3134628

Wei, B., Li, K., Zhou, S., Xue, W., & Tan, G. (2024). An instance based multi-source transfer learning strategy for building's short-term electricity loads prediction under sparse data scenarios. *Journal of Building Engineering*, *85*. https://doi.org/10.1016/j.jobe.2024.108713

Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., & Sun, L. (2022). Transformers in time series: A survey. http://arxiv.org/abs/2202.07125

Woo, G., Liu, C., Kumar, A., Xiong, C., Savarese, S., & Sahoo, D. (2024). Unified training of universal time series forecasting transformers. http://arxiv.org/abs/2402.02592

Xu, B., Hou, R., Ding, X., & Tao, Y. (2018). Residential electric load forecasting method based on mutual information and bp network combination model, 589–593. https://doi.org/10.1145/3297156.3297182

Ye, R., & Dai, Q. (2021). Implementing transfer learning across different datasets for time series forecasting. *Pattern Recognition*, *109*, 107617. https://doi.org/https://doi.org/10.1016/j.patcog.2020.107617

Yuan, Y., Chen, Z., Wang, Z., Sun, Y., & Chen, Y. (2023). Attention mechanism-based transfer learning model for day-ahead energy demand forecasting of shopping mall buildings. *Energy*, *270*. https://doi.org/10.1016/j.energy.2023.126878

Zhang, Z., Zhao, P., Wang, P., & Lee, W. J. (2022). Transfer learning featured short-term combining forecasting model for residential loads with small sample sets. *IEEE Transactions on Industry Applications*, *58*, 4279–4288. https://doi.org/10.1109/TIA.2022.3170385

Appendix

Table A.10: Results for ELD dataset.

Source	Learning Scenario	Metric	NHiTS	Transformer	TSMixer	iTransformer	TimeGPT
Bavaria	Zero-Shot	MAE	3.501	2.850	2.762	1.174	0.037
Bavaria	Zero-Shot	MSE	15.634	10.373	9.903	2.736	0.067
Bavaria	Short TL	MAE	1.275	0.811	0.492	0.406	
Bavaria	Short TL	MSE	2.587	0.961	0.481	0.337	
Bavaria	Full TL	MAE	0.812	0.824	0.394	0.301	
Bavaria	Full TL	MSE	0.951	0.978	0.308	0.211	
GP2	Zero-Shot	MAE	0.645	0.812	0.759	0.451	0.037
GP2	Zero-Shot	MSE	0.693	0.957	0.983	0.450	0.067
GP2	Short TL	MAE	0.702	0.812	0.413	0.300	
GP2	Short TL	MSE	0.763	0.952	0.362	0.221	
GP2	Full TL	MAE	0.507	0.809	0.303	0.266	
GP2	Full TL	MSE	0.470	0.947	0.218	0.180	
ELD	Short training	MAE	0.451	0.813	0.400	0.351	
ELD	Short training	MSE	0.408	0.954	0.352	0.265	
ELD	Full training	MAE	0.404	0.809	0.300	0.266	
ELD	Full training	MSE	0.345	0.947	0.212	0.182	

Table A.11: Results for GP2 dataset.

Source	Learning Scenario	Metric	NHiTS	Transformer	TSMixer	iTransformer	TimeGPT
Bavaria	Zero-Shot	MAE	2.295	2.088	2.067	1.003	0.037
Bavaria	Zero-Shot	MSE	9.312	7.905	8.562	2.493	0.083
Bavaria	Short TL	MAE	0.664	0.681	0.530	0.457	
Bavaria	Short TL	MSE	1.035	1.071	0.862	0.604	
Bavaria	Full TL	MAE	0.651	0.658	0.407	0.364	
Bavaria	Full TL	MSE	0.975	1.025	0.710	0.468	
ELD	Zero-Shot	MAE	0.608	0.656	0.707	0.525	0.037
ELD	Zero-Shot	MSE	0.927	0.987	1.370	0.866	0.083
ELD	Short TL	MAE	0.664	0.656	0.527	0.380	
ELD	Short TL	MSE	1.035	0.987	0.818	0.508	
ELD	Full TL	MAE	0.648	0.649	0.391	0.347	
ELD	Full TL	MSE	0.976	0.976	0.578	0.451	
GP2	Short training	MAE	0.644	0.664	0.513	0.428	
GP2	Short training	MSE	0.969	1.042	0.852	0.587	
GP2	Full training	MAE	0.581	0.649	0.400	0.360	
GP2	Full training	MSE	0.855	0.976	0.596	0.478	

Table A.12: Results for Bavaria dataset.

Source	Learning Scenario	Metric	NHiTS	Transformer	TSMixer	iTransformer	TimeGPT
ELD	Zero-Shot	MAE	2.899_{-2}	2.989_{-2}	2.932_{-2}	0.398	2.200_{-2}
ELD	Zero-Shot	MSE	1.543_{-3}	1.627_{-3}	1.782_{-3}	0.195	2.292_{-2}
ELD	Short TL	MAE	1.436_{-2}	3.022_{-2}	2.181_{-2}	2.156_{-2}	
ELD	Short TL	MSE	5.207_{-4}	1.654_{-3}	1.054_{-3}	8.370_{-4}	
ELD	Full TL	MAE	1.037_{-2}	9.077_{-3}	1.106_{-2}	9.052_{-3}	
ELD	Full TL	MSE	3.258_{-4}	3.137_{-4}	3.557_{-4}	2.552_{-4}	
GP2	Zero-Shot	MAE	3.156_{-2}	3.027_{-2}	3.026_{-2}	0.435	2.200_{-2}
GP2	Zero-Shot	MSE	1.811_{-3}	1.673_{-3}	1.722_{-3}	0.233	2.292_{-2}
GP2	Short TL	MAE	1.055_{-2}	1.074_{-2}	1.229_{-2}	1.399_{-2}	
GP2	Short TL	MSE	4.682_{-4}	4.886_{-4}	5.087_{-4}	4.229_{-4}	
GP2	Full TL	MAE	1.010_{-2}	9.070_{-3}	8.693_{-3}	1.100_{-2}	
GP2	Full TL	MSE	3.177_{-4}	3.141_{-4}	2.691_{-4}	3.091_{-4}	
Bavaria	Short training	MAE	1.004_{-2}	1.120_{-2}	1.125_{-2}	2.723_{-2}	
Bavaria	Short training	MSE	3.408_{-4}	4.945_{-4}	5.016_{-4}	1.315_{-3}	
Bavaria	Full training	MAE	8.917_{-3}	9.164_{-3}	9.207_{-3}	8.256_{-3}	
Bavaria	Full training	MSE	2.776_{-4}	3.146_{-4}	2.871_{-4}	2.394_{-4}	

Declaration of Authorship

I, Benedikt Rein, hereby declare that I have not previously submitted the present work for other examinations. I wrote this work independently. All sources, including sources from the Internet, that I have reproduced in either an unaltered or modified form (particularly sources for texts, graphs, tables, and images), have been acknowledged by me as such. I understand that violations of these principles will result in proceedings regarding deception or attempted deception.

Berlin, July 30, 2024