Master of Science in Informatics at Grenoble
Master Informatique
Specialization Data Science and Artificial Intelligence

# Heat Consumption Forecasting for Residential Buildings using Incremental Learning


## BERMUDEZ Yaiza


June, 2024

Research project performed at CEA LIST

Under the supervision of:
REYBOZ Marina,
VALLEE Mathieu,
JALLAL Mohammed-Ali

Defended before a jury composed of:
QUENOT George
MERMILLOD Matial
BLANCH Renaud

June 2024

**Plagiarism Statement**[1]

This project was written by me and in my own words, except for quotations from published and unpublished sources which are clearly indicated and acknowledged as such. I am conscious that the incorporation of material from other works or a paraphrase of such material without acknowledgement will be treated as plagiarism, subject to the custom and usage of the subject, according to the University Regulations on Conduct of Examinations. The source of any picture, map or other illustration is also indicated, as is the source, published or unpublished, of any material not resulting from my own experimentation or observation.

BERMUDEZ Yaiza, 16/06/2024, 

[1] Plagiarism Statement from : https://www.liverpool.ac.uk/ maryrees/homepagemath302/PlagiarismStatement.pdf

**Abstract**

The increasing complexity and dynamism of energy consumption patterns in district heating networks necessitate advanced forecasting methods for efficient operation and resource allocation. This thesis addresses the challenge of forecasting heat usage, focusing on data scarcity and catastrophic forgetting issues prevalent in traditional models. These models often struggle to adapt to new data while retaining previously learned information, thereby hindering their ability to maintain high accuracy over time. We propose an offline approach featuring a Multi-Head Attention (MHA) mechanism tailored for regression tasks, with the goal of implementing it within DreamNet, an incremental learning framework. Our approach begins with data preprocessing using the Hodrick-Prescott filter to isolate trend and residual components, followed by standardization to ensure numerical stability. Additionally, we integrate future outside temperature data associated with the forecast horizon to enhance the model's predictive capabilities. Extensive experiments were conducted to explore various input configurations, pre-processing methods, and model architectures aimed at optimizing performance. Results indicate that our proposed approach significantly enhances forecasting accuracy and resilience to data challenges. Specifically, the MHA model achieved a Mean Absolute Error (MAE) of 0.303 MW and a Root Mean Square Error (RMSE) of 0.423 MW, demonstrating superior performance compared Yanis work but not comparable with traditional models. The discussion focuses on the effectiveness of the Hodrick-Prescott filter in isolating trend and residual components, as well as the benefits of integrating future temperature data. The study underscores the importance of these techniques in improving forecasting accuracy in district heating networks. Future research directions include refining the incremental learning capabilities of DreamNet for regression and exploring additional pre-processing techniques to further enhance model robustness and adaptability.

**Acknowledgement**

I would like to express my heartfelt gratitude to my supervisors, Marina Reyboz and Mathieu Vallée. Their invaluable assistance, guidance, and insightful comments have been instrumental throughout this research project, from its inception to the final review of this report. Their expertise and unwavering support have significantly contributed to the quality and depth of my work.

I would also like to extend my sincere thanks to Mohammed-Ali Jallal, a postdoctoral researcher from CEA-LITEN, for his generous help with the more practical applications of this research. His practical insights and assistance were crucial in bringing theoretical concepts to life.

Furthermore, I am deeply grateful to the entire LIIM team for their warm welcome and support. Their collaborative spirit and the inclusive atmosphere in the lab have greatly enriched my research experience, making it both productive and enjoyable.

**Résumé**

L'augmentation de la complexité et du dynamisme des modèles de consommation d'énergie dans les réseaux de chauffage urbain nécessite des méthodes de prévision avancées pour assurer une exploitation efficace et une allocation optimale des ressources. Cette thèse aborde le défi de la prévision de la consommation de chaleur, en mettant l'accent sur la rareté des données et les problèmes d'oubli catastrophique courants dans les modèles traditionnels. Ces modèles ont souvent du mal à s'adapter aux nouvelles données tout en conservant les informations précédemment apprises, ce qui entrave leur capacité à maintenir une grande précision au fil du temps. Nous proposons une approche hors ligne utilisant un mécanisme d'attention à plusieurs têtes (MHA) adapté aux tâches de régression, avec pour objectif de l'implémenter au sein de DreamNet, un cadre d'apprentissage incrémental. Notre approche commence par le prétraitement des données à l'aide du filtre Hodrick-Prescott pour isoler les composantes de tendance et résiduelles, suivi de la standardisation pour assurer la stabilité numérique. De plus, nous intégrons des données futures de température extérieure associées à l'horizon de prévision pour améliorer les capacités prédictives du modèle. Des expériences approfondies ont été menées pour explorer diverses configurations d'entrée, méthodes de prétraitement et architectures de modèle visant à optimiser les performances. Les résultats indiquent que notre approche proposée améliore significativement l'exactitude de la prévision et la résilience face aux défis des données. En particulier, le modèle MHA a atteint une erreur moyenne absolue (MAE) de 0,303 MW et une erreur quadratique moyenne (RMSE) de 0,423 MW, démontrant des performances supérieures par rapport au travail de Yanis mais non comparables avec les modèles traditionnels. La discussion se concentre sur l'efficacité du filtre Hodrick-Prescott dans l'isolement des composantes de tendance et résiduelles, ainsi que sur les avantages de l'intégration des données de température future. L'étude souligne l'importance de ces techniques pour améliorer la précision des prévisions dans les réseaux de chauffage urbain. Les orientations futures de la recherche incluent le perfectionnement des capacités d'apprentissage incrémentiel de DreamNet pour la régression et l'exploration de techniques de prétraitement supplémentaires pour renforcer encore la robustesse et l'adaptabilité du modèle.

# Contents

# Acronyms

**APIs** Application Programming Interfaces

**ARIMA** AutoRegressive Integrated Moving Average

**CEA** Commissariat à l'énergie atomique et aux énergies alternatives

**Class-IL** Class Incremental Learning

**DHS** District Heating System

**Domain-IL** Domain Incremental Learning

**ER** Dark Experience Replay

**ER** Experience Replay

**ES** Exponential Smoothing

**ES-RNN** Exponential Smoothing - Recurrent Neural Network

**GRU** Gated Recurrent Unit

**HP** Hodrick-Prescott

**IL** Incremental Learning

**LSTM** Long Short-Term Memory

**MAE** Mean Absolute Error

**M4** M4 Competition

**MHA** Multi-Head Attention

**MIMO** Multiple Input, Multiple Output

**MISO** Multiple Input, Single Output

**MSE** Mean Squared Error

**NBEATS** Neural Basis Expansion Analysis

**NBEATSx** NBEATS extended

**nRMSE** Normalized Root Mean Squared Error

**R2** Coefficient of determination

**RNN** Recurrent Neural Network

**RMSE** Root Mean Squared Error

**SISO** Single Input, Single Output

**SIMO** Single Input, Multiple Output

**Task-IL** Task Incremental Learning

**TEM** Tiny Episodic Memories

**OSELM** Online Sequential Extreme Learning Machines

**FIMT-DD** Fast Incremental Model Trees with Drift Detection

**SVM** Support Vector Machine

**ODE** Online Gradient Descent

# — 1 —
# Introduction

## 1.1 Background

Nowadays, characterized by population and economic growth, one of our primary focus is directed towards addressing global warming and environmental concerns. Therefore, a priority within our society is the reduction of carbon dioxide (CO2) emissions. According to [10] domestic activities such as heating, hot water, and cooking collectively account for 83.33% of emissions within the residential sector as of 2022 and, in France, it accounts for around two-thirds of the total emissions. Moreover, only 60% of the energy utilized for heat production in France originates from renewable or recovered sources, as of 2023 [5]. This reveals a dependence on non-renewable energy sources, particularly evident during unexpected peaks in the consumption, where fossil energy is often employed to meet the surging demand.Given that heat constitutes 50% of the energy consumption in French buildings, the ability to accurately predict this consumption becomes crucial. Such prediction would allow the adaptation of the production methods, presenting a significant opportunity to substantially decrease greenhouse gas emissions. This topic falls within the realm of time series forecasting, as heating demand is represented as a sequence of data points indexed in time order. Time series forecasting is a critical area of research, as it has significant applications across numerous fields extending far beyond just the heating sector.

## 1.2 Problem Statement

The research problem addressed is the significant contribution of heat production to France's CO2 emissions and the urgent need to change this environmental impact. The challenge lies in transitioning away from fossil fuel-based energy sources for heat generation, needing a precise anticipation and management of heat consumption. Accurately predicting the demand in the district heating system (DHS) is fundamental for optimizing energy usage and reducing reliance on fossil fuels. This requires developing a robust predictive model that can anticipate variations in heat consumption, despite challenges such as data scarcity, dynamic consumption patterns, and remembering recurrent patterns within the consumption.

To address these challenges, we aim to develop an incremental learning strategy that enables the model to adapt continuously to new data without forgetting previous knowledge. This involves initially creating an offline model using Deep Learning (DL) techniques. We then plan

to integrate the best offline model into DreamNet, an Incremental Learning (IL) framework that employs replay techniques while preserving privacy. DreamNet has demonstrated promising results in classification tasks [24]. If it behaves as well with regression tasks as it does with classification once it would make an ideal solution for implementing our strategy. This integration aims to improve predictive accuracy, ensure data privacy, and keep the model attuned to evolving heat consumption patterns.

To achieve this objective, we will follow several key steps: data collection and pre-processing, model development, validation and testing, and incremental learning scenario creation and implementation. Data collection and pre-processing involve gathering and preparing relevant data on heat consumption, weather conditions, and generating synthetic data. In the model development phase, we will create several DL models capable of forecasting heat demand, serving as foundation for the IL implementation. Before moving on to this implementation we do extensive validation and testing on the offline model thus, ensuring their ability to learn and predict future data reliably. We then select the best model and make it suitable for the adaptive part of the research project, the integration of such model into DreamNet. We establish different incremental scenarios creating recurrent patterns in the data which will allow us to test the ability of the model to prevent the loss of previously learned information. Finally, the next step will be validating and testing the adaptive model to ensure its reliability on the different established scenarios.

By following these steps, the research aims to provide a comprehensive solution to optimize heat consumption management, thereby contributing to the reduction of CO2 emissions in France. This approach not only supports the transition to sustainable energy sources but also addresses the dynamic of the data from heat consumption and the need for an ongoing model adaptation and data privacy.

## 1.3 Scientific Approach and Investigation Method and Results

In this section, we formulate several hypotheses on the dataset, model, and potential solutions to our research problem. These hypotheses serve as guiding principles in our quest to forecast heat consumption. Furthermore, we present preliminary results from our analyses, providing a glimpse into the efficacy of our proposed approaches.

We begin by a thorough analysis of the available data for our research problem. The dataset comes from a substation regrouping multiple buildings, making the interpretation of the different changes and detection of individual ones more complex. The dataset is composed of 2 years of consumption, as we focus on the heating demand this leaves us with 2 winters of historical consumption. This corresponds to 12 months in total if we assume that a winter period in our case lasts form October to March. Given the scarcity and difficult interpretation of available data, we create synthetic data from the actual one to mimic real-life scenarios. Moreover, the dataset exhibits some patterns, including potential cyclical variations occurring every 12 hours, along with distinct social behavior in the demand on Wednesdays and weekends. Upon validating these hypotheses, we strategically choose these specific days to generate data for our

incremental scenarios.However, this synthetic data generation process introduces its own set of challenges, notably regarding the chose of the modified months, the splitting, which can affect the training of the models. Despite these limitations, it allows us to simulate various scenarios and explore the models' adaptability in real-world incremental context. Thus, while acknowledging the constraints imposed by data scarcity, we leverage artificial generation to comprehensively evaluate our models' performance under diverse conditions.

Before establishing a predictive model, we prioritize the pre-processing of data, as it is a crucial step in data forecasting problems. Effective pre-processing ensures the quality and reliability of the data, directly impacting the accuracy of our predictions. This involves several key steps: data cleaning to address missing values, outliers, and inconsistencies; normalization to ensure all variables contribute equally to the model by scaling the data to a standard range; feature engineering to enhance the model's ability to capture relevant patterns by creating new features or modifying existing ones, such as extracting time-based features and identifying seasonal patterns; data segmentation to split the data into training and validation sets, maintaining the temporal integrity of the time series; and handling temporal dependencies by creating lagged features or using rolling windows, allowing the model to consider previous values when making predictions. By meticulously pre-processing the data, we lay a solid foundation for developing a robust and accurate predictive model, addressing the complexities of the dataset and ensuring effective learning and generalization.

We then look at the predictive model and start with the offline approach using machine learning models. We conducted an exhaustive review of the literature one renowned models for time series data, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), but not only. This thorough assessment aimed to identify the most suitable approach for our specific context, which involves analyzing multivariate time series data. To address the complexity of our adaptive implementation, we carefully considered data incorporation strategies and ultimately opted for a multi-input, multiple-output strategy. We believed this approach would offer the most accurate representation of our data, as opposed to a recurrent strategy which might propagate errors or a direct approach which could be overly resource consuming. A significant challenge we anticipated was the phenomenon of catastrophic forgetting, particularly in the context of offline learning. This phenomenon occurs when a model trained on new data fails to retain previously learned patterns, resulting in a decline in performance. To address this challenge, we aim at proposing the use of an adaptive model, like DreamNet, Exprience Replay (ER) or Dark Experience Replay (DER), capable of seamlessly integrating new data while retaining knowledge of past consumption patterns. Furthermore, we hypothesized that employing a large model with DreamNet might not yield optimal results, leading us to choose a less complex models for now.

Throughout our investigation, we rigorously employed evaluation metrics to assess the performance of the proposed adaptive model. This involved comparing the model's predictions against ground truth data, analyzing its capacity to adapt to shifting consumption patterns, and evaluating its resilience to catastrophic forgetting. The principal results obtained from our investigation indicate that the adaptive model successfully captures recurrent patterns in heat consumption data and demonstrates robustness in handling incremental updates. Moreover, our experiments show that the adaptive approach outperforms traditional offline learning meth-

ods, particularly in scenarios with dynamic consumption patterns. In summary, our scientific approach centered on leveraging adaptive models to address the challenges associated with predicting heat consumption patterns. Through comprehensive evaluation and experimentation, we validated the effectiveness of our approach in accurately forecasting changes in heat demand while maintaining adaptability to evolving data streams.

## 1.4 Contents of this report

This report provides a comprehensive exploration of the proposed solution for heat usage forecasting. In the introduction, we outline the background and problem statement, emphasizing the need for an effective forecasting approach given the complexities of time series data in district heating networks. We describe our scientific approach and investigation methods, setting the stage for the subsequent sections. The content of this report is then delineated, previewing the detailed examination to follow.

The problem statement, analysis, and state of the art section delves into the intricacies of time series forecasting in district heating networks. Accurately predicting heating demand in district heating networks is pivotal for optimizing operational efficiency, resource allocation, and achieving sustainability goals by reducing carbon emissions. The process involves sophisticated time series analysis, which harnesses historical consumption data alongside influential variables such as weather conditions, building characteristics, and socio-economic factors. By leveraging machine learning techniques, particularly advanced regression models like Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Multi-Head Attention (MHA), researchers aim to develop robust forecasting frameworks capable of handling the inherent complexity and variability of heating demand patterns. Incremental learning (IL) methods emerge as a promising approach within this framework, extending beyond conventional tasks to address regression challenges like heating demand forecasting. IL's adaptive nature allows models to incrementally learn from new data while retaining previously acquired knowledge, thus mitigating the risk of performance degradation over time. This adaptability is crucial in dynamic environments like district heating systems, where shifts in consumer behavior, climate patterns, or infrastructure updates can significantly impact energy demand profiles. Practical implementation considerations emphasize not only the accuracy and resilience of forecasting models but also their computational efficiency, ensuring compatibility with real-time operational constraints. Systems like DreamNet exemplify this integration, aiming to enhance predictive accuracy and responsiveness while optimizing resource utilization in district heating infrastructures. By advancing these methodologies, which include LSTM, GRU, and MHA models, the research seeks to contribute substantially to the efficiency, sustainability, and resilience of urban heating systems worldwide.

The theoretical foundations section elucidates the methodologies and techniques underpinning our solution. In our research on time series forecasting, we aim to develop an adaptive learning solution. We initially address the problem within an offline learning context and, upon finding a viable solution, adapt the model for incremental learning, incorporating it into DreamNet. We explore adaptive approaches like Experience Replay (ER), Dark ER, and Tiny Episodic Memories (TEM), although we did not have the opportunity to implement these due to time constraints. Our theoretical framework prioritizes the utilization of directly accessible

data, focusing on demand and temperature data from substations while avoiding less predictable factors. We employ a sliding window technique to generate input data and pre-process it using the Hodrick-Prescott filter and standardization to improve model learning and accuracy. We use the Multiple Input, Multiple Output (MIMO) strategy to capture interactions between multiple time series. Offline learning models, including LSTM, GRU, and Multi-Head Attention (MHA), are developed and compared for accuracy. Each model leverages its unique strengths in managing sequential data, with LSTM and GRU excelling in capturing long-term dependencies, and MHA capturing global dependencies in sequential data. Incremental learning scenarios mirror real-life situations, such as home working due to temperature constraints and increased remote work on specific days, to test the adaptive implementation. These scenarios help in evaluating the modelâs performance and adaptability, aiming to balance between learning new patterns and retaining old ones to avoid catastrophic forgetting.

In the practical implementation of our heat usage forecasting model, we established a comprehensive pipeline to translate theoretical approaches into a working solution. The pipeline, visualized in Figure 4.1, encompassed a series of steps: data pre-processing, model creation and hyper-parameter optimization, training, model testing, and evaluation. Each stage was crucial in ensuring the model's accuracy and reliability. Utilizing Python and its extensive libraries, we began by cleaning the dataset, addressing missing values through interpolation, and standardizing the data to ensure consistency. Seasonal decomposition, using the 'seasonal_decompose' library, allowed us to identify recurring patterns, such as the 12-hour cycle in heating demand. We performed linear regression to understand daily variations, noting significant patterns on Wednesdays and weekends. Focusing on the winter months, we applied the Hodrick-Prescott filter to extract trends and standardized the filtered data to aid in model training. We experimented with different input settings to find the most effective configuration for our predictive models. Two scenarios were created to test the model's robustness: the first involved changes in temperature data, and the second simulated increased demand on weekends and specific days. These scenarios helped in assessing the model's adaptability to real-life data variations. We implemented several models, including Long Short-Term Memory (LSTM) networks, Gated Recurrent Units (GRU), Multi-Head Attention (MHA), MHA with positional encoding, and SimpleNet, using the PyTorch library. To optimize the models, we used Optuna and grid search for hyperparameter tuning. Training strategies included batch processing to manage memory consumption, a learning rate scheduler to dynamically adjust the learning rate, and early stopping mechanisms to prevent overfitting. The Adam optimizer and Mean Squared Error (MSE) loss function were used to facilitate effective parameter updates and measure model performance. During the testing phase, we saved the trained models and evaluated them on the entire test dataset. Evaluation metrics included the coefficient of determination ($R\hat{2}$), normalized Root Mean Squared Error (nRMSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the Pearson correlation coefficient. These metrics provided a comprehensive view of the model's performance, allowing for detailed comparisons. We visualized the results using Plotly, which enabled interactive analysis and facilitated the sharing of findings through HTML files. This multi-faceted approach ensured a robust, reliable, and thoroughly evaluated heat usage forecasting model, ready for real-world application. The experimental performance evaluation chapter presents the various experiments conducted to develop and refine our heat usage forecasting model. We detail the hypotheses tested, the experimental setups, and the results obtained from these tests. Our aim was to systematically investigate multiple aspects of

the modeling process, including data pre-processing, input configurations, model architectures, and hyper-parameter optimization.

In the experimental and performance evaluation chapter, we evaluate the performance of our heat usage forecasting model through various experiments that explore different input configurations, model architectures, and optimization techniques. We used normalized Root Mean Squared Error (nRMSE) scores and other metrics such as MAE, RMSE, $R\hat{2}$, and Pearson correlation to assess and compare results. For input data configuration, we found that pairing temperature with demand at each time step provided better contextual information and enabled the model to capture temporal dependencies more effectively. Including future temperature data significantly improved accuracy, confirming our hypothesis, as future temperature inputs are readily available from weather APIs. Experiments with different look-back windows revealed that a 7-day period balanced historical context richness with model complexity, and including data from the same day in the previous week alongside the past 24 hours enhanced performance. Applying the Hodrick-Prescott (HP) filter to the data improved model performance, particularly for short-term predictions, and models trained with both trend and noise components from the filter outperformed those using raw data. Standardization proved superior to normalization, improving model convergence and stability, reducing the impact of outliers, and ensuring proportional feature contributions, enhancing the model's robustness and reliability. We evaluated various neural network architectures, including LSTM, GRU, Multi-Head Attention (MHA), MHA with positional encoding (MHA_R), and a simple dense layer network (SimpleNet). Hyper-parameter optimization using Optuna fine-tuned parameters such as learning rates, batch sizes, and the number of layers and heads in the MHA model, and we implemented a learning rate scheduler and early stopping to prevent overfitting. The MHA model without positional encoding emerged as the best performer, contrary to our expectations, as the attention mechanism alone was sufficient to capture temporal dependencies effectively. This model consistently outperformed others across all metrics, demonstrating the effectiveness of the attention mechanism in our forecasting task. Through these experiments, we identified the optimal input configurations and model architectures for heat usage forecasting, highlighting the importance of careful data organization, appropriate pre-processing, and robust model selection in achieving accurate and reliable forecasts. Further experiments could explore additional data features or alternative neural network architectures to potentially enhance performance even more.

In discussion and analysis chapter, we discuss the results of our experiments on heat usage forecasting in district heating networks, emphasizing lessons learned and new challenges encountered. One major challenge was determining the optimal data representation to avoid persistence issues and achieve accurate results. Our experiments revealed several key findings. Firstly, a 7-day look-back window for input data struck a good balance between capturing historical context and maintaining model simplicity. Secondly, standardization consistently outperformed normalization in pre-processing, leading to better performance across different model configurations. Including future temperature data as an input significantly improved model accuracy. Table 5.3 shows the global metrics obtained from our models, providing a comprehensive evaluation of their performance. Figures 6.1 and 6.2 illustrate the degradation in prediction accuracy as the forecast horizon extends, which is also reflected in the nRMSE plots. These figures highlight the importance of evaluating models across all forecasting hori-

zons rather than specific time steps. We also addressed the evolution of our results with different data pre-processing strategies, as shown in Figure 6.3, which depicts early predictions using only historical demand. This highlighted the critical role of pre-processing in model accuracy. Our Multi-Head Attention (MHA) model outperformed other models, showing strong performance in MAE, RMSE, and Pearson correlation metrics. However, comparing our results to the state-of-the-art, such as Yanis' results, was challenging due to differences in dataset characteristics and metric scales. Table 6.1 compares our MHA model with other models from the literature. Despite some challenges in direct comparisons due to metric dependencies on data magnitude, our MHA model showed promising results. Lastly, due to time constraints, we could not fully implement the Incremental Learning (IL) methodology. This remains an ongoing aspect of our research, with plans to refine and test IL models on DreamNet in the coming months. Preliminary scenarios involving demand pattern and temperature threshold variations will be tested, with results expected to provide insights into the real benefits of using IL for this application.

Finally, we conclude on our research, we addressed the critical challenge of accurately forecasting heat usage in district heating networks, which is vital for efficient energy management and cost reduction. Our initial objective was to develop a robust offline learning model and subsequently adapt DreamNet for an adaptive learning approach. Although we successfully implemented the offline learning model, we could not fully adapt DreamNet due to time constraints. Further testing is necessary to explicitly demonstrate catastrophic forgetting and validate the adaptive learning approach. Despite this limitation, we developed a custom Multi-Head Attention (MHA) mechanism with 2 layers, 2 heads per layer, and a model dimension of 64, setting the stage for future experimentation. Our research involved an extensive exploration of pre-processing techniques, feature sets, and input configurations using the PyTorch library. We employed rigorous training methods, including batch processing, learning rate scheduling, and hyper-parameter optimization, to refine our models. Our evaluations revealed critical insights, such as the effectiveness of a 7-day look-back window, the superiority of standardization over normalization, and the significant benefits of incorporating future temperature data. These findings highlight essential strategies for enhancing the accuracy of heat usage forecasts. While our research represents a significant step towards improved heat usage forecasting, further investigation is needed to fully realize the potential of adaptive learning models. Future work will focus on finalizing the implementation of incremental learning methodologies, testing their robustness in new scenarios, and optimizing model complexity, size, and computational cost for embedded applications. This project meets the criteria for a Masters Research project through its strong research foundation and contributions to the field. Our in-depth analysis of the scientific question, comparative evaluation of pre-processing techniques, and innovative model implementations demonstrate rigorous inquiry and provide valuable insights. Our work addresses a significant gap in the literature, offering practical solutions and reinforcing its relevance and impact as a Masters Research project.

# — 2 —
# Problem Statement, Analysis and State of the Art

## 2.1 Problem statement

As previously mentioned, an accurate prediction of heating demand is crucial for several reasons: improved planning, efficient resource allocation, significant reductions in carbon emissions, and enhanced energy efficiency. Therefore, our focus is on predicting heating district network demand through time series analysis. By leveraging historical data and relevant influencing factors, we aim to ensure a heating supply that can adapt to varying consumption needs. Our research endeavors the development and implementation of advanced time series prediction techniques, aiming to improve the precision of heating demand forecasts and contribute to the sustainability and resilience of district heating networks.

Moreover, our research places particular emphasis on the practical application of IL methods on this topic. As we will see in the following state of the art most research in incremental learning is focused on classification tasks, we aim to explore its impact on regression tasks. We seek to illustrate how variations in data patterns can affect the accuracy of offline learning models, even after fine-tuning, potentially compromising their performance and proving the interest of an IL approach. Indeed, IL offers a solution that is less vulnerable to this challenge. In the context of the heating district domain, we are not only focused on achieving accurate heat demand predictions but also on developing a solution that can be embedded into existing systems. Therefore, we seek for a longer term forecast, 24 hours ahead with a focus on 3-6 hours ahead, that enables the system to adapt its production using slower but greener energy sources, avoiding reliance on fossil fuels. An embedded solution also means that in addition to performance metrics, the computational cost and efficiency of our final solution are also taken into account. An optimal model should strike a balance between accuracy and computational feasibility to ensure that it can operate effectively within the hardware constraints of embedded systems. Additionally, given that the IL framework DreamNet is already quite complex, our approach is to initially test simpler models. This allows us to establish a solid foundation and understand the baseline performance before progressing to more complex models.

### 2.1.1 Time Series Dynamics in Heating Data : Components and Analysis

Heating data is inherently a time series, consisting of sequential data points collected at regular intervals, each associated with a specific timestamp. A typical representation of a time series, [2] $y_t$ is:

$$y_t = f(t) + \varepsilon_t \tag{2.1}$$

In 2.1, $y_t$ denotes the series value at time $t$, $f(t)$ represents a deterministic function encapsulating the underlying trend, seasonality, and cycles, and $\varepsilon_t$ signifies the stochastic noise component.

The function $f(t)$ encompasses various components: the trend ($T_t$), reflecting the long-term progression ($T_t = \beta_0 + \beta_1 t$ for a linear trend); seasonality ($S_t$), depicting regular patterns that repeat at specific intervals ($S_t = \gamma \sin(\omega t)$ for a sinusoidal seasonal pattern); cyclic patterns ($C_t$), representing irregular, long-term cycles observed in the series; and the noise ($\varepsilon_t$), corresponding to random variation, often modeled as white noise with mean zero and variance $\sigma^2$. Therefore, a time series can be decomposed in **??**:

$$y_t = T_t + S_t + C_t + \varepsilon_t \tag{2.2}$$

This type of data is utilized across numerous fields, leading to increasing interest in time series analysis due to its wide applicability and valuable insights. Accurate forecasting of heating demand is pivotal for optimizing district heating systems' operation and management. Forecasting equations for a time series using past values as input are as follows:

$$\hat{y}_{t+1} = f(y_t, y_{t-1}, y_{t-2}, ..., y_{t-h})$$

$$\hat{y}_{t+2} = f(y_{t+1}, y_t, y_{t-1}, ..., y_{t-h-1})$$

$$\vdots$$

$$\hat{y}_{t+n} = f(y_{t+h}, ..., y_{t+1}, y_t)$$

where h is the historical look-back period and n the horizon forecast.

A prevalent challenge encountered in time series forecasting is the phenomenon of model-induced shifts. When models optimize for minimizing error between the predictions and actual outcome, they often resort to a simplistic strategy of predicting the last observed value from the input window. This is because the last observed value is typically the closest to the next value to be predicted, offering a straightforward approach to minimizing errors. However, this practice poses a significant issue as it signifies that the model has not effectively learned from the data. Essentially, the model relies on a rudimentary strategy persistence that fails to capture the underlying patterns or trends in the data. This challenge is evident in the figures presented in [18], showcasing the difficulty in addressing this issue.

### 2.1.2 Overview of the District Heating Network

District heating is a system where heat is generated from a central source, such as a power plant or a dedicated heating facility, and then distributed through a network of insulated pipes to heat multiple buildings within a local area. This method allows for more efficient energy use compared to individual heating systems in each building. The central heating plant can utilize various energy sources, including natural gas, biomass, geothermal energy, or waste heat from industrial processes.

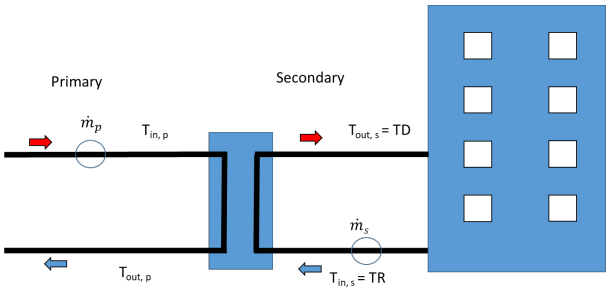


Figure 2.1: A general representation of the district heating substation

We describe the district heating substation in figure 2.1 using specific equations from [1] to determine the power or heat transfer rates. The power $P_p$ in the primary circuit is calculated using the equation2.3, where $\dot{m}_p$ (kg/s) represents the mass flow rate of the carrier fluid in the primary circuit, $c_p$ is the specific heat capacity of the fluid, $T_{\text{in},p}$ is the supplied (inlet) temperature, and $T_{\text{out},p}$ is the return (outlet) temperature. Similarly, the power $P_s$ in the secondary circuit is given by equation 2.4, with $\dot{m}_s$ being the mass flow rate of the secondary fluid, and $T_D$ and $T_R$ representing respectively the out coming temperature from the substation and the incoming temperature in the substation in the secondary circuit. Hence, corresponding to the specific temperature differences relevant to the system.

$$P_p = \dot{m}_p \cdot c_p \cdot (T_{\text{out},p} - T_{\text{in},p}) \tag{2.3}$$

$$P_s = \dot{m}_s \cdot c_p \cdot (T_{\text{out},s} - T_{\text{in},s}) = \dot{m}_s \cdot c_p \cdot (T_D - T_R) \tag{2.4}$$

Furthermore, the power $P_s(t)$ at any given time $t$ in the district heating network can be modeled as in equation 2.5, indicating that $P_s(t)$ is a function of external temperature ($T_{\text{ext}}$), other influencing variables ($H$), and the power values from previous time steps ($P_s(t-1,\ldots,t-d)$). This forms the foundation of our predictive model implementation.

$$P_s(t) = f(T_{\text{ext}}, H, P_s(t-1,\ldots,t-h)) \tag{2.5}$$

where h is the historical look-back.

### 2.1.3 Incremental learning in Machine Learning Paradigms

Incremental Learning (IL) is a machine learning paradigm where the model learns continuously from a stream of data, updating its knowledge over time as new observations become available.

This approach contrasts with traditional batch learning, where the model is trained on a fixed dataset and then applied to new data without further updates.

In IL, there are several variations depending on what aspect of learning is being adapted incrementally, [37]:

**Class Incremental Learning (Class-IL)**: Class-IL focuses on scenarios where the number of classes or categories in the data increases over time. The model must adapt to accommodate new classes without losing its ability to recognize previously learned classes. Traditional models usually become increasingly worse at classifying old classes as it learns new ones. An example could be number recognition's, using the MNIST dataset the model should be able to remember class 0 after learning class 0 and then 1.

**Task Incremental Learning (Task-IL)**: Task-IL requires the algorithm to incrementally learn a sequence of classification tasks (now referred to as 'episodes'). Each task (episode) features different classes, necessitating task identification to determine the applicable classes for a given sample. The algorithm must handle individual tasks (discriminating between classes within an episode) and identify the task (distinguishing between classes from different episodes). Example for the MNIST [1] dataset: Learning about zeros and once initially, then about threes and fours as separate tasks [23].

**Domain Incremental Learning (Domain-IL)**: In Domain-IL, the learning task remains the same, but the domain or distribution of the data changes over time. The model needs to adapt to new environments or contexts while retaining its ability to perform well on previously seen domains. For example, in a weather prediction system, the model might need to adapt to different geographical locations or seasons.

In all these variations of incremental learning, the key challenge is to balance adaptation to new information with the preservation of previously acquired knowledge avoiding catastrophic forgetting. For our research problem we stand in the case of Domain-IL.

## 2.2 State of the Art : Time Series forecasting and IL in District Heating Systems

Having outlined the significance of accurate heating demand prediction and our focus on advanced time series analysis and incremental learning techniques, we now turn our attention to reviewing the existing literature. This review will encompass predictive models for time series specifically applied to district heating systems, as well as the state-of-the-art approaches in incremental learning. By examining these works, we aim to identify the strengths and limitations of current methodologies, thereby informing the development of our innovative predictive framework.

### 2.2.1 Data processing and Forecasting Model Architectures

A well-known issue in time series forecasting is the shift created by the model. When the model aims to minimize its prediction error, it often ends up predicting the last value from the input in

[1] Modified National Institute of Standards and Technology database from : http://yann.lecun.com/exdb/mnist/

the look-back window. This is because the last observed value is typically the closest to the next value to be predicted, making it a safe and easy prediction to minimize errors. However, this approach is problematic because it indicates that the model hasn't learned much from the data. Essentially, the model is relying on the simplest possible strategy persistence which doesn't capture the underlying patterns or trends. we can see such issue in the figures in [18], this is really challenging to overcome and the best way to do so is by not using the real data as input but with some transformation. Hence, a thorough processing of the data is essential, involving for example different techniques like normalization or standardization as well as apply filters. In data preprocessing and signal processing, standardization is commonly employed to transform data to have a mean of 0 and a standard deviation of 1, facilitating algorithms that assume normally distributed data or require standardized feature scales [11]. Min-max normalization scales data to a fixed range, preserving the original distribution while ensuring all features are on the same scale, which is advantageous in neural networks and image processing tasks [40]. High-pass filtering (HP filter) is essential in signal processing to remove low-frequency components while retaining high-frequency variations, thus isolating anomalies and trends in time-series analysis [29]. Advanced techniques such as wavelet transforms and adaptive filters further enhance the efficacy of HP filtering in capturing dynamic signal characteristics.

We chose the Multiple Input, Multiple Output (MIMO) strategy for our approach, [38]. MIMO forecasting is particularly effective, especially given our objective of developing a model suitable for DreamNet [24]. This strategy excels due to its comprehensive approach, utilizing multiple input variables to simultaneously predict multiple output variables. Unlike Single Input, Single Output (SISO), Multiple Input, Single Output (MISO), or Single Input, Multiple Output (SIMO) methods, MIMO forecasting offers unparalleled flexibility in capturing intricate system dynamics [9]. By incorporating various input factors such as future and historical temperature as well as historical heat demand, MIMO models can uncover complex relationships and interactions, potentially leading to more precise forecasts. However, the MIMO approach also introduces increased complexity in model design, training, and interpretation. It requires substantial datasets that encompass all input and output variables, posing challenges in data collection and processing. Despite these complexities, the MIMO strategy shows promise in providing valuable insights into time series forecasting, particularly in domains where understanding the interplay of multiple variables is critical for decision-making.

### 2.2.2 Offline Learning models

The forecast of heat demand (and time series in general) is an increasingly important topic of discussion. Over the years, numerous models utilizing various machine learning and deep learning techniques have been studied for time series forecasting. But historically, deep learning methods initially struggled to outperform classical statistical methods like ARIMA or simpler machine learning approaches in time series forecasting.

RNN-based models have emerged as promising contenders in the field of time series forecasting. These models leverage the inherent sequential nature of time series data, making them well-suited for capturing temporal dependencies and patterns over time. Unlike traditional statistical methods, RNNs have the ability to learn from past observations and use this information to make predictions about future values in the time series [12]. Moreover, ad-

vancements in RNN architectures, such as Long Short-Term Memory (LSTM) networks [21] and Gated Recurrent Units (GRUs), have further enhanced their predictive capabilities. These architectures address the issue of vanishing and exploding gradients commonly encountered in traditional RNNs, enabling them to effectively model long-range dependencies in time series data. The turning point in the field came with the 2019 M4 [26] competition, where the Exponential Smoothing - Recurrent Neural Network (ES-RNN) [36] hybrid model emerged as the winner, surpassing simpler models by significant margins across all metrics. This highlighted the potential of complex and hybrid methods in enhancing forecasting accuracy, although pure machine learning approaches faced issues with overfitting.

Subsequent advancements in the field led to the development of models like Neural Basis Expansion Analysis (NBEATS)[30], which outperformed ES-RNN on various metrics, signaling further progress in time series forecasting. The introduction of NBEATSx [28] further improved forecasting accuracy by incorporating exogenous variables. However, while these models exhibit impressive performance, they may not be suitable for incremental learning goals due to their complexity and resource-intensive nature. Despite the advancements, few comparative studies exist in the literature on District Heating System (DHS). These gaps in the literature motivate our research project, as a comparative study we use an unpublished work from our research laboratory CEA. The work of Yanis Chaigneau [6] is used as a starting point in for research problem. Recent advancements have seen a significant shift towards attention mechanisms, particularly with the introduction of Transformer-based models and their variants like Informer and Encoder-Decoder architectures. These models, leveraging attention mechanisms and activation functions, have shown remarkable performance improvements across various domains including time series forecasting. We aim to try this new approach on the DHS, however, the proposed model in [27], seems too cmplex for our IL goals, so lean toward the use of multi head attention [3].

Transitioning from the literature review on predictive models for time series we look at time series applied to district heating systems, we aim to identify the strengths and limitations of current methodologies. This review will guide the development of our innovative predictive framework for short-term heat demand forecasting within district heating networks. To facilitate our analysis, we turn to the results summarized in Figure **??**, which provides insights into the performance of various machine learning and deep learning models in this domain.

| Reference | Algorithm | Forecast horizon | MAPE | nMAE | RMSE | MAD | R2 |
|---|---|---|---|---|---|---|---|
| [18] | ANN | 72h | 3.2% | | | | \ |
| [19] | AdaBoost | 72h | | 4.7% | | | \ |
| | LSTM | 72h | | 4.1% | | | \ |
| [22] | Strand-based LSTM | 24h | A: 3.48% | B: 6.01% | | | \ |
| | + smoothing | | A: 3.08% | B: 6.38% | | | \ |
| [14] | Kernel Ridge Regression | 24-48h | | | 7.33 kW | 0.63 kW | \ |
| | Linear Regression | 24-48h | | | 9.33 kW | 0.09 kW | \ |
| [35] | FB-Propet and Light GBM | short term | | MAE: 30.88 kW | 50.50 kW | | 0.92 |
| [39] | ENC-DEC LSTM | 24h | 28.14% | | 46 kW | | \ |
| [8] | PB-GRU | 1h | | | | | 0.92 |
| [6] | M-NBEATSx | 24h | | MAE: 0.368 MW | 0.484 MW | nRMSE: 8.48% | \ |
| [6] | XGBOOST | 24h | | MAE: 0.448 MW | 0.579 MW | nRMSE: 10.7% | \ |
| Raport IEA | CNN | 1h | | 4.24% MAE: 39.90 kW | 54.90 kW | RMPSE: 5.84% | \ |

Table 2.1: Summary of forecasting models and their performance metrics

While these techniques have demonstrated promising results, they often struggle to accurately predict peaks or sudden changes in demand, which are critical aspects of interest in forecasting. Moreover, assessing the effectiveness of these models is challenging due to the wide range of metrics used for evaluation. Specifically, many metrics such as MAE and RMSE are data-dependent, making direct comparisons difficult unless computed on identical datasets.

### 2.2.3 Incremental Learning models

In the context of time series, incremental learning appears interesting, as it allows the model to adapt continuously to the evolving data. Thus, it seems to holds promises for forecasting heat demand in district heating networks.

We review some IL frameworks Pietro Buzzega et al. [4] propose Dark Experience Replay (DER) for General Continual Learning. This method integrates rehearsal with knowledge distillation and regularization. It utilizes network logits to maintain consistency with past experiences, thereby enhancing adaptability to shifting data distributions in continuous learning scenarios. Arslan Chaudhry et al. [7] present Tiny Episodic Memories (TEM) as a solution to catastrophic forgetting in neural networks. TEM by periodically revisiting episodic memories of past tasks. This approach aims to balance memory retention with computational efficiency, offering promising results in preserving learned knowledge across multiple tasks without significant performance degradation.

Kim et al. [16] developed a deep learning-based model utilizing LSTM and ARIMA for household water consumption prediction, a domain which is close to forecasting heat demand. These approaches leverage LSTM's capability to handle temporal and sequential data, requiring extensive initial training and periodic re-training to adapt to evolving data patterns. Kenda et al.'s [15] exploration of incremental learning architectures in water management, combines data fusion and LSTM for prediction. Li et al. [20] propose an incremental learning algorithm for SVMs, ensuring model updates while maintaining error within optimal bounds, ideal for non-stationary environments like district heating networks. Rahman et al. [32] introduce a hybrid neural network model blending supervised learning with online updating mechanisms, adept at predicting hourly thermal loads, thus catering to dynamic heat demand prediction. Saeed et al. [34] present an online gradient descent algorithm specialized for non-stationary time series data, facilitating continuous parameter updates. Zheng et al. [41] propose an incremental ensemble learning method using online support vector regression with ensemble techniques, enhancing prediction accuracy through dynamic model updates. Kumar et al. [17] delve into online sequential extreme learning machines (OSELM), designed for real-time heat load prediction, leveraging sequential learning from incoming data streams. Water and heat consumption share challenges regarding real-time data adaptability, resource consumption, long-term performance maintenance, implementation complexity, and flexibility. Advancements in one area often inform progress in the other, emphasizing the need for integrated approaches and continuous refinement.

In our specific field of application, research on the performance of online deep learning methods is scarce. Table 2.2 provides insights into two methods identified in the literature for our use case: Experience Replay (ER) with the Fast Incremental Model Trees with Drift

Detection (FIMT-DD) algorithm. ER aims to enhance learning efficiency by reusing previously stored experiences, albeit its effectiveness is hindered by escalating memory usage associated with the expansion of the buffer size. This drawback makes ER impractical for training on embedded systems due to constraints in computational time and memory storage, thus making DreamNet an alternative solution. Conversely, FIMT-DD serves as the cornerstone model, offering swift incremental learning capabilities while detecting shifts in data distribution over time.

| **Reference** | **Algorithm** | **Horizon** | **MAPE** | **nMAE** | **RMSE** | **MAD** |
|---|---|---|---|---|---|---|
| [14] | ODL-ER | 24-48h | \ | \ | 8.64 kW | 0.31 kW |
| [31] | FIMT-DD | 36h | 4.77 % | \ | \ | \ |

Table 2.2: Performance metrics of different forecasting algorithms

Key takeaways underscore the significance of adaptability, performance, and applicability of these methods. Techniques like online SVMs, adaptive neural networks, and ensemble methods elevate predictive performance by seamlessly integrating new information. These methods could be applied in district heating networks, where real-time prediction and adaptation are paramount for efficient energy management. In conclusion, the integration of incremental learning techniques into heat demand prediction models offers substantial advantages in adaptability and accuracy. The state of the art has enabled us to draw conclusions on various techniques that could be employed and has provided the following theoretical foundation for our proposed solution.

# — 3 —
# Theoretical Foundations for the Solution

In our research on time series forecasting, we aim to develop an adaptive learning solution. To achieve this, we first address the problem within an offline learning context. Once a viable solution is found, we will adapt the model for incremental learning, specifically incorporating it into DreamNet. Additionally, exploring other adaptive approaches, such as Experience Replay (ER), Dark ER and Tiny Episodic Memories (TEM), as mentioned in the state of the art, would be valuable. However, due to time constraints we did not have the opportunity to implement and test these methods. In this chapter, we describe the theoretical framework underlying the solution we used in our approach.

## 3.1 Foundations of the data pre-processing

The data we have access to includes demand, TD, TR, and outside temperature over two years for each substation. Our approach prioritizes the utilization of directly accessible data, avoiding factors like $\dot{m}_s$ ,$T_D$ and $T_R$,2.1 which are hardly predictable and aren't directly accessible at the substation level. By focusing on the demand and temperature data directly obtained from the substations, we ensure a practical and reliable foundation for our forecasting model.

The input data is generated using a sliding window technique with a specified look-back window size. As a result, at each time step, the input data should have the shape

$$[\text{batch size}, \text{look-back window} \times 2 + \text{forecast horizon}]$$

. The forecast horizon is included because we incorporate the future temperature at a given time step to predict the heating demand at that same time step. To address the problem of persistence mentioned in the state of the art, in our approach we pre-process the data. We transform the data using the Hodrick-Prescott filter along with a standardization process. This helps in presenting a modified form of the data, improving the model's ability to learn and make accurate predictions.

### 3.1.1 Hodrick-Prescott filter

The Hodrick-Prescott (HP) filter [13] is a mathematical tool used to separate a time series into a trend component and a cyclical component. The time series $y_t$ can be decomposed in 3.1,

$$y_t = \tau_t + c_t \tag{3.1}$$

where $y_t$ is the observed time series, $\tau_t$ is the trend component and $c_t$ is the cyclical component.

The HP filter estimates the trend component $\tau_t$ by minimizing the objective function 3.2,

$$\min_{\tau_t} \sum_{t=1}^{T} (y_t - \tau_t)^2 + \lambda \sum_{t=2}^{T-1} [(\tau_{t+1} - 2\tau_t + \tau_{t-1})^2] \tag{3.2}$$

where $T$ is the total number of observations in the time series, $\lambda$ is a smoothing parameter that controls the trade-off between the smoothness of the trend and the fit to the data. Higher values of $\lambda$ result in a smoother trend.

The first term in the objective function, $\sum_{t=1}^{T}(y_t - \tau_t)^2$, represents the sum of squared deviations of the observed data from the estimated trend. The second term, $\lambda \sum_{t=2}^{T-1}[(\tau_{t+1} - 2\tau_t + \tau_{t-1})^2]$, represents a penalty on the second differences of the trend, penalizing excessive fluctuations in the trend. By minimizing this objective function, we obtain the estimated trend component $\hat{\tau}_t$. The cyclical component $c_t$ is then obtained as the difference between the observed data and the estimated trend as shown in 3.3

$$c_t = y_t - \hat{\tau}_t \tag{3.3}$$

Overall, the filter separates a time series into a trend component that represents the long-term movements and a cyclical component that represents the short-term fluctuations around the trend.

### 3.1.2 Min-Max Normalization

Min-Max normalization [40] is a technique used to scale data within a specific range, typically [0, 1]. This method transforms the original values linearly, the mathematical formula for Min-Max normalization is given by equation 3.4

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{3.4}$$

where $x$ is the original value, $\min(x)$ is the minimum value in the dataset, $\max(x)$ is the maximum value in the dataset, and $x'$ is the normalized value. This technique is particularly useful when the data features have different scales and units, as it brings them to a common scale without distorting the differences in the ranges of values. Min-Max normalization is widely used in machine learning pre-processing to ensure that the features contribute equally to the model training process, improving the convergence and performance of many algorithms.

### 3.1.3 Standardization

Standardization [11] is a process used in data pre-processing to transform the features of a dataset so that they have a mean of zero and a standard deviation of one. This ensures that the features contribute equally to the analysis, eliminating biases due to differences in scale. Mathematically, standardization is performed using the equation 3.5

$$z = \frac{x - \mu}{\sigma} \tag{3.5}$$

where $z$ is the standardized value, $x$ is the original value, $\mu$ is the mean of the feature, and $\sigma$ is the standard deviation of the feature. By applying this transformation, each feature in the dataset will have a mean of 0 and a standard deviation of 1, making the data more suitable for various machine learning algorithms that assume normally distributed data or are sensitive to the scale of the input features.

### 3.1.4 Forcasting stategy : MIMO

The Multiple Input, Multiple Output (MIMO) strategy extends time series forecasting by incorporating multiple input and output variables simultaneously. This method captures the interactions between multiple time series and predicts multiple future values in a single model.

Consider a multivariate time series with $k$ input variables $\mathbf{X}_t = [x_{1,t}, x_{2,t}, \ldots, x_{k,t}]^T$ and $m$ output variables $\mathbf{Y}_t = [y_{1,t}, y_{2,t}, \ldots, y_{m,t}]^T$. A MIMO [9] model can be expressed in 3.6

$$\mathbf{Y}_{t+1} = f(\mathbf{X}_t, \mathbf{X}_{t-1}, \ldots, \mathbf{X}_{t-p}, \mathbf{Y}_t, \mathbf{Y}_{t-1}, \ldots, \mathbf{Y}_{t-q}) + \varepsilon_t \tag{3.6}$$

where $f(\cdot)$ is a function capturing the relationships between past inputs and outputs, $p$ and $q$ are the lag orders for inputs and outputs, respectively, $\varepsilon_t$ is a vector of error terms.

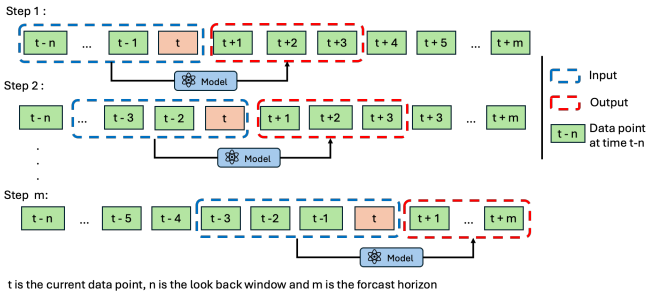


Figure 3.1: A representation of the MIMO strategy

This MIMO approach is advantageous because it can capture the interdependencies between multiple time series and allows for simultaneous prediction of multiple variables, enhancing forecast accuracy and consistency.

## 3.2 Offline Learning

We developed various offline learning models to predict heating demand and compared their accuracy. From the state of the art, we selected three models that struck a balance between size and accuracy for our needs. Specifically, we chose to study an LSTM, a GRU, and a Multi-Head Attention model, which had been tested in Transformers and Informer architectures but not as standalone model.

### 3.2.1 Long-Short Term Memory

Long Short-Term Memory (LSTM) networks are a potent tool for time series forecasting due to their capacity to capture long-term dependencies and manage sequential data effectively.

Unlike traditional recurrent neural networks (RNNs), LSTMs incorporate a memory cell that can retain information over extended periods, allowing them to recall past observations and context vital for making precise predictions in time series data. This is facilitated by gating mechanisms within LSTMs, including the input gate ($i_t$), forget gate ($f_t$), and output gate ($o_t$), which regulate the flow of information through the network and enable selective updating of the memory cell. The equations governing these gates are as follows:

$$\begin{aligned} I_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i), \\ F_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f), \\ O_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o). \end{aligned} \tag{3.7}$$

Additionally, LSTMs adeptly manage the flow of gradients during training, mitigating the vanishing gradient problem commonly encountered in deep learning models. This capability is facilitated by the LSTM's architecture, which includes the memory cell ($c_t$) and the hidden state ($h_t$), updated according to the following equations:

$$\begin{aligned} C_t &= F_t \odot C_{t-1} + I_t \odot \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\ H_t &= O_t \odot \tanh(C_t) \end{aligned} \tag{3.8}$$

These equations enable LSTMs to effectively manage long sequences of data without losing critical information, making them versatile and suitable for various types of time series data, from stock price prediction to weather forecasting. Overall, the combination of memory cells, gating mechanisms, and gradient flow control renders LSTMs highly effective for capturing complex patterns and dependencies in time series data, making them a popular choice for time series forecasting tasks across different domains.

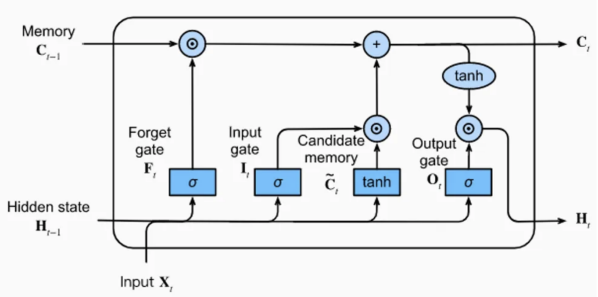


Figure 3.2: Architecture of LSTM [1]

### 3.2.2 Gated Recurrent Unit

Gated Recurrent Unit (GRU) networks are another powerful tool for time series forecasting, known for their simpler architecture compared to LSTMs while still being effective at capturing long-term dependencies and handling sequential data. GRUs combine the functionalities of the input and forget gates in LSTMs into a single update gate ($z_t$) and simplify the memory cell

[1] Image from: https://medium.com/@ottaviocalzone/an-intuitive-explanation-of-lstm-a035eb6ab42c

update process. The equations governing the update gate and the candidate hidden state ($h'_t$) in GRUs are as follows:

$$\begin{aligned} z_t &= \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z) \\ r_t &= \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r) \\ h'_t &= \tanh(W_{xh}x_t + W_{rh}(r_t \odot h_{t-1}) + b_h) \end{aligned} \tag{3.9}$$

Here, $r_t$ represents the reset gate, which controls how much of the previous hidden state to forget. The candidate hidden state $h'_t$ is computed using both the current input and the reset gate-modulated previous hidden state.

Subsequently, GRUs employ a gating mechanism to combine the current candidate hidden state $h'_t$ with the previous hidden state $h_{t-1}$ to produce the updated hidden state $h_t$:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot h'_t \tag{3.10}$$

This equation allows the GRU to selectively update the hidden state based on the update gate $z_t$, thereby determining how much information from the candidate hidden state should be incorporated into the new hidden state.

GRUs effectively manage the flow of gradients during training, similar to LSTMs, enabling them to learn from long sequences of data without encountering the vanishing gradient problem. Their simpler architecture and comparable performance make them an attractive option for time series forecasting tasks across various domains.

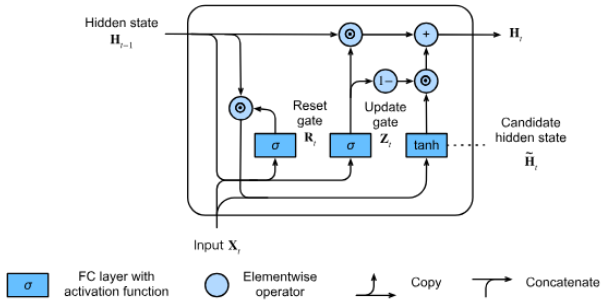


Figure 3.3: Architecture of GRU [2]

### 3.2.3 Multi-Head Attention for Time Series

Multi-Head Attention (MHA) is a key component of Transformer models, renowned for its ability to capture global dependencies in sequential data, making it a potent tool for time series forecasting. MHA operates by computing attention scores between each element in a sequence, allowing the model to focus on relevant information while filtering out noise. The core idea of MHA is to compute multiple attention heads in parallel to capture diverse aspects of the data and enhance model expressiveness.

The attention mechanism in MHA involves three main steps: computing query ($Q$), key ($K$), and value ($V$) matrices from the input sequence. These matrices are then used to calculate

[2]Image from: https://d2l.ai/chapter$_r ecurrent - modern/gru.html$

attention scores, which determine the importance of each element in the sequence. The attention scores are subsequently used to weight the values, producing the output of the attention mechanism.

The attention mechanism can be mathematically represented as follows:

Given an input sequence $X$ of length $N$, the query ($Q$), key ($K$), and value ($V$) matrices are computed as linear transformations of $X$:

$$\begin{aligned} Q &= XW_Q \\ K &= XW_K \\ V &= XW_V \end{aligned} \tag{3.11}$$

where $W_Q$, $W_K$, and $W_V$ are learnable weight matrices.

The attention scores ($A$) are then computed as the dot product of the query and key matrices, scaled by the square root of the dimension of the key vectors ($d_k$):

$$A = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \tag{3.12}$$

The output of the attention mechanism ($Y$) is obtained by multiplying the attention scores by the value matrix:

$$Y = AV$$

This process is typically repeated multiple times with different sets of learnable weight matrices to compute multiple attention heads in parallel. The outputs of these attention heads are concatenated and linearly transformed to produce the final output of the MHA layer.

MHA's ability to capture complex dependencies in sequential data, combined with its parallel computation of multiple attention heads, makes it a powerful tool for time series forecasting tasks, enabling models to effectively learn and exploit temporal patterns in the data.

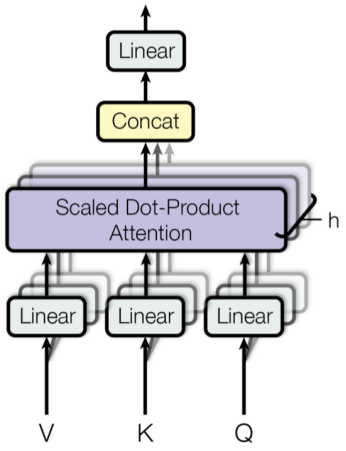


Figure 3.4: Architecture of MHA [3]

To enhance our Multi-Head Attention (MHA) mechanism, we add a positional encoding layer that embeds positional information into the input tokens.

[3]Image from: https://paperswithcode.com/method/multi-head-attention

Positional encoding is a technique used in transformer models to incorporate the order of input tokens, which is crucial for sequence processing tasks. Unlike recurrent neural networks (RNNs), transformer models do not inherently account for the positions of tokens in a sequence. Positional encoding provides a way to inject information about the positions of tokens into the model.

Mathematically, positional encoding involves creating a set of vectors that are added to the input embeddings to provide the model with information about the position of each token. The used positional encoding scheme involves sine and cosine functions of different frequencies.

For a sequence of length $n$ and an embedding dimension $d$, the positional encoding vector for a position $pos$ and a dimension $i$ is defined as follows:

$$\begin{aligned} \mathrm{PE}_{(pos,2i)} &= \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) \\ \mathrm{PE}_{(pos,2i+1)} &= \cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) \end{aligned} \tag{3.13}$$

Here $pos$ is the position in the sequence (ranging from 0 to $n-1$), $i$ is the dimension index (ranging from 0 to $d-1$), $\mathrm{PE}_{(pos,2i)}$ is the positional encoding for the even dimensions, $\mathrm{PE}_{(pos,2i+1)}$ is the positional encoding for the odd dimensions.

The rationale behind using sine and cosine functions is that they provide unique values for each position and dimension, enabling the model to distinguish between different positions in the sequence. Additionally, these functions have desirable mathematical properties that help the model generalize to longer sequences than those seen during training.

The positional encoding vectors are added to the input embeddings before feeding them into the transformer model:

$$\mathrm{Input}_{\text{with PE}} = \text{Input Embeddings} + \text{Positional Encodings}$$

This addition allows the model to incorporate position information into the learned representations, enabling it to understand the order and relative positions of tokens in the sequence.

## 3.3 Foundations of the Incremental Learning approach

For the IL implementation, we utilize the same pre-processing techniques as previously described, given that there are no constraints on applying these methods incrementally. This includes the computation of the filter as well as standardization and min-max normalization. However, it remains crucial to empirically validate this theoretical approach to ensure its effectiveness in a the IL setting.

### 3.3.1 The scenarios

To test the adaptive implementation, we need to establish incremental scenarios that mirror real-life situations. Due to insufficient data and limited understanding of the existing data, we

opted to generate new one from the real data to facilitate the application of these scenarios. The generated data is represented as in Figures 4.8, 4.9

where each block is the incoming data over time for our incremental learning approach. The labels on the data in 3.5 allows us to track where the data comes from in the time line of the scenario. In real life this data structure would be represented by a sudden change in the demand in block 2 thus the model retrains on it and then the consumption comes back to normal in block 4. In this situation, we want the model to able to remember the "normal" pattern as well as the sudden change in case it happens again. This is why our scenarios have been though such that the new patterns in the demand are not just punctual events but recurrent ones in real life.

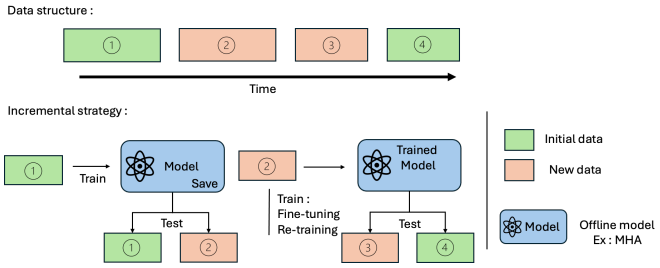


Figure 3.5: Theoretical data and scenario representation

This data structure is used for the incremental model as well as also for the offline model one, where it first learns from unmodified data and attempts to predict both unmodified and modified instances. This will serve as our upper bound for evaluating the adaptive model's performance. Then, we fine-tune the offline model on the modified data, as depicted in the incremental strategy in Figure 3.5, then test again on both unmodified and modified instances. Theoretically, the outcome of such experience is that : after retraining, the model should have better performance on the modified data, while its performance on the unmodified data deteriorates. This phenomenon is commonly referred to as catastrophic forgetting.

### Scenario 1 : home working due to temperature constraints

In this scenario, our objective is to establish a recurring pattern based on the outside temperature. The thought behind this is that, in real life, if the outside temperature is low, people might tend to work from home. Low temperatures could also indicate severe weather conditions, such as snow, making it difficult to go to work. To achieve this, we modify the data by identifying all days where the average temperature falls below 5 degrees Celsius. On these identified days, we replace the residuals on the given day, which represent the social behavior in the consumption, with those from weekends. This modification aims to incorporate a consistent pattern in the data based on temperature variations, mimicking the effects of remote working. We chose this temperature threshold, even though it is not highly realistic, due to the limited availability of data.

To implement this, we perform on the weekend days a linear regression to determine the power consumption $P(k)$ as a function of the external temperature $T_{\text{ext}}(k)$ as shown in 3.14 coming from [1].

$$P(k) = a \cdot T_{\text{ext}}(k) + b + \text{res}_{\text{we}}(k) \tag{3.14}$$

Here, $k \in [0, 48]$ represents the half-hour index of the day, and $\text{res}_{\text{we}}(k)$ is the "sociological profile" residual, reflecting human behavior patterns that do not depend on weather (e.g., whether people are at home or not).

Then the weekdays with average temperatures below 5 degrees Celsius, also we use the previously computed a and b and replace $\text{res}_{\text{sem}}(k)$ with $\text{res}_{\text{we}}(k)$ to simulate the effect of the weekend sociological profile on these days and compute the demand. By applying this approach, we create a modified dataset that reflects a recurrent pattern influenced by outside temperature, allowing us to test the model's ability to adapt and predict under these altered conditions.

#### Scenario 2 : home working specific to Wednesdays

In this scenario, we introduce a modification to the data by amplifying the demand on weekends by a certain factor and then after a certain time we also multiply by that same factor Wednesdays. The rationale behind this adjustment is to mimic people staying at home during weekends and the effects of increased remote work on Wednesdays day. By boosting the demand, we seek to simulate the behavioral shifts associated with a higher number of individuals working from home on Wednesdays. This modification enables us to delve into how the model responds to and performs under these altered conditions, offering valuable insights into its resilience and adaptability in real-world settings influenced by evolving work habits.

### 3.3.2 DreamNet

DreamNet [25] is indeed a privacy-preserving model, utilizing an innovative approach to continual learning while preserving privacy. Its architecture employs two interconnected networks, the Learning Net and Memory Net, to facilitate continual learning without forgetting previous knowledge. The process initiates with the Learning Net, which learns real features from the current class of data and pseudo-features from previously learned classes. These pseudo-features are generated by the Memory Net through a reinjection sampling procedure, wherein random noise is introduced and the resulting output is reinjected multiple times to generate pseudo-examples as we can see in Figure 3.6. The Memory Net essentially serves as a repository for storing the weights of the model after learning from previous tasks or datasets. This stored information enables the model to recall and utilize prior knowledge when learning new tasks, thus mitigating the issue of catastrophic forgetting.

The architecture's unique feature lies in its Auto-Hetero associative Artificial Neural Network (ANN) structure, which combines auto-encoding and supervised classification capabilities. This combination allows the model to replicate input information while effectively classifying data. By generating pseudo-examples from previously learned classes, DreamNet ensures the preservation of learned functions over multiple learning cycles without the need for retaining actual data from previous classes. This innovative framework not only facilitates continual learning but also prioritizes privacy by operating in a data-free manner, thus offering a promising solution for various applications requiring both continual learning and privacy preservation.

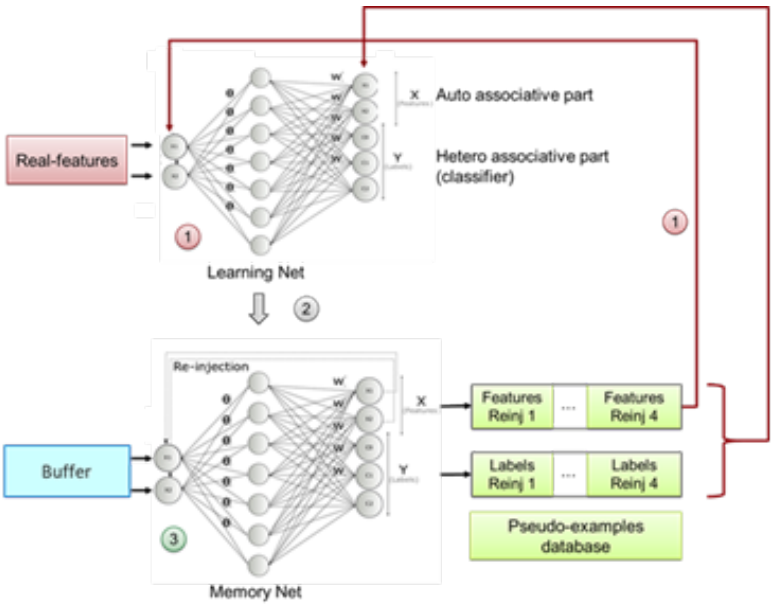


Figure 3.6: Architecture of DreamNet form [25]

Our aim is to tailor DreamNet to suit our regression problem by substituting the Artificial Neural Network (ANN) with our Multi-Head Attention (MHA). To achieve this, we require an auto-hetero output mechanism tailored specifically for MHA. Before proceeding with adapting DreamNet for our regression, it's essential to establish the presence of catastrophic forgetting and demonstrate the relevance of incremental learning and DreamNet for our solution. We use the following scenarios to do so.

# — 4 —
# Practical implementation

In this chapter we explain the implementation of the previous theoretical approches, to do so we established a pipeline, Figure 4.1. Within our pipeline, we outline a sequence of steps commencing with data pre-processing, followed by model creation and hyperparameter optimization, training, model testing, and evaluation, culminating in the analysis of the obtained results. Each of these steps is detailed in the sections of this chapter.

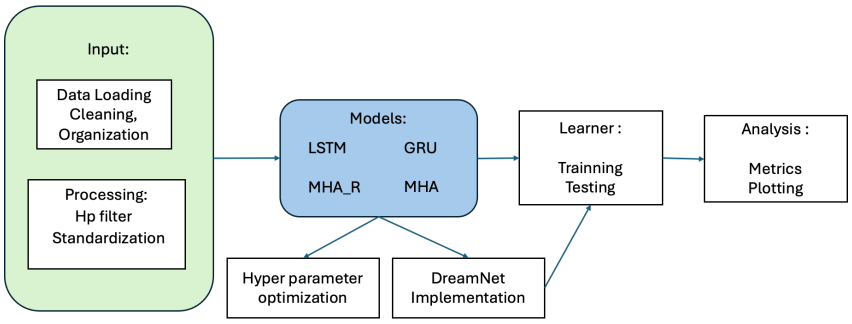


Figure 4.1: Description of the pipeline we follow for our implementation

The implementation of the different models and data processing,is done in Python, we use its extensive libraries for data analysis and machine learning tasks.

## 4.1 The Data

We analyzed of the substation data (figure 4.2, followed by the loading the dataset. We cleaned the data by addressing missing values through interpolation and pre-processed it, including organizing the feature and standardization. Finally, we partitioned the dataset into training, validation, and testing sets to facilitate model training and evaluation.

### 4.1.1 Data anlalysis

We utilized the Python library seasonal_decompose to conduct an in-depth analysis of our dataset, implementing a lag of 12 hours to capture the recurring patterns effectively. Seasonal decomposition dissects the time series data into its essential components: trend, seasonal, and residual. This allowed us to gain valuable insights into the underlying structure of the data.

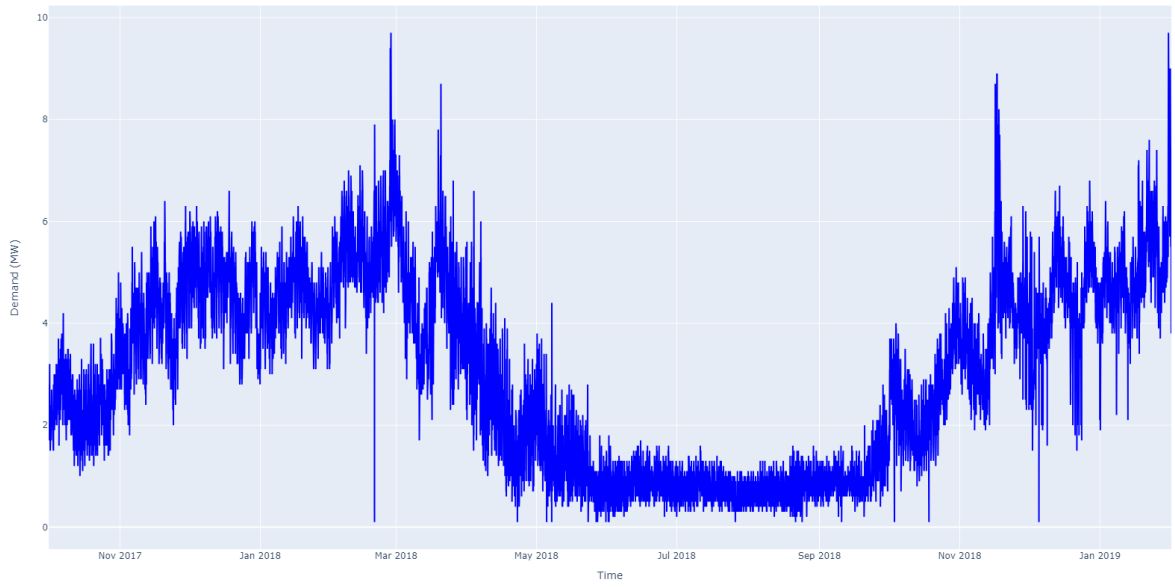


Figure 4.2: Substation data representation

The analysis, depicted in Figure 4.3, highlighted a recurrent pattern in the data occurring every 12 hours, aligning with the alternating day and night consumption cycles typically observed in heating demand.

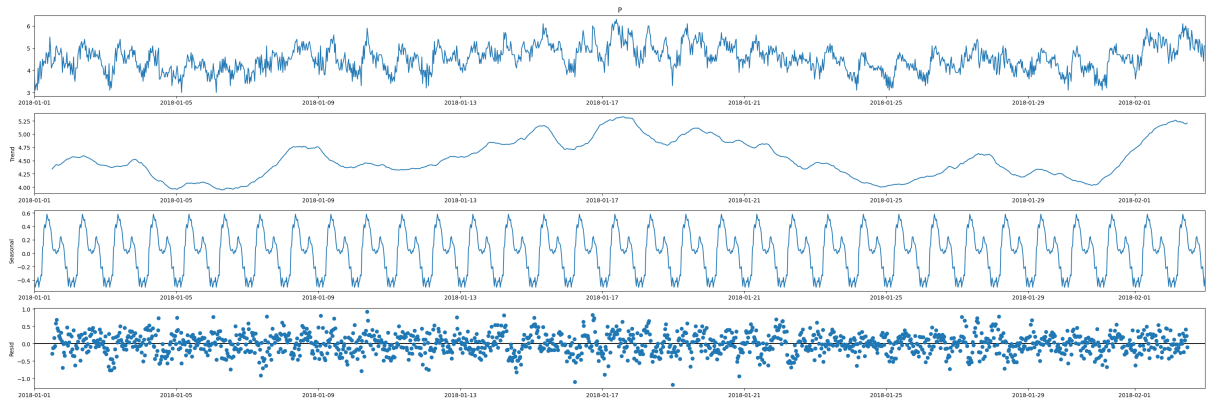

Figure 4.3: Representation of the decomposition with a lag of 12

We conducted a linear regression analysis on the data and calculated the average residuals for each day of the week. These residuals represent the sociological pattern in heat consumption. As depicted in Figure 4.4, we observed distinct behaviors on Wednesdays and weekends compared to other days of the week, aligning with our expectations. This analysis is instrumental in generating artificial data for the incremental scenarios, providing valuable insights into the daily variations in heat consumption patterns.

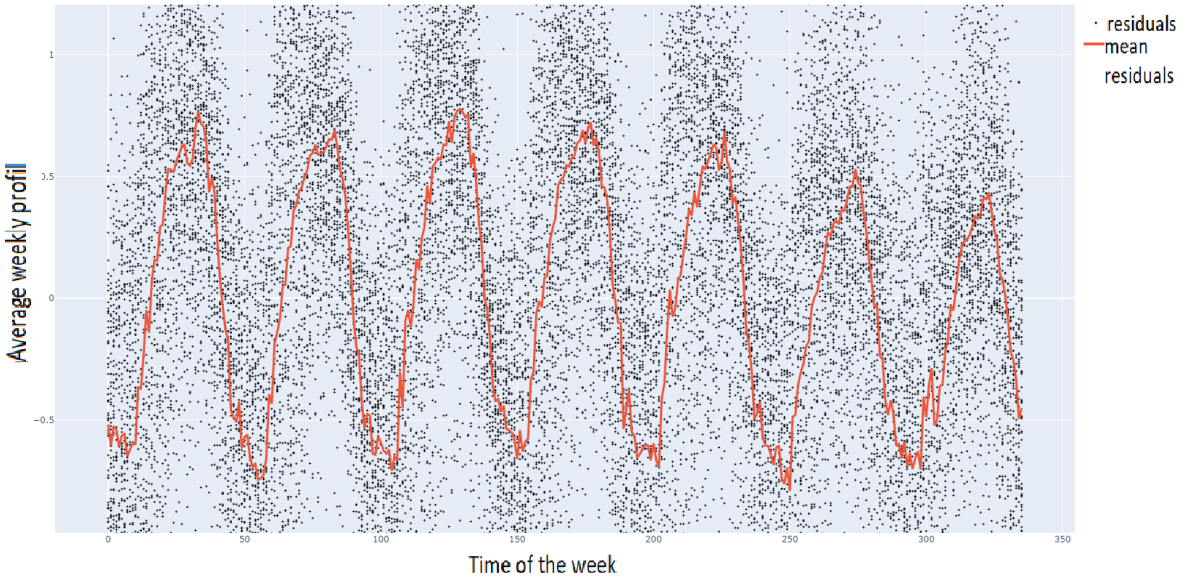


Figure 4.4: Representation of the weekly social habits in the substation

### 4.1.2 Data Processing

To forecast heat usage accurately, especially considering the multi-functional nature of the heating network supplying both heating and hot water for sanitary purposes, we narrow our focus to the winter months.

In the data processing stage, we apply the Hodrick-Prescott (HP) filter to the data to extract the underlying trend (figure 4.5). From this filtered data, we concatenate the trend, residual, and temperature for each step of the look-back window, along with the future temperature for the predicting horizon. This comprehensive approach ensures that our predictive model incorporates essential factors influencing heat usage. We did multiple test on the lamb to choose for the filter and chose lamb = 6.25 , which corresponds to (1600/4**4) for annual data suggested by Ravn and Uhlig [33](see Annexes).

The filtered data is then standardized to ensure that the features are on a similar scale, which aids in the convergence of the optimization algorithm during model training. This process is essential for models that are sensitive to the scale of input features, as it prevents certain features from dominating the learning process due to their larger magnitudes, in our case the temperature. By standardizing the data, we ensure that each feature contributes proportionally to the model's learning process, resulting in more stable and reliable predictions.

The data is typically split as illustrated in Figure 4.6 for the majority of tests. However, in the incremental scenario, this splitting approach may need adjustment due to data scarcity. Additionally, to improve the accuracy of our predictions, we explore various input settings. We experiment with different combinations of past historical demand and temperature data, as well as future temperature projections. Although positional encoding did not yield significant improvements, we believe that integrating the demand and temperature data from the same day

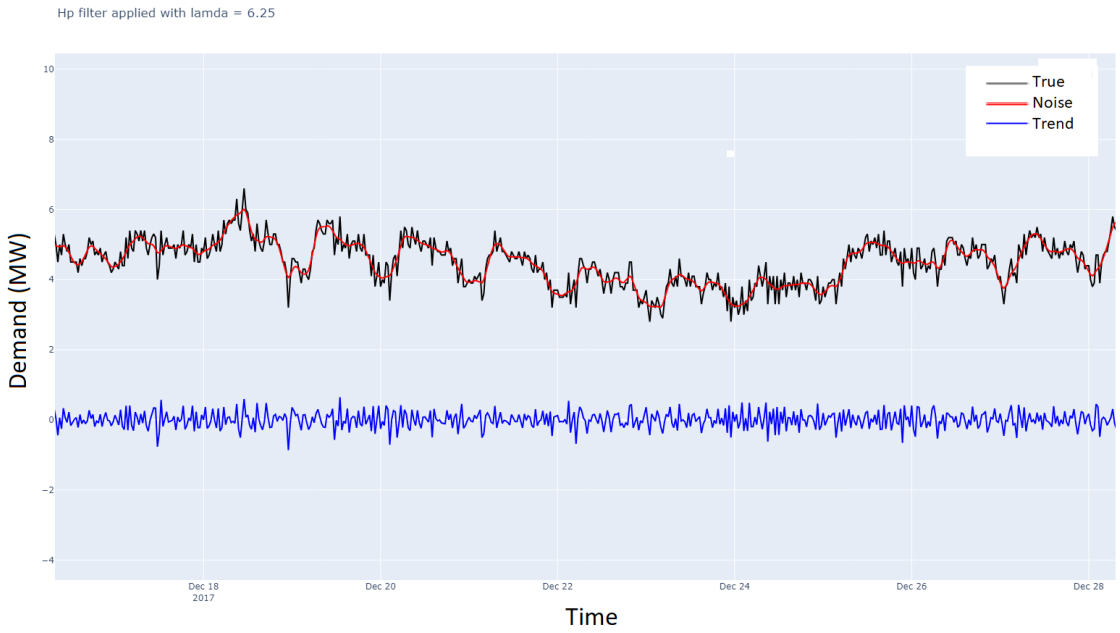


Figure 4.5: Representation of the computation of the filter on the data

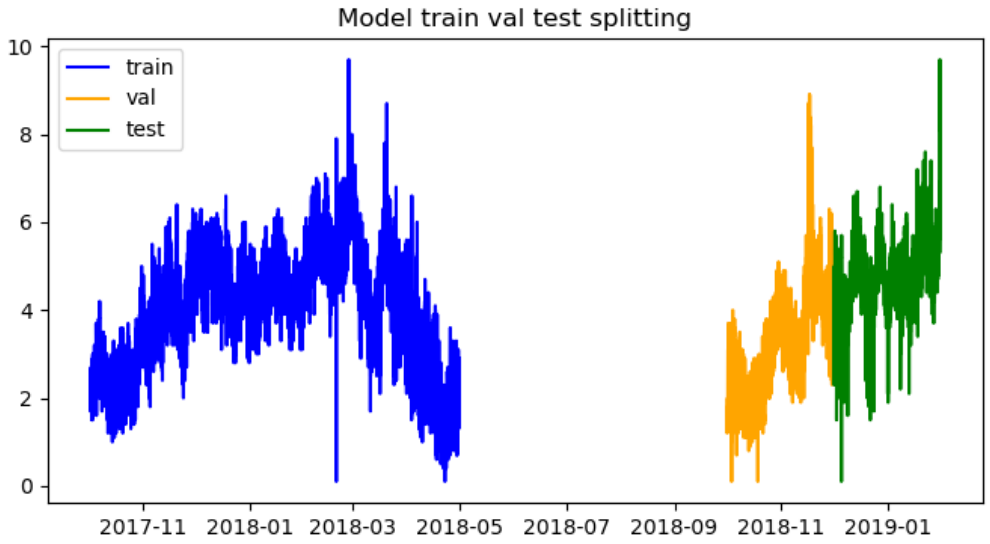


Figure 4.6: A representation of the splitting of the data

of the week from the previous week could enhance the performance of our models, particularly for the Multi-Head Attention (MHA) architecture. This improvement has the potential to offer the model valuable contextual information and historical patterns, it represents a favorable compromise between providing the model with the past seven days data and solely relying on the past 24 hours. Excessive input, could decrease the model's peformance as well as add unnecessary complexity, which is undesirable for DreamNet's architecture.

### 4.1.3 Scenarios Implementation

As described previously, we establish two distinct scenarios that simulate real-life changes in the available dataset. This approach allows us to compare real data with synthetic data, facilitating the identification of these changes more efficiently.

To implement the first scenario, we began by ensuring that the dataset included days with an average temperature below five degrees Celsius. Figure 4.7 illustrates the average temperature for each day, highlighting the days that meet this criterion.


Figure 4.7: Representation of the mean temperature by day

In the first scenario, depicted in Figure 4.8, we observe a specific moment in time where the data containing temperatures below five degrees is changed. This scenario will helps see if the model is capable of adapting to such a change given that it has access to the temperature.

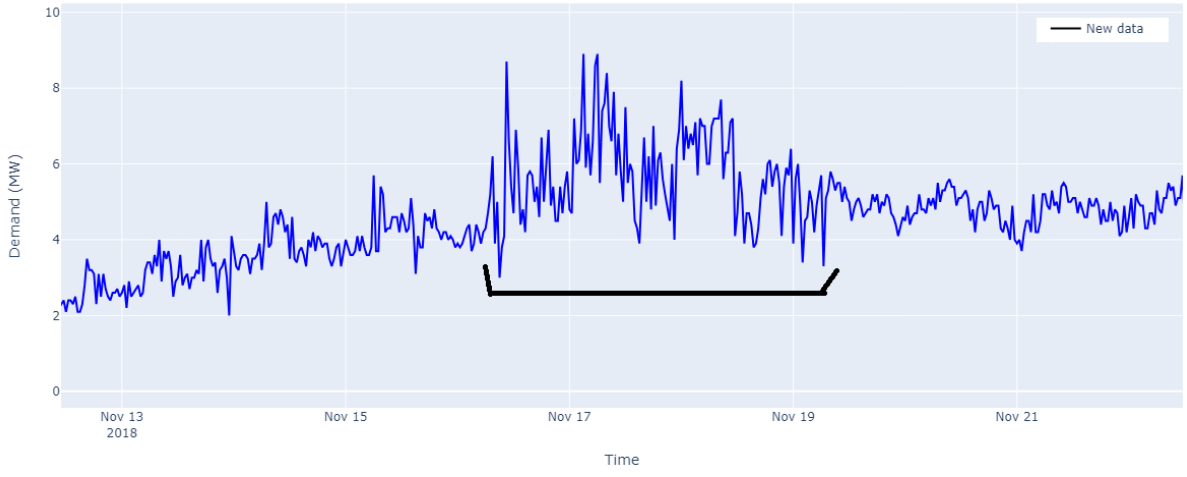


Figure 4.8: Representation of the scenario 1

The second scenario is illustrated in Figure4.9, where we simulate an increase in demand on weekends by using a factor of 1.5 and observe the impact of increasing the demand on

Wednesdays starting at a specific time step. This scenario is designed to reflect realistic patterns in data usage and consumption, which often fluctuate based on weekly cycles and specific days.

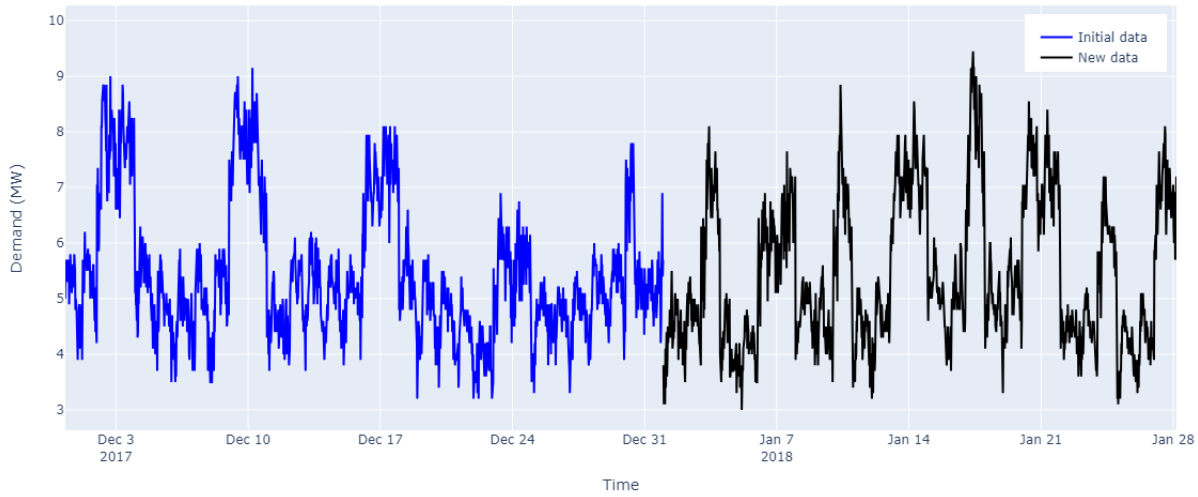


Figure 4.9: Representation of the scenario 2

By thoroughly testing these scenarios, we aim to develop a robust framework capable of adapting to real-world changes in data availability, enhancing the accuracy and reliability of our predictive models.

## 4.2 The models

For implementing the models, we relied on the PyTorch library, utilizing its functionalities for LSTM and GRU architectures. Additionally, we developed our own implementations of Multi-Head Attention (MHA) and positional encoding. This approach allowed us to tailor the models to our specific requirements and experiment with different configurations to optimize performance. By coding our own MHA and positional encoding, we gained a deeper understanding of these components and adapted them to suit the characteristics of our dataset and the objectives of our heat usage forecasting task. This hands-on approach facilitated fine-tuning and experimentation, enabling us to derive insights and refine our models for improved predictive capabilities.

### 4.2.1 Architecture

The table 5.2 provides a comprehensive summary of the various architectures of our models. Each model architecture is designed to handle different aspects of time series forecasting, leveraging distinct neural network components and configurations. The table includes detailed descriptions of the Multi-Head Attention (MHA) based models, Long Short-Term Memory (LSTM) networks, Gated Recurrent Unit (GRU) models, and the enhanced MHA with positional encoding (MHA_R) as well as a simple model made of dense layer for comparison. The specifics of each architecture, including the layers and their configurations, are outlined to provide a clear understanding of the structural differences and unique features of each model.

| Model | Details |
|---|---|
| LSTM | (lstm): LSTM(input_size, hidden_size, num_layers, batch_first=True)<br>(fc_out): Linear(in_features=hidden_size, out_features=output_size, bias=True) |
| GRU | (gru): GRU(input_size, hidden_size, batch_first=True)<br>(linear): Linear(in_features, out_features, bias=True)<br>(relu): ReLU()<br>(out): Linear(in_features, out_features, bias=True) |
| MHA | (mha_layers): ModuleList<br>(0-1): 2 x MultiHeadAttention<br>(W_q): Linear(in_features, out_features, bias=True)<br>(W_k): Linear(in_features, out_features, bias=True)<br>(W_v): Linear(in_features, out_features, bias=True)<br>(W_o): Linear(in_features, out_features, bias=True)<br>(linear): Linear(in_features, out_features, bias=True)<br>(output_layer): Linear(in_features, out_features, bias=True) |
| MHA_R | (positional_encoding): PositionalEncoding()<br>(linear): Linear(in_features, out_features, bias=True)<br>(out) : MHA model |
| SimpleNet | (fc1): Linear(in_features, out_features, bias=True)<br>(sig): Sigmoid()<br>(fc2): Linear(in_features, out_features, bias=True)<br>(relu): ReLU()<br>(fc3): Linear(in_features, out_features, bias=True) |

Table 4.1: Architecture of the different models

In our quest to optimize the performance of our models, we employed two powerful techniques: Optuna and grid search. Optuna is a hyperparameter optimization framework that automates the search for optimal hyperparameters. It works by iteratively exploring the hyperparameter space, guided by the results of previous trials, to identify the combination of hyperparameters that yields the best performance. Optuna uses a technique called Bayesian optimization, which efficiently balances exploration and exploitation to converge on the optimal solution with minimal computational resources. By leveraging Optuna, we were able to systematically search the hyperparameter space and identify the optimal configuration for our models, leading to improved predictive accuracy and performance. Grid search, on the other hand, is a brute-force technique that exhaustively searches the hyperparameter space by evaluating every possible combination of hyperparameters. While grid search is computationally expensive and less efficient than Bayesian optimization, it guarantees finding the best combination of hyperparameters within the specified search space. By performing grid search, we were able to comprehensively explore the hyperparameter space and evaluate the performance of different configurations. This allowed us to gain insights into how different hyperparameters affect the model's performance and identify the optimal configuration for our specific task.

Overall, by combining Optuna and grid search, we were able to fine-tune our models and optimize their performance for heat usage forecasting. These techniques enabled us to efficiently explore the hyperparameter space, identify the optimal configuration, and achieve superior predictive accuracy, ultimately enhancing the effectiveness of our models in real-world

applications.

### 4.2.2 Training and validation phase

In our training process, we employed several strategies to optimize training efficiency and prevent overfitting. Firstly, adopting a batch training approach involved dividing the dataset into smaller, more manageable batches. This allows to alleviate the memory constraints but also facilitated parallel processing if needed, enhancing the overall efficiency of the training process. Additionally, we implemented a learning rate scheduler to dynamically adjust the learning rate during training. By monitoring the model's performance on the training and validation data, the scheduler updates the learning rate over time, ensuring that the model's parameters were updated effectively without oscillating or diverging. This adaptive learning rate adjustment optimized convergence, accelerating the learning process and improving training efficiency. To further safeguard against overfitting and ensure the model's generalization ability, we incorporated early stopping mechanisms into our training pipeline. Early stopping allowed us to monitor the validation loss during training and halt the training process if the loss failed to improve over a predefined number of epochs. This prevented the model from excessively fitting to the training data, thereby promoting better generalization to unseen data and enhancing the model's predictive performance.

Moreover, we utilized the Adam optimizer and Mean Squared Error (MSE) loss function to optimize parameter updates and quantify the model's performance, respectively. The Adam optimizer's adaptive learning rate method enabled efficient parameter optimization, while the MSE loss function provided a measure of the model's accuracy in predicting heat usage. In our training regimen, we employ the Adam optimizer and Mean Squared Error (MSE) loss function to optimize the model's parameters and quantify its performance, respectively. The Adam optimizer is a popular choice for training deep neural networks due to its adaptive learning rate method. It combines the benefits of two other popular optimizers, RMSprop and AdaGrad, by maintaining separate adaptive learning rates for each parameter and utilizing momentum to accelerate convergence. Adam dynamically adjusts the learning rate based on the first and second moments of the gradients, allowing it to adaptively scale the learning rate for each parameter. This adaptive behavior enables Adam to effectively handle a wide range of learning rates and converge quickly to a good solution. On the other hand, the Mean Squared Error (MSE) loss function is commonly used in regression tasks to measure the discrepancy between the predicted values and the actual targets. It calculates the average squared difference between the predicted and true values across all data points, providing a measure of the model's accuracy in predicting continuous outcomes. The use of MSE loss in our training process allows us to quantify the model's performance in terms of how closely its predictions align with the ground truth values. By minimizing the MSE loss during training, we aim to optimize the model's parameters and improve its ability to accurately forecast heat usage.

By integrating these strategies into our training process, we effectively optimized memory usage, facilitated efficient learning, and prevented overfitting, ultimately improving the performance and robustness of our models for heat usage forecasting.

### 4.2.3 Testing phase

Before commencing testing, we take the precautionary step of saving the trained model to safeguard its parameters and architecture for future reference, analysis, or deployment. This precaution ensures that we have ready access to the model for any subsequent evaluations, deployments, or refinement processes.

In our testing phase, we adopt a thorough evaluation approach by subjecting our model to the entire test dataset at once. This holistic evaluation allows us to gain a comprehensive understanding of the model's performance across the entire dataset. To gauge the effectiveness of our model, we compute various evaluation metrics, offering insights into different facets of its performance. Upon completion of testing, we transition to visualizing the results using Plotly, a robust data visualization library known for its interactive plots and support for exporting to HTML files. Plotly's interactive capabilities empower users to delve into the data and model predictions interactively, facilitating in-depth analysis. Moreover, exporting plots to HTML files enhances sharing and collaboration, as HTML files are universally accessible across different platforms without the need for specialized software. By harnessing Plotly for visualizing the results, we can delve deeper into the model's performance, identifying any discernible patterns or anomalies in the predictions. This visualization step enriches the interpretability of the results and enables informed decision-making regarding model enhancements or further experimentation. Additionally, we not only plot the results but also track the evolution of evaluation metrics over the prediction horizon, providing valuable insights into the model's performance trends.

### 4.2.4 Evaluation

To evaluate the performance of our heat demand forecasting model, we employed several metrics: $R^2$, normalized Root Mean Squared Error (nRMSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the Pearson correlation coefficient. These metrics were chosen because there is no standard metric universally accepted in the literature, and we aimed to make our results as comparable as possible. Metrics like MAE and RMSE are data-dependent, which is why we also included nRMSE, $R^2$, and the Pearson coefficient for a more comprehensive evaluation.

**Coefficient of Determination ($R^2$)**

The $R^2$ (coefficient of determination) measures the proportion of variance in the dependent variable that is predictable from the independent variables. It provides an indication of how well the predicted values match the observed values. The formula is:

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

where $y_i$ are the observed values, $\hat{y}_i$ are the predicted values, and $\bar{y}$ is the mean of the observed values. An $R^2$ value closer to 1 indicates a better fit.

**Root Mean Squared Error (RMSE)**

The RMSE quantifies the average magnitude of the errors between predicted and observed values. It gives a sense of how well the model's predictions match the actual values. The formula is:

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

Lower RMSE values indicate better model performance, with the error measured in the same units as the target variable.

**Normalized Root Mean Squared Error (nRMSE)**

The nRMSE normalizes the RMSE by the mean of the observed values, making it independent of the scale of the data. This allows for better comparison across different datasets. The formula is:

$$\text{nRMSE} = \frac{\text{RMSE}}{\bar{y}}$$

where $\bar{y}$ is the mean of the observed values. A lower nRMSE indicates better model performance.

**Mean Absolute Error (MAE)**

The MAE measures the average absolute errors between predicted and observed values, providing a straightforward interpretation of the average error magnitude. The formula is:

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

Lower MAE values indicate better model performance, representing the average magnitude of errors in the same units as the target variable.

**Pearson Correlation Coefficient**

The Pearson correlation coefficient evaluates the linear correlation between the observed and predicted values. It ranges from -1 to 1, where 1 implies a perfect positive correlation, -1 implies a perfect negative correlation, and 0 implies no correlation. The formula is:

$$r = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2 \sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}}$$

where $\bar{\hat{y}}$ is the mean of the predicted values. A higher absolute value of the Pearson coefficient indicates a stronger linear relationship between observed and predicted values.

By using these metrics, we ensure a comprehensive evaluation of our model's performance. $R^2$ provides insight into the proportion of variance explained by the model. RMSE and MAE give a direct measure of prediction errors, while nRMSE normalizes these errors for better comparison across datasets. The Pearson correlation coefficient helps in understanding the linear relationship between predictions and actual values. This multi-metric approach allows us to assess different aspects of model performance, ensuring robustness and facilitating comparison with other studies in the literature.

# — 5 —

# Experimental Performance Evaluation or validation of solution

In this chapter, we discuss the various experiments conducted to develop and refine our heat usage forecasting model. To establish our final model, we conducted a series of experiments across several dimensions, such as input data configuration, model architecture, and training optimization. We present the nRMSE scores to evaluate the experiments, although all the evaluation metrics discussed earlier were computed and considered in drawing conclusions form each experiment. An example of how the metrics are computed for all the experiments can be found in the annexes.

## 5.1 Experiments on the model's input

In this section the model used for the experiments on the input is the MHA with the hyper parameters presented before. Various inputs using different combination of the available data have been tested during our experimentation, indeed we have access to m, TD, TR, demand, and outside temperature from the substation data. Although as mentioned before, future TD and TR cannot be computed in advance, but we tried to incorporate historical TD and TR data in the input. However, the results did not show notable improvements, instead it increased the complexity of the input data.

### 5.1.1 Organization of the data

As mentioned earlier, the available data on substation includes 2 years of the demand and related temperature, at each time step. Given that the attention mechanism relies heavily on how an element of the sequence attends regarding with the others, the order in which the input data is constructed should be important. We organize the input in two different ways to test which one yields the best model results. The first organization involves taking the historical temperature (where $Temp\_t - n$ represents the outside temperature at time t-n) for a given look-back period (n = 7days) and concatenating it with the historical demand (where $P\_t$ represents the demand at time t), resulting in a sequence 5.1

$$[Temp\_t - n, ..., Temp\_t, P\_t - n, ..., P\_t] \tag{5.1}$$

The second organization involves concatenating the historical temperature at each time step with the demand for a given look-back period (n = 7days), resulting in a sequence 5.2

$$[Temp\_t - n, P\_t - n, ..., Temp\_t, P\_t] \tag{5.2}$$

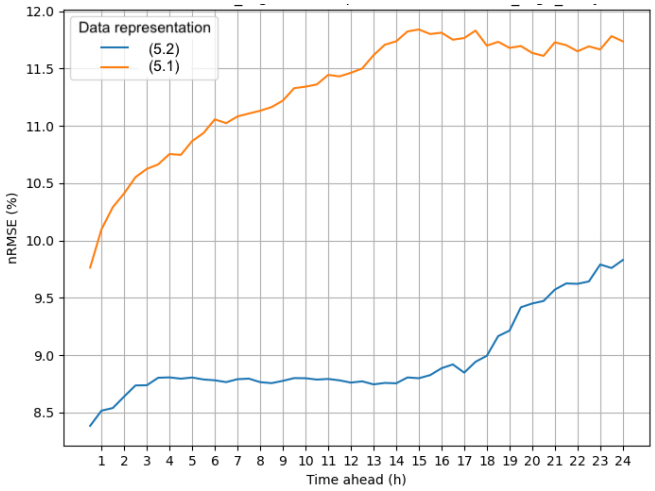


Figure 5.1: A representation of the performance of the model across time ahead prediction with 2 different data organizations

As shown in Figure5.1, it's evident that the model performs best with the latter organization, aligning with our expectations. This arrangement ensures that at each time step the temperature and corresponding demand are associated by the model, providing better contextual information. As a result, the model can better capture temporal dependencies and make more accurate predictions. Therefore, from now on, the data will be organized this way for the next experiments.

Here, is an example of how we conclude on the experience results, in table 5.1 we can see that the best results are achieved with the new data representation established a the second representation in the experiments and the HP filter.

| Metric | No New Data & No HP filter | No New Data & HP filter | New Data & No HP filter | New Data & HP filter |
|---|---|---|---|---|
| MAE (MW) | 0.347060 | 0.342012 | 0.308385 | **0.306549** |
| RMSE (MW) | 0.513 | 0.510 | 0.427 | **0.424** |
| nRMSE (%) | 11.393 | 11.321 | 9.026 | **8.965** |
| R2 | 0.699 | 0.703 | 0.704 | **0.708** |
| Pearson | 0.839 | 0.840 | 0.840 | **0.844** |

Table 5.1: Global Scores of the Model on Different Data Settings

### 5.1.2 Interest of using the future temperature

Given the strong correlation between heating demand and outside temperature, we hypothesized that including future temperature would significantly improve the performance. Thus,

with use the data organization from the previous experiment concatenate the future temperature of the horizon window (m = 48), giving by 5.3

$$[Temp\_t - n, P\_t - n, ..., Temp\_t, P\_t, Temp\_t + 1, ..., Temp\_t + m] \tag{5.3}$$

As anticipated, our hypothesis proved correct. Figure 5.2 illustrates that incorporating future temperature indeed led to notable improvements in model accuracy. Obtaining future temperature data is relatively straightforward through weather APIs, enabling accurate predictions and enhancing the model's predictive capabilities.

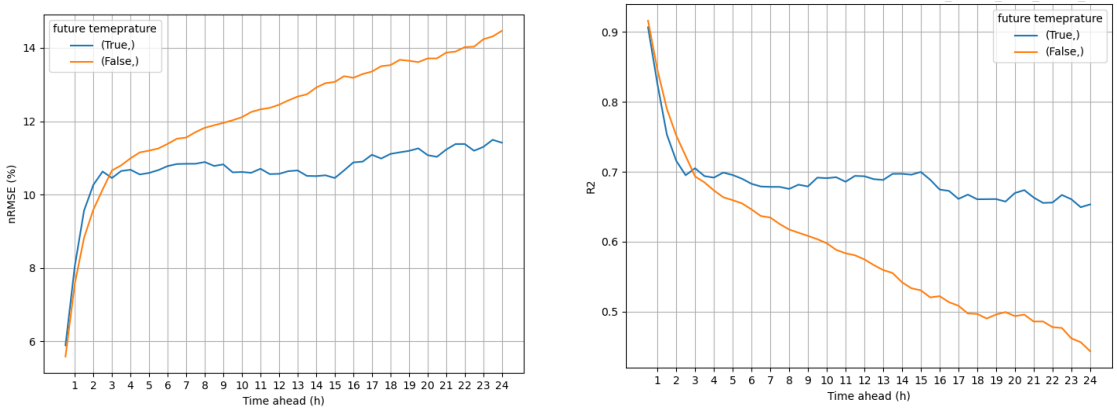


Figure 5.2: Representation of the evolution of the models performance on the different time ahead predictions when using the future temperature as inputs metrics: nRMSE(left) R2(right)

### 5.1.3 Analysis of the different look-back periods

We conducted experiments with varying lengths of historical data to determine the optimal look-back window to reach the balance of a trade off between the richness of historical context and model's complexity.

As shown on the right of Figure 5.3, the model achieves its best performance with a look-back window of 7 days. To try and reduce the size of the input we hypothesize that a 24-hour look-back with the data from the same day of the past week could performe as good as the 7-days look-back. To validate this hypothesis, we conducted further tests shown on the left of Figure 5.3, where we added the data from the 7th day to all other look-back periods,

effectively including both the past 7th day and the past 24 hours as input. The results demonstrate that this modification outperforms the model using the entire past 7 days, confirming our hypothesis.

From now on the look-back period used in the next experiments is 7, since the frequency our data is 30min this makes a look-back of 7*24*2 = 366 data points.

### 5.1.4 Interest of using Hodrick-Prescott (HP) filter

As represented in Figure 5.4, applying the filter to the data consistently improved results on our model, particularly noticeable in shorter-term predictions ranging from 30 minutes to 1 hour and 30 minutes. To further analyze the impact of different components of the filtered data,

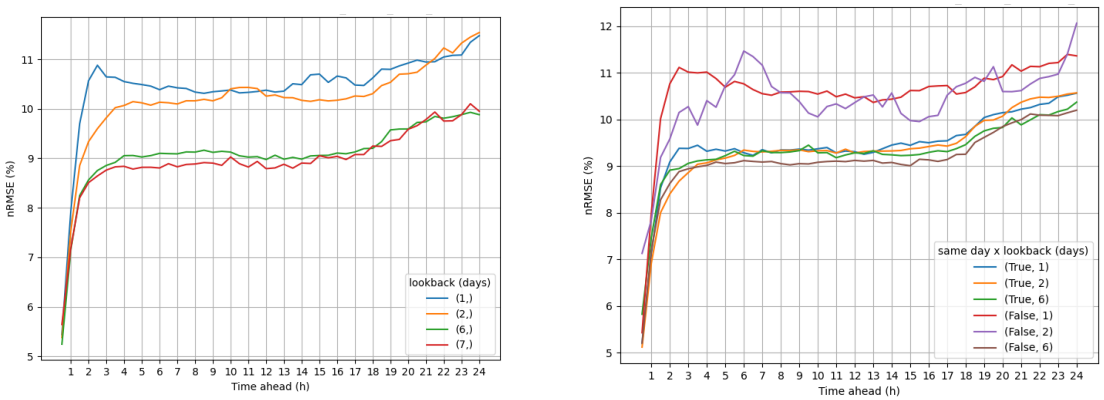


Figure 5.3: Representation of the evolution of the models performance on the different time ahead predictions when using the future temperature as inputs

we compared the performance of models trained with only the trend component versus those trained with both the trend and the remaining noise. Additionally, we conducted a study on the remaining noise and found it to exhibit characteristics consistent with white noise (refer to the Annex for detailed findings).

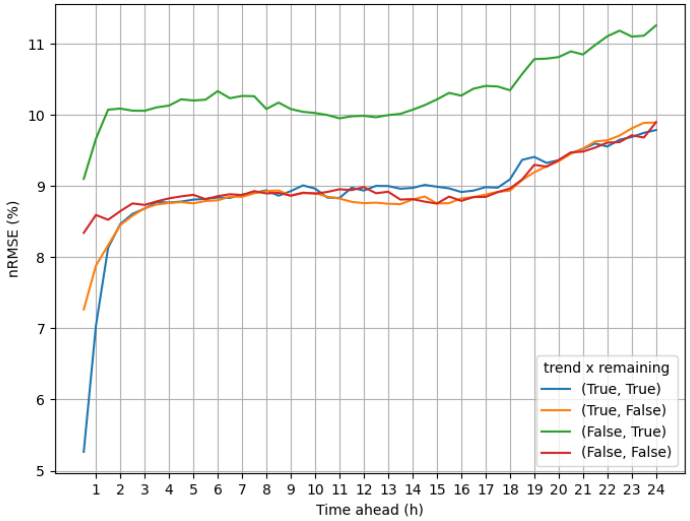


Figure 5.4: Representation of model performance across the different time ahead predictions using the HP filter

In Figure 5.4, when both trend and remaining are set to false, it indicates that we are using standardized raw data, as will be shown in the next experiment. This approach achieves good results because the data is pre-processed. Using raw data can lead to a persistence/shift problem in the model's predictions, so we assume that data should always be pre-processed for time series prediction.

From here we always use the filtered data using the trend and the remaining, unless written otherwise.

### 5.1.5 Difference between Standardization and Normalization

In our experimentation, we applied several preprocessing techniques on the input, such as standardization, normalization, and the Hodrick-Prescott (HP) filter, to assess their effects on model performance. Standardization adjusts the data to have a mean of zero and a standard deviation of one, which often proves effective in improving the convergence and stability of neural network models. In contrast, normalization scales the data to a range between zero and one, which may not be as effective in certain contexts, particularly when dealing with features with varying magnitudes. The superior performance of standardization underscores its importance as a preprocessing step in enhancing model performance and facilitating efficient training and convergence. Further analysis revealed that standardization effectively reduces the impact of outliers and ensures that each feature contributes proportionally to the model's learning process, leading to more reliable and robust predictions. This finding emphasizes the significance of careful pre-processing in optimizing the performance of machine learning models for complex tasks like heat usage forecasting.[**?**]

In Figure 5.5, we present an experiment combining the HP filter and data pre-processing. The best performance is achieved when using standardization processing. This is likely because the magnitude of our data impacts the min-max normalization process.

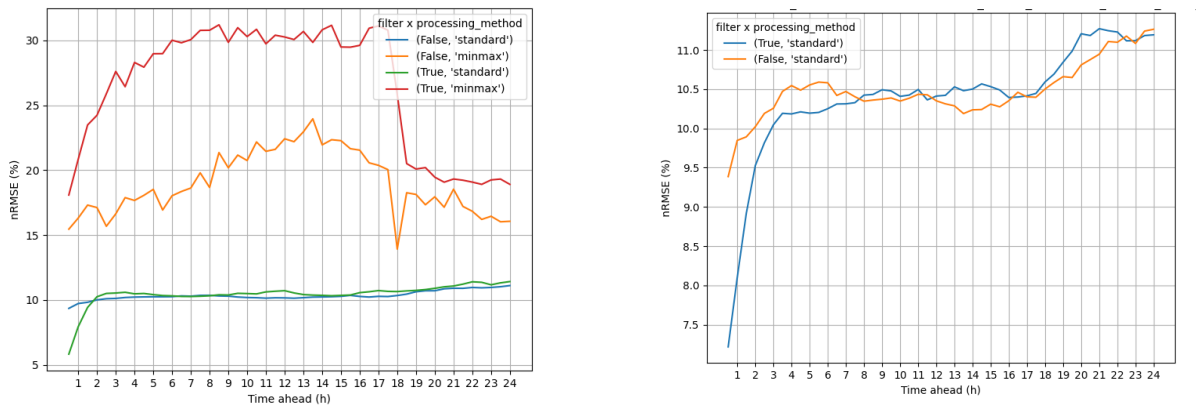


Figure 5.5: Representation of model performance across the different time ahead predictions using the HP filter

## 5.2 Experiments on the different models

Based on the results from previous experiments, we set the look-back period to 7 days, apply the HP filter and standardization, and use the second data organization method.

### 5.2.1 Architecture

For each of our models, we performed hyper-parameter optimization using OPTUNA, a Python library, to fine-tune parameters such as learning rates, batch sizes, and the number of layers and heads in the MHA model. We tested different optimizer, including Adam and Adagrad, as well as various loss functions like Mean Squared Error (MSE) and cross-entropy, to identify the best

combination for our forecasting task. On Table 5.2 we can the final architecture returned by optuna for each model. Additional, experiences can be found in the annexes.

| Model | Values |
|---|---|
| LSTM | (lstm): input_size = 144, hidden_size = 128, num_layers=2, output_size = 48 |
| GRU | (gru): input_size = 144, hidden_size = 128, num_layers=2, output_size = 48 |
| MHA and MHA_R | input_size = 144, d_model = 64, num_heads = 2, num_layers = 2, output_size = 48 |
| SimpleNet | (fc1): Linear(in_features=1056, out_features=128, bias=True)<br>(sig): Sigmoid()<br>(fc2): Linear(in_features=128, out_features=64, bias=True)<br>(relu): ReLU()<br>(fc3): Linear(in_features=64, out_features=48, bias=True) |

Table 5.2: Architecture of the different models

Additionally, we implemented a learning rate scheduler to dynamically adjust the learning rate during training and used early stopping to prevent over-fitting and stagnation in validation loss. By comparing the training and validation loss of our models with and without early stopping, we observed overfitting in the absence of early stopping and a plateau on the validation in the absence of the scheduler.

### 5.2.2 Performance across models

We then compare the performance of various neural network architectures: Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), our custom Multi-Head Attention (MHA), Multi-Head Attention with Positional Encoding (MHA_R), and a combined GRU and LSTM model. Additionally, we compare these results with a simple network comprising three dense layers (ref Annexe) and a persistence model that returns the last observed data point. These serve as benchmarks to evaluate our models' performance. These comparisons aim to identify the most effective model for accurately forecasting substation demand based on historical temperature and demand data.

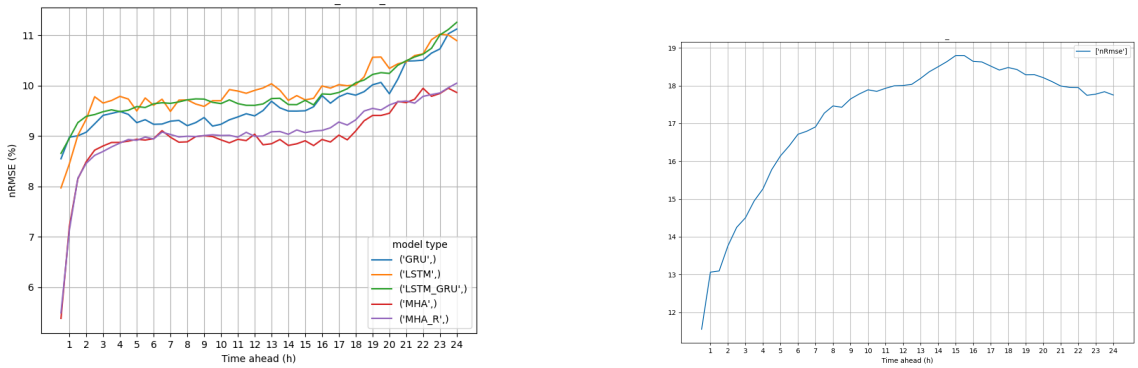


Figure 5.6: Representation of model performance across the different time ahead predictions using the HP filter

We hypothesized that adding positional encoding to the input sequence would improve the model's performance. From Figure 5.6, we observe that the Multi-Head Attention (MHA)

model without positional encoding performs the best. This result is unexpected, as we anticipated the positional encoding to enhance the model's ability to capture temporal dependencies. However, given the look-back period of 7 days, it appears that the model has sufficient data to establish these dependencies effectively using just the attention mechanism. The performance gap between the MHA model with positional encoding and the one without it is minimal, suggesting that the attention mechanism alone is quite robust. Nevertheless, we could explore adding the encoding of the timestamp to see if it yields better results, but we didn't have enough time to test this. On the right side of Figure 5.6, we can see the performance of the persistence model, been the worse results a model could achieve while minimizing its MSE Loss.

To summarize our experience with the offline learning models, we refer to Table 5.3, which shows that the MHA model outperforms all others across all metrics.

| **Model** | **MHA** | **LSTM** | **GRU** | **MHA_R** | **SimpleNet** | **Persistence** |
|---|---|---|---|---|---|---|
| **MAE (MW)** | **0.306** | 0.350 | 0.340 | 0.313 | 0.328 | 0.573 |
| **RMSE (MW)** | **0.423** | 0.479 | 0.464 | 0.432 | 0.445 | 0.778 |
| **nRMSE (%)** | **8.945** | 10.137 | 9.804 | 9.129 | 9.421 | 17.275 |
| **$R^2$** | **0.709** | 0.626 | 0.650 | 0.771 | 0.677 | 0.309 |
| **Pearson** | **0.844** | 0.791 | 0.811 | 0.836 | 0.832 | 0.645 |

Table 5.3: Comparison of different models based on various metrics

# — 6 —
# Analysis of the obtained results

Throughout this research project, the most challenging part has been finding the right data representation for the input to avoid encountering persistence issues while achieving good results.

## 6.1 Summary of results

In our experiments, several significant findings emerged, illuminating effective strategies for heat usage forecasting in district heating networks. One of the most critical insights was the importance of input data processing. We discovered that a 7-day look-back window provided the best results for our forecasting models, achieving an optimal balance between capturing historical context and maintaining model simplicity. Additionally, our experiments underscored the superiority of standardization over normalization for data pre-processing. Standardization consistently led to better performance across various model configurations and input settings. Another pivotal enhancement was the integration of future temperature data, which significantly improved the accuracy of our models.Using these insights, we tested our models extensively. Table 5.3 presents the global metrics obtained across all forecasting horizons and models, showcasing the results of our experiments. These metrics offer a comprehensive evaluation of our models' performance. Figures 6.1 and 6.2 provide visual examples of predictions made using the Multi-Head Attention (MHA) model. These figures highlight the degradation in prediction accuracy as the forecast horizon extends to 1 hour and 3 hours ahead, respectively. This degradation is also evident in the normalized Root Mean Square Error (nRMSE) plots. Consequently, we compare the global metrics across all forecasting horizons rather than focusing on specific time steps, offering a more holistic view of model performance over time.

## 6.2 Discussion

One of the main problems we encountered during the research project was regarding the data representation, we will now present the evolution of the results obtained during this project using different types of data pre-processing. In figure 6.3 we can see the prediction of our model with the first data representation used which included only the historical demand. It became clear that pre-processing the data is one of the most important steps because it can completely change a model's predictions.

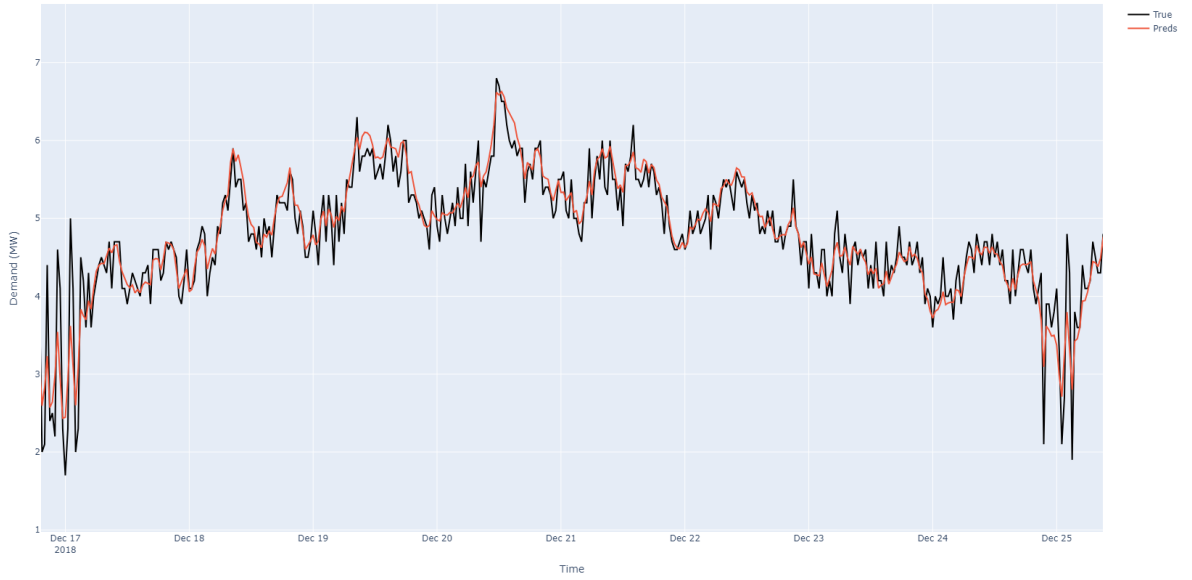


Figure 6.1: Prediction 1 hour ahead using MHA

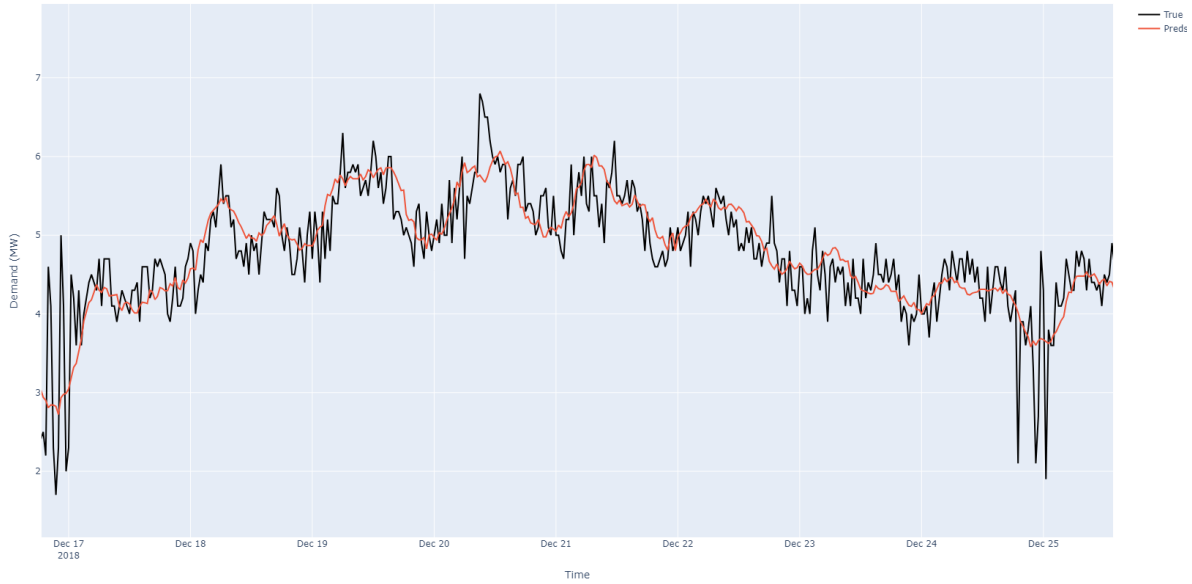


Figure 6.2: Prediction 3 hour ahead using MHA

Overall, our MHA model demonstrates superior performance compared to other models. To contextualize our findings, we compared our results with the state-of-the-art, using Yanis' [6] results as a reference point since we operate on the same dataset. While our MHA model exhibits strong performance, especially in terms of MAE, RMSE, and Pearson correlation, further comparison with machine and incremental learning approaches from the literature underscores the effectiveness of our approach. It's important to acknowledge that direct comparisons across models can be complex due to variations in dataset characteristics and modeling techniques. Additionally, it is difficult to compare since most of the metrics are data-dependent, making direct comparisons challenging. We can really compare our results with the ones obtained by

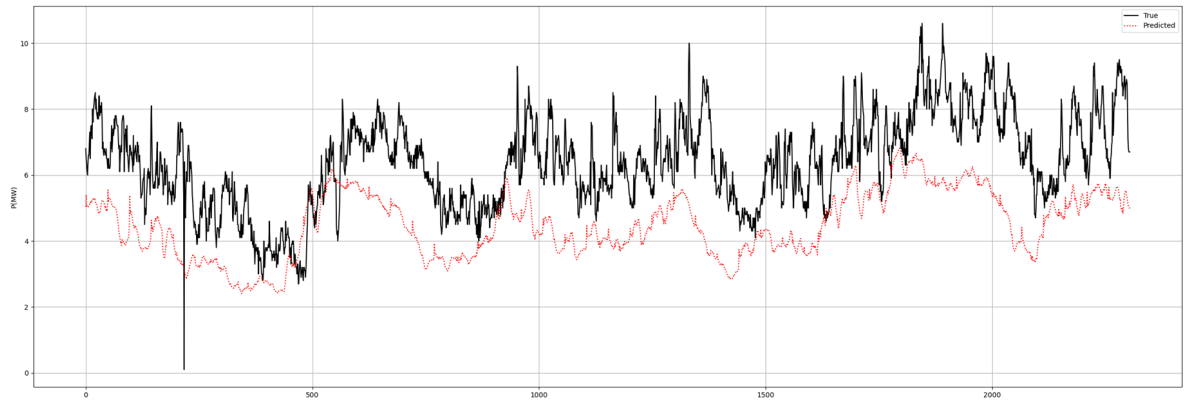


Figure 6.3: Prediction 1 hour ahead using MHA and the initial data representation

Yanis because as explained before the metrics used in the state of the art depend on the magnitude of the data, ours is in MW and the state of the art in KW so their is not much to compare our selves on this matter. This makes it really difficult for use to situate ourselves within the literature.

| Metric | MAE (MW) | RMSE (MW) | nRMSE (%) | $R^2$ | Pearson |
|---|---|---|---|---|---|
| **MHA** | **0.303** | **0.423** | 8.945 | **0.709** | **0.844** |
| **NBEATSx** | 0.368 | 0.484 | **8.48** | \ | \ |
| **KKR** | \ | 0.007 | \ | \ | \ |
| **LR** | 0.039 | 0.009 | \ | \ | \ |
| **CNN** | 0.337 | 0.054 | \ | \ | \ |
| **ENC-DEC LSTM** | \ | 0.046 | \ | \ | \ |
| **ODL-ER** | \ | 8.64 | \ | \ | \ |
| **FIMT-DD** | MAPE: 4.77% | \ | \ | \ | \ |
| **FB-Prophet + Light GBM** | 0.308 | 0.505 | \ | \ | **0.92** \ |

Table 6.1: Comparison of different models from the state of the art and performance metrics of forecasting algorithms

Due to time constraints, we were unable to fully implement the Incremental Learning (IL) methodology within the scope of this project. However, this remains an ongoing aspect of our research, which we aim to complete by the project's conclusion in 2 months. Our future work includes refining the established scenarios and creating new ones to test both our current model and the IL model we plan to implement on DreamNet. The initial steps towards implementing IL on DreamNet have been undertaken, but comprehensive testing has not yet been conducted. Our goal is to thoroughly test and validate these implementations to ensure they perform well under various scenarios. Specifically, we will focus on refining the scenarios involving variations in demand patterns and temperature thresholds, as these have shown significant promise in preliminary analyses. We anticipate having preliminary results from these scenarios ready for presentation at the defense of our research project. These results will provide valuable insights into the real interest of using IL for this application.

# — 7 —
# Conclusion

This research addresses the critical challenge of accurately forecasting heat usage in district heating networks, which is essential for efficient energy management and cost reduction. Our initial goal was to establish a robust offline learning model for this task, followed by the adaptation of DreamNet for an adaptive learning approach. Although we successfully implemented the offline learning model, time constraints prevented us from fully adapting DreamNet. Consequently, further testing is necessary to explicitly demonstrate catastrophic forgetting and validate the efficacy of the adaptive learning approach. Despite this limitation, we managed to develop a custom Multi-Head Attention (MHA) mechanism with 2 layers of multi-head attention, 2 heads per layer, and a model dimension of 64. This first implementation on DreamNet lays the foundation for future experimentation and refinement. Our investigation involved a comprehensive exploration of pre-processing techniques, feature sets, and input configurations, leveraging the PyTorch library for model implementation. Through rigorous training, incorporating batch processing, learning rate scheduling, and hyper-parameter optimization, we refined our models to achieve strong performance. Evaluation involved thorough testing on the entire dataset, yielding insights into the effectiveness of different model configurations. Notably, we observed the importance of a 7-day look-back window, the superiority of standardization over normalization, and the substantial gains from integrating future temperature data. These findings highlight critical strategies for enhancing the accuracy of heat usage forecasts. While our research represents a significant step towards improved heat usage forecasting, further investigation is warranted to fully realize the potential of adaptive learning models in this domain. Our future work includes finalizing the implementation of incremental learning methodologies and developing new scenarios to test their robustness. Additionally, as we aim for an embedded solution, future efforts will focus on optimizing model complexity, size, and computational cost, ensuring the feasibility of an embedded implementation.

The project fulfills the criteria for a Masters Research project through its strong research foundation and contributions to the field. Our thorough analysis of the scientific question behind our research problem demonstrates rigorous inquiry and meaningful contributions to the field. As evidenced by our review of the state-of-the-art, there are not many works in the literature that address the dual challenges of both offline and incremental forecasting in district heating networks. This gap underscores the novelty and significance of our work. We have delved deeply into the intricacies of heat usage forecasting, exploring both traditional and cutting-edge methodologies. Our comparative analysis of standardization versus normalization for data preprocessing, the implementation of various look-back windows, and the integration

of future temperature data are all testament to the comprehensive nature of our research. These efforts not only highlight our commitment to addressing complex research questions but also contribute valuable insights and methodologies to the existing body of knowledge in the field. In summary, although our work is not done, our project not only addresses a significant gap in the literature but also provides practical solutions and methodologies, reinforcing its relevance and impact as a Masters Research project.

# — A —
# Appendix

## A.1 Insight of the persistence issue

We encountered a persistent issue related to data processing that affected many models, as depicted in Figure A.1. Although the metrics, particularly the Pearson coefficient, appeared promising, the models failed to produce meaningful predictions. Instead, they merely returned the last input data point, resulting in a consistent shift in the time-ahead forecast. This issue led to a degradation in the metrics as the forecast horizon increased, highlighting a significant challenge in our modeling approach.

## A.2 Lamb parameter on Hp filter study

As shown in Figure A.2,A.3, A.4 the lamb that best represents the data is lamb = 6.25.

## A.3 Grid Search

Here we can find an example of the grid search conducted on the different models after retrieving the results from the optuna optimization.

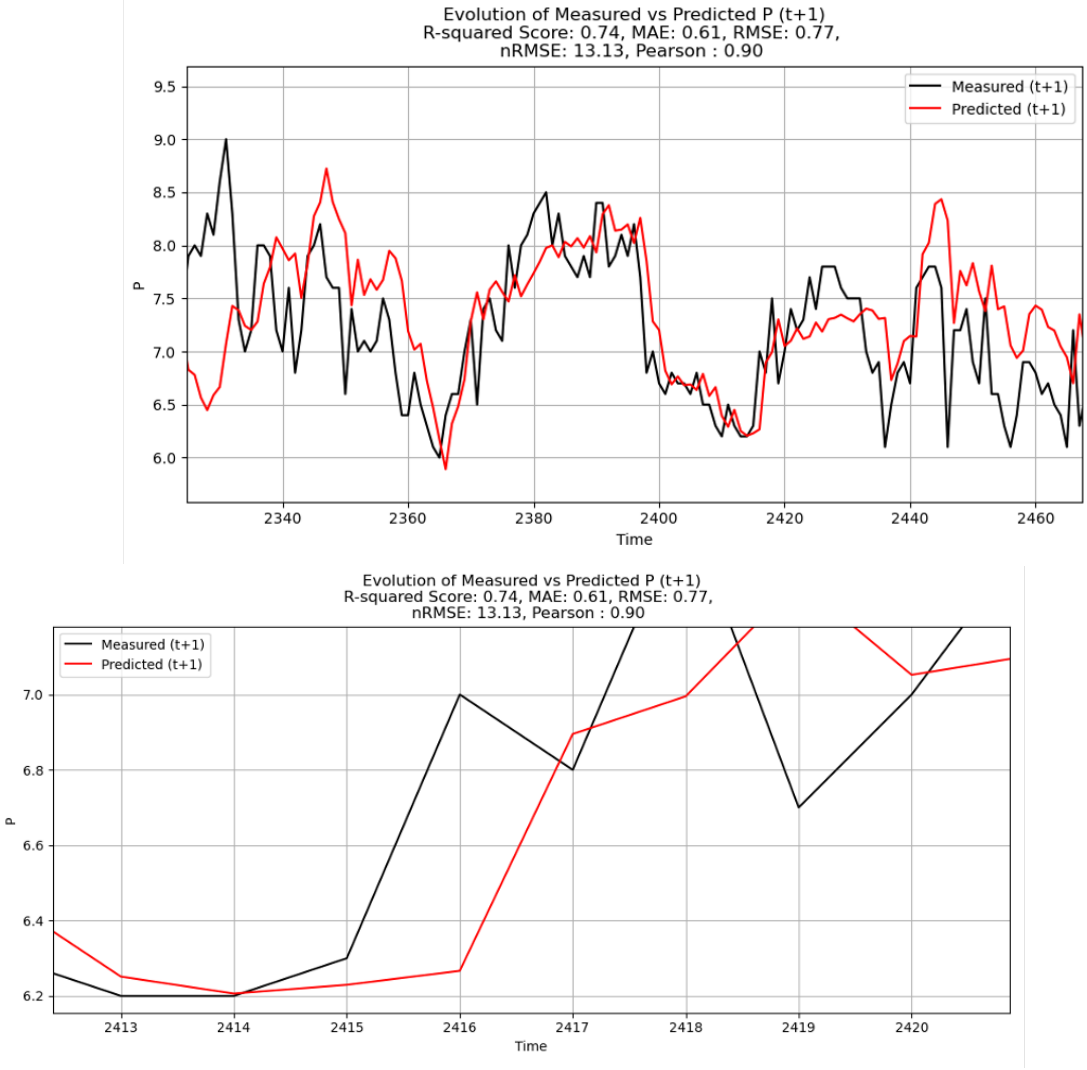


Figure A.1: A representation of the persistence problem encountered

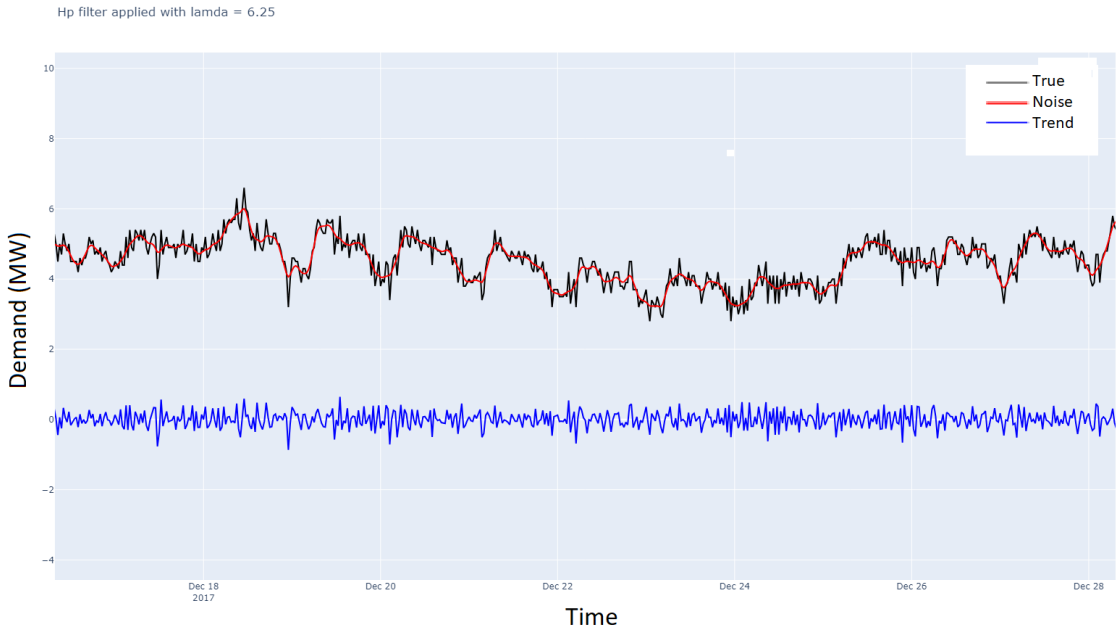


Figure A.2: HP filter with lamb = 6.25

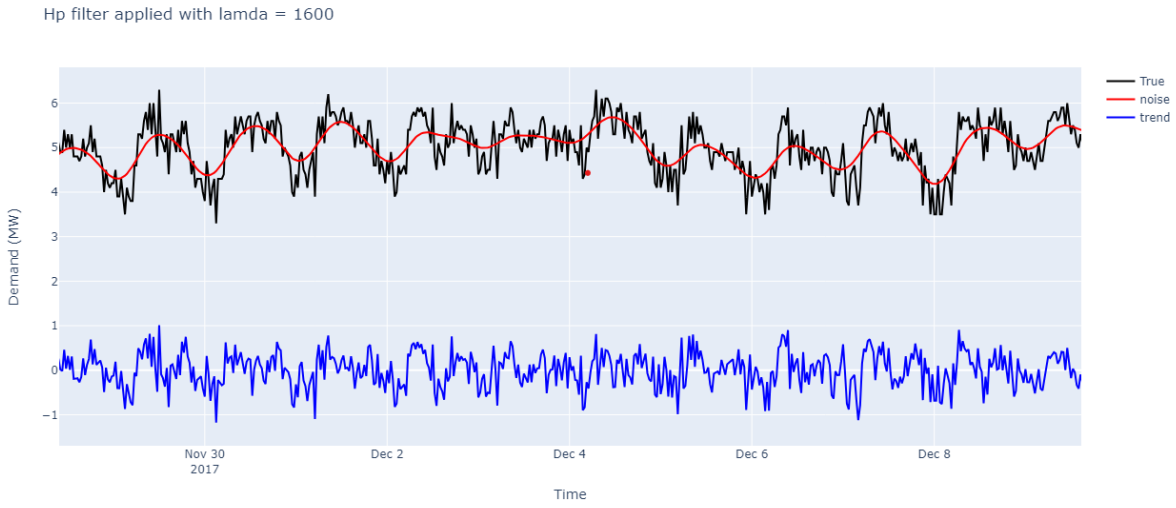


Figure A.3: HP filter with lamb = 1600

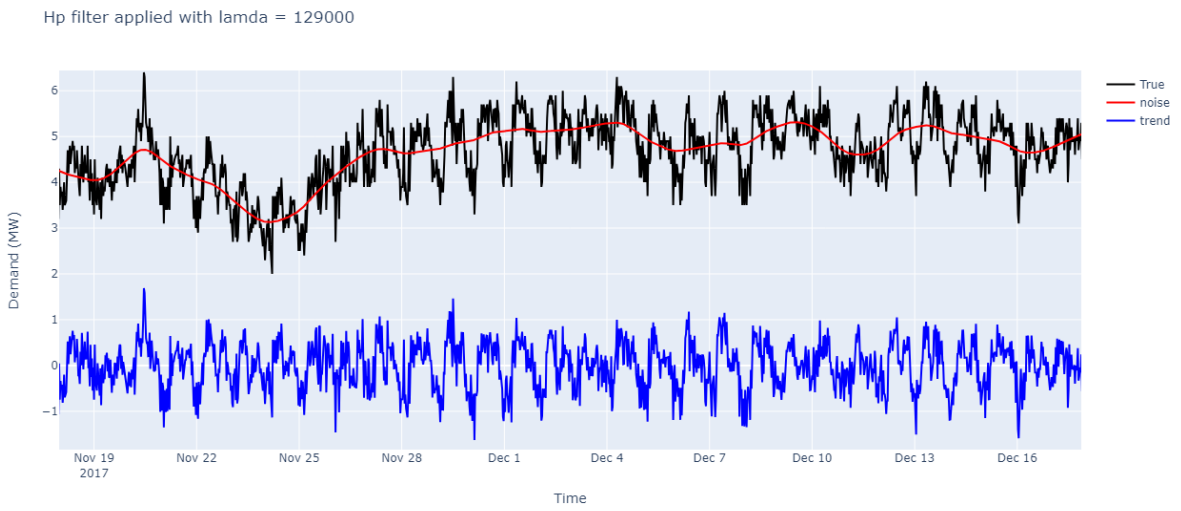


Figure A.4: HP filter with lamb = 129000

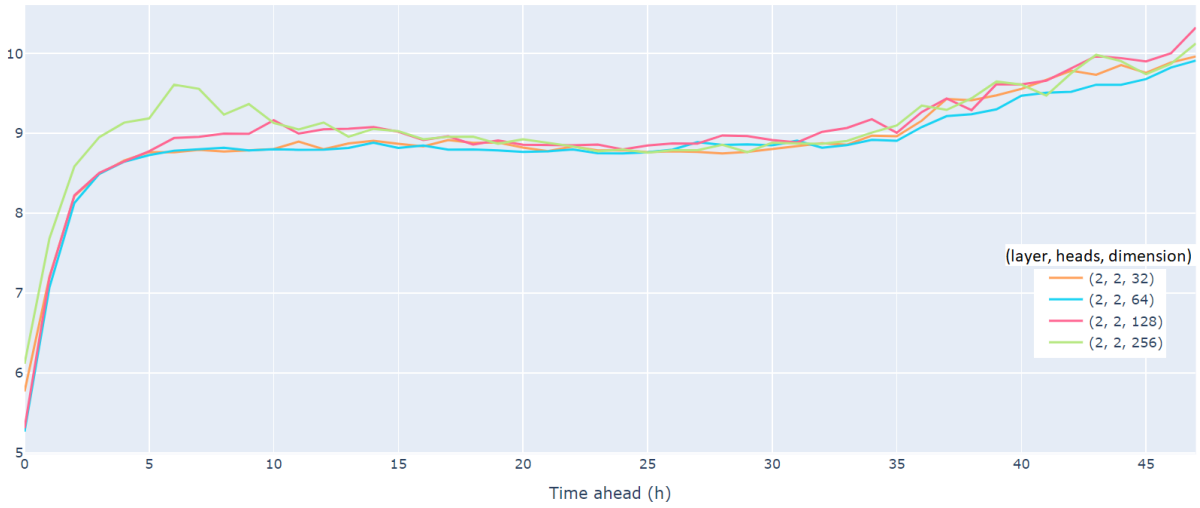


Figure A.5: Representation of the grid search conducted on the MHA model

# Bibliography


[1] R. BaviÃ¨re and M. VallÃ©e. Optimal temperature control of large scale district heating networks. In *Energy Procedia, 16th International Symposium on District Heating and Cooling, DHC2018, 9–12 September 2018, Hamburg, Germany*, volume 149, pages 69–78, 2018. See equations (2) and (3).

[2] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis: Forecasting and Control*. Wiley, 2008.

[3] Seok-Jun Bu and Sung-Bae Cho. Time series forecasting with multi-headed attention-based deep learning for residential energy consumption. *Energies*, 13(18), 2020.

[4] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *arXiv*, 2004.07211, 2020.

[5] Cerema. District heating and cooling in france.

[6] Yanis Chaigneau. Load forecasting in energy systems with machine learning algorithms.

[7] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, et al. Continual learning with tiny episodic memories. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1257–1266, Long Beach, California, USA, 2019. PMLR.

[8] Laura Boca de Giuli, Riccardo Scattolini, and Alessio La Bella. Physics-based neural network modelling, predictive control and lifelong learning applied to district heating systems.

[9] Imadeldin Elmutasim. A brief review of massive MIMO technology for the next generation. *Int. Arab J. Inf. Technol.*, 20(2):262–269, 2023.

[10] General Commission for Sustainable Development. GHG emissions from the energy industry.

[11] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd edition, 2009.

[12] Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. Recurrent neural networks for time series forecasting: Current status and future directions. *CoRR*, abs/1909.00590, 2019.

[13] Robert Hodrick and Edward Prescott. Postwar u.s. business cycles: An empirical investigation. *Journal of Money, Credit and Banking*, 29(1):1–16, 1997.

[14] Neele Kemper, Michael Heider, Dirk Pietruschka, and JÃ¶rg HÃ¤hner. Forecasting of residential unitâs heat demands: a comparison of machine learning techniques in a real-world case study.

[15] K. Kenda et al. Computer architectures for incremental learning in water management. *Sustainability*, 2022.

[16] J. Kim et al. Development of a deep learning-based prediction model for water consumption at the household level. *Water*, 14(9):1512, 2022.

[17] Sachin Kumar, Saibal K. Pal, and Ram Pal Singh. A novel method based on extreme learning machine to predict heating and cooling load through design and structural attributes. *Energy and Buildings*, 176:275–286, 2018.

[18] Teresa Kurek, Artur Bielecki, Konrad Åwirski, Konrad Wojdan, MichaÅ Guzek, Jakub BiaÅek, RafaÅ Brzozowski, and RafaÅ Serafin. Heat demand forecasting algorithm for a warsaw district heating network. 217:119347.

[19] Stefan Leiprecht, Fabian Behrens, Till Faber, and Matthias Finkenrath. A comprehensive thermal load forecasting analysis based on machine learning algorithms. 7:319–326.

[20] C. Li, K. Liu, and H. Wang. The incremental learning algorithm with support vector machine based on hyperplane-distance. *Applied Intelligence*, 34:19–27, 2011.

[21] Benjamin Lindemann, Timo Müller, Hannes Vietz, Nasser Jazdi, and Michael Weyrich. A survey on long short-term memory networks for time series prediction. *Procedia CIRP*, 99:650–655, 2021. 14th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 15-17 July 2020.

[22] Junyu Liu, Xiao Wang, Yan Zhao, Bin Dong, Kuan Lu, and Ranran Wang. Heating load forecasting for combined heat and power plants via strand-based LSTM. 8:33360–33369.

[23] Marion Mainsant. *Direction: Martial Mermillod, Marina Reyboz, Christelle Godin*. ThÃ¨se de doctorat, UniversitÃ© Grenoble Alpes, 12 2023.

[24] Marion Mainsant, Martial Mermillod, Christelle Godin, and Marina Reyboz. A study of the dream net model robustness across continual learning scenarios. page 824. IEEE.

[25] Marion Mainsant, Miguel Solinas, Marina Reyboz, Christelle Godin, and Martial Mermillod. Dream net: a privacy preserving continual learning model for face emotion recognition.

[26] Spyros Makridakis. The m4 competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, Volume:Pages, 2019.

[27] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers.

[28] Kelvin Olivares et al. N-beatsx: Extensions to n-beats for interpretable and exogenous time series forecasting. *Conference Paper*, 2021.

[29] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall, 2nd edition, 1999.

[30] Boris Oreshkin et al. N-beats: Neural basis expansion analysis for time series forecasting. *International Conference on Learning Representations (ICLR)*, 2020.

[31] Spyridon Provatas. Faculty of computing blekinge institute of technology SE 371 79 karlskrona, sweden.

[32] Aowabin Rahman, Vivek Srikumar, and Amanda D. Smith. Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks. *Applied Energy*, 212:372–385, 2018.

[33] Morten O. Ravn and Harald Uhlig. On adjusting the hodrick-prescott filter for the frequency of observations. *The Review of Economics and Statistics*, 84(2):371–375, 2002.

[34] W. Saeed. Frequency-based ensemble forecasting model for time series forecasting. *Computational and Applied Mathematics*, 41(66):66, 2022.

[35] Asim Shakeel, Daotong Chong, and Jinshi Wang. District heating load forecasting with a hybrid model based on LightGBM and FB-prophet. 409:137130.

[36] Slawek Smyl. A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, Volume:Pages, 2019.

[37] Gido M. van de Ven, Tinne Tuytelaars, and Andreas S. Tolias. Three types of incremental learning. *Nat. Mac. Intell.*, 4(12):1185–1197, 2022.

[38] Gonzalo MartÃn-RoldÃ¡n Villanueva, Mark Dougherty, Jonathan Atkinson, and Siril Yella. Householdâs energy consumption and production forecasting: A multi-step ahead forecast strategies comparison.

[39] Milan ZdravkoviÄ, Ivan ÄiriÄ, and Marko IgnjatoviÄ. Explainable heat demand forecasting for the novel control strategies of district heating systems. 53:405–413.

[40] Y. Zhang and Q. Yang. Scalable high performance deep learning platform based on min-max normalization. *International Journal of Computers Communications & Control*, 13(4):607–620, 2018.

[41] J. Zheng, F. Shen, H. Fan, et al. An online incremental learning support vector machine for large-scale data. *Neural Computing & Applications*, 22:1023–1035, 2013.