



École Polytechnique Fédérale de Lausanne

Urban Heat Islands Magnitude Prediction with Graph Machine
Learning, and Impact on Building's Energy Consumption

by Christophe Roger

Master Thesis

Approved by the Examining Committee:

Prof. Dr. Gabriele Manoli
Thesis Advisor

Prof. Dr. Stefan Broennimann
External Expert

Dr. Martin Hendrick
Thesis Supervisor

EPFL ENAC IA URBES
BP 3236 (Bâtiment BP)
Station 16
CH-1015 Lausanne

January 31, 2025

Acknowledgments

I would like to express my gratitude to Professor Gabriele Manoli for giving me the opportunity to complete my thesis in the URBES laboratory, and to Dr. Martin Hendrick for his invaluable advice and support throughout this journey. I am also very thankful to the Institute of Geography at the University of Bern for providing the data used in this work.

A special thanks goes to my dear friends Violette, Liselotte, and Lola, whose warm hospitality in Lausanne was very precious whenever I needed to come.

Lastly I am also grateful to my girlfriend Elsa, whose encouragement and understanding over the past four months gave me great support, even when though I was not the best in sharing the details of my research, and to my family who also helped me and encouraged me along this journey.

Lausanne, January 31, 2025

Christophe Roger

Abstract

The Urban Heat Island (UHI) phenomenon refers to the elevated temperatures observed in urban areas compared to their surrounding rural regions. This effect is due to urban characteristics such as reduced vegetation and the use of low-albedo materials, and has significant implications for human health, energy consumption, and atmospheric pollution. With climate change expected to amplify the UHI effect through rising global temperatures, there is an urgent need for models to characterize and predict UHI in order to assist decision-makers in developing adaptation and mitigation strategies.

Machine learning offers a powerful approach to studying and forecasting UHI. This thesis introduces a novel methodology for the hourly prediction of UHI intensity using a dense network of temperature measurement devices installed in the Bern region. The proposed method, called the Spatio-Temporal Fusion Graph Attention Network (STFGAN), uses the graph structure of the sensor network to capture spatial dependencies, incorporates an attention mechanism to emphasize critical connections, and employs gated recurrent units to process temporal features prior to fusion. The method achieves a Root Mean Square Error (RMSE) of 0.65°C , placing it among the most accurate hourly air temperature prediction models within the urban canopy layer and reducing error by 14% compared to a baseline XGBoost model trained on the same data.

Building on these predictions, a model was developed to generalize UHI intensity predictions across the entire domain, rather than being limited to the specific locations of the temperature sensors. This allowed for an evaluation of the UHI effect on building energy consumption using the cooling degree hours method. Results indicate that cooling loads have increased across most urban areas, with some zones experiencing up to a 30% rise compared to a reference station in Zollikofen, a rural area near Bern.

The findings demonstrate the effectiveness of the STFGAN methodology for hourly UHI intensity predictions and confirm the substantial impact of UHI on cooling demands in urban environments.

Contents

Acknowledgments	2
Abstract	3
1 Introduction	10
1.1 Global situation	10
1.2 Literature review	11
1.2.1 Urban heat island phenomena	11
1.2.2 Spatial machine learning and its application to UHI modeling	12
1.2.3 Buildings energy consumption relation with temperature and Urban Heat Island (UHI)	13
1.3 Objectives and Research questions	14
1.4 Novelties	15
2 Background	16
2.1 Machine learning techniques description	16
2.1.1 XGBoost Algorithm	16
2.1.2 Gated recurrent unit	18
2.1.3 Feature fusion	19
2.1.4 Graph machine learning	20
3 Data and Methodology	23
3.1 Data	23
3.1.1 Study Area and context	23
3.1.2 Low-cost measurement devices	24
3.1.3 Target definition: Temperature delta	25
3.1.4 Urban morphology	27
3.1.5 Spatial and demographic properties	27
3.1.6 Environmental characteristics	28
3.1.7 Temporal features	29
3.1.8 Graph construction	30
3.2 Methods: Machine learning models	30
3.2.1 XGBoost	31
3.2.2 Spatio-temporal Fusion graph attention network	31
3.2.3 Model evaluation	32
3.3 Buildings energy consumption and domain generalization	33
3.3.1 Building's energy consumption	33

3.3.2	Domain generalization	33
3.4	Implementation and parameters fine-tuning	34
3.4.1	Parameters fine-tuning	34
4	Results	37
4.1	UHI intensity predictions	37
4.1.1	Performance on the training set	37
4.1.2	Performance on the test set	39
4.1.3	Models insights	44
4.2	Building energy consumption	47
5	Discussion	50
5.1	Spatio-Temporal Fusion Graph Attention Network (STFGAN) performance	50
5.2	Domain generalization	53
5.3	Impact of UHI on the cooling load	53
5.4	Limitations and further research	54
6	Conclusion	56
	Bibliography	57
	Appendix	64

List of Figures

2.1	Schema of the gradient boosting process. At each iteration, a new decision tree is fitted on the residuals of the previous tree, therefore reducing the error at each iteration. The final prediction is a combination of the prediction of all the trees[61]	17
2.2	Gated Recurrent Unit, fully gated version (Wikipedia)	19
2.3	Difference between euclidean and non-euclidean data [69]	20
2.4	Illustration of a 3-heads attention mechanism by node 1 [76]	22
3.1	Comparison of Bern elevation and local climate zone classification.	24
3.2	Distribution of the daytime and nighttime average temperature at each stations over the four summers.	24
3.3	Data availability. Blue or red means that the Low-Cost Measurement Device (LCD) is functioning at the date, white means that the LCD is either not installed or that the data is not available. Blue is the training data and red is the testing data. The validation for the XGBoost model was performed with a 5-folds cross validation on the train set, while it was done using the 2022 summer for the STFGAN (see 3.4)	25
3.4	Overview of the stations with (a) the locations of the LCDs and (b) the MeteoSwiss Zollikofen weather station.	26
3.5	Distribution of temperature deltas (ΔT) for daytime (orange) and nighttime (blue) periods, showing the frequency of temperature differences between the urban stations and the rural reference at Zollikofen.	27
3.6	Examples of urban morphology in (a) the city center and (b) the surroundings of Bern. The yellow circles represent the LCD with the 100m buffer, and the red polygons are individual buildings.	28
3.7	NDVI map of Bern	29
3.8	Final graph structure used for the model implementation	30
3.9	Graph attention network architecture	32

3.10	Training diagram of the STFGAN for one time step. An epoch is the processing of all the time steps. Blue represents the spatial processing, orange is the temporal processing, purple is the feature fusion and green is the prediction and loss computing.	35
3.11	hyper parameters fine-tuning using 5-fold cross validation grid search. The mean score is the average R^2 of the 5 folds.	36
3.12	train set loss evolution and test set metrics evolution with the epochs. The Loss metric is the mean squared error	36
4.1	RMSE of (a) XGBoost and (b) STFGAN on the train set. Grey circles indicate the LCDs where no data is available to compute the RMSE. The color intensity represents the magnitude of the RMSE with a lower RMSE indicating a better performance in the predictions	38
4.2	odels predictions vs true values. The colors represent the point density, and the dashed line is the identity line, where the predicted values are equal to the true values. Points closer to this line indicate better model performance, while deviations from the line suggest prediction errors.	38
4.3	Map of the mean squared error at each LCD for the test set. Grey circles are LCD without data for 2022	39
4.4	Models predictions vs true values. The colors represent the point density, and the dashed line is the identity line, where the predicted values are equal to the true values. Points closer to this line indicate better model performance, while deviations from the line suggest prediction errors.	40
4.5	RMSE distribution on the three categories of LCD. Category A is for LCDs with a complete timeseries, B is for LCDs with an incomplete time series and C is for LCDs with no training timeseries	41
4.7	Plotted LCDs in Figure 4.6. The category indication is the type of LCD, and the number is the LCD ID	42
4.6	Results of each model in different configurations. The type of LCD plotted is indicated bu the category in each sup-title. The first three plot represent three LCDs that were installed in Bern's center, but at different times. The last two are LCDs that are next to each other but in the exterior of Bern. See Figure 4.7. The 'True' line on plot 1 and 2 is the same due to the lack of testing data for LCD 47.	43
4.8	(a) MdAPE and (b) MAE over 300 hours.	44
4.9	XGBoost features importance for (a) With ΔT lag and (b) without ΔT lags. The total gain represents the feature ability to generate prediction relatively to other features	45
4.10	Attention weights computed by the GATConv layers. The attention weights are averaged over all the time steps. (a) Represent the first layer weights and (b) the second layer	46
4.11	Visualization of the average given attention weights for each node. A larger average attention weight means a more important node	46
4.12	Difference in the average given attention weight between layer one and layer two.	47

4.13	Difference in CDHs between each station and the reference station over time.	47
4.14	Feature importance of the XGBoost model trained on the summer 2022 predictions	48
4.15	Relation between the static station features and the cumulated Cooling degree hours (CDH) over the 2022 year. The shaded area represents the 95% confidence interval	48
4.16	Comparison of the cumulated CDH differential between the Zollikofen weather station and (a) the stations, (b) the Bern region	49
5.1	Comparison of the metrics' evolution: (a) Shows the metric value where only targets exceeding the threshold are considered. (b) Shows the metric value where only targets below the threshold are considered.	52
1	Comparison of the error for each LCD	68

List of Tables

3.1	Model hyperparameters for XGBoost and STFGAN.	36
4.1	Performance metrics for the different models on the train set.	37
4.2	Performance metrics (RMSE, MAE, R^2) for the different models on the test set. The day/night Median Absolute Percentage Error (MdAPE) is not corresponding to the	39
4.3	RMSE details (RMSE _{min} , RMSE _{max} , RMSE _{var}) for the different models on the test set. . . .	40
4.4	Features of the plotted LCDs in Figure 4.7	42
4.5	Metrics of the XGBoost model used to extend the UHI predictions to the study area. The train set is composed of the UHI intensity predictions for 2022 and 80% of the stations. The test set is the remaining 20% stations	48
5.1	Comparison between GNN [48], GATG [49] and STFGAN	51
1	Feature List	67

Chapter 1

Introduction

1.1 Global situation

Nowadays, about 55% of the world's population lives in urban areas. The proportion reaches 80% in developing countries and is projected to be around 68% globally by 2050 [1]. The global urbanization leads to several challenges in various aspects of a sustainable urban planning : mobility, health, logistics, food security, greenhouse gases, and what is of interest, temperature [2]–[4].

A significant factor contributing to these challenges is the extensive use of fossil fuels since the Industrial Revolution [5], which has led to the release of large quantities of CO₂ and other greenhouse gases, causing an atmospheric warming through radiative forcing. This warming has reached 1.1°C during the period 2011–2020 compared to 1850–1900 levels [5].

Although temperatures are rising unevenly across the globe, urban areas are disproportionately affected because the temperature rise is further aggravated by urban morphology, limited vegetation, and other characteristics specific to cities [6]. This effect is called the urban heat island (UHI) effect. Urban heat islands lead to higher mortality due to heat stroke and other related health issues[7]–[9], higher air conditioning (A/C) consumption [10], or increased air pollution [11].

All these impacts can be mitigated if urban planners address effectively the urban heat island effect, by greening the city [12] or by using highly reflective materials for construction [13], [14].

Although the UHI causes and mitigation strategies are well known, the UHI phenomenon persists. Given its strong dependence on city-specific factors, there is a need for efficient and accurate modeling and predictive methods. Although physical simulations have traditionally been the primary approach, they are constrained by high computational costs and limited generalization capabilities. Recently, research has increasingly shifted towards leveraging machine learning (ML) techniques. ML is a powerful tool for capturing non-linear behaviors in data while significantly reducing computational demands compared to physical simulations at larger scales. Applying ML to model Urban Heat Islands (UHI) and their underlying causes has become a prominent area of study [15], [16].

The following section examines the UHI phenomenon and various modeling techniques, as well as their

impact on building energy consumption.

1.2 Literature review

1.2.1 Urban heat island phenomena

[17]. Research on urban heat islands spans a wide range of topics, including methods of measurement and their impacts on health, pollution, and vegetation growth. Studies on the UHI effect go back to the early 19th century [18] where the amateur meteorologist L. Howard started to study London city temperatures [19]. During two decades, Howard collected temperature records at three points in the surrounding countryside of London and compared them to the data from the Royal Society, located in London's city center. From his observations, he concluded that the air temperature is proportional to urban density and development. Successive studies in different cities in the 19th and early 20th century established that the temperature difference between urban and rural environment is indeed existent, and also deduced that the magnitude of an existing temperature difference is more dependent on the the local measuring environment rather than the size of the city[20].

In 1927, innovations in the measuring methodology led to the first mapping of the urban-rural temperature difference in Vienna [21]. The methodology, consisting of measuring the temperature along a car-trip with a thermometer mounted above the car, was soon replicated for the city of Karlsruhe [22] from where the term *urban heat island* originated.

In the mid-20th century, more experimentation led to the first regression model to link temperature differences between urban and rural areas to weather conditions (e.g. cloud cover) in Uppsala, Sweden [23]. Along this regression model, the first surface energy balance framework to try to understand the physical phenomenons leading to the urban heat island effect [24]. Numerous studies followed around the world, with a more formal approach to understand the physical causes of the urban heat island phenomena.

Nowadays, the literature on UHI focuses on 4 areas : the causes, the consequences, the modeling methods and the mitigation paths [17].

Several techniques exist for modeling UHI, varying in their objectives and the data they use. Historically, in-situ temperature measurements were used to model the UHI effect. However, with the increasing availability of remote sensing data and the limited spatial resolution of in-situ measurements, most UHI studies now rely on satellite data. The major challenge faced with satellite-based data is the low temporal resolution often encountered. The maximum temporal resolution is 1 to 2 days, obtained with the TERRA and AQUA satellites [25]. Modeling can be split in two different approach : physics-based (energy balance, 3D models) and measurement-based (weather stations, remote sensing) with the main goal of the studies being to find the links between impacting variables and the UHI intensity.

The methods also differ in the technique to directly asses the UHI effect. While some use the difference between the temperatures measured in the urban context and the temperatures measured in neighboring rural areas, other directly use Land Surface Temperature (LST) as a proxy [25]. The methods can be categorized into two main groups: experimental and empirical:

- **Experimental models** : These methods are useful because they overcome a key limitation of empirical

models : the reliance on data availability. They also reflect more diverse and complex conditions. The main limitation is often the computational cost. Kim and Brown [26] identifies three types of experimental studies :

1. Simulations with **atmospheric numerical models** : The main methods are the Weather Research and Forecasting (WRF) model and the Urban Canopy Model (UCM). While the WRF model is a widely used numerical model to simulate and forecast weather, the UCM integrates various heating and radiation sources to simulate the UHI effect.
 2. Simulations with **computer softwares** : CFD-based methods to study the energy balance at a street scale. UHIs are explained from the physical perspective.
 3. Simulations with **physical scale models** : Wind-tunnel facilities with reduced scale models. Effective to study energy balance or the urban geometry effect, but it neglects other factors
- **Empirical models** : Empirical approaches rely on observations of the components that characterize the UHI effect. Statistical methods are then applied to derive models from these observations. However, they face two primary limitations: data availability and model interpretability. The resulting models often take the form of regression or correlation analyses. Bi-variate regression is commonly used to link the UHI effect with weather data or urban morphology, while multiple regression analysis helps identify the factors that are most or least sensitive to the UHI effect [27]–[29]. Additionally, correlation analysis is primarily used to examine the direct relationships between the UHI effect, solar radiation intensity, and other influencing factors. Machine learning models can be classified as empirical.

Machine learning methods have surged in 2012 [30] with the idea of finding important factors in the formation of UHI effect and building predictive models, using more elaborate statistical methods than regression.

1.2.2 Spatial machine learning and its application to UHI modeling

Spatial Machine Learning (SML) focuses on the application of machine learning techniques to spatial datasets, such as environmental or geographic data [31]. While traditional machine learning approaches can approximate spatial relationships using proxies like distance to specific points, SML provides methods tailored to spatial structures. These include Convolutional Neural Network (CNN) for gridded (raster) data and graph-based machine learning methods for non-gridded data, such as data from a network of weather stations [32].

The use of machine learning for UHI modeling began in 2012 with Mirzaei, Haghighat, Nakhaie, *et al.* [30], who developed an artificial neural network to predict indoor temperatures resulting from the UHI effect in Montreal. Their model incorporated features such as wind velocity, building occupancy, vegetation ratio, and meteorological variables. In 2017, Malings, Pozzi, Klima, *et al.* [33] implemented a Gaussian process model to predict temperature fields. Subsequently, Hardin, Liu, Cao, *et al.* [34] used a random forest regression-based kriging model to predict air temperature using environmental features such as humidity and wind speed. The temporal aspect of UHI modeling was first addressed by Han, Ang, Malkawi, *et al.* [35], who employed a Recurrent Neural Network (RNN) to enhance localized weather predictions and building performance simulations. Since 2023, the number of studies leveraging machine learning for UHI-related tasks has surged. For example, Shafi, Jain, and Zaman [36] used algorithms like Naive Bayes, Support Vector Machine (SVM), Random Forest, and Gradient Boosting to understand the impact of Land use and Land cover (LULC) on LST.

Similarly, Alam [37] combined Machine Learning (ML) with spatial modeling to predict LULC changes and their effects on UHI. Rao, Tassinari, and Torreggiani [38] investigated the UHI-land use relationship under climate change using satellite data and gradient boosting regression, finding significant impacts of vegetation index changes on the LST. For the city of Bern, Burger, Gubler, and Brönnimann [39] proposed a land-use regression to model the nocturnal air temperature fields over several years.

Other studies have used explainable ML methods for UHI analysis. For instance, Ming, Liu, Li, *et al.* [40] proposed a combination of geographically weighted regression, random forest, and Shapley Additive Explanations (SHAP) analysis to examine spatial disparities in UHI effects. Liu, Wang, and Bai [41] used Gradient Boosting and SHAP to analyze nonlinear relationships between environmental features and UHI, identifying greening as a key mitigation strategy. Additionally, Tanoori, Soltani, and Modiri [42] and Li, Yigitcanlar, Nepal, *et al.* [43] developed models integrating advanced ML methods like Convolutional Long Short-Term Memory (ConvLSTM) and spatiotemporal convolutional neural networks (STCNN) for long-term trend prediction and urban planning insights.

Graph Neural Network (GNN) emerged following the introduction of geometric machine learning by Bronstein, Bruna, LeCun, *et al.* [44], which extended deep learning methods to non-Euclidean domains. GNNs have since been applied to different spatiotemporal tasks, including traffic state predictions [45], electricity load forecasting [46], and environmental data modeling [47]. Despite their widespread use in spatiotemporal modeling, the application of GNNs to UHI prediction remains limited. Two notable examples are Yu, Li, Huang, *et al.* [48], who employed GNNs to model street-scale temperatures using a network of weather stations in Chicago, and Yu, Shi, and Xu [49] with the development of a Graph Attention Network with a Gated Recurrent Unit (GATG) to predict temperature at a country scale.

The increasing use of SML and GNN methods in UHI modeling shows how these approaches can deal with complicated spatial and temporal interactions. This improves our ability to predict and mitigate UHI effects, for example buildings energy consumption.

1.2.3 Buildings energy consumption relation with temperature and UHI

Various methods exist to model the energy consumption changes caused by UHI. The building sector, which represents about 40% of Switzerland's global final energy consumption [50], is a significant focus of decarbonization efforts. Buildings energy consumption is influenced by numerous factors, including occupancy patterns and structural characteristics, but outdoor temperature plays a pivotal role. It drives heating and cooling demands, which represent nearly 40% of the total energy consumption [51]. Consequently, understanding and monitoring UHI effects are critical for applications such as electrical load forecasting [10].

The methodology to account for UHI-related Building Energy Consumption (BEC) typically involves three steps: collecting data to quantify UHI, estimating BEC, and reporting UHI's impact [10]. Two main modeling approaches are employed for this purpose: top-down and bottom-up methods. Bottom-up methods, in turn, include physics-based and statistical models:

- **Physics-Based Models (PBM):** These models simulate BEC by applying thermal transfer principles, relying on detailed data about external temperatures and building characteristics. PBMs are primarily used at highly localized scales (e.g., individual buildings) to provide high-resolution temporal energy consumption estimates. Toparlar, Blocken, Maiheu, *et al.* [52] utilized a Building Energy Simulation

(BES) software to assess the impact of the UHI effect on a building, varying characteristics such as orientation, form, materials, and usage.

- **Statistical Models:** These models establish statistical relationships between BEC and ambient temperature. They are scalable and relatively easy to implement but depend on the availability of energy consumption data for calibration. While statistical models can operate on larger spatial scales, their accuracy diminishes as the scale increases. For example, Radhi and Sharples [53] uses a regression method to quantify and predict the impact of the increased urban Cooling Degree Days (CDD) on the electricity consumption in Bahrain, and show that using the temperatures measures in the airport would lead to an error of up to 6%.

Once a relationship between energy consumption and temperature is established, UHI effects are generally expressed as a percentage change. A significant challenge in this field is the limited availability of high-resolution spatial data with extensive coverage. Studies have reported increases in cooling energy consumption ranging from 10% to 120%, alongside reductions in heating energy demand between 3% and 45%, with the effects being more pronounced in urban centers compared to rural areas [10].

Sailor [54] highlights that Heating Degree Days (HDD) and CDD are the primary climate variables influencing electricity consumption. They report that even a small temperature increase can lead to a rise in electricity consumption by as much as 70%. Similarly, França, Almeida, Lucena, *et al.* [55] developed two machine learning models: one for daily and another for monthly electrical load forecasting—considering the spatial distribution of weather stations to account for the UHI effects. Their findings indicate a monthly electricity consumption increase of approximately 7000 MWh for a 3°C temperature rise for the metropolitan area of Rio de Janeiro. Comparably, Santamouris, Cartalis, Synnefa, *et al.* [56], Zhou, Zhuang, Yang, *et al.* [57], and Guattari, Evangelisti, and Balaras [58] collectively contribute to understanding the impact of the UHI effect on energy demand, with each study addressing different aspects and scales of the phenomenon. Santamouris, Cartalis, Synnefa, *et al.* [56] provides a broad review of UHIs influence on city-wide electricity demand and peak power usage. They emphasize that factors such as air conditioning prevalence, building thermal quality, and electricity load composition shape the extent of cooling demand. This aligns with Zhou, Zhuang, Yang, *et al.* [57], who highlight how urban morphology influences energy consumption, particularly through reduced heating demand with higher ambient temperatures. Together, these studies underline the importance of local characteristics, such as building stock and urban form, in determining the energy impact of temperature variations. Expanding on these findings, Guattari, Evangelisti, and Balaras [58] focus on a specific case study in Rome, examining localized UHI effects by comparing urban and suburban temperature data. Their results validate the broader trends reported by Santamouris, Cartalis, Synnefa, *et al.* [56], showing a significant increase in the cooling energy demand (30%) alongside a reduction in the heating demand (11%). These studies demonstrate that while UHI effects universally increase cooling demand, the magnitude and balance with reduced heating demand depend on localized factors such as urban morphology, building characteristics, and climatic conditions. They emphasize the need for integrated approaches that account for these variables when assessing UHI impacts and formulating mitigation strategies.

1.3 Objectives and Research questions

The literature review outlines the current state of research on the Urban Heat Island (UHI) phenomenon, its causes, impacts, modeling methods, and mitigation strategies. Although machine learning and deep learning

techniques have been extensively applied to air temperature and UHI phenomena, newer methods, such as graph-based approaches, remain under-explored, particularly in a context with high spatial and temporal resolution. The following research questions are addressed in this thesis:

1. How can graph-based machine learning methods, combined with temporal dynamics processing, improve UHI modeling performance, and how do they compare to traditional approaches?
2. What is the impact of the UHI effect on Bern's region building energy consumption

1.4 Novelties

The UHI phenomena in Bern has been widely studied by the *Centre for Climate change Research* of the University of Bern, with the *Urban Climate Bern* project, and both the spatial and temporal (daily) patterns were analyzed, mainly using land-use regression, proving the existence of the UHI effect in the city of Bern. Thanks to this project, a high spatial and temporal resolution dataset of the urban temperatures of Bern during four consecutive summers is available [59].

The main contributions of this study lie in the development and application of an advanced machine learning technique to model the UHI phenomena. A key novelty, in this context, is the design and implementation of an attention-based graph neural network combined with a temporal memory bank. This approach enables the effective modeling of spatial and temporal interactions among weather stations, producing accurate hourly predictions of UHI intensity time series. This architecture is used for predicting UHI intensity time series at a hourly horizon by leveraging a dense network of temperature-measuring stations in the city of Bern, Switzerland.

This Master project address current literature gaps in the application of advanced machine learning techniques for UHI modeling, providing innovative methods for leveraging spatial and temporal interactions to achieve accurate and high-resolution predictions of urban heat dynamics. This study will also evaluate and quantify the impact of the UHI effect on the cooling load in the area of Bern.

Chapter 2

Background

In this chapter, we provide an overview of the machine learning techniques employed in the study. These methods span a range of approaches, from gradient-boosting algorithms to advanced neural network architectures and graph-based methods, each selected for its unique strengths in handling different aspects of UHI modeling. By combining these techniques, we aim to leverage their complementary capabilities to improve prediction accuracy and interpretability.

The methods are organized as follow: starting with a conventional technique (gradient-boosted decision trees), moving through temporal modeling (Gated recurrent unit), then introducing fusion and how it enhances performance, and concluding with graph machine learning, which explicitly captures spatial relationships.

2.1 Machine learning techniques description

2.1.1 XGBoost Algorithm

Extreme Gradient Boosting (XGBoost) is a powerful and widely-used machine learning algorithm specifically designed for high speed and performance. It builds upon the principles of gradient boosting by introducing various optimizations that enhance its scalability and computational efficiency, making it suitable for both small and large-scale datasets. It was introduced formally in 2016 by Chen and Guestrin [60].

The main idea behind gradient boosting is to create a group of simple models (in the case of XGBoost, decision trees) that work together in an iterative process to improve predictions. Each new tree in the sequence is built to correct the errors made by the previous trees, helping the model become more accurate over time. Unlike regular decision trees, which can sometimes be too "unstable" (high variance) or too simplistic (high bias), the gradient boosting framework helps reduce these problems by focusing on fixing the mistakes of earlier trees, making the model more reliable and precise.

XGBoost improves upon traditional gradient boosting methods through several key innovations. It incorporates L_1 (Lasso) and L_2 (Ridge) regularization to penalize overly complex models and prevent overfitting,

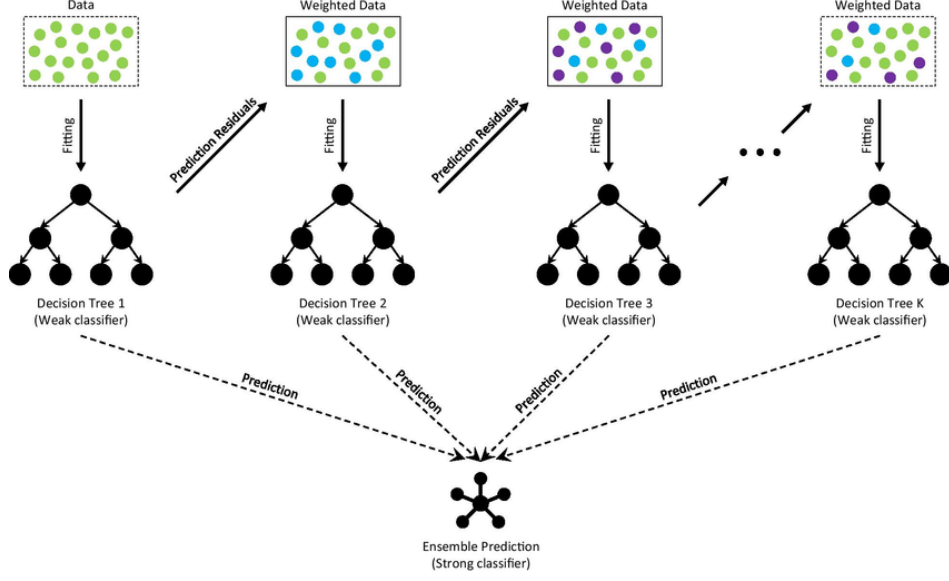


Figure 2.1: Schema of the gradient boosting process. At each iteration, a new decision tree is fitted on the residuals of the previous tree, therefore reducing the error at each iteration. The final prediction is a combination of the prediction of all the trees[61]

enhancing generalization capabilities, to get better performance on unseen data. Lasso (Least absolute shrinkage and selection operator) is a regression method that consists of shrinking the coefficients of less important features to exactly zero, effectively performing feature selection. This is achieved by adding the L_1 -norm penalty term (sum of the absolute value of the coefficient) to the loss function, which encourages sparsity in the model. Ridge regression, on the other hand, adds an L_2 -norm penalty term (sum of the square of the coefficients) to the loss function, which discourages large coefficient values but does not shrink them to zero. By combining both Lasso and Ridge regularization, XGBoost achieves a trade-off between feature selection and coefficient shrinkage, leading to better handling of overfitting while maintaining model complexity.

The algorithm supports parallelized tree construction, significantly reducing training time compared to conventional boosting methods. Additionally, XGBoost employs a technique known as *max-depth pruning* or the *depth-first approach* to cut branches that do not contribute to better predictions, helping the model to stay efficient by focusing on improving the branches of the tree that enhance accuracy. Furthermore, XGBoost uses a method called weighted quantile sketch which computes approximate quantiles of each feature to find the best split at the decision nodes, even when the data is imbalanced, ensuring better performance.

The output is given by Chen and Guestrin [60]:

$$\hat{y}_i = \phi(\mathbf{x}_i) = \sum_{k=1}^K f_k(\mathbf{x}_i), \quad f_k \in \mathcal{F}, \quad (2.1)$$

where \mathcal{F} is the space of regression trees. To obtain the set of functions f_k the following regularized objective

is minimized:

$$\mathcal{L}(\phi) = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k), \quad (2.2)$$

where $l(y_i, \hat{y}_i)$ represents the loss function measuring the difference between the predicted \hat{y}_i and actual outcomes y_i , and $\Omega(f_k)$ is the regularization term defined as:

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \|w\|^2 \quad (2.3)$$

with γ controlling the complexity of the tree (number of leaves T) and λ governing the regularization of leaf weights w_j .

Gradient tree boosting is a modification of equation 2.2 where the second order approximation of the function f_k is made to obtain a quickly optimizable loss function (proposed by Friedman, Hastie, and Tibshirani [62]). The following loss function to minimize at the step t is obtained:

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^n \left[g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i) \right] + \Omega(f_t) \quad (2.4)$$

While XGBoost also optimizes other parameters to improve the performance of the model, the mathematical details can be found in [60]. XGBoost has been widely used in all sorts of machine learning tasks, and derived into libraries for numerous programming languages, thus making it a very popular choice for machine learning tasks. the library `xgboost` in python is used here.

While XGBoost excels at capturing non-linear relationships and interactions, it lacks the ability to effectively model temporal dependencies inherent in sequential data. To address this limitation, the Gated Recurrent Unit (GRU) is introduced, a type of RNN specifically designed to capture long-term temporal patterns and dependencies in time-series data.

2.1.2 Gated recurrent unit

The GRU model is designed to efficiently process sequential data, such as time series, by selectively remembering important past information. It processes each element of the sequence one at a time, updating its internal state to capture the relationships and patterns in the data over time.

The GRU consists of two main components: the update gate and the reset gate. The update gate determines how much of the previous hidden state (the model's memory) should be kept for future predictions, while the reset gate decides which parts of the previous memory should be forgotten. The reset gate's output is then combined with the current input and the previous memory to form a new "gate" that updates the memory. This new memory is then combined with the update gate to produce the model's updated hidden state, allowing it to maintain relevant information and forget what's not needed.

For illustration Figure 2.2 describes how a GRU cell (the fully gated version) works.

At each step, the GRU cell processes two main inputs:

- h_{t-1} : This is the "memory" from the previous step. It contains information the network has learned so far.

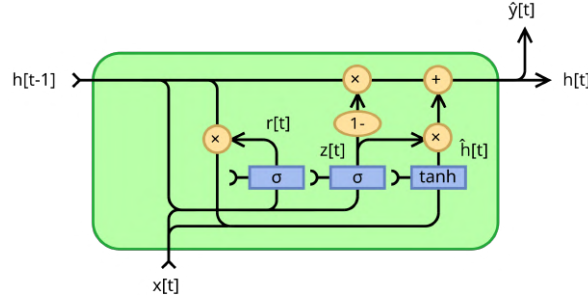


Figure 2.2: Gated Recurrent Unit, fully gated version (Wikipedia)

- x_t : This is the new input data at the current step.

It has two main gates: the reset gate r_t and the update gate z_t . The reset gate r_t decides how much of the past information to forget, while the update gate z_t controls how much of the new information to add. It uses a mathematical function called the sigmoid (σ) to produce a value between 0 and 1:

- A value close to 0 means "forget."
- A value close to 1 means "keep."

The new memory (Candidate Memory) \tilde{h}_t is computed using the reset gate and the previous old memory (hidden state) h_{t-1} , combined with the current input x_t , passed through a function called \tanh , which helps in shaping the data.

Compared to other sequence processing architecture, the GRU address the vanishing gradient problem observed in basic RNN architectures, while simplifying the Long Short-Term Memory (LSTM) architecture, with less parameters and a lower computational cost [63].

Even if the GRU effectively captures the dependencies in temporal (and sequential) data, its performance can be further enhanced by incorporating additional contextual information. This is where feature fusion comes into play, combining diverse data sources or features to enrich the model's input and provide a more comprehensive understanding of the underlying patterns.

2.1.3 Feature fusion

Feature fusion is the process of combining features from different sources of information to obtain a unified and extensive view of the input features. It can allow to capture the context at different scales, e.g. for image recognition [64], or to fuse spatial and temporal information for spatio-temporal tasks, in traffic prediction [65], sea surface temperature prediction [66] or action recognition [67].

While feature fusion enrich the model's input with the additional context, the spatial dependencies and interactions are not considered. To address this, graph-based machine learning (GML) deal with this limitation by using the graph representation of the input data to model the spatial interactions at different scales, offering a comprehensive approach to UHI modeling.

2.1.4 Graph machine learning

Graph Machine Learning (GML) is useful to process any kind of data that can be represented as graph. A graph is a set of nodes (vertices) that represent entities and edges (links) representing the relationships between the entities. Nodes and edges can have properties associated with them, such as input features, locations for nodes, or distance and correlation between nodes for edges. The graph is formally described as:

$$G = (V, E) \quad (2.5)$$

where V is the set of nodes and E the set of edges [68]. The edges can be either directed or undirected, and may carry a weight. The graph can also be represented as an adjacency matrix A where $A_{ij} = 1$ if there is an edge between the nodes i and j , and 0 otherwise ($A_{ij} = \pm 1$ if the graph is directed + indicates directed link from node i to j).

GML uses the spatial relations between the nodes (represented with the edges) to make predictions. It works similarly to a CNN, but can be applied to non-Euclidean data (Figure 2.3).

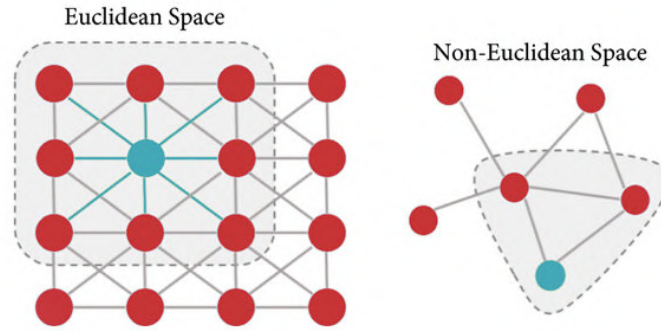


Figure 2.3: Difference between euclidean and non-euclidean data [69]

A graph neural network is designed in two main steps. First, the characterization of the graph and, second, the design of the loss function [68]. The graph can be:

- Directed or undirected : edges can have as feature its direction (e.g.m 'an arrow' from node i to node j)
- Homogeneous or heterogeneous : The nodes and edges types can (or not) be different
- Static or dynamic : The graph can (or not) evolve over time

The loss function design depends on two factors: the type of task and the training setting. The task type can be:

- Node-level: Prediction is made for individual nodes, either as a classification (discrete values) or regression (continuous values)
- Prediction is made for the edges (i.e., determining properties of the relationships between nodes, such as whether there is an edge or predicting its weight)

- Graph-level: Prediction is made for the entire graph, such as classifying a graph into categories, performing regression, or clustering graphs

The training setting can be:

1. Supervised: The provided training data is labeled
2. Semi-supervised: A small proportion of the training data is labeled. This leads to two further settings: transductive, where the model predicts the label of nodes seen during training, and inductive, where the model predicts the label of new nodes
3. Unsupervised: No labels are provided (e.g., clustering)

Knowing these information, the architecture of the GNN can be initiated. The main components of the GNN architecture is the propagation module, the pooling module and the sampling module if the graph is very large. The propagation modules are the equivalent of the convolution operator to the graph domain, and are categorized in two parts : spectral-based and spatial-based. Spectral-based propagation modules leverage the spectral representation of graphs (i.e. the associated matrix representation) and define a convolution operator in the spectral domain (spectral networks [70], ChebNet [71] or Graph Convolutional Network (GCN) [72]). The pooling module is designed to create a subgraph from the base graph in order to reduce the set of edges and nodes while keeping the maximum of information from the base graph [73]. The sampling module helps the processing of large-scale datasets by reducing the size of the processed data.

Spatial-based approaches define the convolution operator directly using the graph topology. The major issue is to maintain the local invariance of CNN (patterns can be recognized regardless of their position in the input space) while having a non-euclidean domain. An example of a spatial-based approach is the GraphSAGE framework [74], where the convolution operator is defined as a weighted operation on the features of the node and its neighbors (mean, LSTM or pooling). It adapts on large graph by sampling a fixed number of neighbors for each node.

More advanced spatial-based approaches, such as the attention-based approach, have been introduced to improve performance. The attention mechanism, first introduced by Bahdanau [75] for weighting the importance of components in sequences, is applied in GML to derive the weights associated to each neighbor of a node. This enhances performance by focusing on the most relevant neighbors. Graph Attention Network (GAT) [76] incorporates the attention mechanism into the propagation step. The hidden state \mathbf{h}_v^{t+1} (feature vector of the node v at layer $t + 1$) of a node v is given by

$$\mathbf{h}_v^{t+1} = \rho \left(\sum_{u \in \mathcal{N}_v} \alpha_{vu} \mathbf{W} \mathbf{h}_u^t \right), \quad (2.6)$$

where

$$\alpha_{vu} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^T [\mathbf{W} \mathbf{h}_v \| \mathbf{W} \mathbf{h}_u]))}{\sum_{k \in \mathcal{N}_v} \exp(\text{LeakyReLU}(\mathbf{a}^T [\mathbf{W} \mathbf{h}_v \| \mathbf{W} \mathbf{h}_k]))}. \quad (2.7)$$

\mathcal{N}_v is the set of neighbors of node v , ρ is the activation function, α_{vu} is the attention coefficient from neighbor u to node v , \mathbf{W} is the learnable weight matrix that transforms the features of neighbor u and LeakyReLU is a non linear function that allow the negative part to be very small instead of being 0.

In simplified terms, the updated feature of the node v is a weighted combination of the transformed

features of its neighbors, with weights α learned through the attention mechanism. The attention mechanism computes the attention score of node u to node v by determination of the similarity of their features (through the learnable weight matrix \mathbf{W}) and with the learnable attention vector \mathbf{a} that learns the important features for the specific task. The attention mechanism can also be multi-head (several representations of the attention mechanism are learned in parallel and then concatenated) to improve the model's performance and stability.

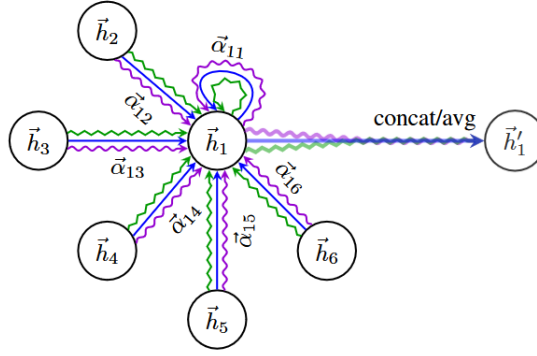


Figure 2.4: Illustration of a 3-heads attention mechanism by node 1 [76]

Chapter 3

Data and Methodology

Understanding and predicting UHI effects require a comprehensive approach that integrates diverse datasets and advanced machine learning methods. This chapter describes the study area and the various data sources used in this work, including temperature measurements, urban morphology, geographic characteristics, environmental features, and temporal variables. Additionally, the methodology for constructing the graph representation of the sensor network is described, setting the basis for applying graph machine learning techniques to model and analyze the UHI dynamics in Bern.

3.1 Data

3.1.1 Study Area and context

The studied area is the city of Bern, and its surroundings, located in central western Switzerland, on the Swiss Plateau, 20 kilometers north of the Bernese alps. Bern is crossed by the Aare river, which forms a kind of canyon in the city. The two highest points in the Bern area are Gurten (south of Bern), with an elevation of 864 meters and the Bantiger with an elevation of 947 meters. The average elevation of the area is 553 meters and the lowest elevation point is 498 meters. Bern is the 5th most populated city in Switzerland, with around 130 000 inhabitants in the city and around 400 000 in the greater area in 2020 [77].

To characterize the land cover, the local climate zone classification is adopted [78] composed by 10 urban classes and 7 natural classes. As shown on Figure 3.1b, the city of Bern is composed of a compact mid-rise city center (dense mix of 3 to 9 stories buildings with low to no vegetation) surrounded by open mid-rise and low-rise (open arrangements of mid-rise and low-rise buildings with abundance of pervious land cover). The mountains surrounding Bern are covered in dense forest and its worth noting the large industrial area in the north-east of the city, classified as large low-rise (open arrangements of large low-rise buildings, with impervious land cover).

Currently classified as having a continental climate with no dry season and hot summers according to the Köppen-Geiger classification [79], Bern is projected to transition to a temperate climate with no dry season and hot summers by 2100, driven by climate change and an expected increase of 4°C in mean air temperature

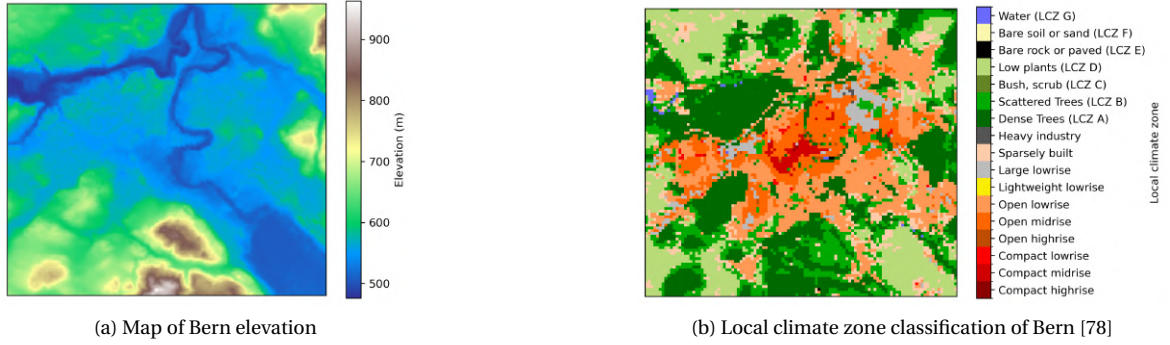


Figure 3.1: Comparison of Bern elevation and local climate zone classification.

[79]. The annual average temperature in Bern's canton is 8.8 °C with an average yearly precipitation of 1059 mm during the period 1981-2010 [80].

The dataset used in this study contains measurements for four summers: 2019, 2020, 2021, and 2022. During these periods, the average temperature is 19.44°C, with a nighttime mean of 16.2°C and a daytime mean of 21.85°C, Figure 3.2. The number of tropical nights ($T_{avg} > 20^{\circ}\text{C}$) recorded is 20 in 2019, 11 in 2020, 8 in 2021, and 20 in 2022.

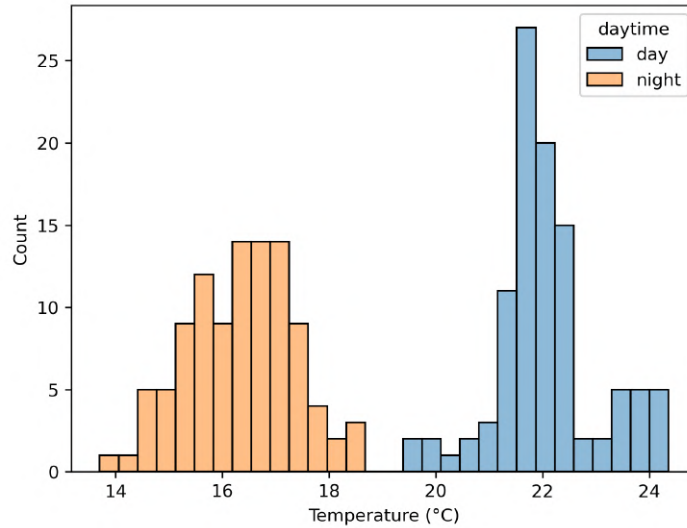


Figure 3.2: Distribution of the daytime and nighttime average temperature at each stations over the four summers.

3.1.2 Low-cost measurement devices

The temperature-measuring station network originates from the study by Gubler, Christen, Remund, *et al.* [81], which installed a dense network of LCD in Bern to evaluate intra-urban temperature differences during the summer of 2018. The LCD are located 3 meters above ground and measure the temperature with 10

minutes intervals. For this study, the temperature data was hourly averaged. They showed that the LCD approach can be used to model UHI patterns at a very fine scale while not being costly. The device network consisted of 79 measurement devices at the beginning, and in total 103 were installed, with 2 of them being in Zollikofen, near the MeteoSwiss reference weather station (Figure 3.4).

The locations of the station are very diverse, and cover the entire metropolitan area (one station near the airport in Belp, a few stations in Koniz or Ostermundigen). Figure 3.3 presents the data availability through the studied period (2019-2022). Each red vertical line represents the separation between two consecutive summers. The data is separated in two : the train set (2019, 2020, 2021) that will be used to train the models and the test set (2022) that will be used to evaluate the models performance.

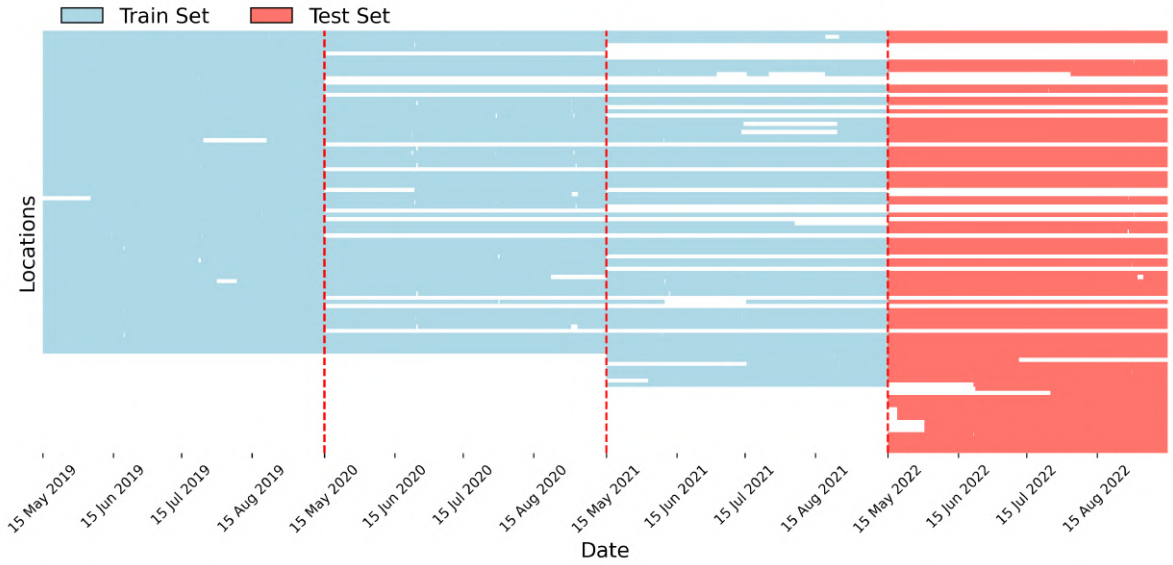


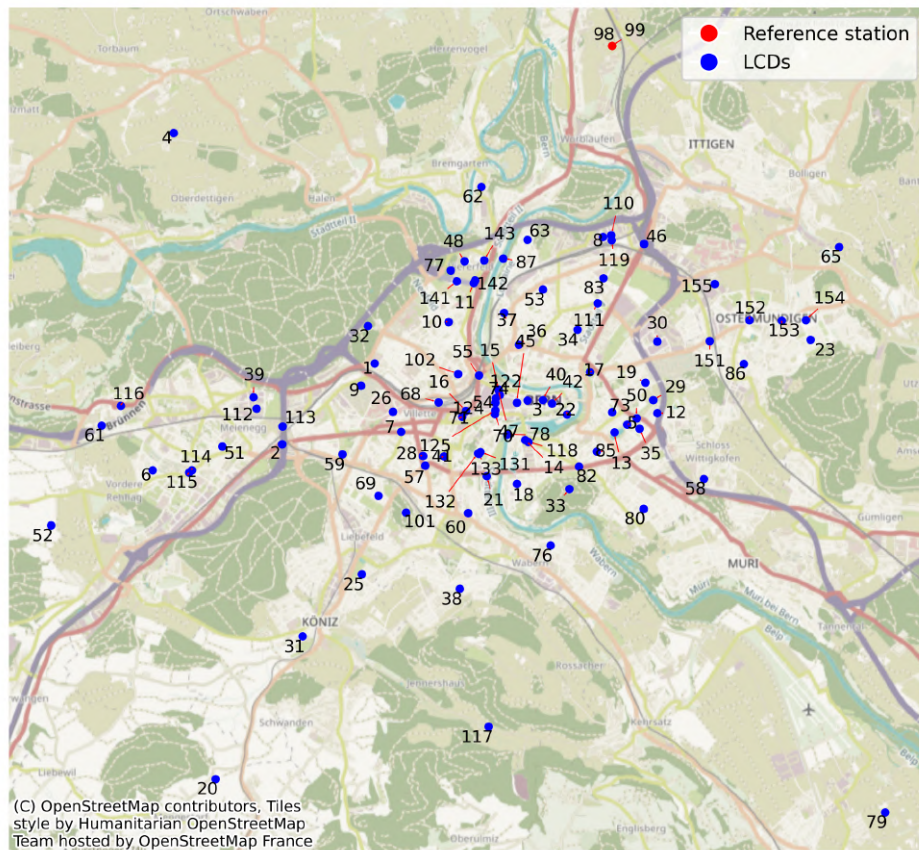
Figure 3.3: Data availability. Blue or red means that the LCD is functioning at the date, white means that the LCD is either not installed or that the data is not available. Blue is the training data and red is the testing data. The validation for the XGBoost model was performed with a 5-folds cross validation on the train set, while it was done using the 2022 summer for the STFGAN (see 3.4)

3.1.3 Target definition: Temperature delta

As indicated previously the urban heat island intensity is defined by the temperature difference between the urban area and its rural surroundings. The rural surroundings temperature is defined as the temperature from the Zollikofen weather station. Then, at every time step t , the temperature delta for another station i is computed with :

$$\Delta T_{i,t} = T_{i,t} - T_{\text{Zollikofen},t} \quad (3.1)$$

Figure 3.5 illustrates the distribution of these temperature deltas for daytime and nighttime periods, highlighting the variability in UHI intensity across different period of the day.



(a)



(b)

Figure 3.4: Overview of the stations with (a) the locations of the LCDs and (b) the MeteoSwiss Zollikofen weather station.

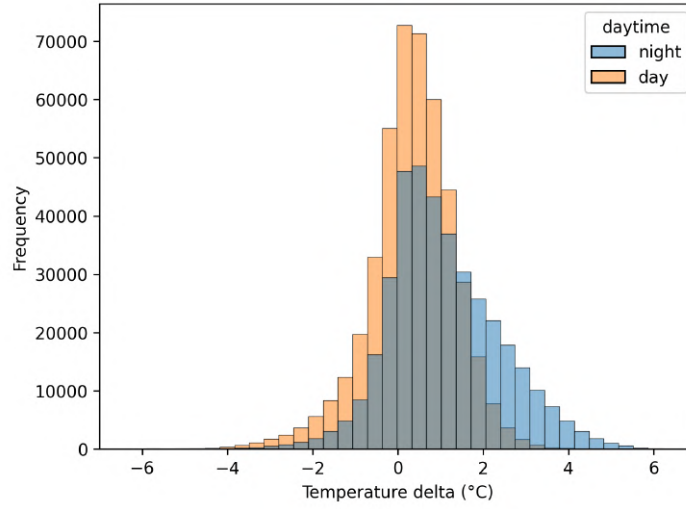


Figure 3.5: Distribution of temperature deltas (ΔT) for daytime (orange) and nighttime (blue) periods, showing the frequency of temperature differences between the urban stations and the rural reference at Zollikofen.

3.1.4 Urban morphology

Urban morphology is a key factor in understanding how city structures influence urban weather dynamics. This subsection describes the features extracted from the Swiss3DBuildings dataset [82] to characterize building distribution and density.

From this dataset, three features were extracted:

- **Buildings height** : Extracted from the 3D shape of the building, and then averaged inside a 100 meters buffer around each LCD
- **Buildings count** : The number of buildings inside a 100 meter buffer around each LCD
- **Construction density** : the average distance between the buildings at three level : the 50 nearest buildings, the 100 nearest and the 200 nearest. For each LCD, the average distance of all the buildings inside the buffer is computed.

Figure 3.6 illustrates the differences of urban structure Bern can have between the city center and the surroundings.

3.1.5 Spatial and demographic properties

In the following, we introduce metrics that describe the relationship between each LCD and its surrounding environment, including features such as distance to the city center, population distribution, elevation, and land cover. These features aimed to characterize urban morphology within its larger geographic setting.

The features considered include:

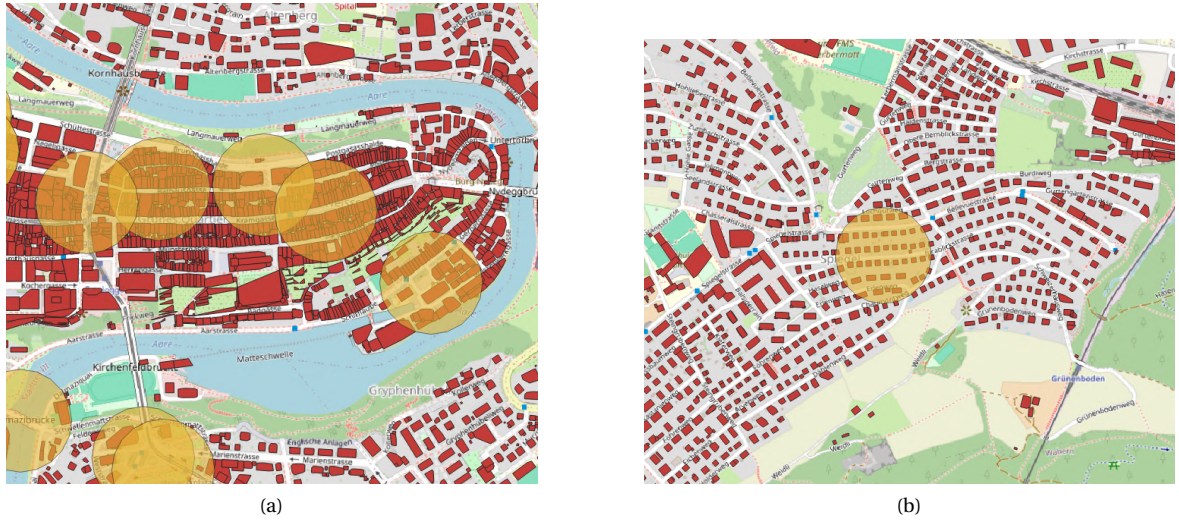


Figure 3.6: Examples of urban morphology in (a) the city center and (b) the surroundings of Bern. The yellow circles represent the LCD with the 100m buffer, and the red polygons are individual buildings.

- **Distance to Bern center** : a LCD in Bern center (Log 3, see Figure 3.4a) was chosen as the reference to represent the center of the city, and then the Euclidean distance was used to compute the distance between each station and the reference station.
- **Population**: the population data is taken from the STATPOP dataset of Geoadmin [83] in 2022. This dataset counts the population in each inhabited 2500m² grid cell in Switzerland. To each LCD is associated the population of the grid cell it belongs.
- **Elevation**: the elevation data is derived from NASA Shuttle Radar Topography Mission (SRTM) dataset [84] with a resolution of 30 meters, Figure 3.1a.
- **Land cover**: the land cover (Figure 3.1b) is obtained from the local climate zone classification dataset from Demuzere, Kittner, Martilli, *et al.* [78]. It consists of 17 classes, 10 urban classes and 7 natural classes, with a resolution of 100 meters, originally designed as a framework to study urban heat islands.
- **Distance to the nearest green space**: The green spaces with an area above one hectare were extracted from the swissTLM3D dataset [85] with QGIS, and then the distance from each LCD to the nearest green space was computed.

3.1.6 Environmental characteristics

Environmental characteristics are crucial to assess the natural and built components of cities and their influence on urban climate. Key indicators for understanding the environmental context of urban areas were computed:

- **NDVI** : Normalized differentiated vegetation index. This index is a proxy for vegetation density. It is

computed as follow from remote sensing data :

$$NDVI = \frac{NIR - Red}{NIR + Red} \quad (3.2)$$

where NIR and R are the spectral reflectance measurements taken from, respectively, the near-infrared and red regions of the spectrum. The data used originates from the Sentinel-2 satellite [86] and has a resolution of 10 meters. It is the averaged values of all the 2023 images of Sentinel with less than 10% of cloud cover, Figure3.7.

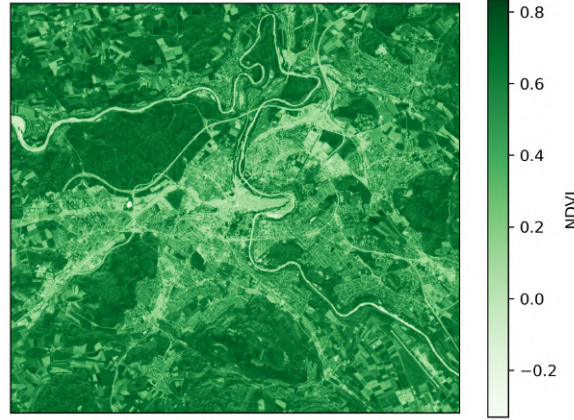


Figure 3.7: NDVI map of Bern

- **Albedo** : The albedo is a measure of the surface reflectance, indicating the total fraction of the sunlight that is diffusively radiated. It is computed as follow :

$$\alpha = \frac{J_e}{E_e} \quad (3.3)$$

J_e being the radiosity of the surface and E_e the irradiance received by the surface. The data used originates from the MODIS dataset [87], and has a resolution of 500 meters. Similarly as the NDVI, the values were averaged over 2023.

3.1.7 Temporal features

Several temporal and weather-related characteristics were included as exogenous variables to account for the time of day, the month, and prevailing weather conditions. Specifically, the hour and month were added to represent temporal variations. Since weather measurements are not available for each LCD, global weather data from the reference weather station was used. The following weather features, obtained from MeteoSwiss [88], were incorporated:

- Vapor and atmospheric pressure
- Global irradiation and hourly sunshine duration

- Maximum, minimum and average hourly temperature
- Hourly precipitations
- Hourly humidity
- Hourly dew point
- Hourly wind speed and direction

3.1.8 Graph construction

The graph representing the network of LCDs is constructed using the `knn_graph` function from the `torch_cluster` library [89]. The resulting graph is directed and includes self-loops (each node is connected to itself). The features used to compute distances between nodes are population, average building height, elevation, distance to the center of Bern, and distance to the nearest park or forest. A decision was made to assign 10 neighbors to each node (including itself) as a trade-off between ensuring representativity and managing computational resources, resulting in a graph with 101 nodes and 1,010 edges, Figure 3.8.

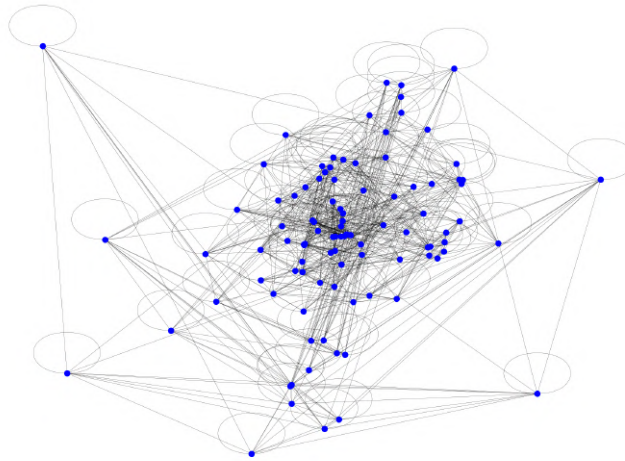


Figure 3.8: Final graph structure used for the model implementation

3.2 Methods: Machine learning models

Two methods were used to predict the temperature delta timeseries : one XGBoost model used as a baseline predicting model to assess the performance of and a STFGAN. Both models were trained using the same data, although the format of the data was adapted to each model, and on a T4 GPU (using Google Colab service).

3.2.1 XGBoost

Among all possible ML algorithms, XGBoost was chosen for several reasons: First, it is very quick to train on large datasets, because it uses second-order approximation to compute the gradient of the loss and it supports parallelization at multiple steps of the process (tree construction and histogram-based approach for numerical features). Then, from Wang, Yang, Chen, *et al.* [16], it is one of the best performing ML algorithm on temporal air temperature prediction studies in the period 2019-2022, and, finally, because it is very easy to implement thanks to the xgboost library in python [60].

3.2.2 Spatio-temporal Fusion graph attention network

Due to the spatio-temporal nature of the UHI phenomenon, the STFGAN was specifically designed to capture both its temporal and spatial dimensions.

First, a linear layer transforms the initial feature set into a higher-dimensional space, allowing the model to learn more complex feature representations. Then, to account for the spatial interactions, two Graph attentional operator (GATConv) layers [76] were used to propagate each node information to its neighbors. These layers propagate information from each node to its neighbors, enabling the model to capture the relationships between neighboring nodes in the graph. The first layer captures immediate neighbor information, aggregating features from directly connected nodes, while the second layer extends this process by capturing information from the neighbors of those neighbors. This propagation allows the model to learn both local and more global spatial relationships within the graph. The attention mechanism (see 2.1) within the GATConv layers automatically assigns weights to each feature and each link, prioritizing the most relevant information based on the importance of the connections and node attributes.

The temporal dimension is processed for by a temporal memory component which is composed of a linear layer to transform the input features into a hidden state, and a GRU-based module (described in 2.1) to process the sequential temporal representations. This memory component captures the temporal dependencies across multiple lagged inputs, allowing the model to effectively encode patterns over time. Additionally, an attention mechanism is applied to dynamically weigh the contributions of past hidden states (from the memory bank) relative to the current features. The attention mechanism employs a multi-layer structure, including a linear transformation, a Tanh activation, and a final Softmax layer to compute the attention weights. These weights are used to emphasize the most relevant parts of the memory bank when combining it with the current input features. Finally, the memory bank is updated by shifting the sequence of hidden states and appending the current feature representation, ensuring a rolling context of the most recent inputs. This approach maintains temporal coherence while integrating historical and present information. Both representations are then concatenated to and passed through linear layers to obtain the final predictions.

The loss function is the mean squared error (Equation 3.4), and, although all the nodes are predicted, due to the lack of data availability for certain nodes, the loss function is computed only on nodes with available data for the processed time step. The gradient is computed and propagated backward among all learnable parameters.

The architecture of the STFGAN model is presented in Figure 3.9.

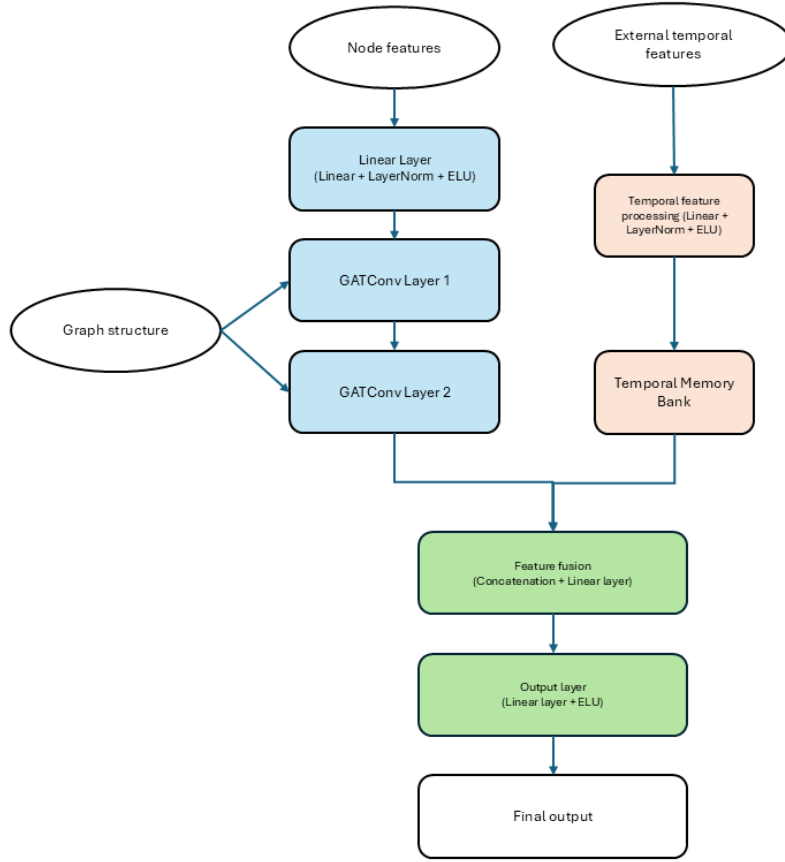


Figure 3.9: Graph attention network architecture

3.2.3 Model evaluation

To evaluate models performances, two metrics are used : the Mean Squared Error (MSE), the Mean Absolute Error (MAE), the coefficient of determination (R^2) and the MdAPE. The MdAPE is chosen to avoid the issue of the Mean Absolute Percentage Error (MAPE) which can become excessively large when the true values are close to zero.

$$\begin{aligned}
 \text{RMSE} &= \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \\
 \text{MAE} &= \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \\
 R^2 &= 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \\
 \text{MdAPE} &= \text{median} \left(\frac{|y_i - \hat{y}_i|}{|y_i|} \times 100 \right)
 \end{aligned} \tag{3.4}$$

Here, y_i represents the actual observed values, \hat{y}_i represents the predicted values and \bar{y}_i is the mean of the actual observed values. An interesting point is that we can test the models on several configuration: depending on the LCD, the training data (i.e. the summers of 2019 to 2021) is either complete, sparse or inexistent, and evaluate the performance on these three different categories.

3.3 Buildings energy consumption and domain generalization

In this work, buildings energy consumption is estimated based on the ML predictions of the UHI intensity (temperature deltas) at every LCD location. For that, the UHI intensity predictions are generalized to the entire study area allowing to evaluate the impact on buildings energy consumption in the Bern area via the cooling degree hours method.

3.3.1 Building's energy consumption

To evaluate the impact of the UHI effect on buildings energy consumption, the CDH are used as a proxy of the buildings energy needs [90]–[92]. They are computed as follows :

$$CDH_h = \begin{cases} T_{cell,h} - T_{base}, & \text{if } T_{cell,h} > T_{base} \\ 0, & \text{if } T_{cell,h} \leq T_{base} \end{cases} \quad (3.5)$$

with $T_{cell,h}$ the temperature of a given cell at time h and T_{base} the base temperature. The base temperature is usually taken between 10°C and 28°C [93]–[96]. The definition of a cooling day by MeteoSwiss is a day where "the average temperature of the day is 18.3°C or higher" [97]. Therefore the base temperature value taken here is 18.3°C

Due to the very heterogeneous nature of the constructions (commercial vs residential, house vs buildings, age of construction or type of isolation), it is difficult to derive precisely the cooling energy consumption increase (or decrease) due to the UHI effect, and therefore the CDH are used as a proxy of the cooling energy demand. After computing the CDH at each stations over the 2022 summer, the difference between each station and the reference station is made to evaluate the impact of the UHI effect on the cooling load, and by extension, to the building energy consumption.

3.3.2 Domain generalization

Domain generalization aims to develop models that perform well on unseen data from different but related domains, addressing the challenge of generalizing knowledge beyond the specific data on which models are trained. In this study, domain generalization is particularly important to enable predictions across the entire city rather than being limited to the locations of the low-cost devices (LCDs).

Several techniques exist for domain generalization [98]–[100], and while they have shown success in general domains, spatial generalization remains more challenging due to these unique spatial characteristics [98]. Basic techniques to address spatial heterogeneity include geographically weighted regression, kriging [101], [102] or land-use regression [39].

To perform the spatial generalization in this study, a ML approach based on XGBoost is developed. The domain (Bern area) is divided into a 150x150 grid and the features that are discussed earlier are attributed to each grid cell. The obtained cells have a resolution of 70m. Then, a XGBoost algorithm is trained on grid cells containing one or several stations using data from the STFGAN model to provide temperature predictions for all stations over the year 2022. The training is done on 80% of the stations and the testing is done on the remaining 20%.

The XGBoost regression model is then applied to all grid cells at each time step, enabling generalization beyond the station locations to the entire study area. This generates hourly temperature delta maps for the entire Bern area, which are used to compute the CDH differential between a reference weather station and the broader study area. By cumulating CDHs over the summer, the total differential across the entire study area is obtained, providing insights into spatial temperature variations and their seasonal impacts

3.4 Implementation and parameters fine-tuning

As described earlier, both models were created using python, with the libraries pandas [103], xgboost [60], scikit-learn [104], pytorch [105] and pytorch-geometric [106]. To make sure that the results were comparable, both models were executed on a T4 GPU provided by the Google Colab service [107]. The XGBoost model training took 20 seconds, whereas one epoch of the STFGAN model required one minute, resulting in a total training time of approximately one hour.

The data used for the XGBoost model consists of a data frame of 611666 samples. Categorical variables (land cover and month) were encoded as one-hot variables and the hour was transformed into two continuous cyclical variables using the following formula :

$$\text{hour sin} = \sin\left(\frac{\text{hour} * 2\pi}{24}\right) \quad (3.6)$$

$$\text{hour cos} = \cos\left(\frac{\text{hour} * 2\pi}{24}\right) \quad (3.7)$$

While the dataframe for the XGBoost model training contains all the necessary variables, it is a bit more difficult for the STFGAN due to the temporal memory block needs. To train the STFGAN, an iteration was done in all time steps and at each time step all the learnable matrices were updated (causing high computational cost). Therefore, the data set used for the STFGAN was composed of one data frame with the node features (static), one dataframe with the edges, one dataframe with the temporal variables at each timestep (with the hour encoded as in the XGBoost dataframe) and one dataframe containing the target at each time step to compute the loss. At each epoch, the model is trained and updated over all the time steps. This process is illustrated in Figure 3.10. The loss is computed at each timestep and averaged over all the time steps to compute the epoch loss.

3.4.1 Parameters fine-tuning

For the XGBoost model, a 5-fold cross validation grid search was performed to find the best hyper parameters. This method separates the data into 5 subsets and fits, for each possible parameter combination, a XGBoost model using 4 of the 5 subsets. This model is then evaluated on the remaining subset. The performance of

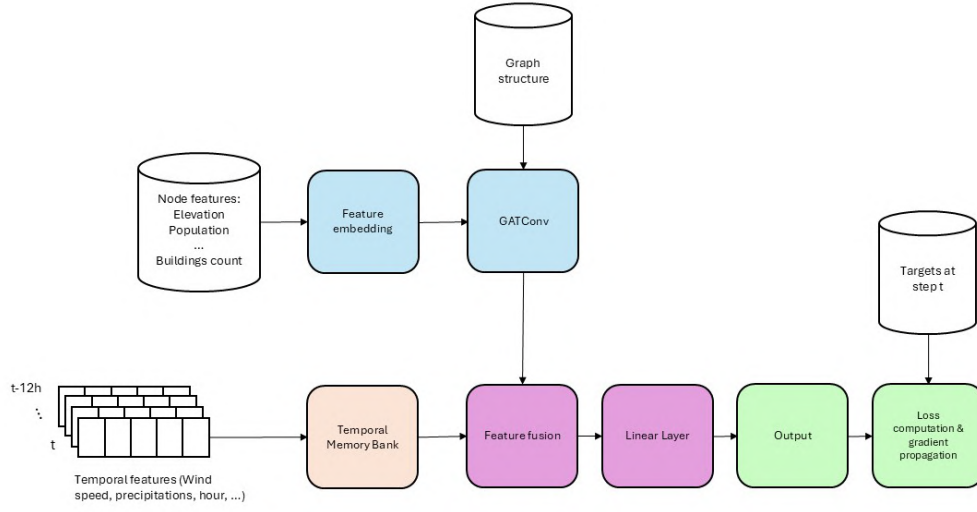


Figure 3.10: Training diagram of the STFGAN for one time step. An epoch is the processing of all the time steps. Blue represents the spatial processing, orange is the temporal processing, purple is the feature fusion and green is the prediction and loss computing.

the parameter combination is the average R^2 score over the 5 folds. The parameters that were tuned are the learning rate, the maximum depth of the trees and the number of trees. The grid search was performed using the scikit-learn library [104], and the results are shown in Figure 3.11.

For the STFGAN, the hyper parameters that could be tuned were the number of attention heads of the GATConv layers, the dimension of the hidden states, the number of lag steps in the temporal memory bank, the dropout rate to avoid over-fitting, the activation function and the number of epochs. Due to the high computational cost of training the model, the fine-tuning of the parameters was difficult to assess. Default values were chosen for all the hyper parameters excepted the number of epochs. To obtain the optimal number of epochs, the model was trained over the first three summers and validated on the 2022 summer to limit overfitting. The training was stopped when the validation loss did not decrease in 10 consecutive epochs.

The Figure 3.12 outlines the evolution of the training loss and the prediction MSE. The used training loss was the mean squared error and the prediction MSE was used to check over-fitting over the training by doing early stopping. The parameters chosen are detailed in Table 3.1

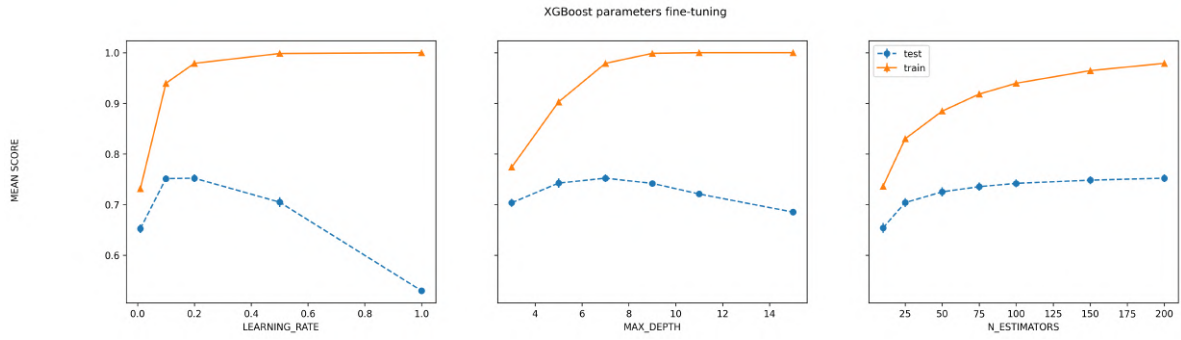


Figure 3.11: hyper parameters fine-tuning using 5-fold cross validation grid search. The mean score is the average R^2 of the 5 folds.

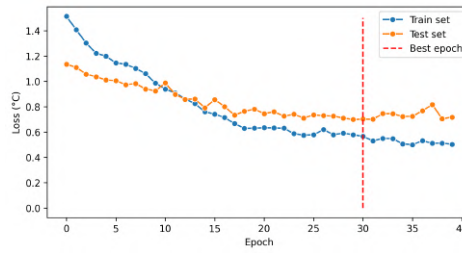


Figure 3.12: train set loss evolution and test set metrics evolution with the epochs. The Loss metric is the mean squared error

XGBoost Parameters	STFGAN Parameters
Learning rate: 0.2	Attention heads: 4
Max depth: 7	Hidden state dimension: 128
Number of trees: 200	Temporal lag steps: 12 hours
	Dropout rate: 0.1
	Number of GATConv layers: 2
	Activation function: ELU [108]
	Optimizer: AdamW [109]

Table 3.1: Model hyperparameters for XGBoost and STFGAN.

Chapter 4

Results

This chapter presents the results obtained using the methods and data described above. The objective is to evaluate the relevance of the spatio-temporal graph neural network approach to predict the UHI effect at a city scale, and to demonstrate the impact of the UHI effect on the energy consumption of buildings.

First, the results of the UHI intensity predictions across all LCDs using the STFGAN and XGBoost methods are presented, along with the models insights. Then, the results of the spatio-temporal modeling of UHI impact on buildings energy consumption will be shown.

4.1 UHI intensity predictions

4.1.1 Performance on the training set

The first step of the model's performance evaluation is to check the accuracy of the fitted model predictions on the train set (i.e. on the data that was used to train the model). This is a primary indication of the model performance, although it might not indicate how good will the model be on the test set, due to over-fitting. Table 4.1 summarizes the metrics of the different models. The best model on the train set is by far the XGBoost trained with the delta T lags. It achieves a RMSE of 0.44°C with very good consistency on all the stations (variance of the error of 0.006°C). The STFGAN and the other XGBoost model have comparable metrics. The Figures 4.1 and 4.2 illustrate the predictions accuracy. The lower performance of the STFGAN on the training set, despite its greater complexity, may suggest underfitting due to insufficient training duration.

Model	RMSE (°C)	MAE (°C)	R ²	RMSE _{min}	RMSE _{max}	RMSE _{var}
XGBoost without delta T lag	0.60	0.38	0.78	0.30	1.8	0.067
XGBoost with delta T lag	0.44	0.33	0.88	0.26	0.80	0.006
STFGAN	0.57	0.39	0.80	0.25	1.76	0.05

Table 4.1: Performance metrics for the different models on the train set.

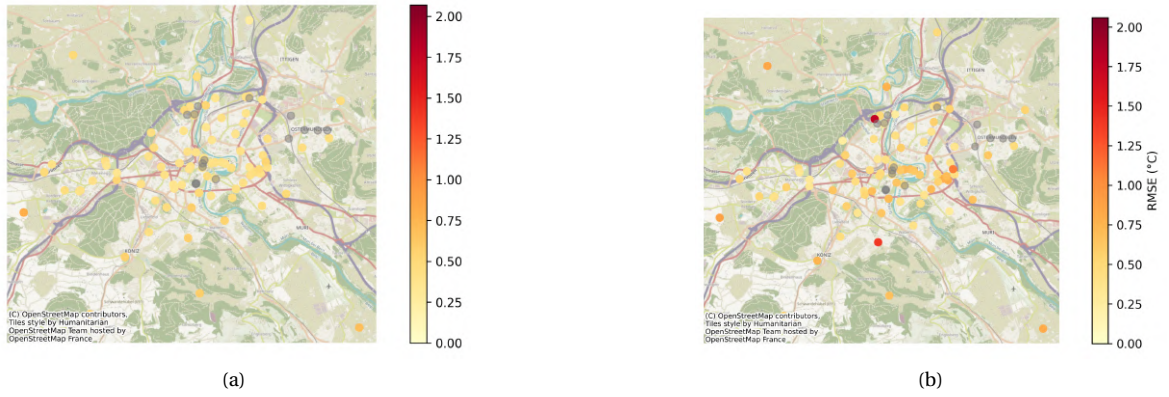


Figure 4.1: RMSE of (a) XGBoost and (b) STFGAN on the train set. Grey circles indicate the LCDs where no data is available to compute the RMSE. The color intensity represents the magnitude of the RMSE with a lower RMSE indicating a better performance in the predictions

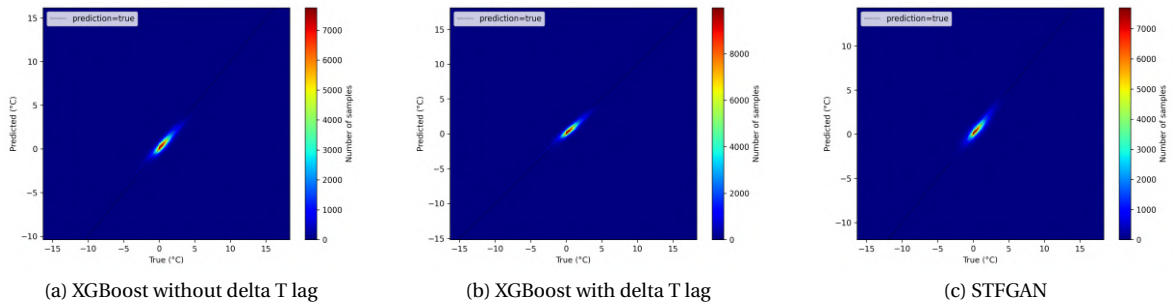


Figure 4.2: odels predictions vs true values. The colors represent the point density, and the dashed line is the identity line, where the predicted values are equal to the true values. Points closer to this line indicate better model performance, while deviations from the line suggest prediction errors.

4.1.2 Performance on the test set

A novel technique to predict UHI intensity was introduced in this report (Section 3.2), using an architecture designed to capture both spatial dependencies and temporal patterns. This approach demonstrated a substantial improvement compared to the baseline model, XGBoost. As shown in Table 4.2, the STFGAN model achieved a 12% reduction in the root mean squared error, a 18% decrease in the mean absolute error and 12% increase in the coefficient of determination obtained by comparing the true values with the predicted values over the whole summer of 2022.

As shown in Figure 4.3, the STFGAN outperforms in terms of RMSE for most of the LCDs. However, there are a few cases where its predictions are more erroneous compared to the baseline model. The baseline model demonstrates greater consistency across all stations, which is reflected in the lower $RMSE_{var}$ achieved by the XGBoost model compared to the STFGAN (Figure 4.3). This trade-off between accuracy and consistency highlights the potential limitations of the STFGAN model in specific cases. A full comparison of RMSE values across all stations is provided in the appendix (Figure 1). When compared to the metrics of the models on the training set, the STFGAN is the model with the lower overfitting, with a 14% increase in the Root Mean Squared Error (RMSE) on the test set compared to the test set. In comparison, we observe a 68% increase of the RMSE for the XGBoost with temperature delta lags. While this result indicates a lower overfitting, it might also confirm the underfitting hypothesis.

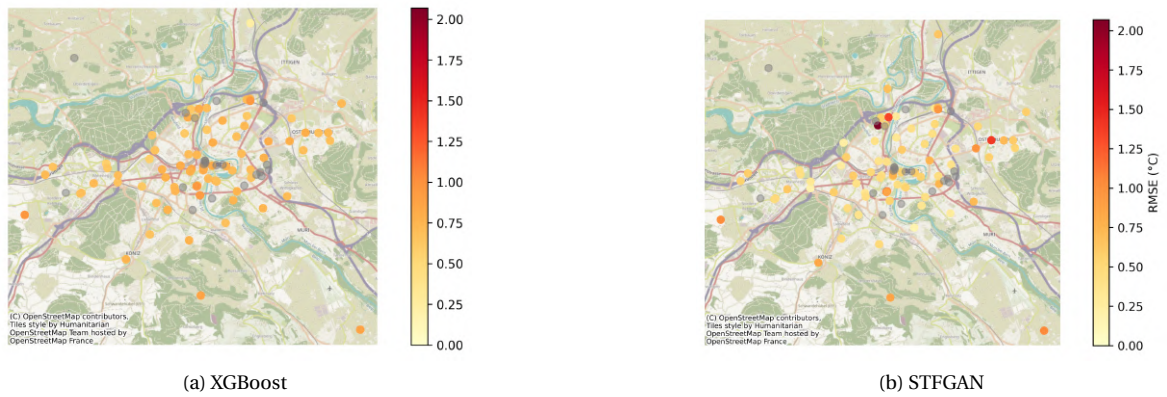


Figure 4.3: Map of the mean squared error at each LCD for the test set. Grey circles are LCD without data for 2022

Model	RMSE (°C)	MAE (°C)	R ²	MdAPE (ΔT)	MdAPE (T)	MdAPE day/night (ΔT)
XGBoost without delta T lag	0.79	0.60	0.62	53%	2.3%	73%/51%
XGBoost with delta T lag	0.74	0.55	0.67	49%	2.1%	65%/52%
STFGAN	0.65	0.45	0.75	25%	1.5%	37%/20%

Table 4.2: Performance metrics (RMSE, MAE, R²) for the different models on the test set. The day/night MdAPE is not corresponding to the

The LCDs are divided into three categories: those with a complete time series (2019–2022), those with an incomplete time series due to being installed after 2019 or removed before 2022, and those installed in

Model	RMSE _{min}	RMSE _{max}	RMSE _{var}
XGBoost without delta T lag	0.30	1.37	0.026
XGBoost with delta T lag	0.24	0.96	0.01
STFGAN	0.16	2.07	0.09

Table 4.3: RMSE details (RMSE_{min}, RMSE_{max}, RMSE_{var}) for the different models on the test set.

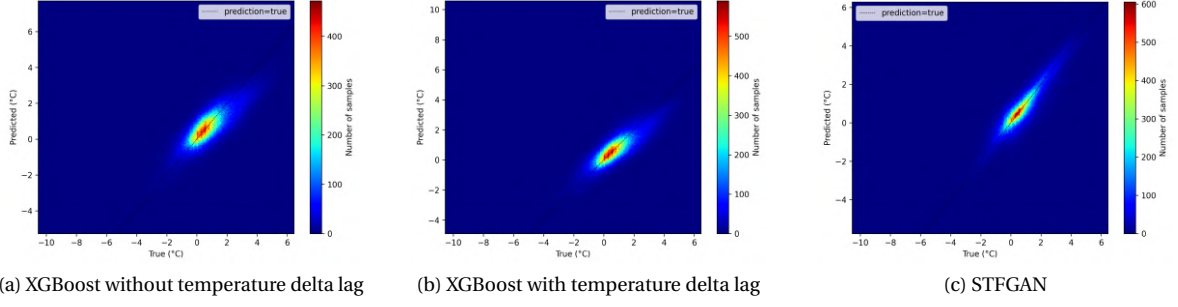


Figure 4.4: Models predictions vs true values. The colors represent the point density, and the dashed line is the identity line, where the predicted values are equal to the true values. Points closer to this line indicate better model performance, while deviations from the line suggest prediction errors.

2022, which lack training data. Figure 4.6 provides a detailed comparison of the models' predictions across these categories (A, B and C), for LCDs mapped in Figure 4.7. The selected LCDs are in two distinct locations: the first three are in the center of Bern (and more specifically along the same street: Weisenhaupplatz), and the last two in the outskirts of the city. They illustrate the model advantages and drawbacks for the different categories of LCD. The features of the different LCDs are in Table 4.4. The poor performance of STFGAN on LCD 141 (category C) could be attributed to differences in its features compared to LCD 48, resulting in the two not being connected in the graph. Consequently, predictions for LCD 141 rely on contributions from more distant LCDs, which may not share the same temperature profile as LCD 48. This contrasts with LCD 124 (also category C), which is connected to nodes with similar temperature profiles. In this case, STFGAN demonstrates its effectiveness compared to the XGBoost model, particularly in scenarios with high UHI intensity that deviate from previous patterns. Here, the unseen LCD benefits from contributions from its connected nodes, as illustrated by the peak observed at the start of 29/06. More generally, the STFGAN have better accuracy for LCDs seen during the training, but a worse accuracy on unseen LCDs (see Figure 4.5).

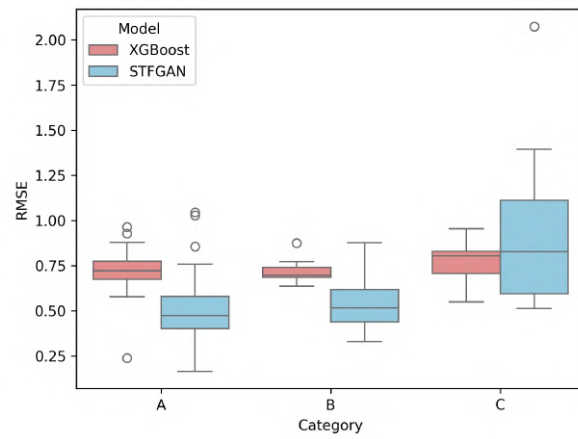


Figure 4.5: RMSE distribution on the three categories of LCD. Category A is for LCDs with a complete timeseries, B is for LCDs with an incomplete time series and C is for LCDs with no training timeseries

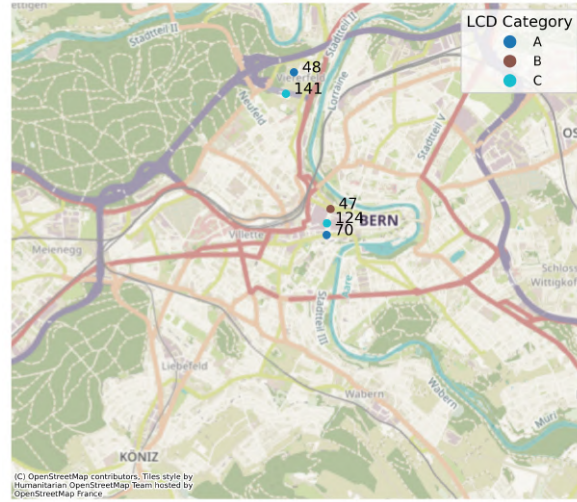


Figure 4.7: Plotted LCDs in Figure 4.6. The category indication is the type of LCD, and the number is the LCD ID

Log_Nr	124	47	70	141	48
population	16	0	3	0	0
avg_height (m)	17.9	22.7	18.3	0	4.4
elevation (m)	552	546	555	571	571
d_50_avg (m)	46.5	58.2	58.2	0	44.4
d100_avg (m)	66.9	83.5	83.0	0	94.4
d200_avg (m)	95.7	119	116	0	251
LCZ	2	5	2	12	12
Albedo	0.11	0.11	0.11	0.13	0.13
NDVI	0.04	0.07	0.08	0.48	0.73
build_count	106	50	53	0	44
dist_to_center (m)	430	413	469	1837	2026
dist_to_forest (m)	779	612	926	75	86

Table 4.4: Features of the plotted LCDs in Figure 4.7

Figure 4.8 shows an analysis of the MdAPE and MAE over time. The Figure reveals that STFGAN performs better in nighttime predictions compared to daytime. Specifically, the global MdAPE for nighttime is 20%, while for daytime it increases to 37%. This difference may be attributed to the lower UHI intensity during the day, which increases the absolute percentage error for the same absolute error, as well as a greater variability in temperature distribution across stations, making daytime predictions more challenging.

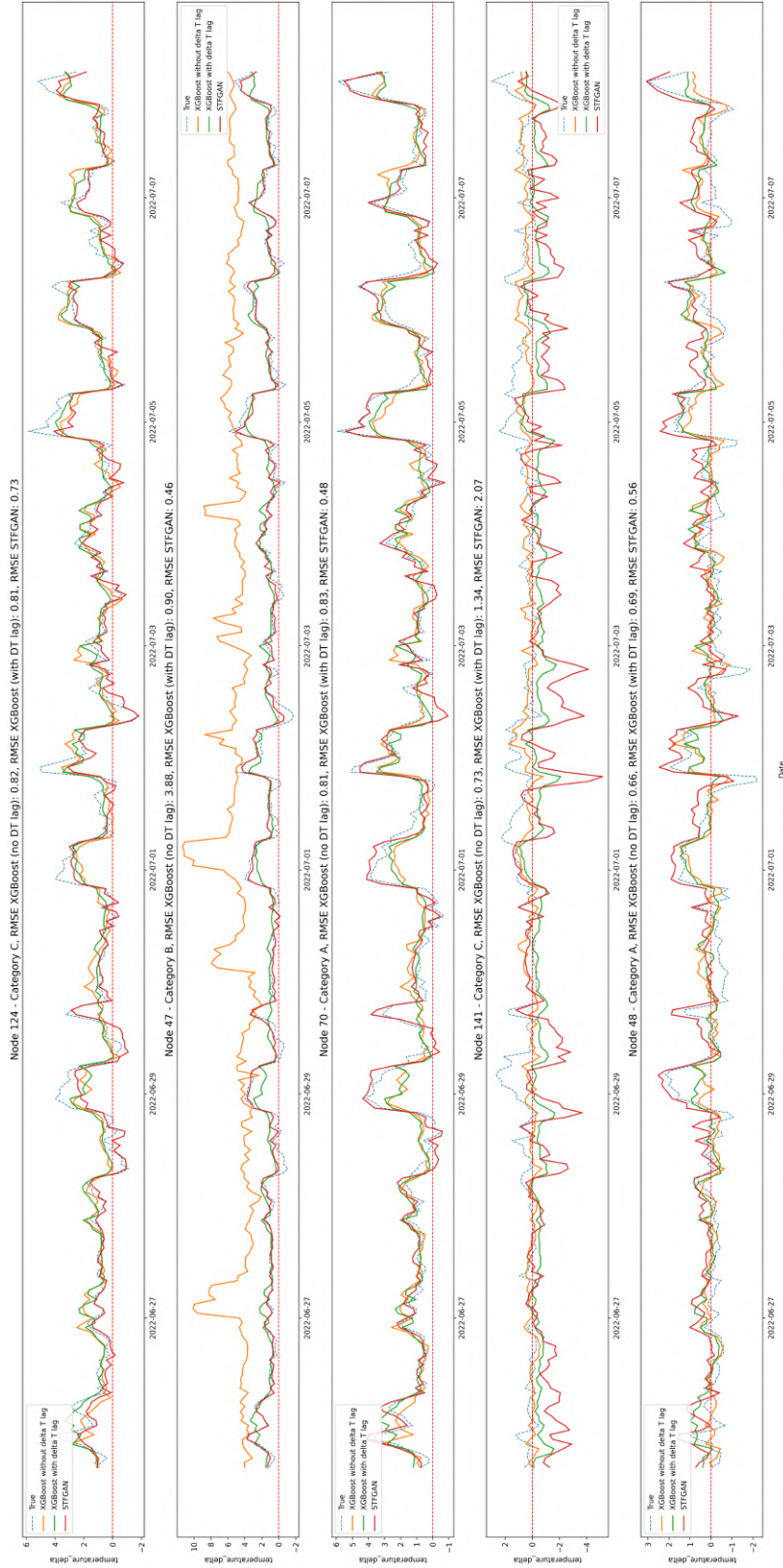


Figure 4.6: Results of each model in different configurations. The type of LCD plotted is indicated by the category in each sup-title. The first three plots represent three LCDs that were installed in Bern's center, but at different times. The last two are LCDs that are next to each other but in the exterior of Bern. See Figure 4.7. The 'True' line on plot 1 and 2 is the same due to the lack of testing data for LCD 47.

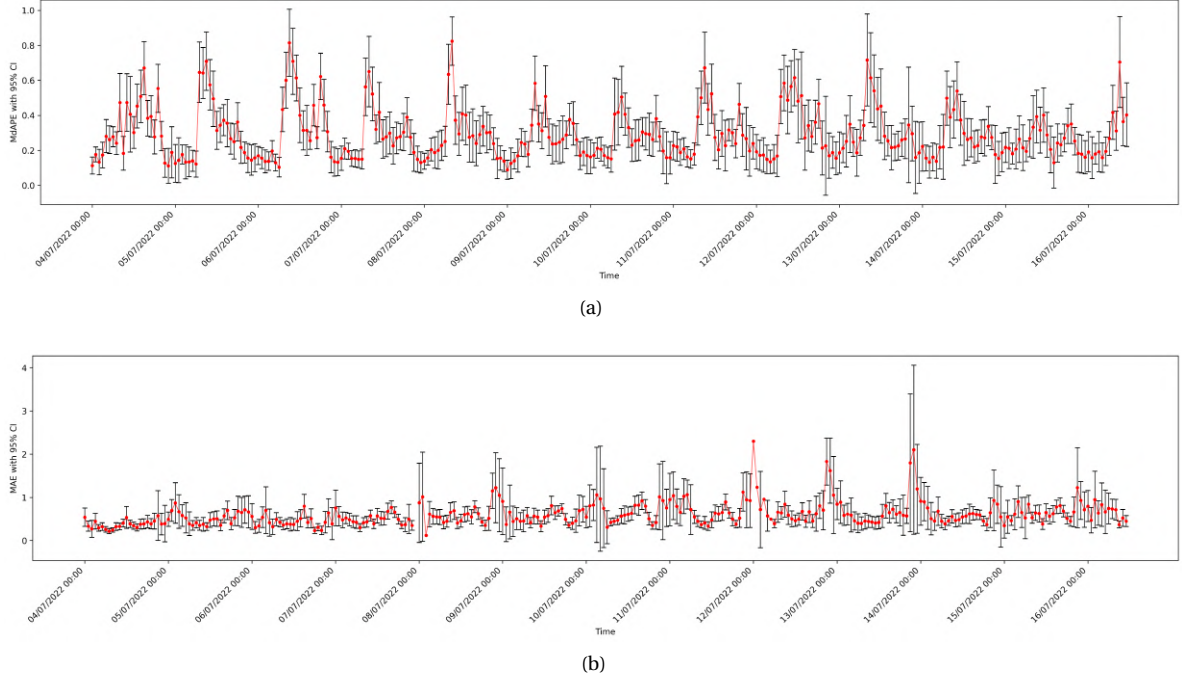


Figure 4.8: (a) MdAPE and (b) MAE over 300 hours.

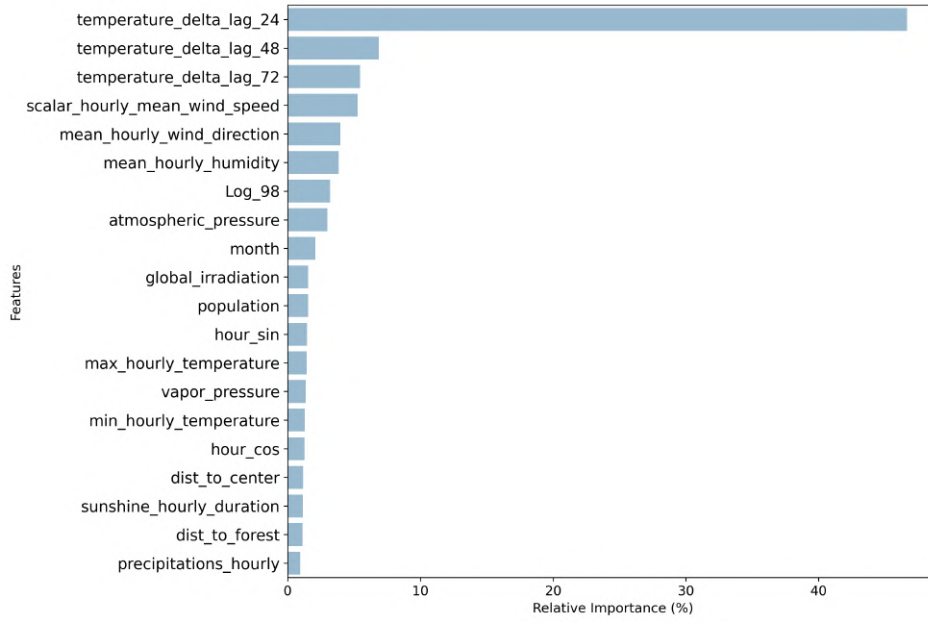
4.1.3 Models insights

Two methods are used to obtain information from the models training. For the XGBoost, the feature importance are inspected, and for the STFGAN, the attention weights are analyzed.

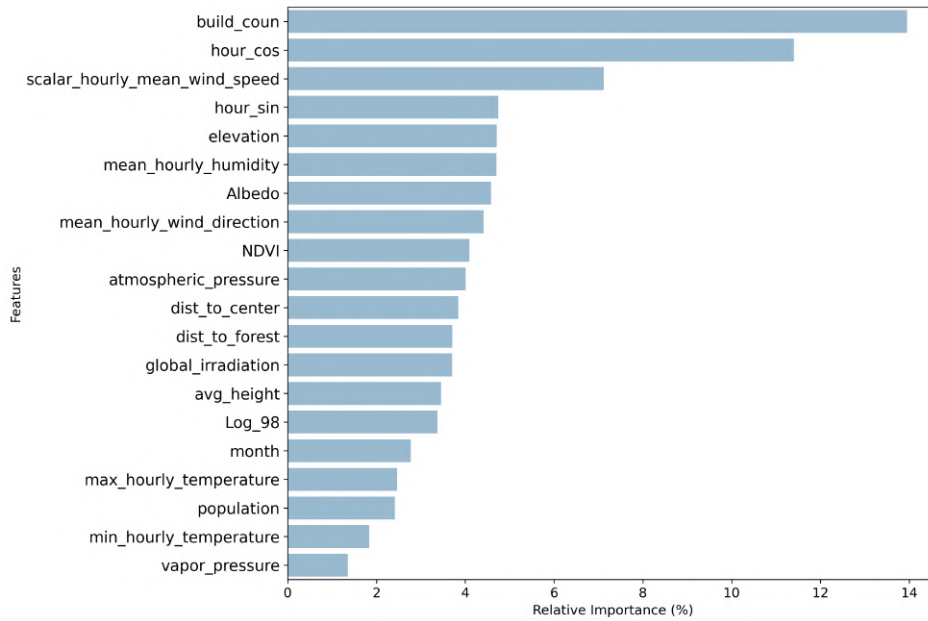
Two XGBoost models were trained: one that includes the temperature lag feature to capture temporal information, and another without the temperature lag for comparison. The feature importance analysis (Figure 4.9) reveals two key insights. First, in the model with the temperature lag, this feature emerges as the most significant predictor, dominating over all other features. Second, in the model without the temperature lag, urban morphology—specifically the building count around each station—becomes the primary predictor, followed by vegetation presence or absence, as represented by the NDVI. These results highlight the critical role of temporal information when available and underline the influence of urban and vegetation characteristics in its absence.

To gain insights into the importance of connections within the graph, an attempt was made to interpret the attention weights from the GATConv layer. These weights, which are updated at each time step during training, were averaged across all time steps to provide an overall view of edge importance. For bidirectional edges, the weights were summed. It is worth noting that the total received weight for each LCD is always normalized to 1, while the total given weight is not necessarily constrained to this sum.

Figures 4.10 and 4.11 present, respectively, the attention weight of each edge and the average given attention weight for each LCD. While the first attention layer weights could potentially reflect the significance of neighboring nodes for a given LCD, the second layer weights, more abstract, captures global and higher-order dependencies. These weights are based on the feature representations learned by the first layer.



(a)



(b)

Figure 4.9: XGBoost features importance for (a) With ΔT lag and (b) without ΔT lags. The total gain represents the feature ability to generate prediction relatively to other features

The first-layer attention weights seem to suggest that for certain nodes—particularly those located farther from the city center—distant neighbors may hold greater importance. This could hint at shared temporal and spatial patterns in UHI intensity. The second-layer weights, however, are less evenly distributed and harder to

interpret meaningfully.

Another metric examined is the average given attention, as shown in Figure 4.12, which reflects the average weight assigned to a node by its neighbors. While a few nodes appear to be more influential than others, no clear or consistent spatial pattern could be identified from this analysis.

This exploration represents a preliminary attempt to interpret the attention weights generated by the model. While some patterns have been visualized, additional work would be necessary to fully understand the implications of these weights and their relationship to feature importance.

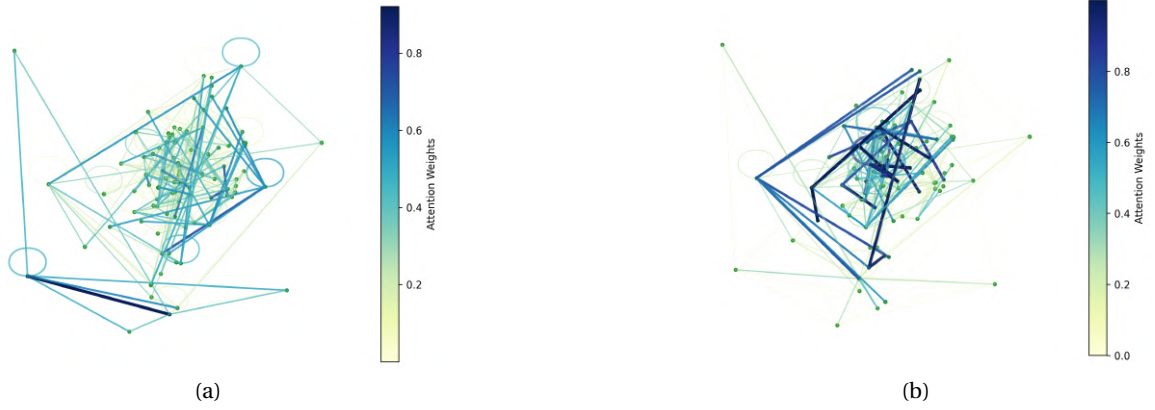


Figure 4.10: Attention weights computed by the GATConv layers. The attention weights are averaged over all the time steps. (a) Represent the first layer weights and (b) the second layer

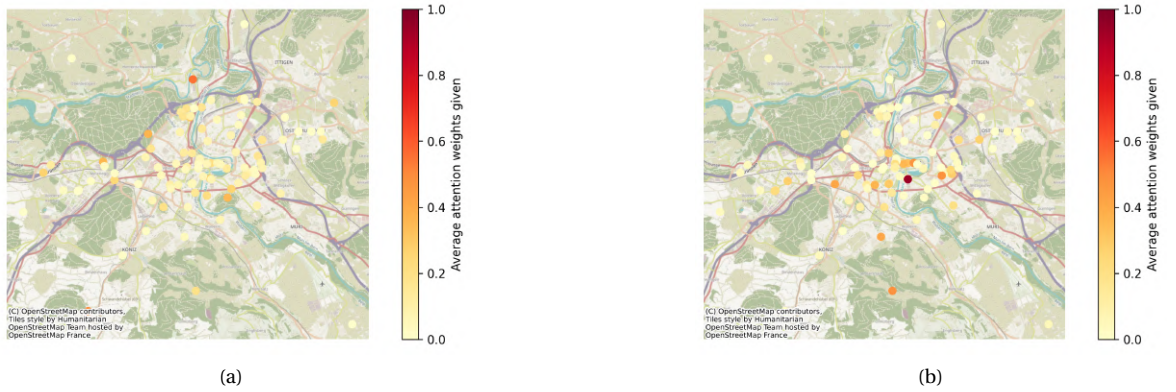


Figure 4.11: Visualization of the average given attention weights for each node. A larger average attention weight means a more important node

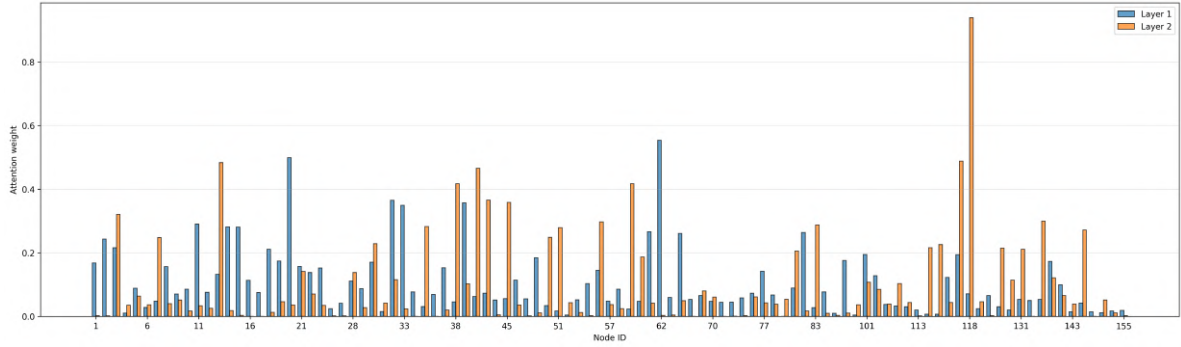


Figure 4.12: Difference in the average given attention weight between layer one and layer two.

4.2 Building energy consumption

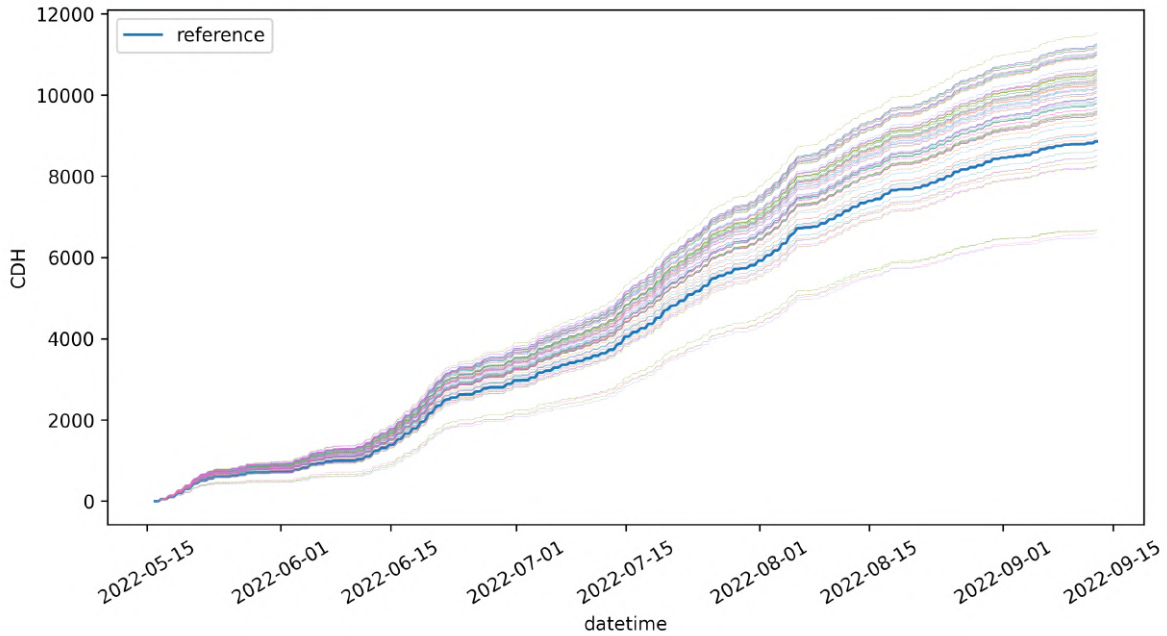


Figure 4.13: Difference in CDHs between each station and the reference station over time.

The evolution of the cumulative CDHs over the summer of 2022 shown in Figure 4.13 represents the differential in cooling load for the urban area compared to the reference station (shown with the blue line). The largest cumulated CDHs increase are for the LCDs located in Bern's city center (in the old city area), where the cumulated CDHs increase by around 30% compared to Zollikofen (see Figure 4.16a).

The next step was to generalize the UHI intensity to the entire study area to construct a global map of CDH differential over the 2022 summer. The metrics of the trained XGBoost model are described in Table 4.5. The features importance of the model are shown in Figure 4.15.

The Figure 4.16 highlights the differences in the cooling load between the reference weather station in

	RMSE	MAE	R ²
Train set	0.37	0.29	0.90
Test set	0.53	0.40	0.79

Table 4.5: Metrics of the XGBoost model used to extend the UHI predictions to the study area. The train set is composed of the UHI intensity predictions for 2022 and 80% of the stations. The test set is the remaining 20% stations

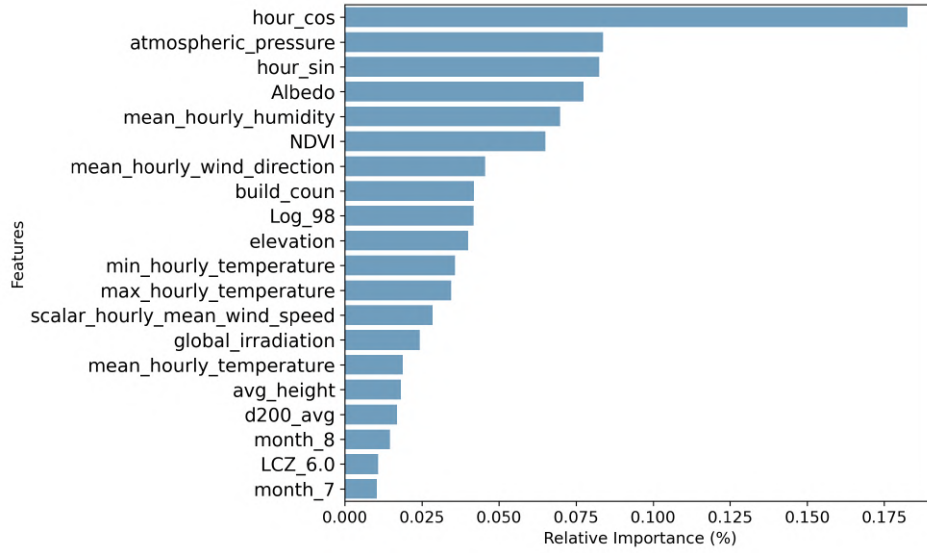


Figure 4.14: Feature importance of the XGBoost model trained on the summer 2022 predictions

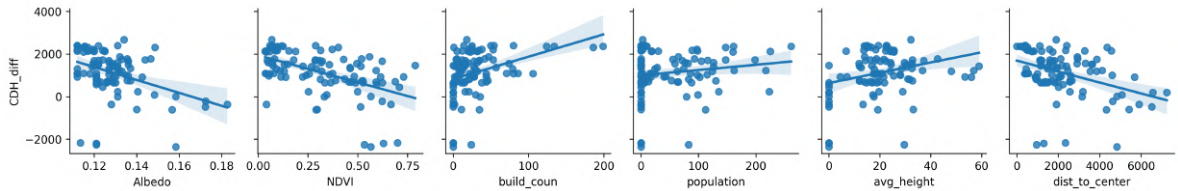


Figure 4.15: Relation between the static station features and the cumulated CDH over the 2022 year. The shaded area represents the 95% confidence interval

Zollikofen, the city of Bern, and the surroundings. There is a clear increase of the CDH due to the urban infrastructures, and the green spaces, as the Dählhölzli zoological park, located in the south of the city center, have a decreasing impact on the cooling load. The Aar river also seems to decrease the impact of the UHI effect on the cooling load. Figure 4.15 shows the relation between the stations features and the cumulated CDH. NDVI, albedo and the distance from Bern's city center reduce the cooling load while the building density increase it.

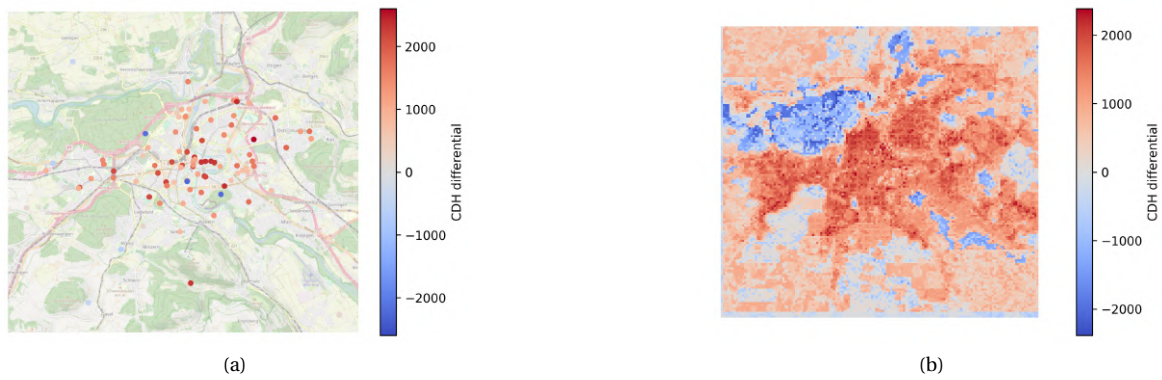


Figure 4.16: Comparison of the cumulated CDH differential between the Zollikofen weather station and (a) the stations, (b) the Bern region

Chapter 5

Discussion

This study aims to evaluate the performance of a state-of-the-art deep learning algorithm, using graph machine learning, on the prediction of the UHI effect and to use these predictions to assess the impact of the UHI effect on the cooling load across the Bern area.

The discussion will focus on the comparison of the STFGAN model compared to the literature, its limitations and its strengths, and on the UHI effect on the cooling load.

5.1 STFGAN performance

Machine learning methods to predict UHI intensity is a recent focus in the scientific literature [15]. Machine learning no longer is a supporting tool study the UHI effect, but is becoming a useful tool to predict accurately the UHI effect. Conventional methods like random forest, artificial neural network (ANN) or support vector machine [15] have been used and have proven their efficiency, specifically to determine spatial patterns and characteristics of UHI. While the temporal nature of UHI was under looked, maybe due to the lack of accurate and extensive data, it is becoming the next step in the improvement of UHI predictions via ML techniques. In recent studies, Han, Ang, Malkawi, *et al.* [35] used a RNN to generate localized weather data, with the aim of improving buildings energy modeling, Han, Chong, Lim, *et al.* [110] proposes a novel approach that combines kriging with a LSTM to account both for the spatial variations with the kriging layer and the temporal pattern (with the LSTM layer), to predict the microclimate variations at really low scale (1x1 meter and 10 minutes, for the extent of a university campus), using 14 weather stations.

Although the field of spatio-temporal UHI prediction have been explored, the use of a graph-based approach is still relatively nascent, offering significant potential for capturing the complex spatial and temporal dependencies inherent in UHI phenomena. The performance of a graph attention-based architecture, with temporal dimension incorporation, has already been showed by Yu, Shi, and Xu [49], with an attention-based graph neural network combined with a gated recurrent unit to predict air temperatures at a country scale. Similarly, Yu, Li, Huang, *et al.* [48] builds a graph neural network to predict the street-scale temperature in Chicago, with no temporal processing, and using the GraphSAGE operator instead of attention-based graph convolutional layers. The STFGAN approach integrates both methods to create a graph attention-based neural network with a temporal memory bank to predict the local temperature deltas at a city scale. The main

difference is the integration of both explicit spatial features, an explicit temporal processing via the GRU, and the direct prediction of air temperature difference between the urban and rural area, instead of the air temperature in the urban canopy layer. The characteristics of the three different methods are summed up in Table 5.1.

	GNN [48]	GATG [49]	STFGAN
Temporal Dynamics	Implicit via node features, no explicit temporal memory.	Explicit via GRU	Explicit with temporal features embedding and GRU.
Graph Layers	GraphSAGE ¹ .	GATConv ²	GATConv ²
Feature Fusion	Early, implicit fusion in node features.	Early, implicit fusion in node features	Late, explicit fusion after independent processing ³ .
Temporal Data Handling	Limited to snapshot-based processing.	Gated recurrent unit to handle long-term temporal patterns	Explicit temporal memory for long-term dynamics (models sequential dependencies).
Model Flexibility	Simpler, suitable for static or quasi-static data.	Simpler model for large regions, no spatial features needed	Better for dynamic and temporally sparse data.
Spatial features	Land cover, tree and urban height, vegetation coverage, urban coverage, NDVI, LAI, distance to lake Michigan	Latitude and longitude	Land cover, buildings height, density, and count, yearly averaged NDVI, population, yearly averaged albedo, distance to city center and distance to nearest forest
Temporal features	Weather features (soil, surface a 2m air temperature, humidity, wind speed, irradiance), time features (hour, day, month)	Weather features (air pressure, wind speed, humidity, water level and temperature) and time features (hour, day and month)	Weather features (pressure, humidity, wind speed and direction, irradiance, precipitation, air temperature, hourly sunshine duration) and time features (hour and month)

Table 5.1: Comparison between GNN [48], GATG [49] and STFGAN

¹ GraphSAGE (SAmple and aggreGatE): An inductive framework that efficiently generates node embeddings for unseen data using node feature information.[74]

² GATConv: Graph Attention Convolution Layer [76].

³ Independent processing refers to processing nodes' spatial features before fusion.

The STFGAN achieves a root mean square error (RMSE) of 0.65°C, placing it within the lower range for a hourly horizon air temperature predictions [16]. Comparable studies report an average RMSE of 1.35±0.60°C on temperature prediction. In contrast, the GNN demonstrates an RMSE of 0.93°C, while the GATG achieves 0.50°C. These results indicate that incorporating temporal processing significantly enhances the performance of graph-based models for predicting city microclimate, although the micro climate's variability is more unpredictable than a country-scale climate variability. The study by Yu, Li, Huang, *et al.* [48], which employs a similar setup involving a dense network of measuring devices at a local scale and utilizes a comparable method (Graph Neural Network), provides a basis for comparison in terms of MdAPE. Yu, Li, Huang, *et al.* [48] reports an MAE of 0.66°C for air temperature predictions. Considering the urban heat island (UHI) magnitude is estimated to range between 1°C and 3°C for cities with populations exceeding 1 million [111], the MAPE can be roughly calculated as $MAE/\Delta T$. This translates to a MAPE range of 22% to 66%. With the STFGAN

achieving a MdAPE of 37%, this demonstrates its strong performance and highlights the added advantages of incorporating temporal processing and an attention mechanism.

The comparison with the baseline model (XGBoost) shows the better performance of the STFGAN on the same dataset, with a 12% decrease in the RMSE. The good performance of the STFGAN is particularly important for medium values of UHI intensity (2 to 4°C). This is illustrated in Figure 5.1, which depicts when only the true UHI intensity exceeds (Figure 5.1a) or is less than (Figure 5.1b) the considered threshold. When only targets exceeding the threshold are considered, the XGBoost error metrics increase at a faster rate than the STFGAN metrics until the 5°C threshold. For the opposite threshold, similar behavior is observed between both models.

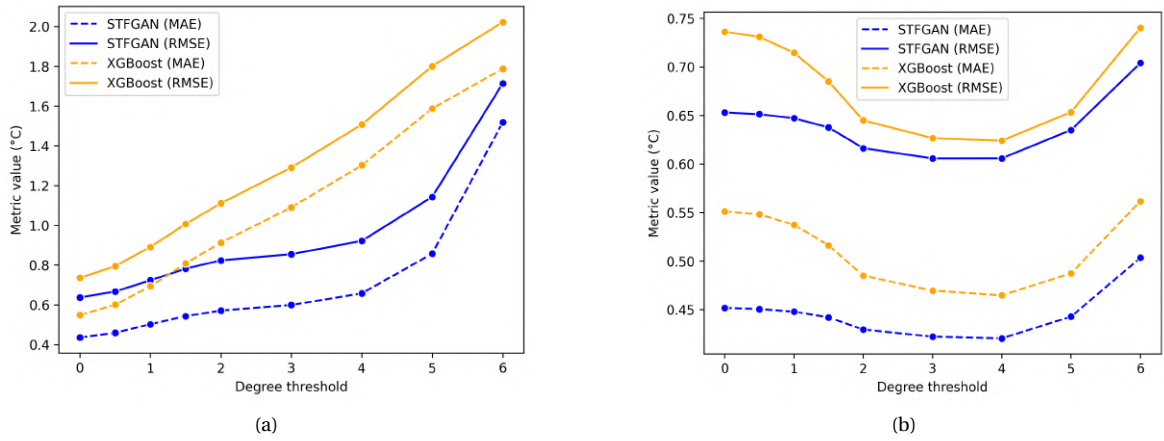


Figure 5.1: Comparison of the metrics' evolution: (a) Shows the metric value where only targets exceeding the threshold are considered. (b) Shows the metric value where only targets below the threshold are considered.

Although the overall accuracy of the STFGAN model surpasses that of the XGBoost method, one area for improvement lies in its performance on unseen data. Specifically, 16 LCDs installed in 2022 had no corresponding training data. On these 16 LCDs, the performance metrics for the two methods are as follows: STFGAN achieved an MAE of 0.63°C and an RMSE of 0.86°C, while XGBoost achieved an MAE of 0.58°C and an RMSE of 0.78°C.

To improve this issue, an interesting point of discussion is the graph construction process. The results show that the worst predicted node is around nodes with a similar temporal pattern and good predictions, but as their features differ completely, they are not neighbors in the graph. It has been decided, arbitrarily, for each node, to link the 10 nearest neighbors, to use the proximity in several spatial features (population, elevation, ...) while the absolute distance could have been used, or a mix of both methods. The decision to use feature proximity was motivated by the hypothesis that stations with similar local environment would have a similar behavior and by the existence of isolated stations. To create the graph representation, Yu, Shi, and Xu [49] create a link when the absolute distance between two nodes is below a certain threshold while Yu, Li, Huang, *et al.* [48] computes a metric that combines the spatial similarity and the distance to add a link if the metric is above a threshold.

Li, Yu, Liu, *et al.* [112] reviews the different graph construction methods for spatio-temporal data and outlines the need to account for non-spatial proximity in weather prediction tasks (specifically to account for the urban heat island phenomena). Another method could be to automatically learn the graph structure during

the training, with a complete graph structure and a learnable edge matrix.

5.2 Domain generalization

Using the STFGAN predictions for 2022, an XGBoost model is constructed to generate an hourly map of UHI intensity across the entire Bern area. The XGBoost algorithm has been previously applied to spatial interpolation tasks. For instance, Wang, Wu, and Wu [113] utilized an XGBoost model to predict PM2.5 concentrations in Shanghai based on data from 19 measuring stations, while Katayama, Yasuda, and Fuse [114] demonstrated that XGBoost achieved comparable performance to traditional spatial interpolation methods, such as kriging and inverse distance weighted regression, in a spatio-temporal traffic prediction context.

The XGBoost model enables spatial interpolation of temperature delta predictions while preserving time and weather features, which have proven to be key predictors of UHI intensity over time. Although the model metrics are promising, with an R^2 of 0.79 for stations not included in training, this performance may overestimate the interpolation accuracy for the entire area. This limitation is due to spatial heterogeneity and the under-representation of certain area-specific conditions by the available stations, for example, the over-representation of urban land cover compared to other types of land cover in the dataset.

From the hourly UHI intensity map, after adding the temperature of the reference station, the cooling degree hours are computed for each grid cell, and cumulated over the summer. The CDH differential between the reference and the grid cells represents the impact of the UHI on the cooling load.

Although this approach effectively leverages the temporal dependencies in the UHI effect, it may fall short in capturing the spatial relationships between distant and closer locations.

5.3 Impact of UHI on the cooling load

The impact of the UHI effect on the buildings energy consumption has been largely studied [10], and the increase in the cooling energy needs for buildings has a median of 18.7%. Although several methods exist, the main issue is often to obtain data on the buildings energy consumption with and without the UHI effect, and it is complicated to generalize the results to a whole city. To bypass this issue, a global measure of the cooling load, cooling degree hours, is used here. This method is used by Schatz and Kucharik [115] where an increase of 25% in the cooling degree days is found in the city of Madison, USA. According to the Koppen climate classification [79], Madison has a Dfb (Warm-summer humid continental) climate and Bern has a Cfb (Temperate oceanic) bordering a Dfb climate (using climate normals for both cities). In Bern, the cooling degree hours observe a median increase of 15%, with the city center experiencing an increase of around 30%.

The spatial pattern of UHI impact on the cooling load across the Bern region is clearly depicted by the generalized cumulated cooling degree hours for the entire study area (Figure 4.16b). Almost all urban areas are impacted, with a strong increase in the cooling load, up to a 30% compared to the reference station. Figure 4.15 outlines the relation between the features and the cooling load increase. The Normalized Difference Vegetation Index (NDVI), Albedo and distance to Bern's center decrease the cumulated CDH differential, while

the buildings density, characterized by the the building count and the average buildings height, increases the cooling differential. The population has little to no impact on the cooling load increase, but this might be due to the attribution of the population feature to each station. Each station was attributed the population of the grid cell in which it was located in the STATPOP dataset [83], and this resulted in a lot of stations with a null population.

These results align with previous studies on the impact and mitigation of the UHI effect. Promoting green cities by expanding green spaces to enhance NDVI, along with increasing albedo through the use of reflective materials, has been shown to mitigate the UHI effect [116], [117], a finding that is also confirmed for Bern.

The cooling load differential could be crucial in the modeling of the energy consumption of Bern. If the energy consumption simulations are made with the temperature measured at the Zollikofen weather station, without accounting for the UHI effect, major discrepancies between the simulated energy consumption and the observed consumption could be seen [118], and this discrepancy is set to increase with higher heatwaves frequency, and higher global temperatures.

5.4 Limitations and further research

The original study that installed the LCDs network throughout Bern's region conducted an evaluation of the performance of the temperature measures, revealing a mainly positive bias in the temperature measurements during the day, specifically in the urban and sub-urban areas due to the reduced wind velocities caused by the presence of trees and buildings. This positive bias could artificially increase the cooling degree hours evaluation and impact the temperature data modeling by the STFGAN and the XXGBoost models. Due to limited computational resources, the STFGAN could not be fine-tuned and trained extensively, which might cause underfitting and reduce its performance. Although extrapolation of the cumulative CDHs to the entire study area provides valuable information on the impact of the UHI effect on the cooling load, it is difficult to assess its accuracy. This extrapolation is based on a XGBoost model constructed from a limited set of localizations which might not be representative of the climate in the whole area. The lack of data on precise energy consumption do not allows to validate the CDH method and to go further by quantifying the real energy consumption differential due to the UHI.

Further research could focus on improving and extending the results obtained. Firstly, the study of different graph construction processes (such as a mix of absolute distance and feature similarity, or a graph where all nodes are linked and the edges' weight is learned during the training) and the analysis of their impact on prediction accuracy may help optimize the model structure. In addition, a promising direction is to use accurate local predictions to generate unseen locations predictions, taking into account the most important features and relationships identified in the attention matrices. Refining STFGAN's hyper parameters and comparing its performance with other graph-based approaches will further validate its effectiveness. Moreover, applying the same methodology to winter data may enable us to fully understand seasonal variations, by balancing the observed increase in air-conditioning hours in summer with the decrease in heating hours in winter. To extend its usefulness, the model could be adapted to predict over a longer time horizon (e.g., 4 hours), making it a potential early-warning tool for extreme heat events. Another promising approach would be to incorporate forecasted future weather data alongside the measured weather data to enhance the prediction accuracy.

The application of this framework to other cities would further validate its approach and demonstrate

its broader utility. In addition, the spatial and temporal interpretability of the STFGAN process could be enhanced through advanced methods. For example, attribution [119] can highlight important edges and nodes in graph neural networks, while the GAtt edge attribution method [120] is specifically designed to improve the explainability of graph attention networks.

Future research on the impact of UHI on building energy consumption could focus on developing a model that connects building characteristics, location, temperature, and energy consumption under the influence of the UHI effect, provided the necessary data is available. Additionally, analyzing the effects of various climate change projections could provide valuable insights and support decision-makers in mitigating these impacts.

Chapter 6

Conclusion

This thesis aims at developing a new methodology to predict the intensity of the UHI effect at a city scale, and more specifically in the Bern urban area. The new methodology is a combination of a gated recurrent unit and a graph attention network to account both for the temporal processes and the spatial dependencies. The developed method, called the spatio-temporal fusion graph attention network (STFGAN) leverages the so-called attention mechanism which automatically attributes weights to edges and features and combines the processed spatial features with a representation of the temporal features after their processing through a temporal memory bank. This method proves to be in the higher range of performance in hourly temperature predictions, with a RMSE of 0.63°C, and beats an XGBoost model trained on the same data, improving the performance by 15% and showing the relevance of using a graph-based approach. This algorithm can be used to generate predictions in locations with no sensor, predict the UHI intensity at H+1 or, via the weights generated by the attention mechanism, find the most important features and edges in the graph.

Using the UHI intensity predictions from the STFGAN model, another XGBoost algorithm is built and trained to extrapolate the UHI intensity predictions to the entire study area. Combined with the reference temperature, a hourly map of the cooling degree hours can be constructed, and the impact of the UHI analyzed.

The cooling degree hours modeling highlights the cooling load differential between the rural surroundings of Bern and its urban areas. Due to the UHI effect, the cooling load notably increased during the summer of 2022, particularly in central Bern. Green spaces and higher albedo were found to mitigate this increase, providing valuable insights for decision-makers.

Further research could improve the graph-based approach by fine-tuning the several hyper parameters, and test different methods of graph construction. Moreover, this methodology should be applied to other locations to validate its accuracy

Bibliography

- [1] U. N. D. of Economic and S. Affairs, *World Urbanization Prospects: The 2018 Revision*. United Nations, 2019. [Online]. Available: <https://www.un-ilibrary.org/content/books/9789210043144>.
- [2] N. B. Grimm, S. H. Faeth, N. E. Golubiewski, C. L. Redman, J. Wu, X. Bai, and J. M. Briggs, “Global change and the ecology of cities”, *Science*, vol. 319, no. 5864, pp. 756–760, 2008. DOI: 10.1126/science.1150195. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-38949138832&doi=10.1126%2fscience.1150195&partnerID=40&md5=a96e242b2c46ea6f80943559aa3f376b>.
- [3] W. Qing, “Urbanization and global health: The role of air pollution”, *Iranian journal of public health*, vol. 47, no. 11, p. 1644, 2018.
- [4] B. Giles-Corti, A. Vernez-Moudon, R. Reis, G. Turrell, A. L. Dannenberg, H. Badland, S. Foster, M. Lowe, J. F. Sallis, M. Stevenson, *et al.*, “City planning and population health: A global challenge”, *The lancet*, vol. 388, no. 10062, pp. 2912–2924, 2016.
- [5] Core Writing Team, H. Lee and J. Romero (eds.), *Climate Change 2023: Synthesis Report. Contribution of Working Groups I, II and III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Geneva, Switzerland: IPCC, 2023, pp. 35–115. DOI: 10.59327/IPCC/AR6-9789291691647.
- [6] Z. Qiao, N. Wang, J. Chen, X. Xu, L. Liu, and D. Han, “Understanding the differences in the contribution and impact of urbanization on urban warming during heatwave and non-heatwave periods in china”, *Journal of Cleaner Production*, vol. 474, p. 143 626, 2024.
- [7] C. Heaviside, H. Macintyre, and S. Vardoulakis, “The urban heat island: Implications for health in a changing environment”, *Current environmental health reports*, vol. 4, pp. 296–305, 2017.
- [8] S. E. Cleland, W. Steinhardt, L. M. Neas, J. J. West, and A. G. Rappold, “Urban heat island impacts on heat-related cardiovascular morbidity: A time series analysis of older adults in us metropolitan areas”, *Environment international*, vol. 178, p. 108 005, 2023.
- [9] W. T. K. Huang, P. Masselot, E. Bou-Zeid, S. Fatichi, A. Paschalis, T. Sun, A. Gasparrini, and G. Manoli, “Economic valuation of temperature-related mortality attributed to urban heat islands in european cities”, *Nature communications*, vol. 14, no. 1, p. 7438, 2023.
- [10] X. Li, Y. Zhou, S. Yu, G. Jia, H. Li, and W. Li, “Urban heat island impacts on building energy consumption: A review of approaches and findings”, *Energy*, vol. 174, pp. 407–419, 2019.
- [11] A. Piracha and M. T. Chaudhary, “Urban air pollution, urban heat island and human health: A review of the literature”, *Sustainability*, vol. 14, no. 15, p. 9234, 2022.
- [12] A. M. M. Irfeey, H.-W. Chau, M. M. F. Sumaiya, C. Y. Wai, N. Muttill, and E. Jamei, “Sustainable mitigation strategies for urban heat island effects in urban areas”, *Sustainability*, vol. 15, no. 14, p. 10 767, 2023.

- [13] H. Akbari, C. Cartalis, D. Kolokotsa, A. Muscio, A. L. Pisello, F. Rossi, M. Santamouris, A. Synnefa, N. H. Wong, and M. Zinzi, "Local climate change and urban heat island mitigation techniques—the state of the art", *Journal of Civil Engineering and Management*, vol. 22, no. 1, pp. 1–16, 2016.
- [14] H. Akbari and D. Kolokotsa, "Three decades of urban heat islands and mitigation technologies research", *Energy and buildings*, vol. 133, pp. 834–842, 2016.
- [15] S. Ghorbany, M. Hu, S. Yao, and C. Wang, "Towards a sustainable urban future: A comprehensive review of urban heat island research technologies and machine learning approaches", *Sustainability*, vol. 16, no. 11, p. 4609, 2024.
- [16] H. Wang, J. Yang, G. Chen, C. Ren, and J. Zhang, "Machine learning applications on air temperature prediction in the urban canopy layer: A critical review of 2011–2022", *Urban Climate*, vol. 49, p. 101 499, 2023.
- [17] S. Cheval, V. Amihăesei, Z. Chitu, A. Dumitrescu, V. Falcescu, A. Irasoc, D. Micu, E. Mihulet, I. Ontel, M. G. Paraschiv, *et al.*, "A systematic review of urban heat island and heat waves research (1991–2022)", *Climate Risk Management*, p. 100 603, 2024.
- [18] I. D. Stewart, "Why should urban heat island researchers study history?", *Urban Climate*, vol. 30, p. 100 484, 2019.
- [19] L. Howard, *The climate of London: deduced from meteorological observations made in the metropolis and at various places around it*. Harvey, Darton; J., and A. Arch; Longman; Hatchard; S. Highley; R. Hunter, 1833, vol. 3.
- [20] J. v. Hann, "Über den temperaturunterschied zwischen stadt und land [on the temperature difference between town and country]", *Österreichischen Gesellschaft für Meteorologie, Zeitschrift*, vol. 20, pp. 457–462, 1885.
- [21] W. Schmidt, "Die verteilung der minimum temperaturen in der frostnacht des 12 mai 1927 im gemeindegebiet von wien", *Fortschritte der Landwirtschaft*, vol. 2, pp. 681–686, 1927.
- [22] A. Peppler, "Das auto als hilfsmittel der meteorologischen forschung", *Das Wetter*, vol. 46, pp. 305–308, 1929.
- [23] Å. Sundborg, "Climatological studies in uppsala: With special regard to the temperature conditions in the urban area", (*No Title*), 1951.
- [24] T. R. Oke, "The energetic basis of the urban heat island", *Quarterly journal of the royal meteorological society*, vol. 108, no. 455, pp. 1–24, 1982.
- [25] D. Zhou, J. Xiao, S. Bonafoni, C. Berger, K. Deilami, Y. Zhou, S. Frolking, R. Yao, Z. Qiao, and J. A. Sobrino, "Satellite remote sensing of surface urban heat islands: Progress, challenges, and perspectives", *Remote Sensing*, vol. 11, no. 1, p. 48, 2018.
- [26] S. W. Kim and R. D. Brown, "Urban heat island (uhi) intensity and magnitude estimations: A systematic literature review", *Science of the Total Environment*, vol. 779, p. 146 389, 2021.
- [27] M. Kolokotroni and R. Giridharan, "Urban heat island intensity in london: An investigation of the impact of physical characteristics on changes in outdoor air temperature during summer", *Solar energy*, vol. 82, no. 11, pp. 986–998, 2008.
- [28] Y.-H. Kim and J.-J. Baik, "Daily maximum urban heat island intensity in large cities of korea", *Theoretical and Applied Climatology*, vol. 79, pp. 151–164, 2004.

- [29] F.-Y. Lin, K.-T. Huang, T.-P. Lin, and R.-L. Hwang, "Generating hourly local weather data with high spatially resolution and the applications in bioclimatic performance", *Science of the Total Environment*, vol. 653, pp. 1262–1271, 2019.
- [30] P. A. Mirzaei, F. Haghighat, A. A. Nakhaie, A. Yagouti, M. Giguère, R. Keusseyan, and A. Coman, "Indoor thermal condition in urban heat island–development of a predictive tool", *Building and Environment*, vol. 57, pp. 7–17, 2012.
- [31] Y. Casali, N. Y. Aydin, and T. Comes, "Machine learning for spatial analyses in urban areas: A scoping review", *Sustainable cities and society*, vol. 85, p. 104 050, 2022.
- [32] S. Zhao, Z. Chen, Z. Xiong, Y. Shi, S. Saha, and X. X. Zhu, "Beyond grid data: Exploring graph neural networks for earth observation", *IEEE Geoscience and Remote Sensing Magazine*, 2024.
- [33] C. Malings, M. Pozzi, K. Klima, M. Bergés, E. Bou-Zeid, and P. Ramamurthy, "Surface heat assessment for developed environments: Probabilistic urban temperature modeling", *Computers, Environment and Urban Systems*, vol. 66, pp. 53–64, 2017.
- [34] A. Hardin, Y. Liu, G. Cao, and J. Vanos, "Urban heat island intensity and spatial variability by synoptic weather type in the northeast us", *Urban climate*, vol. 24, pp. 747–762, 2018.
- [35] J. M. Han, Y. Q. Ang, A. Malkawi, and H. W. Samuelson, "Using recurrent neural networks for localized weather prediction with combined use of public airport data and on-site measurements", *Building and Environment*, vol. 192, p. 107 601, 2021.
- [36] M. Shafi, A. Jain, and M. Zaman, "Applying machine learning algorithms on urban heat island (uhi) dataset", in *International Conference on Innovative Computing and Communications: Proceedings of ICICC 2022, Volume 3*, Springer, 2022, pp. 725–732.
- [37] M. Alam, "Towards sustainable and livable cities: Leveraging remote sensing", *Machine Learning, and Geo-Information Modelling to Explore and Predict Thermal Field Variance in Response to Urban Growth [J]. Sustainability*, vol. 15, 2023.
- [38] P. Rao, P. Tassinari, and D. Torreggiani, "Exploring the land-use urban heat island nexus under climate change conditions using machine learning approach: A spatio-temporal analysis of remotely sensed data", *Heliyon*, vol. 9, no. 8, 2023.
- [39] M. Burger, M. Gubler, and S. Brönnimann, "Modeling the intra-urban nocturnal summertime air temperature fields at a daily basis in a city with complex topography", *PLoS climate*, vol. 1, no. 12, e0000089, 2022.
- [40] Y. Ming, Y. Liu, Y. Li, and Y. Song, "Unraveling nonlinear and spatial non-stationary effects of urban form on surface urban heat islands using explainable spatial machine learning", *Computers, Environment and Urban Systems*, vol. 114, p. 102 200, 2024.
- [41] Q. Liu, J. Wang, and B. Bai, "Unveiling nonlinear effects of built environment attributes on urban heat resilience using interpretable machine learning", *Urban Climate*, vol. 56, p. 102 046, 2024.
- [42] G. Tanoori, A. Soltani, and A. Modiri, "Machine learning for urban heat island (uhi) analysis: Predicting land surface temperature (lst) in urban environments", *Urban Climate*, vol. 55, p. 101 962, 2024.
- [43] F. Li, T. Yigitcanlar, M. Nepal, K. N. Thanh, and F. Dur, "A novel urban heat vulnerability analysis: Integrating machine learning and remote sensing for enhanced insights", *Remote Sensing*, vol. 16, no. 16, p. 3032, 2024.
- [44] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond euclidean data", *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

- [45] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, “Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4883–4894, 2019.
- [46] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, “Connecting the dots: Multivariate time series forecasting with graph neural networks”, in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 753–763.
- [47] X. Geng, X. He, L. Xu, and J. Yu, “Graph correlated attention recurrent neural network for multivariate time series forecasting”, *Information Sciences*, vol. 606, pp. 126–142, 2022.
- [48] Y. Yu, P. Li, D. Huang, and A. Sharma, “Street-level temperature estimation using graph neural networks: Performance, feature embedding and interpretability”, *Urban Climate*, vol. 56, p. 102 003, 2024.
- [49] X. Yu, S. Shi, and L. Xu, “A spatial–temporal graph attention network approach for air temperature forecasting”, *Applied Soft Computing*, vol. 113, p. 107 888, 2021.
- [50] Swiss Federal Office of Energy (SFOE), *Buildings - energy efficiency*, Accessed: 2024-11-27, 2024. [Online]. Available: <https://www.bfe.admin.ch/bfe/en/home/efficiency/buildings.html>.
- [51] M. González-Torres, L. Pérez-Lombard, J. F. Coronel, I. R. Maestre, and D. Yan, “A review on buildings energy information: Trends, end-uses, fuels and drivers”, *Energy Reports*, vol. 8, pp. 626–637, 2022.
- [52] Y. Toparlar, B. Blocken, B. Maiheu, and G. Van Heijst, “Impact of urban microclimate on summertime building cooling demand: A parametric analysis for antwerp, belgium”, *Applied Energy*, vol. 228, pp. 852–872, 2018.
- [53] H. Radhi and S. Sharples, “Quantifying the domestic electricity consumption for air-conditioning due to urban heat islands in hot arid regions”, *Applied energy*, vol. 112, pp. 371–380, 2013.
- [54] D. J. Sailor, “Relating residential and commercial sector electricity loads to climate—evaluating state level sensitivities and vulnerabilities”, *Energy*, vol. 26, no. 7, pp. 645–657, 2001.
- [55] G. B. França, V. A. d. Almeida, A. J. d. Lucena, L. d. Faria Peres, H. F. d. Campos Velho, M. V. d. Almeida, G. G. Pimentel, K. d. N. Cardozo, L. B. C. Belém, V. F. V. V. de Miranda, *et al.*, “Urban heat island and electrical load estimation using machine learning in metropolitan area of rio de janeiro”, *Theoretical and Applied Climatology*, pp. 1–15, 2024.
- [56] M. Santamouris, C. Cartalis, A. Synnefa, and D. Kolokotsa, “On the impact of urban heat island and global warming on the power demand and electricity consumption of buildings—a review”, *Energy and buildings*, vol. 98, pp. 119–124, 2015.
- [57] Y. Zhou, Z. Zhuang, F. Yang, Y. Yu, and X. Xie, “Urban morphology on heat island and building energy consumption”, *Procedia Engineering*, vol. 205, pp. 2401–2406, 2017.
- [58] C. Guattari, L. Evangelisti, and C. A. Balaras, “On the assessment of urban heat island phenomenon and its effects on building energy performance: A case study of rome (italy)”, *Energy and Buildings*, vol. 158, pp. 605–615, 2018.
- [59] M. Burger, M. Gubler, and S. Brönnimann, “High-resolution dataset of nocturnal air temperatures in bern, switzerland (2007–2022)”, *Geoscience Data Journal*, 2024.
- [60] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system”, in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.
- [61] H. Deng, Y. Zhou, L. Wang, and C. Zhang, “Ensemble learning for the early prediction of neonatal jaundice with genetic features”, *BMC medical informatics and decision making*, vol. 21, pp. 1–11, 2021.

- [62] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors)", *The annals of statistics*, vol. 28, no. 2, pp. 337–407, 2000.
- [63] I. D. Mienye, T. G. Swart, and G. Obaido, "Recurrent neural networks: A comprehensive review of architectures, variants, and applications", *Information*, vol. 15, no. 9, p. 517, 2024.
- [64] H. Jiang, C. Zhang, and M. Wu, "Pedestrian detection based on multi-scale fusion features", in *2018 International Conference on Network Infrastructure and Digital Content (IC-NIDC)*, IEEE, 2018, pp. 329–333.
- [65] Y. Wang, Q. Ren, and J. Li, "Spatial-temporal multi-feature fusion network for long short-term traffic prediction", *Expert Systems with Applications*, vol. 224, p. 119 959, 2023.
- [66] S. Hou, W. Li, T. Liu, S. Zhou, J. Guan, R. Qin, and Z. Wang, "Must: A multi-source spatio-temporal data fusion model for short-term sea surface temperature prediction", *Ocean engineering*, vol. 259, p. 111 932, 2022.
- [67] J. Liu, A. Shahroudy, D. Xu, A. C. Kot, and G. Wang, "Skeleton-based action recognition using spatio-temporal lstm network with trust gates", *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 3007–3021, 2017.
- [68] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications", *AI open*, vol. 1, pp. 57–81, 2020.
- [69] U. A. Bhatti, H. Tang, G. Wu, S. Marjan, and A. Hussain, "Deep learning with graph convolutional networks: An overview and latest applications in computational intelligence", *International Journal of Intelligent Systems*, vol. 2023, no. 1, p. 8 342 104, 2023.
- [70] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs", *arXiv preprint arXiv:1312.6203*, 2013.
- [71] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering", *Advances in neural information processing systems*, vol. 29, 2016.
- [72] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks", *arXiv preprint arXiv:1609.02907*, 2016.
- [73] D. Grattarola, D. Zambon, F. M. Bianchi, and C. Alippi, "Understanding pooling in graph neural networks", *IEEE transactions on neural networks and learning systems*, vol. 35, no. 2, pp. 2708–2718, 2022.
- [74] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs", *Advances in neural information processing systems*, vol. 30, 2017.
- [75] D. Bahdanau, "Neural machine translation by jointly learning to align and translate", *arXiv preprint arXiv:1409.0473*, 2014.
- [76] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, *et al.*, "Graph attention networks", *stat*, vol. 1050, no. 20, pp. 10–48 550, 2017.
- [77] *Actualisation 2020 des niveaux géographiques non-institutionnels*, FR. Neuchâtel: Office fédéral de la statistique (BFS), Mar. 2024, p. 32, ISBN: 978-3-303-21058-1. [Online]. Available: <https://dam-api.bfs.admin.ch/hub/api/dam/assets/30665779/master>.
- [78] M. Demuzere, J. Kittner, A. Martilli, G. Mills, C. Moede, I. D. Stewart, J. van Vliet, and B. Bechtel, "A global map of local climate zones to support earth system modelling and urban-scale environmental science", *Earth System Science Data*, vol. 14, no. 8, pp. 3835–3873, 2022.

- [79] H. E. Beck, N. E. Zimmermann, T. R. McVicar, N. Vergopolan, A. Berg, and E. F. Wood, “Present and future köppen-geiger climate classification maps at 1-km resolution”, *Scientific data*, vol. 5, no. 1, pp. 1–12, 2018.
- [80] N. C. for Climate Services (NCCS), *Changements climatiques dans le canton de berne: Ce que l'on sait et ce qui est attendu dans le futur*, version 1.0, 15 pages, 2021. [Online]. Available: <https://www.nccs.admin.ch/nccs/en/home/regions/kantone/bern.html>.
- [81] M. Gubler, A. Christen, J. Remund, and S. Brönnimann, “Evaluation and application of a low-cost measurement network to study intra-urban temperature differences during summer 2018 in bern, switzerland”, *Urban climate*, vol. 37, p. 100817, 2021.
- [82] Federal Office of Topography swisstopo, *swiss3Dbuildings 2.0*, <https://www.swisstopo.admin.ch>, Free geodata and geoservices of swisstopo. Usage requires attribution: ©swisstopo, 2020.
- [83] B. GEOSTAT, “Statistique de la population et des ménages (statpop)”, 2010, Accès: 2024-09-23. [Online]. Available: <https://www.bfs.admin.ch/bfs/de/home/dienstleistungen/geostat.html>.
- [84] NASA JPL, *NASA Shuttle Radar Topography Mission Global 1 arc second*, Distributed by NASA EOSDIS Land Processes Distributed Active Archive Center, Accessed 2024-12-19, 2013. [Online]. Available: <https://doi.org/10.5067/MEASURES/SRTM/SRTMGL1.003>.
- [85] Office fédéral de topographie swisstopo, *Swisstlm3d*, 2024. [Online]. Available: <https://www.swisstopo.admin.ch/fr/modele-du-territoire-swisstlm3d>.
- [86] E. S. A. (ESA), *Copernicus sentinel-2 msi: Multispectral instrument, level-2a*, Accessed: 2024-12-19, 2023. [Online]. Available: <https://scihub.copernicus.eu>.
- [87] C. Schaaf and Z. Wang, *Mcd43a3 modis/terra+aqua brdf/albedo daily l3 global - 500m v006 [data set]*, Accessed 2025-01-29, 2015. [Online]. Available: <https://doi.org/10.5067/MODIS/MCD43A3.006>.
- [88] Météo Suisse, 2025. [Online]. Available: <https://www.meteosuisse.admin.ch/>.
- [89] *Pytorch Cluster*, https://github.com/rusty1s/pytorch_cluster, Accessed: 2025-01-30.
- [90] F. Salata, S. Falasca, V. Ciancio, G. Curci, S. Grignaffini, and P. de Wilde, “Estimating building cooling energy demand through the cooling degree hours in a changing climate: A modeling study”, *Sustainable Cities and Society*, vol. 76, p. 103518, 2022.
- [91] R. G. Quayle and H. F. Diaz, “Heating degree day data applied to residential heating energy consumption”, *Journal of Applied Meteorology and Climatology*, vol. 19, no. 3, pp. 241–246, 1980.
- [92] R. Castaño-Rosa, R. Barrella, C. Sánchez-Guevara, R. Barbosa, I. Kyprianou, E. Paschalidou, N. S. Thomaidis, D. Dokupilova, J. P. Gouveia, J. Kádár, *et al.*, “Cooling degree models and future energy demand in the residential sector. a seven-country case study”, *Sustainability*, vol. 13, no. 5, p. 2987, 2021.
- [93] O. Büyükalaca, H. Bulut, and T. Yılmaz, “Analysis of variable-base heating and cooling degree-days for turkey”, *Applied Energy*, vol. 69, no. 4, pp. 269–283, 2001.
- [94] L. Yang, J. C. Lam, and C. L. Tsang, “Energy performance of building envelopes in different climate zones in china”, *Applied energy*, vol. 85, no. 9, pp. 800–817, 2008.
- [95] P. Borah, M. K. Singh, and S. Mahapatra, “Estimation of degree-days for different climatic zones of north-east india”, *Sustainable Cities and Society*, vol. 14, pp. 70–81, 2015.
- [96] International Energy Agency, *Average cooling degree days in g20 economies, 1979-2022*, IEA, Paris, 2023. [Online]. Available: <https://www.iea.org/data-and-statistics/charts/average-cooling-degree-days-in-g20-economies-1979-2022> (visited on 01/17/2025).

- [97] MeteoSwiss, *Cooling day — weather and climate from a to z*, Accessed: 2025-01-17, n.d. [Online]. Available: <https://www.meteoswiss.admin.ch/weather/weather-and-climate-from-a-to-z/cooling-day.html>.
- [98] D. Yu, G. Bai, Y. Li, and L. Zhao, “Deep spatial domain generalization”, in *2022 IEEE International Conference on Data Mining (ICDM)*, IEEE, 2022, pp. 1293–1298.
- [99] D. Koldasbayeva, P. Tregubova, M. Gasanov, A. Zaytsev, A. Petrovskaya, and E. Burnaev, “Challenges in data-driven geospatial modeling for environmental research and practice”, *Nature Communications*, vol. 15, no. 1, p. 10 700, 2024.
- [100] J. Li and A. D. Heap, “A review of comparative studies of spatial interpolation methods in environmental sciences: Performance and impact factors”, *Ecological Informatics*, vol. 6, no. 3-4, pp. 228–241, 2011.
- [101] J. P. Kleijnen, “Kriging metamodeling in simulation: A review”, *European journal of operational research*, vol. 192, no. 3, pp. 707–716, 2009.
- [102] A. Maissen, F. Techel, and M. Volpi, “A three-stage model pipeline predicting regional avalanche danger in switzerland (ravafcast v1. 0.0): A decision-support tool for operational avalanche forecasting”, *Geoscientific Model Development*, vol. 17, no. 21, pp. 7569–7593, 2024.
- [103] T. pandas development team, *Pandas-dev/pandas: Pandas*, version latest, Feb. 2020. DOI: 10.5281/zenodo.3509134. [Online]. Available: <https://doi.org/10.5281/zenodo.3509134>.
- [104] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python”, *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [105] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, “Pytorch: An imperative style, high-performance deep learning library”, *Advances in neural information processing systems*, vol. 32, 2019.
- [106] M. Fey and J. E. Lenssen, *Fast graph representation learning with pytorch geometric*, MIT License, 2019. [Online]. Available: https://github.com/pyg-team/pytorch_geometric.
- [107] Google Inc., *Google colab*, Accessed: 2025-01-30, 2021. [Online]. Available: <https://colab.research.google.com>.
- [108] D.-A. Clevert, “Fast and accurate deep network learning by exponential linear units (elus)”, *arXiv preprint arXiv:1511.07289*, 2015.
- [109] I. Loshchilov, “Decoupled weight decay regularization”, *arXiv preprint arXiv:1711.05101*, 2017.
- [110] J. Han, A. Chong, J. Lim, S. Ramasamy, N. H. Wong, and F. Biljecki, “Microclimate spatio-temporal prediction using deep learning and land use data”, *Building and Environment*, vol. 253, p. 111 358, 2024.
- [111] A. Tzavali, J. P. Paravantis, G. Mihalakakou, A. Fotiadi, and E. Stigka, “Urban heat island intensity: A literature review”, *Fresenius Environmental Bulletin*, vol. 24, no. 12b, pp. 4537–4554, 2015.
- [112] Y. Li, D. Yu, Z. Liu, M. Zhang, X. Gong, and L. Zhao, “Graph neural network for spatiotemporal data: Methods and applications”, *arXiv preprint arXiv:2306.00012*, 2023.
- [113] Z. Wang, X. Wu, and Y. Wu, “A spatiotemporal xgboost model for pm2. 5 concentration prediction and its application in shanghai”, *Heliyon*, vol. 9, no. 12, 2023.

- [114] H. Katayama, S. Yasuda, and T. Fuse, “Comparative validation of spatial interpolation methods for traffic density for data-driven travel-time prediction”, *International Journal of Intelligent Transportation Systems Research*, vol. 20, no. 3, pp. 830–837, 2022.
- [115] J. Schatz and C. J. Kucharik, “Urban heat island effects on growing seasons and heating and cooling degree days in madison, wisconsin usa”, *International Journal of Climatology*, vol. 36, no. 15, pp. 4873–4884, 2016.
- [116] L. Kleerekoper, M. Van Esch, and T. B. Salcedo, “How to make a city climate-proof, addressing the urban heat island effect”, *Resources, Conservation and Recycling*, vol. 64, pp. 30–38, 2012.
- [117] K. Deilami, M. Kamruzzaman, and Y. Liu, “Urban heat island effect: A systematic review of spatio-temporal factors, data, methods, and mitigation measures”, *International journal of applied earth observation and geoinformation*, vol. 67, pp. 30–42, 2018.
- [118] S. Magli, C. Lodi, L. Lombroso, A. Muscio, and S. Teggi, “Analysis of the urban heat island effects on building energy consumption”, *International Journal of Energy and Environmental Engineering*, vol. 6, pp. 91–99, 2015.
- [119] B. Sanchez-Lengeling, J. Wei, B. Lee, E. Reif, P. Wang, W. Qian, K. McCloskey, L. Colwell, and A. Wiltischko, “Evaluating attribution for graph neural networks”, *Advances in neural information processing systems*, vol. 33, pp. 5898–5910, 2020.
- [120] Y.-M. Shin, S. Li, X. Cao, and W.-Y. Shin, “Revisiting attention weights as interpretations of message-passing neural networks”, *arXiv preprint arXiv:2406.04612*, 2024.

Data and code availability

<https://github.com/urbes-team/STFGAN>

List of Acronyms

NDVI Normalized Difference Vegetation Index

UHI Urban Heat Island

LST Land Surface Temperature

WRF Weather Research and Forecasting

UCM Urban Canopy Model

SML Spatial Machine Learning

CNN Convolutional Neural Network

RNN Recurrent Neural Network

SHAP Shapley Additive Explanations

ML Machine Learning

ConvLSTM Convolutional Long Short-Term Memory

GNN Graph Neural Network

GATG Graph Attention Network with a Gated Recurrent Unit

HDD Heating Degree Days

CDD Cooling Degree Days

PBM Physics-Based Models

BEC Building Energy Consumption

GRU Gated Recurrent Unit

XGBoost Extreme Gradient Boosting

GML Graph Machine Learning

GCN Graph Convolutional Network

LSTM Long Short-Term Memory

GAT Graph Attention Network

LCD Low-Cost Measurement Device

LULC Land use and Land cover

GATConv Graph attentional operator

STFGAN Spatio-Temporal Fusion Graph Attention Network

CDH Cooling degree hours

SVM Support Vector Machine

BES Building Energy Simulation

MSE Mean Squared Error

MAE Mean Absolute Error

RMSE Root Mean Squared Error

MAPE Mean Absolute Percentage Error

MdAPE Median Absolute Percentage Error

Figures & Tables

Table 1: Feature List

Description	Variable name	Type	Value Range	Source
Building's height	avg_height	Numerical	[0,59]	Swiss3DBuildings [82]
Building's count	build_coun	Numerical	[0, 199]	Swiss3DBuildings [82]
Average distance of the 50, 100 and 200 nearest buildings [m]	dX_avg	Numerical	[0,1001]	Swiss3DBuildings [82]
Population	population	Numerical	[0, 261]	Federal Statistical Office [83]
NDVI	NDVI	Numerical	[0.04, 0.79]	Sentinel [86]
Albedo	Albedo	Numerical	[0.11, 0.20]	MODIS [87]
Elevation [m]	elevation	Numerical	[498, 656]	NASA SRTM [84]
Local climate zones	LCZ	Categorical	[1, 17]	Wudapt [78]
Distance to Bern's city center [m]	dist_to_center	Numerical	[0, 7245]	-
Distance to the nearest green space [m]	dist_to_forest	Numerical	[0, 1131]	SwissTLM3D [85]
Temperature at the reference stations [°C]	Log_98	Numerical	[2.37, 35.9]	UNIBE [81]
Average hourly vapor pressure [hPa]	vapor_pressure	Numerical	[5.4, 24.7]	MeteoSwiss [88]
Global irradiation [W/m²]	global_irradiation	Numerical	[0, 1034]	MeteoSwiss [88]
Atmospheric pressure [hPa]	atmospheric_pressure	Numerical	[935, 967]	MeteoSwiss [88]
Maximum hourly temperature [°C]	max_hourly_temperature	Numerical	[2.6, 35.4]	MeteoSwiss [88]
Minimum hourly temperature [°C]	min_hourly_temperature	Numerical	[1.9, 34.8]	MeteoSwiss [88]
Hourly precipitations [mm]	precipitations_hourly	Numerical	[0, 23.6]	MeteoSwiss [88]
Average hourly humidity [%]	mean_hourly_humidity	Numerical	[21.6, 100]	MeteoSwiss [88]
Hourly sunshine duration [h]	sunshine_hourly_duration	Numerical	[0, 1]	MeteoSwiss [88]
Hourly dew point [°C]	hourly_dew_point	Numerical	[-1.8, 21.6]	MeteoSwiss [88]
Average hourly wind speed [m/s]	scalar_hourly_mean_wind_speed	Numerical	[0, 7.9]	MeteoSwiss [88]
Average hourly wind direction [°]	mean_hourly_wind_direction	Numerical	[0, 360]	MeteoSwiss [88]
Hour of the day	hour	Numerical	[0, 24]	UNIBE [81]
Month of the year	month	Categorical	[5, 9]	UNIBE [81]

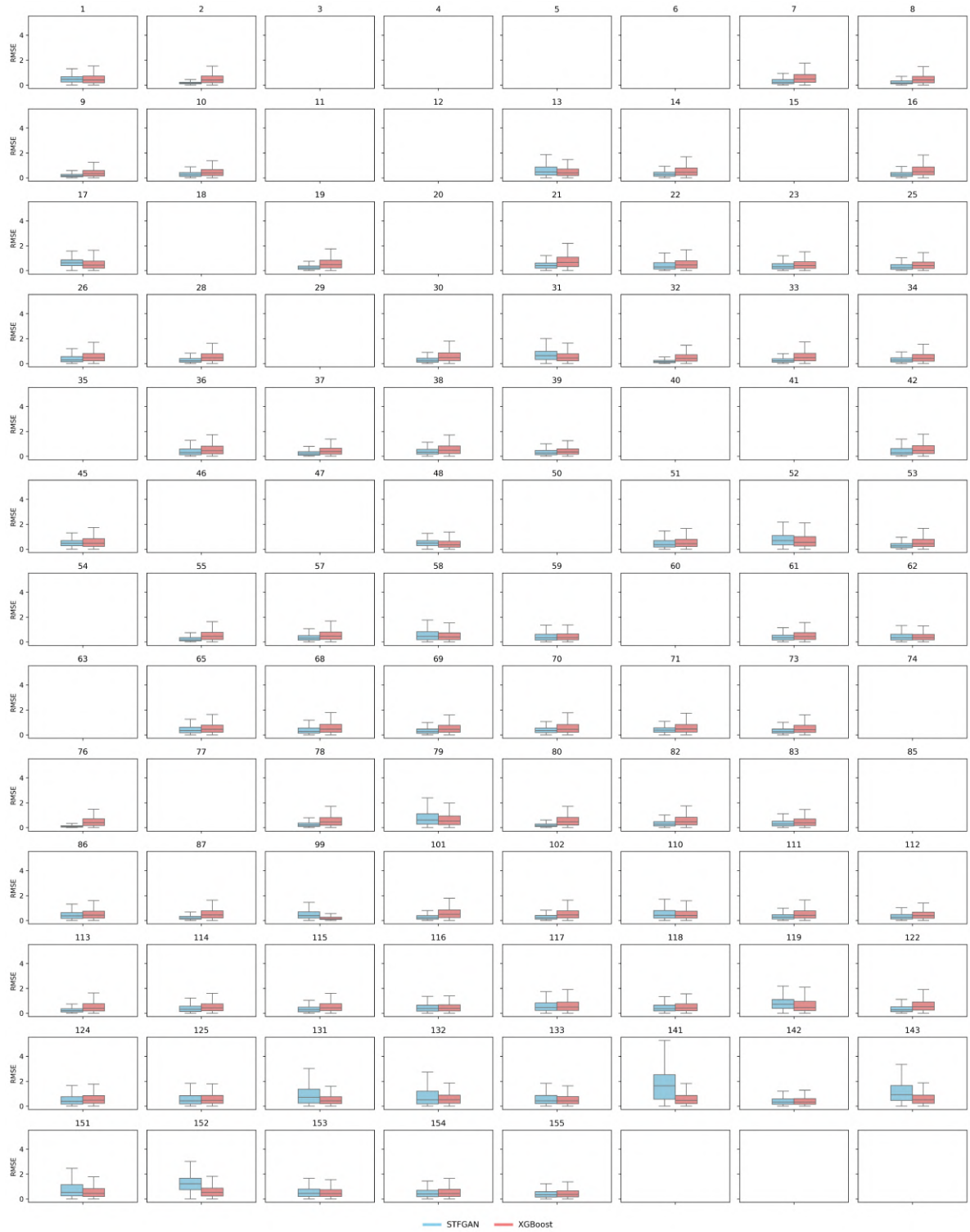


Figure 1: Comparison of the error for each LCD