2024

Ambient temperature modelling with ECOSTRESS and private weather stations

Gaurav Khatri

AMBIENT TEMPERATURE MODELLING WITH ECOSTRESS AND PRIVATE WEATHER STATIONS

Gaurav Khatri

A THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science
in
The Department of Computer Science
to
The Graduate School
of
The University of Alabama in Huntsville

May 2024

Approved by:

Dr. Huaming Zhang, Research Advisor/Committee Chair
Dr. Leiqiu Hu, Research Advisor/Committee Member
Dr. Deepak Acharya, Committee Member
Dr. Letha Etzkorn, Department Chair
Dr. Rainer Steinwandt, College Dean
Dr. Jon Hakkila, Graduate Dean

Abstract

AMBIENT TEMPERATURE MODELLING WITH ECOSTRESS AND PRIVATE WEATHER STATIONS

Gaurav Khatri

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science

Computer Science

The University of Alabama in Huntsville

May 2024

This thesis explores the development and application of a novel data architecture for predicting ambient temperatures across US cities, focusing on integrating multi-source data *i.e.*, ECOSTRESS land surface temperatures, urban surface properties, and crowdsourced weather data. The methodology is designed for scalability and adaptability across different urban regions, employing rigorous data quality control to enhance prediction accuracy. The validation of this model across diverse urban settings, demonstrated through rigorous RMSE comparisons and spatial mapping, validates its superiority over traditional models. Through experiments in diverse climatic conditions in Madison, Wisconsin, and Las Vegas, Nevada, the study assesses the model's generalizability and effectiveness in capturing spatio-temporal temperature variations. This study aims to contribute to urban heat island mitigation and sustainable urban planning, setting a benchmark for future research in urban climatology.

Acknowledgements

I am truly grateful to everyone who has played a part in bringing this project to fruition. I want to express my heartfelt thanks to my supervisors, Dr. Huaming Zhang and Dr. Leiqiu Hu for their invaluable mentorship, unwavering support, and constant motivation throughout this project. With their vast expertise, immense patience, and unwavering dedication, they have played a pivotal role in shaping the direction of my research and guiding me through any obstacles I faced. Without their invaluable contributions, this study would not have been possible. Their invaluable perspectives have enriched my understanding and added depth to my research findings. I am truly grateful for their generosity and willingness to be a part of this project.

I would like to express my sincere gratitude to The Department of Computer Science for their generous provision of resources, facilities, and infrastructure, which have greatly contributed to the successful completion of my project.

The accommodating research environment and collaborative atmosphere have played a crucial role in my achievements. In addition, I would like to extend my appreciation to my colleagues, friends, and family members for their unwavering support, encouragement, and understanding. Their constant presence and patience have been a source of motivation and inspiration throughout my journey.

Finally, I am deeply thankful to the authors, researchers, and scholars whose previous work has paved the way for my study. Their valuable insights, discoveries, and contributions have been invaluable in shaping my research.

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

1.1 Background

Understanding and predicting the spatio-temporal variations in the urban canopy layer is critical for mitigating the adverse effects of urban heat islands (UHIs) and ensuring sustainable urban development. High air temperature within hot summer months poses significant challenges to human health, energy consumption, and environmental quality [15]. Global warming has resulted in higher extreme temperatures worldwide and will likely continue to increase the extreme temperatures in the future[14]. Such extreme conditions have pronounced effects on our society such as increased mortality rates, disruptions in energy infrastructures, and risk of accidents[32]. Accurately predicting these variations at higher resolutions enables proactive measures to be taken, such as issuing heatwave warnings, implementing urban greening strategies, and optimizing energy use. This study examines the current advancements in spatio-temporal prediction of urban temperature, focusing on data sources, prediction methods, and key challenges.

While the rural landscape is dominated by the same homogeneous cover of vegetation and soil, the urban area has a higher level of both spatial and temporal heterogeneity in the surface properties. This diversity is not only confined

to structures with different thermal properties, non-absorbent roadways, and intermittent green spaces, but it also involves the interaction of energy absorption, reflection, and re-radiation. Due to this effect, which is called the "urban heat island", the temperature in some areas can be much higher than in the neighboring environments [15]. Also, the "canyon effect" caused by high buildings renders heat and reduces airflow which makes the temperature contrast within the urban canopy worsen[5]. Furthermore, anthropogenic heat sources such as transportation and industrial processes contribute to elevated urban temperatures. The interplay of these factors, along with complex microclimates within urban canyons, results in larger and more dynamic temperature variations compared to rural areas [3]. Due to these complex processes and their interactions, accurately predicting urban temperature at higher resolutions presents significant challenges.

1.2 Research Problem

Although traditional weather stations are really useful for understanding large-scale temperature patterns, they lack the complexity to adapt to the high-resolution needs of urban areas. The accurate depiction of urban temperature requires the proper modeling of various spatial and temporal properties discussed above. However, the weather stations are unevenly distributed and situated in non-representative places such as rooftops and across larger distances, that do not capture the micro-climate variations within and among the urban settings[6].

As a result, the continuous field of urban temperature cannot be interpreted from the limited data points hindering the overall analysis and prediction.

Recent methodologies such as remote sensing provide valuable insights into surface properties and land use [27], and the use of mobile sensor networks helps to fill gaps in traditional networks. Through a combination of traditional methods with cutting-edge technologies, we can get to a place where urban temperature can be assessed properly for informed planning against heat stress, construction of heat mitigation strategies, and better public health, particularly in our rapidly changing cities. In recent years, the availability of low-cost weather devices has also enabled the prospect of utilizing low-cost crowdsourced data to understand temperature patterns more effectively. This is especially true for urban areas with a denser network of such weather devices, commonly called "private weather stations" (pws). However, such private weather stations (pws) are highly prone to erroneous data *i.e.*, releasing data from faulty devices, devices inside the buildings, and devices in unusual places, compromising the overall data quality. Cleaning these data and enriching it with publicly available satellite observations enables us to predict ambient temperatures across larger regions accurately.

In conclusion, urban temperature modeling is crucial for us to mitigate the risks of extreme summer temperatures. Traditional methods often fail to capture the complex spatial and temporal variations within urban landscapes at higher resolutions. This presents a unique opportunity to utilize remote sensing data with surface properties and crowdsourced observations from private weather stations to enhance urban predictions at higher resolutions.

1.3 Approach

Building upon established research exploring the relationship between Land Surface Temperature (LST), near-surface air temperature, and urban surface characteristics, we propose a novel methodology for spatial-temporal prediction of temperature at a higher resolution of 70 meters and a temporal resolution of 24 hours, that leverages the synergy of these diverse data sources. We use different regression and ensemble techniques to model temperature accurately and compare the results. At first, the different data sources are processed and combined into a single source. Given the nature of pws data, they are subjected to systematic filtering criteria described in Chapter 3 as specified in the implementations of CrowdQC package[7]. Once combined, the observations are aggregated at an hourly level for each month filtering out the noise from lower temporal resolutions.

Due to the diurnal pattern of temperature observations, hour as a feature will always have a higher weightage in simple models. Hence, residuals are calculated for each station by subtracting the hourly mean to remove the undue influence of the hour parameter in the decision trees, as discussed further in Chapter 3.2. This eliminates the higher feature importance for the hour parameter. Finally, different models are trained and experimented to find the most optimal parameters for better scores and improved visual representations.

This overall methodology offers several key advancements, including:

1. **Multi-source data integration**: We utilize Ecosystsem Spaceborne Thermal Radiometer Experiment on Space Station (ECOSTRESS) LST data,

urban surface properties, and crowdsourced weather data to develop comprehensive regression models, improving the overall prediction accuracy. ECOSTRESS is an ongoing scientific experiment, launched in 2018 in collaboration with Jet Propulsion Laboratory (JPL) in which a radiometer mounted on the International Space Station (ISS) measures the temperature of plants growing in specific locations on Earth over a solar year. Through these observations, the experiment intends to understand global events such as heat waves and their overall impacts.

2. **Scalability and adaptability**: The architecture is designed to be scalable and adaptable to different regions. By combining crowdsourced data, ECOSTRESS, and Urban Surface data, all of which are publicly available, we can build models applicable to any city of choice. The residual temperature calculations in combination with the hybrid training architecture that we designed enable us to utilize the same architecture across different regions.

3. **Rigorous data quality control**: We implement a rigorous quality control pipeline to ensure the accuracy and reliability of the crowdsourced data before using them in our models. Although there are several frameworks such as the CrowdQC package available for crowdsourced data, we provide experimental verification that those procedures are still not enough and suggest additional quality control mechanisms such as Quantile Based Anomaly removal for improved robustness.

4. **Diurnal Maps**: The architecture aims to generate hourly temperature maps at the monthly level, providing detailed spatiotemporal information.

5. **High-resolution prediction**: Our goal is to achieve 70-meter resolution predictions, significantly higher than typical weather predictions. Although high-resolution predictions are really important to effectively understand urban climate conditions, conventional weather stations are expensive to operate at such levels. LST although easily available at higher resolutions, by itself is not the single most suitable parameter to estimate air temperature as it deviates significantly during extreme conditions and is also not available at finer temporal resolutions to incorporate urban climate variations [32]. Hence, the capacity to incorporate these different data sources presents interesting applications.

To validate our approach, we will initially focus on two contrasting regions: Madison, Wisconsin - typically cooler and moist, and Las Vegas, Nevada which is hot and dry. By testing in these diverse environments, we can assess the generalizability of our model and its ability to handle different climatic and urban surface conditions.

1.4 Thesis Organization

The overall thesis is organized into six chapters. Chapter 1 deals with the primary introduction to the research topic, the necessary background, and the

research problem. We discuss a bit about our research approach to summarize how we aim to solve the research problem and the key advancements of this study.

Chapter 2 gives a brief introduction to the relevant literature review and significant shortcomings of all the studies done till now, based upon which we can build more complex models. It also gives an overview of the regression techniques that we discuss in this research and their mathematical foundations.

Chapter 3 goes into great detail about the technique employed for this research, including the study area. This research work is an application of Computer Science knowledge in the Earth Science domain (*i.e.*, prediction of ambient temperature). This chapter provides in-depth explanations of computer science components such as Regression Algorithms, Machine Learning, Random Forests, Residual Methods, Neural Networks, Ensemble Methods, etc.

The fourth chapter presents the overall results of the experiments performed across Las Vegas and Madison for the summer months (*i.e.*, June, July, and August). We implement different algorithms *i.e.*, Linear Regression, Random Forest, Gradient Boosting Regression, and Extreme Gradient Boosting to plot predicted temperature maps for the entire domain and discuss the shortcomings at each step. Then we also discuss a hybrid training procedure that combines multiple Gradient Boosting to give the best results and explain our reasoning on why it works much better. We also dive a bit into Artificial Neural Networks and discuss why additional methods such as Geospatially Weighted Regression Techniques are not pursued in this study given the nature of our data. Chapter six finally provides the conclusion of our study and its possible future directions.

Chapter 2 Background and Related Works

2.1 Supervised Machine Learning

Supervised learning is a fundamental machine learning approach in which a model learns a mapping between labeled input data and desired outputs. This "learning" process entails examining labeled training instances, which are made up of pairs of input data and associated goal values. The model then applies this information to produce predictions for previously unseen data, to successfully generalize to new contexts, enabling applications like image recognition, spam filtering, and house price prediction. While offering interpretability and high accuracy, supervised learning relies on substantial labeled data, susceptible to overfitting and inheriting biases if present. Supervised learning can be broadly categorized into two categories: *i.e.*, classification (predicting categories) and regression (predicting continuous values).

In our study, the problem of predicting temperature values at unsampled locations can be modeled as a regression problem. This warrants the use of several regression algorithms discussed below.

2.2 Linear Regression

Regression is a supervised learning technique that models the relationship between a dependent variable (what is predicted) and one or more independent variables (predictors). This relationship is commonly stated as a continuous function, which allows us to predict the target value for previously unobserved data points.

Regression models are classified into several categories, each with its own set of assumptions and applications. However, the underlying principle for linear regression, the most popular type, may be described using the following equation:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p + \epsilon, \tag{2.1}$$

where y is the dependent variable, $x_1, x_2, \ldots, x_n$ are the independent variables, $\beta_0, \beta_1, \beta_2, \ldots, \beta_n$ are the regression coefficients, and ϵ is the error term representing unexplained variability.

2.3 Polynomial Regression

Polynomial regression is a form of regression analysis in which the relationship between the independent variable x and the dependent variable y is modeled as an n-degree polynomial function [17]. Unlike linear regression, polynomial regression allows us to model nonlinear relationships between the independent and dependent variables by introducing higher-order terms (*e.g.,* $x^2, x^3, \ldots, x^n$). This flexibility enables us to capture more complex patterns in the data. Polynomial

regression can be implemented using various techniques, such as ordinary least squares (OLS) regression, gradient descent, or polynomial basis functions. The general equation for polynomial regression is:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_n x^n + \varepsilon,$$

where y is the dependent variable, x is the independent variable, $\beta_0, \beta_1, \ldots, \beta_n$ are the coefficients of the polynomial, ε is the error term.

2.4 Spatial Regression

Spatial regression is a statistical technique used to model the relationships between variables that are observed at different locations in space. It accounts for the spatial autocorrelation present in the data, which means that nearby observations tend to be more similar to each other than observations farther apart. Spatial regression models often incorporate spatial weights matrices to capture the spatial relationships between observations [4]. These weights matrices specify the degree of association between observations based on their spatial proximity[20]. Some common methods include Ordinary Least Squares (OLS), Spatial Autoregressive (SAR), and Spatial Error Models (SEM). Spatial regression is widely used in fields such as geography, environmental science, economics, and public health to analyze spatial data and understand the spatial patterns of phenomena.

The general equation for spatial regression can be written as:

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \ldots + \beta_p x_{pi} + \epsilon_i,$$

where y_i is the dependent variable at location i, $x_{1i}, x_{2i}, \ldots, x_{pi}$ are the independent variables at location i, $\beta_0, \beta_1, \ldots, \beta_p$ are the coefficients, ϵ_i is the error term, which accounts for unexplained variability and spatial autocorrelation.

2.5 Temporal Regression

Temporal regression, similar to spatial regression, is a statistical method used to model the relationships between variables over time. It accounts for the temporal autocorrelation present in the data, meaning that observations collected closer in time tend to be more similar to each other than observations collected further apart. Temporal regression models often incorporate time-series techniques to account for the sequential nature of the data. Some examples include autoregressive integrated moving average (ARIMA) models, autoregressive conditional heteroskedasticity (ARCH) models, and vector autoregression (VAR) models, among others. The choice of temporal regression model depends on the specific characteristics of the data, such as trend, seasonality, and stationarity.

The general equation for temporal regression can be expressed as:

$$y_t = \beta_0 + \beta_1 x_{1t} + \beta_2 x_{2t} + \ldots + \beta_p x_{pt} + \epsilon_t,$$

where y_t represents the dependent variable at time t, $x_{1t}, x_{2t}, \ldots, x_{pt}$ are the independent variables at time t, $\beta_0, \beta_1, \ldots, \beta_p$ are the coefficients, ϵ_t denotes the error term, capturing unexplained variability and temporal auto-correlation.

2.6 Geographically Weighted Regression (GWR)

Geographically Weighted Regression (GWR) is a spatial regression technique that extends traditional regression by allowing the regression coefficients to vary locally across space [28]. This means that the relationship between the dependent and independent variables can differ depending on the location we're analyzing. This flexibility makes GWR particularly useful for modeling spatially non-stationary processes where relationships might change geographically.

GWR estimates separate regression equations for each observation in the dataset, considering the values of neighboring observations within a defined bandwidth. Each neighboring observation contributes to the local regression with a weight based on its distance from the target observation. Weights typically decline smoothly with increasing distance, often using a kernel function like Gaussian or bi-square.

The general form of the GWR model can be expressed as:

$$y_i = \beta_0(u_i, v_i) + \sum_{j=1}^{n} \beta_j(u_i, v_i) x_{ij} + \epsilon_i, \tag{2.2}$$

where y_i is the dependent variable value for observation i, (u_i, v_i) are the spatial coordinates of observation i, $\beta_0(u_i, v_i)$ is the intercept for the local regression at (u_i, v_i), $\beta_j(u_i, v_i)$ are the local regression coefficients for independent variable j at (u_i, v_i), x_{ij} is the value of independent variable j for observation i, n is the total number of observations , ϵ_i is the error term for observation i.

2.7 Decision Trees

Decision trees are a popular and intuitive machine learning algorithm used for both classification and regression tasks. They model decisions based on a series of if-else conditions and are particularly useful for understanding and interpreting the underlying patterns in the data[19]. The decision tree algorithm recursively partitions the dataset based on the values of features. It selects the best feature to split the data at each internal node, aiming to maximize information gain or minimize impurity. The process continues until certain stopping criteria are met, such as reaching a maximum depth or having a minimum number of samples in each node. The ability to model non-linear relationships effectively well enables Decision Trees to have very high importance in spatial regression[26].

Additionally, there are specific properties of Decision Trees that make them highly useful for our study. Primarily, the spatial variation in ambient temperature is highly regulated by a few surface properties such as impervious fraction, tree fraction, etc., which enables the ability to create robust trees that can easily depict such relationships[23]. Additionally, it has been observed that some surface properties such as water fraction will substantially affect Ta only after exceeding a certain threshold[29]. The splitting process in Decision Trees automatically enables it to recognize these threshold values, thus being able to create general models for multiple cities.

In our study, we will mostly explore ensemble-based methods which combine multiple Decision Trees, since a single decision tree has stability and capacity

issues. We implement different ensemble-based methods such as Random Forest, Gradient Boosting, and Extreme Gradient boosting which aggregate multiple weak decision trees in different ways, which is discussed in Chapter 3.

2.8 Related Study

Traditionally, weather stations by themselves offer limited spatial coverage, hindering high-resolution spatio-temporal prediction of temperature in complex urban environments. Relevant studies such as Li *et al.* [13] have specifically highlighted the need for multiple data sources for proper modeling of such spatiotemporal variations. They emphasized the potential of dense networks for capturing local variations in temperature, providing valuable insights into the heterogeneity of the urban thermal environment. Approaches to high-density temperature monitoring have also been tried, including methods such as vehicle-mounted temperature sensors [21], but these methods are costly to maintain and scale across regions. These limitations have driven researchers toward adopting alternative data sources for accurate predictions. Venter *et al.* [24] and Shandas *et al.* [22] acknowledged the value of satellite data (LANDSAT, LiDAR) for high-resolution temperature mapping, offering a broader spatial perspective and enabling the identification of large-scale thermal patterns. Land Surface Temperature (LST) observations have also been used to aid high-resolution studies [18], offering valuable information for temperature modeling enabling us to develop more accurate and comprehensive models. However, LST by itself is not the best parameter to estimate extreme temperatures as its relationship with air temperature deviates

significantly during peak summer months [9]. Moreover, satellites have an observational duration of weeks to months at infrequent time intervals, making it difficult to have a continuous overview of the LST patterns over a region. This is where the choice of ECOSTRESS observations improves the quality of the input data in our study since ECOSTRESS overpasses the same location at different hours within a few days, resulting in a more comprehensive LST pattern for input [11].

Additional research has been done by Zumwald *et al.* [31] to explore the use of Private Weather Station (PWS) data for urban temperature monitoring, providing valuable ground-level information but requiring careful consideration of potential uncertainties and biases inherent in such data. As PWS data are highly prone to errors, providing a scalable system for data assessment and quality control is crucial. Significant research has been done by Fenner *et al.* [7] to establish a general framework to control the quality of crowdsourced air temperature observations, which we have implemented as a primary quality control scheme. However, their approach only considers the fact that errors in temperature measurements are independent of other features, which is not always the case. During our experiments, it became evident that rejecting anomalous temperature observations based on feature values is important to improving model accuracy as discussed in Chapter 3.

In addition to these data sources, various methods have been researched over the years for spatio-temporal temperature prediction. Traditional methods including physical models based on physical principles are highly accepted but

have the drawback of incurring high computational cost and a coarse resolution, thus limiting the applicability [12]. These studies are highly dependent on observations from available weather stations which limits the application of these studies to only areas with denser observation networks. Statistical interpolation via methods such as Kriging [16] seemed promising in different domains, yet encountered issues in correctly reproducing non-linear spatio-temporal behaviors for climate applications as discussed by Federico *et al.* in their study [2].

With the advent of Machine Learning (ML) and cheaper computing power, several studies have been done to adapt common ML models for spatio-temporal predictions. Although different models ranging from Multiple Linear Regression to Decision Trees and Neural Networks have been tried, tree-based models have usually outperformed the other methodologies in 60% of the studies[26]. Simpler methods such as Linear Regression are unable to model the complex non-linear relationship between different urban properties. Neural Networks tend to work well for temporal predictions, but these models require large training data to prevent overfitting. Additionally, the lack of long-term historical observations limits their applicability for a lot of regions that lack such data. However, Decision Trees and their variations have a great capacity to model non-linear relationships in spatial predictions effectively given the fact that spatial variation is highly regulated by few features enabling decision trees to make optimal splits and also the inherent capacity of decision trees to model the threshold relationship of temperature and independent features [29] [23].

However, most of the studies are focused on limited areas and do not provide sufficient demonstrations of applicability across regions of different geographical properties and periods [26]. Stenka *et al.* [25] have extensively focused on nighttime predictions within Germany combining crowdsourced temperature data with LST obtained from LANDSAT and geodata to train ML models. However daytime predictions are extremely important for creating urban mobility policies to support regions with extreme temperatures. From a Computer Science point of view, tree-based models tend to underestimate high temperatures observed during day-time hours and overestimate low temperatures observed during night-time hours [32][30], resulting in larger error values during certain hours of the day, resulting in practical difficulty to develop a robust ML model that provides accurate predictions for different hours of the day across different regions.

Zumwald *et al.* [32] demonstrated the applicability of machine learning to generate high-resolution (10 m x 10 m) urban air temperature by using PWS data with spatial and meteorological predictors for 1 day. Additional works by Federico *et al.* [2] have been done to further explore the use of deep learning networks for spatio-temporal prediction for synthetic datasets and a real-world study that uses complex meteorological station networks. Although the framework is promising, the real-world application needs further validation using diverse real-world environmental datasets.

Our study expands on these recent approaches and provides significant advances in terms of prediction errors and applicability across diverse geographical regions. This study provides diurnal predictions at a monthly level providing

stable results for further study of urban climate conditions such as heat index, pollutant study, and environmental justice. We build upon the quality control mechanism for crowdsourced data provided by Fenner *et al.* [8] and implemented by recent studies by Zumwald *et al.* [31] and Federico *et al.* [2], by incorporating feature-based anomaly detection to remove extensive outliers, resulting in improved prediction errors compared to previous studies. We then build and compare several ensemble models to suggest a novel ensemble architecture for this application. Based on this model, we produce spatiotemporal diurnal maps for 70-meter resolution levels at monthly levels for June, July, and August across multiple years and multiple US cities (*i.e.*, Madison, Las Vegas).

Chapter 3. Methodology

3.1 Study Area

For this study, we focus on two areas, Madison, Wisconsin (43.07N, -89.38E), and Las Vegas, Nevada (36.17N, -115.14E). The Madison downtown area is situated alongside two lakes: Lake Mendota and Lake Monona. This presents an interesting challenge for the study, as the diurnal temperature pattern of water and nearby land bodies is expected to show different behaviors during different hours of the day. For example, during nighttime, water bodies are comparatively warmer compared to the surrounding landmass. Similarly, during peak afternoons, water bodies are relatively cooler compared to the landmass. This ensures that the model will be easily scalable across areas with different surface conditions and also enables easy visual analysis of the model prediction.

We also make a conscious choice of using LST data from NASA's Ecosystem Spaceborne Thermal Radiometer Experiment on Space Station (ECOSTRESS) observations as it provides detailed temperature images of the Earth's surface at a high resolution of 70 meters [11]. Additionally, it is a low-orbit satellite, with frequent observations at different timestamps, unlike LANDSAT measurements. This lets us capture the diurnal variations in surface measurements over multiple years, improving the model capacity.

Our study will be focused only on peak summer months, *i.e.*, June, July, and August.

3.2 Dataset and Processing Pipeline

One major component of this study is to build a config-driven scalable data pipeline that can be migrated across any region within the US to produce a reliable data source for any location and period of choice. The entire pipeline downloads the relevant data for the region of choice, processes the data, aggregates it, and combines it with multiple sources to be ready for final modeling. The overall workflow of this pipeline is shown in Figure 3.1.

The different layers of this pipeline are described below.

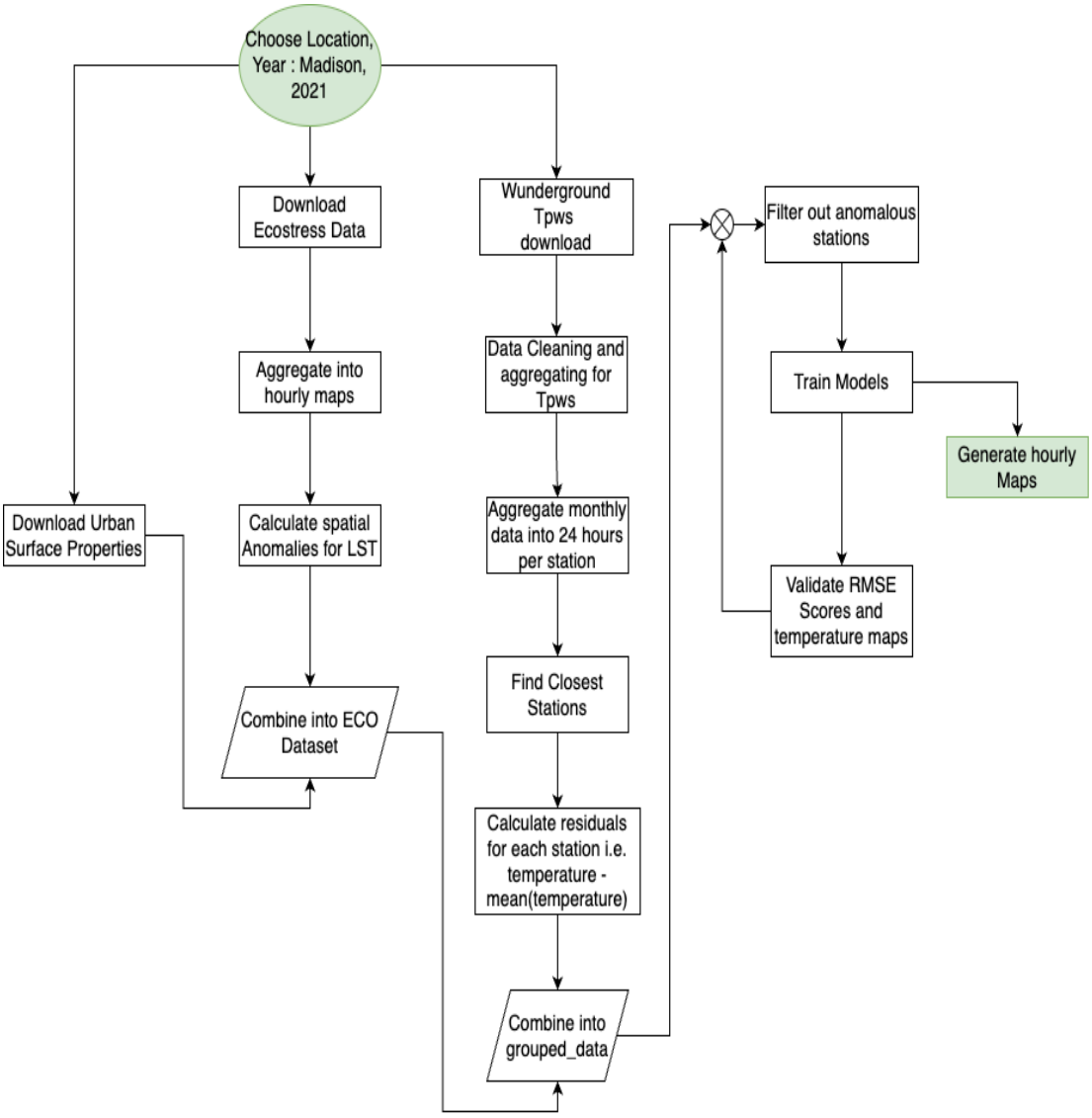

Figure 3.1: Pipeline Diagram.

3.2.1 Data Ingestion Layer

In this layer, are primarily focused on gathering required data at a reasonable speed. The overall working of this layer can be described as follows:

1. **Download $\mathbf{T_{pws}}$ Data**: At first, we acquire ground-based near-surface air temperature data (T_{pws}) for the chosen location and year from Wunderground API (eg. Madison, 2021). This data needs to be cleaned for erroneous stations *i.e.*, missing data, invalid values, etc. For the cleaning, we incorporate primary data cleaning based on the methodologies described in the Crowd QC+ paper [7] to flag anomalous values based on gross error,spike-dip test, temporal consistency, and visual check procedures. Additionally, the initial data is in 15-minute frequency, which is then aggregated on an hourly basis, making the data more robust to outlier observations. Each observation is identified by Station ID, Date, Timestamp, and Temperature value. The data summary from this step is shown in 3.1.

2. **Download ECOSTRESS Data**: The pipeline retrieves satellite-based land surface temperature (LST) data from the ECOSTRESS mission for the same location and year and stores it as GeoTIFF files. For uniformity, the temperature is converted to Celsius degrees. Values are assigned as N/A for either the cloudy/cloud-contaminated pixels or missing values. ECOSTRESS images are known for issues of geometric distortions and some

Table 3.1: Initial Dataset.

Location	Month	Total Observations	Number of Stations
Madison	June,2021	77967	118
	July,2021	85364	121
	August,2021	84539	119
Las Vegas	June,2021	113837	169
	July,2021	124782	173
	August,2021	124753	172

images are not well aligned. Thus, we georeference these images to make sure that they are well in line with the rest of the images.

3. **Download Urban Surface Data**: The grid products are then generated for the urban land surface properties dataset, including, impervious fraction, tree canopy fraction, land cover categories, building footprint fraction, and building height, for each single city using National Land Cover Dataset (NLCD)[1] products of 30-meter spatial resolution. Because of the differences in spatial resolution and grids between ECOSTRESS images and NLCD products, the NLCD products were aggregated to match the spatial resolution (70 meters) as well as the grids of ECOSTRESS images. The Urban Surface Data was aggregated from the NLCD Land Cover dataset, Microsoft Building Footprints, and Microsoft Building Footprints with Heights data sources. This results in one GeoTIFF file for each surface property.

The descriptions of these fields are given in Table 3.2.

Table 3.2: Urban Surface Properties.

Field Name	Field Description
Latitude	Latitude in degree format
Longitude	Longitude in degree format
valueImperviousfraction	Impervious fraction of surface
valueTreefraction	Tree fraction of surface
valueBuildingheight	Height of building if available
valueNearestDistWater	Distance to nearest water source
valueWaterfraction	Fraction of water surface
valueLandcover	Land cover ratio
valueBuildingfraction	Building area ration

3.2.2 Data Processing Layer

This stage deals we deal with the further processing of the different data sources such that they can be easily aggregated into a unified data source that can be fed to machine learning models. The ground and satellite observations are handled separately owing to the nature of the data, which is summarized below:

3.2.2.1 ECOSTRESS Data Processing

The Land Surface Temperature data is arranged such that we have 78 image files for Las Vegas ranging from the year 2019-2022. Similarly, Madison has 45 such files which accounts for clear surface observations. The files are then transformed into the same dimensions (681*681 pixels) and aggregated at an hourly level, which results in 24 image files. This enables us to understand diurnal pattern variations more easily. Since each of the original files corresponded to different dates and weather conditions, we calculate the spatial deviation in the feature and use it to train our models instead of the original LST value. The calculation is done as follows :

$$\mathbf{LST_{adjusted}} = \mathbf{LST} - \mathbf{LST_{mean}},$$

Any missing hour will be interpolated via average such that we have 24 image files for the entire domain, giving us the complete diurnal pattern for Land Surface Temperature. Each pixel in the image file corresponds to a 70m resolution and now has the following attributes as described in Table 3.3.

Table 3.3: LST Dataset Attributes.

Data Attribute	Attribute Description
Hour	Hour of the day
Latitude	Latitude value
Longitude	Longitude value
Adjusted LST	LST deviation

The Dataset from Table 3.3 will be combined with the dataset from Table 3.2 to give us a comprehensive urban dataset with hourly LST variations.

3.2.2.2 PWS Data Processing

In this step, the pws data is aggregated by station, month, and hour, such that each station has 24 observations for each month. This is done because the daily variation in temperature is highly volatile and hence it becomes really difficult to fine-tune models at a daily level. Hence, using monthly aggregated data and using it to spatially interpolate temperature for 24 hours creates stable models. The aggregated datasets obtained at the end of this step can be summarized below in Table 3.4.

We then create a new feature called 'closest_station_temperature' which is the average of recorded temperature for 3 closest observation stations for that timestamp. The introduction of this new feature is a very clever mechanism for us to integrate spatial relations into the model, which highly weights the temperature

values of nearby regions. This extra feature is really important for test data, as the test data will not have any recorded temperature values. Then we utilize a two-step data cleaning operation in this layer to filter out erroneous data. Finally, we use the distance tree and the aggregated dataset to prepare training data as described below.

Table 3.4: Final Data Count.

Location	Month	Total Observations
Madison	June,2021	2592
	July,2021	2616
	August,2021	2592
Las Vegas	June,2021	3552
	July,2021	3960
	August,2021	3888

3.2.2.3 KD Tree Construction

A KD tree, or "K-dimensional tree," is a data structure used to organize points in a K-dimensional space. It is especially useful for doing nearest-neighbor searches and range queries on multidimensional datasets. The general algorithm for KD Tree works as follows :

1. **Partitioning Space**: The first step in building a KD tree involves partitioning the space along alternating axes.

2. **Selecting Splitting Plane**: At each level of the tree, a splitting plane is selected perpendicular to one of the coordinate axes.

3. **Dividing Data** Once the splitting plane is chosen, the data points are divided into two subsets based on their positions relative to the splitting plane

4. **Recursive Construction**: The process of partitioning and subdividing continues recursively until each partition contains a small number of points or until a certain depth of the tree is reached

The total time complexity of building the K-D tree is O(n log n) and the total complexity of the nearest neighbor search in a K-D tree is O(log n) on average, where n is the number of points in the tree. This search complexity is a huge boost for test data, where we need to find the nearest neighbors for 463,761 data points in each domain.

3.2.2.4 Data Cleaning

In this step, we perform exploratory data analysis to analyze the overall training data pattern for a given location. Given the erroneous nature of private weather stations, there could be outlier stations as shown in Figure 3.2 for Madison. So, we employ a two-step procedure to remove these outliers.

1. **Statistical Outlier Detection**: In this step, any station whose mean values differ by more than 20% from the nearest 3 stations will be automatically removed.

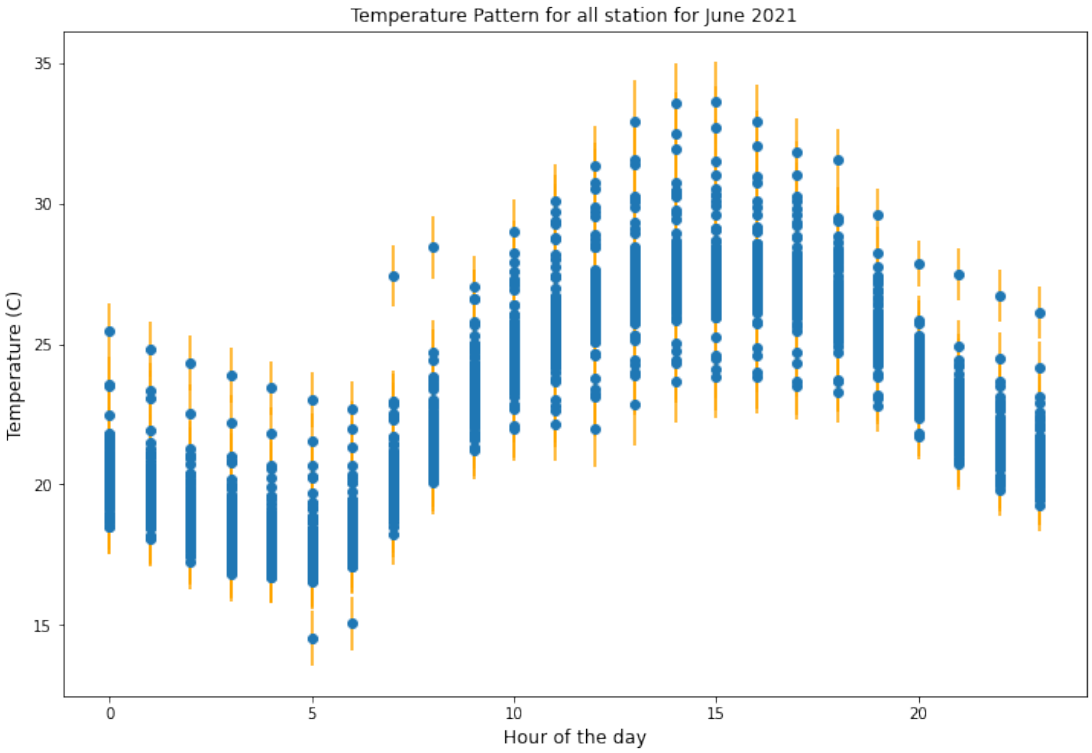

Figure 3.2: Outliers in Training Data for Madison.

2. **Null Value Handling**: Any station with more than 35% of missing values will be removed.

As a result, we get a better temperature distribution after this step as shown in Figure 2.3.

3.2.2.5 Residual Data Correction

Upon initial data examination, we observed a pronounced diurnal pattern in temperature, consistent with expectations as seen in Figure 3.3. Notably, specific hours of the day exhibit higher temperatures than others. While this

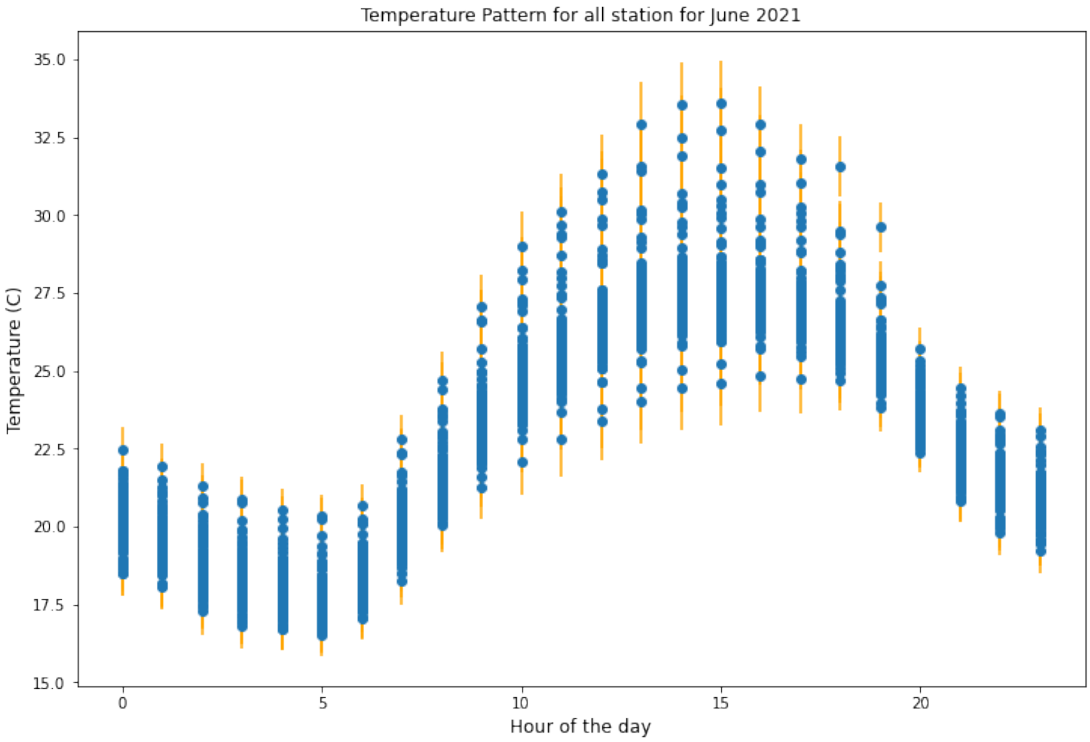

Figure 3.3: Removed Outliers in Training Data.

pattern is inherent, it can unduly influence predictive models, leading to inflated temperature attributions during certain hours. Consequently, temperature maps derived from such data may inaccurately reflect true temperature distributions. To mitigate this, we recalibrated our temperature values by subtracting the hourly mean values corresponding to each station. This correction yielded more accurate and reliable results in our output images, enhancing the fidelity of our temperature predictions. Furthermore, following the aforementioned data adjustment, the importance of the hour variable diminished. This refinement enabled us to employ a unified comprehensive model across different hours, thereby enhancing the efficiency and effectiveness of our modeling approach.

$$Residual_{hour-i} = Temperature(x, y) - Average(Temperature(x, y))$$

3.2.3 Data Modeling / Visualization Layer

In this section, we present a comprehensive overview of the data visualization and modeling layer, delineating its operational framework and iterative processes. Initially, the layer undertakes anomaly detection to filter out aberrant stations, ensuring data integrity and reliability. Following this, machine learning models are trained on the refined dataset to uncover underlying patterns and relationships. The details of machine learning models used are presented in the following sections. Subsequently, the layer generates output maps in the form of temperature heatmaps based on the model predictions, facilitating visual interpretation of spatial trends. Integral to the process, the Root Mean Square Error scores and output maps are rigorously validated to assess accuracy and inform re-calibration if discrepancies arise. This iterative approach ensures continual refinement and optimization of the models and outputs, with re-calibration prompting a return to the initial stage to refilter and process until desired results are obtained.

3.2.3.1 Anomalous Station Filtering

In this step, additional data filtering is performed based on quantile deviations for each of the urban features and temperature values. If we take a closer look at mean temperature variations across stations based on input features, we can see that some stations always deviate significantly from other stations as

shown in Figure 3.4. So we apply a systematic filter that flags such anomalous stations across all 24 hours, ensuring consistency and comprehensiveness in the analysis. Specifically, stations exhibiting outlier characteristics outside of [3,97] quantile ranges for respective hours are identified. This is then aggregated across all the hours in the present study and the top 4 outliers across all stations are systematically excluded from the dataset. This meticulous curation process not only enhances the quality of the dataset but also facilitates more nuanced insights into temperature variations and environmental influences across the study area. Based on different experiments, we observed that such data filtering enabled us to get rid of rural hot spots in the prediction map.

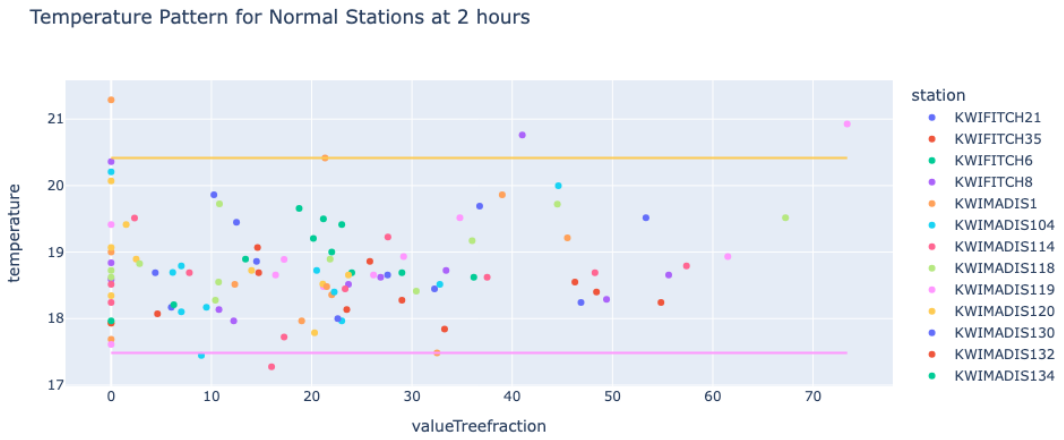

Figure 3.4: Stations Outside Quantile Range: Madison.

As a result, we get an even better representation of the diurnal temperature across stations, resulting in a robust temperature as shown in Figure 3.5.

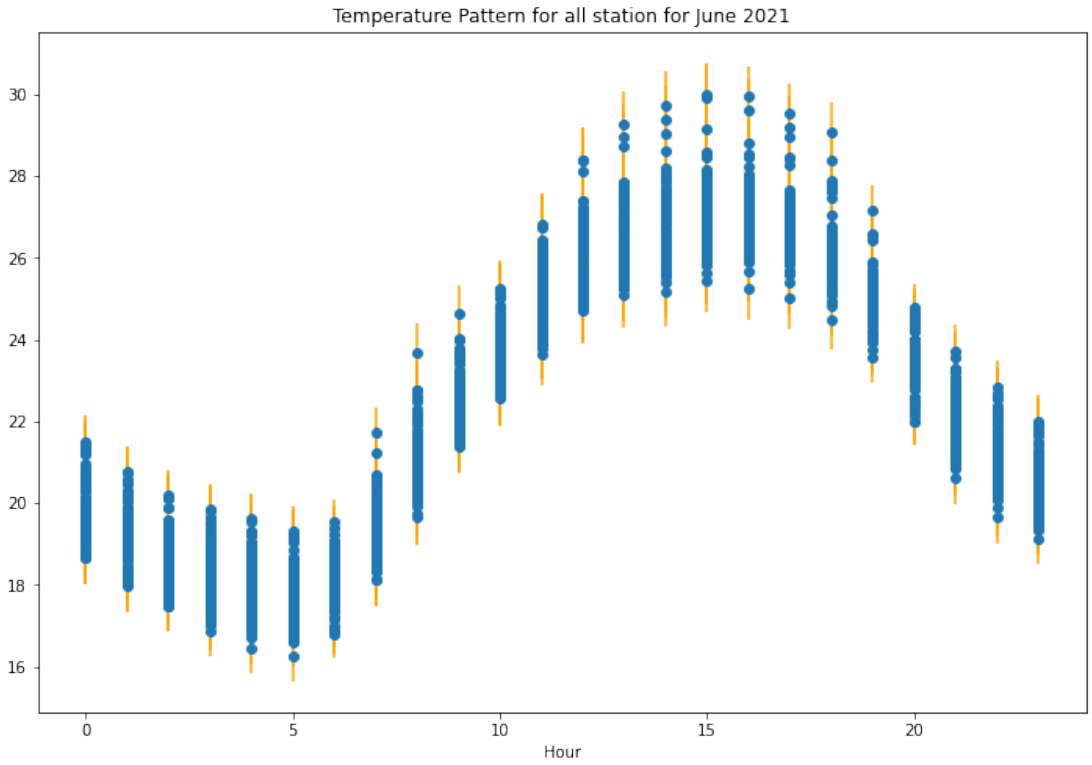

Figure 3.5: Temperature pattern after Quantile filtering.

3.2.3.2 Training Models

In this step, the output data from quantile filtering will be finally fed into different machine learning algorithms of choice. The following machine learning algorithms have been experimented within this study :

3.2.3.2.1 Random Forest

Random Forest is a versatile and powerful machine-learning algorithm that is widely used for both classification and regression tasks. It belongs to the ensemble learning family, which combines multiple base learners to make predictions.

Random Forest operates by constructing a multitude of decision trees during the training phase. Each decision tree is built using a subset of the features and a bootstrap sample of the training data, introducing randomness to the model. During prediction, the Random Forest aggregates the predictions of individual trees to arrive at the final output. This ensemble approach improves the robustness and generalization ability of the model, mitigating overfitting and capturing complex relationships in the data. Random Forest is known for its flexibility, scalability, and resistance to overfitting, making it a popular choice for various machine learning tasks across domains such as finance, healthcare, and natural language processing. A sample Random Forest is shown in Figure 3.6.

3.2.3.2.2 Gradient Boosting

Gradient Boosting Algorithm (GBM) is a powerful technique for machine learning, used for both regression and classification tasks. It is an ensemble method that combines predictions from many weak learners, like decision trees, to create a stronger learner. Unlike Random Forest, which uses independent trees in parallel, GBM builds trees sequentially. Each new tree learns from the errors of the previous trees. GBM minimizes a loss function by adding trees to the ensemble one at a time, where each tree corrects errors made by prior trees. This is explained further in Equation 3.7. By repeatedly minimizing the loss function, Gradient Boosting creates a more accurate learning model.

3.2.3.2.3 eXTreme Gradient Boosting (XGBoost)

XGBoost (Extreme Gradient Boosting) is a powerful and efficient machine

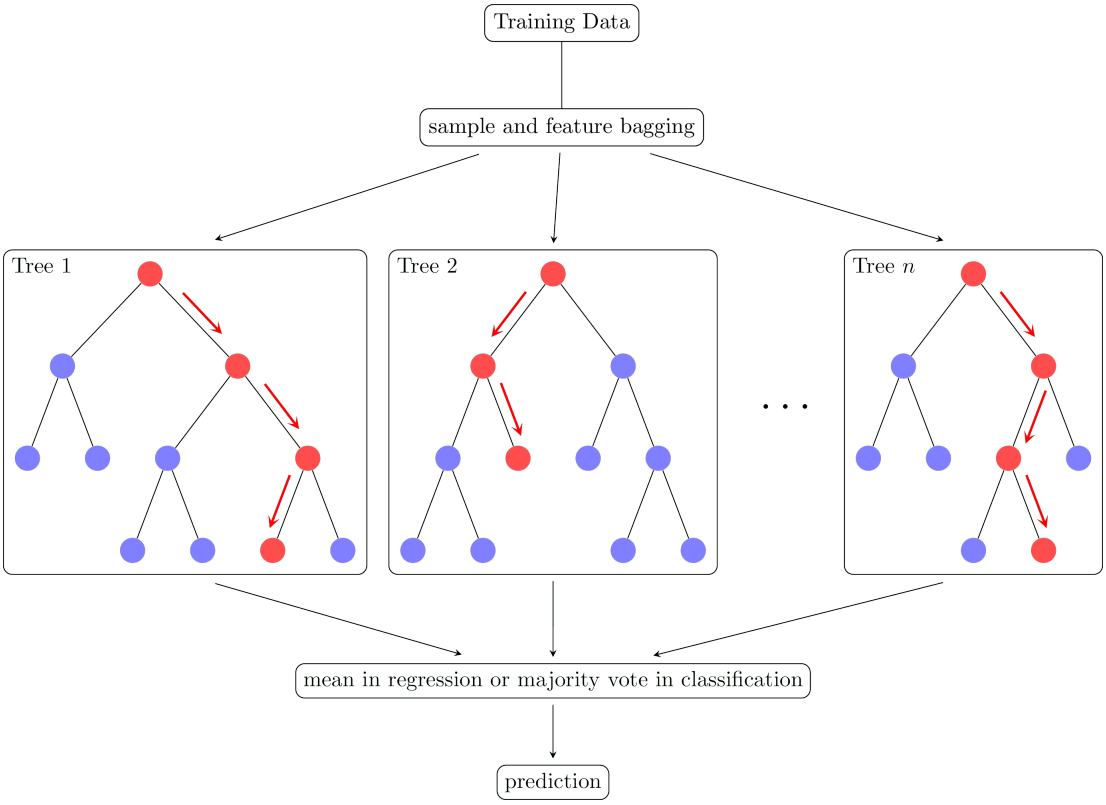

Figure 3.6: Random Forest Algorithm.
Source: `https://tikz.net/random-forest`

learning algorithm used for supervised learning tasks, especially in structured/tabular data scenarios. It belongs to the family of boosting algorithms, known for their high predictive performance and ability to handle large datasets. XGBoost iteratively builds a strong predictive model by combining the outputs of multiple weak learners, usually decision trees, to improve predictive accuracy as shown in Figure 3.8.

The basic algorithm for XGBoost works as follows :

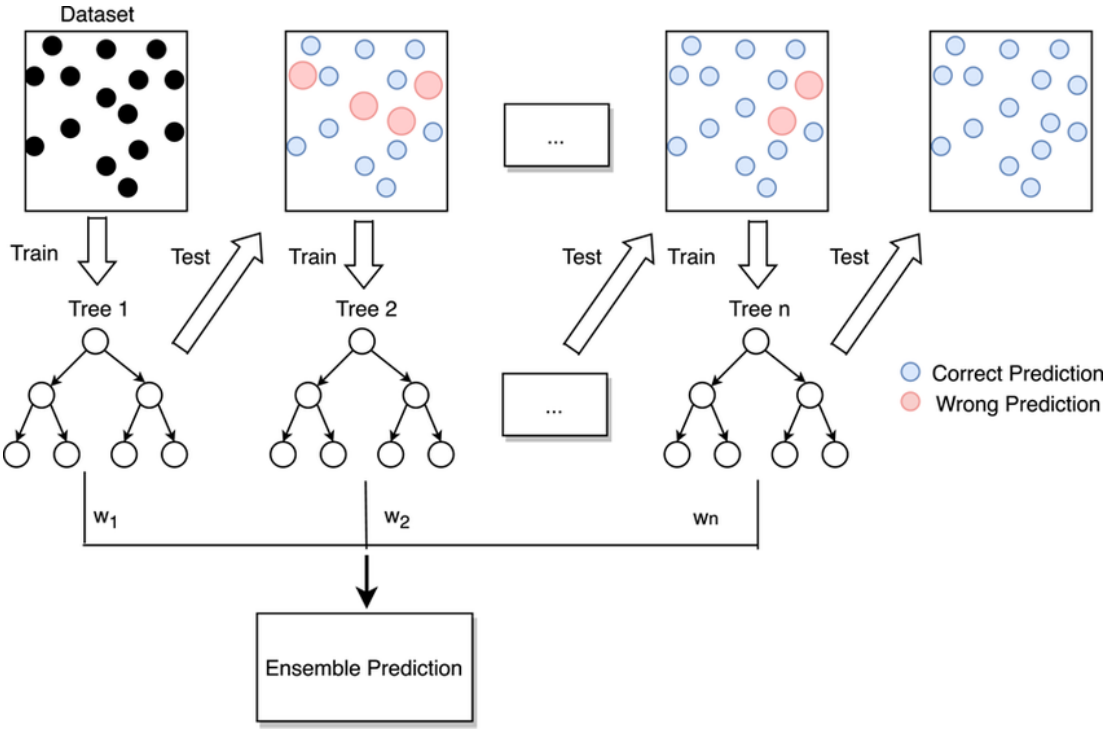

Figure 3.7: Gradient Boosting Algorithm.
Source: `https://www.researchgate.net/publication/351542039`

1. **Initialize with a constant value**: XGBoost starts by initializing the model with a constant value, usually the mean of the target variable for regression problems, or the logarithm of the odds ratio for binary classification problems.

2. **Iterative tree building**: XGBoost builds a series of decision trees sequentially, with each subsequent tree trying to correct the errors made by the previous ones. It uses a gradient-boosting framework where each tree is fitted on the residuals (the differences between the predicted and actual values) of the preceding trees.

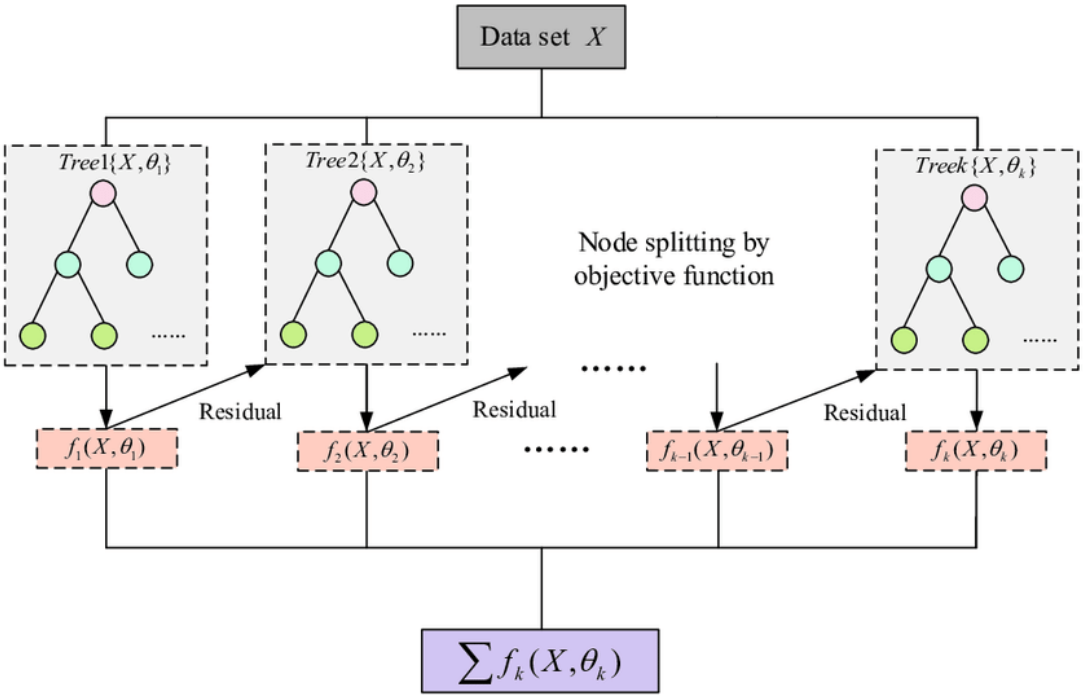

Figure 3.8: XGBoost Algorithm.
Source: `https://www.researchgate.net/publication/351542039`.

3. **Optimization of the objective function**: XGBoost optimizes a specific objective function, typically a combination of a loss function and a regularization term, to find the best split points in each decision tree and to prevent overfitting. The objective function guides the model to minimize prediction errors and complexity simultaneously.

4. **Pruning and regularization**: XGBoost incorporates techniques like pruning and regularization to prevent overfitting and improve generalization performance. Regularization parameters control the complexity of individual trees and the overall ensemble, helping to achieve a balance between bias and variance.

5. **Gradient descent optimization**: XGBoost uses a gradient descent optimization algorithm to minimize the objective function iteratively. It updates the model parameters in the direction that reduces the value of the objective function, gradually converging towards the optimal solution.

Overall, XGBoost's ability to handle complex interactions, deal with missing values, and optimize computational efficiency makes it a popular choice for a wide range of machine learning tasks, including classification, regression, and ranking problems. It has become a standard tool in many data science competitions and real-world applications due to its exceptional performance and scalability.

3.2.3.2.4 Artificial Neural Network

Artificial Neural Networks (ANNs) are computational models inspired by the structure and function of biological neural networks in the human brain. ANNs consist of interconnected nodes, called neurons, organized in layers. Each neuron performs a simple computation, and the network as a whole can learn complex patterns and relationships from data through a process called training.

Let's denote the input to the neural network as $\mathbf{x}$, and the output as $\hat{y}$. The neural network consists of multiple layers, including an input layer, one or more hidden layers, and an output layer. Each layer l contains $n^{[l]}$ neurons, and the output of neuron j in layer l is denoted as $a_j^{[l]}$.

The computation in each neuron is represented by the following equations:

$$z_j^{[l]} = \sum_{i=1}^{n^{[l-1]}} w_{ij}^{[l]} a_i^{[l-1]} + b_j^{[l]}$$
$$a_j^{[l]} = g(z_j^{[l]}),$$

where $w_{ij}^{[l]}$ is the weight connecting neuron i in layer $l-1$ to neuron j in layer l, $b_j^{[l]}$ is the bias of neuron j in layer l, and $g(\cdot)$ is the activation function.

The output of the neural network is computed as $\hat{y} = a_1^{[L]}$, where L is the index of the output layer.

The training of neural networks is typically performed using an algorithm called backpropagation, which involves the following steps:

1. **Forward Propagation**: Compute the output of the network for a given input by propagating the input forward through the network.

2. **Compute Loss**: Calculate the difference between the predicted output $\hat{y}$ and the actual output y, using a suitable loss function.

3. **Backward Propagation**: Compute the gradients of the loss function with respect to the weights and biases of the network using the chain rule of calculus.

4. **Update Parameters**: Update the weights and biases of the network in the opposite direction of the gradients to minimize the loss function using optimization algorithms such as gradient descent.

5. **Repeat**: Repeat the process with different input samples until the network converges to a satisfactory solution.

The success of artificial neural networks in various applications, including image recognition, natural language processing, and pattern recognition, has made them one of the most widely used machine learning techniques today.

3.2.3.3 Preparing Raster Outputs

Once the model is trained, the goal is to predict the temperature for each pixel of size 70 meters in the entire domain. Since our output image resolution is set at 681*681 pixels, this accounts for a total of 463,761 predictions. We generate one such image for each hour, resulting in 24 images of size 681*681 pixels. This is accomplished in two steps. At first, the visualization layer creates a prediction table for each unique latitude and longitude value of the entire domain. Once the table is calculated, we create an empty image array of required dimensions. Then the entire prediction matrix will be mapped to each pixel in the map one by one. Once the pixel mapping is completed, the final output map is plotted as in Figure 3.9.

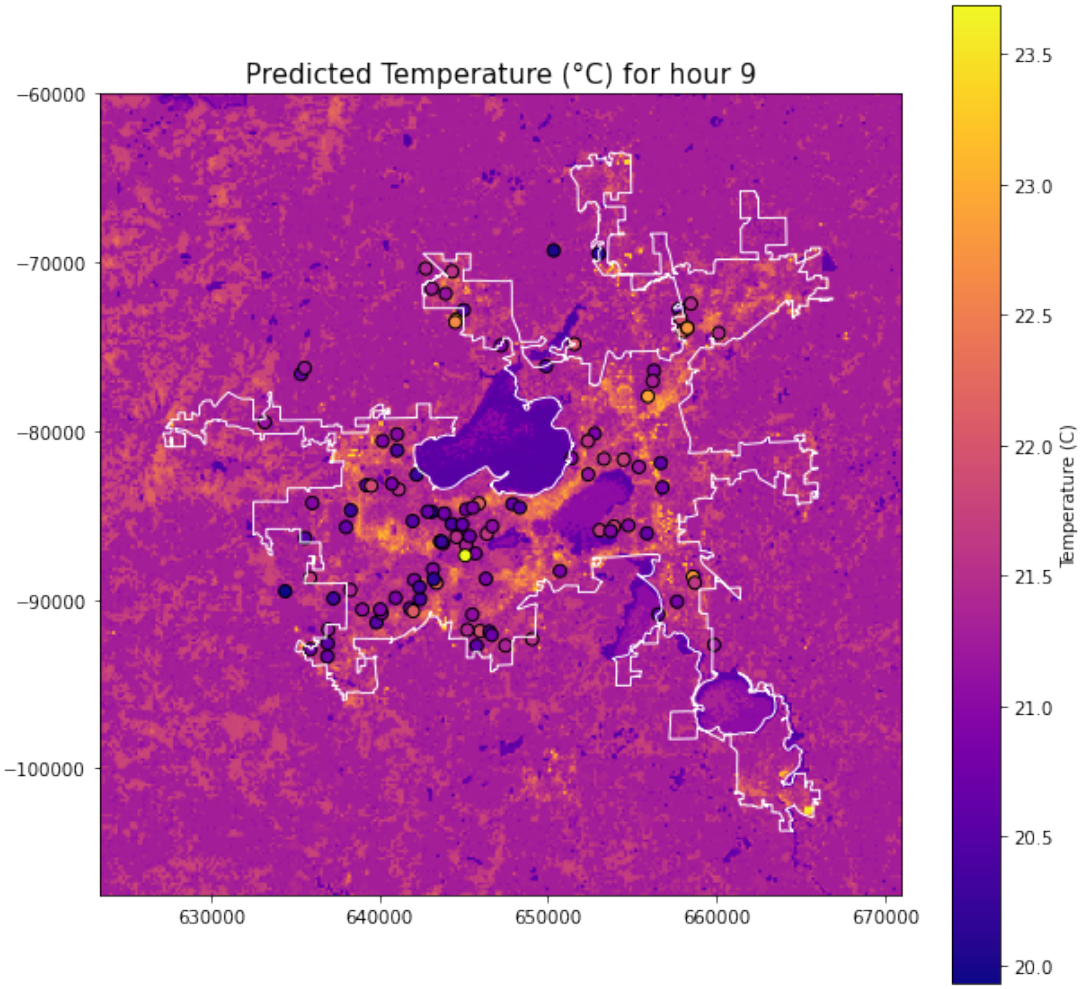

Figure 3.9: Sample Output Domain Map for Madison.

Chapter 4. Experiments and Results

In this section, we provide a complete comparative analysis of various machine learning models used in our study to predict temperature values. Each model was rigorously trained and evaluated on the monthly aggregated dataset. Except as otherwise stated, the results reported herein are specific to Madison City in June 2021. Additional thoughts and deductions are given in the appendix section. Each model's performance was evaluated using the Root Mean Square Error (RMSE), which provided crucial insights into its effectiveness and applicability within the context of our research.

The standard training procedure for all the algorithms tested works as follows :

1. Fetch aggregated training data for a given location, month, and year.

2. Split the dataset for training and testing in a 4:1 ratio. Furthermore, the train data would be split for cross-validation. Additionally, for different algorithms, the train-test split is done with the same random seed for consistent result comparisons.

3. Plot the RMSE diagram, average temperature diagram, and feature importance diagram (if available) and readjust the data processing pipeline based on this.

4. Run predictions for the entire domain and visualize the raster diagrams for 24 hours for a visual check on the quality of results.

4.1 Linear Regression

We use the aggregated dataset to construct a simple Linear Regression model to act as a baseline. This would give us a good starting point before approaching more complex methods. The average RMSE score is 0.72. The results of this step can be summarized in Fig 4.1 and Fig 4.2.

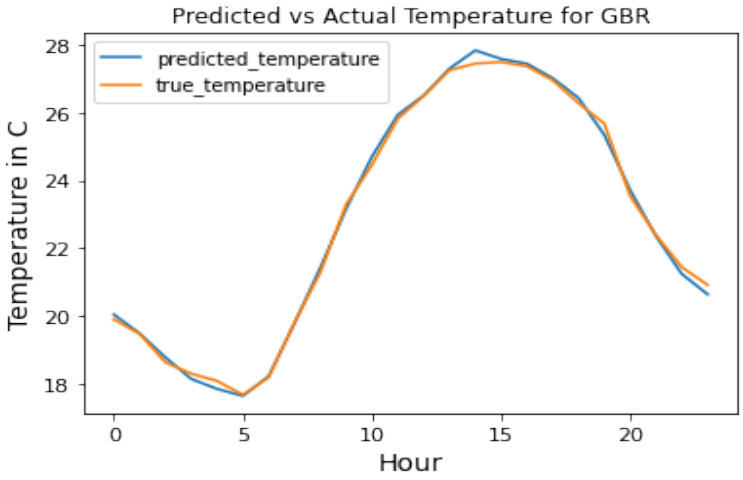

Figure 4.1: Linear Regression: Predicted Vs Actual Temperature.

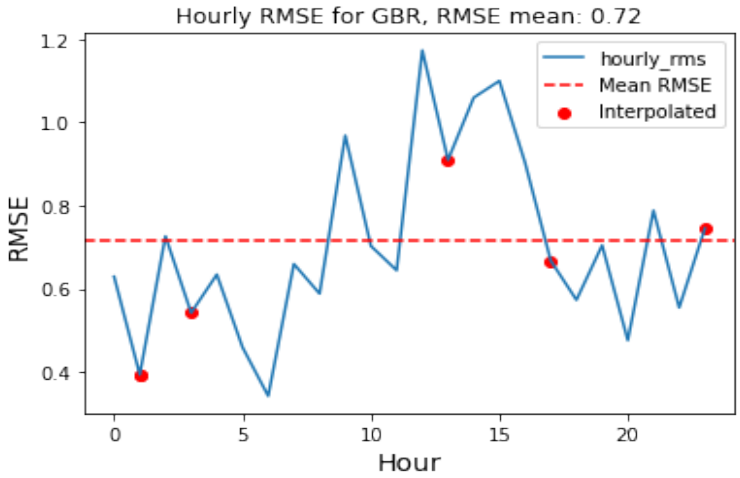

Figure 4.2: Linear Regression: Hourly Root Mean Square Error.

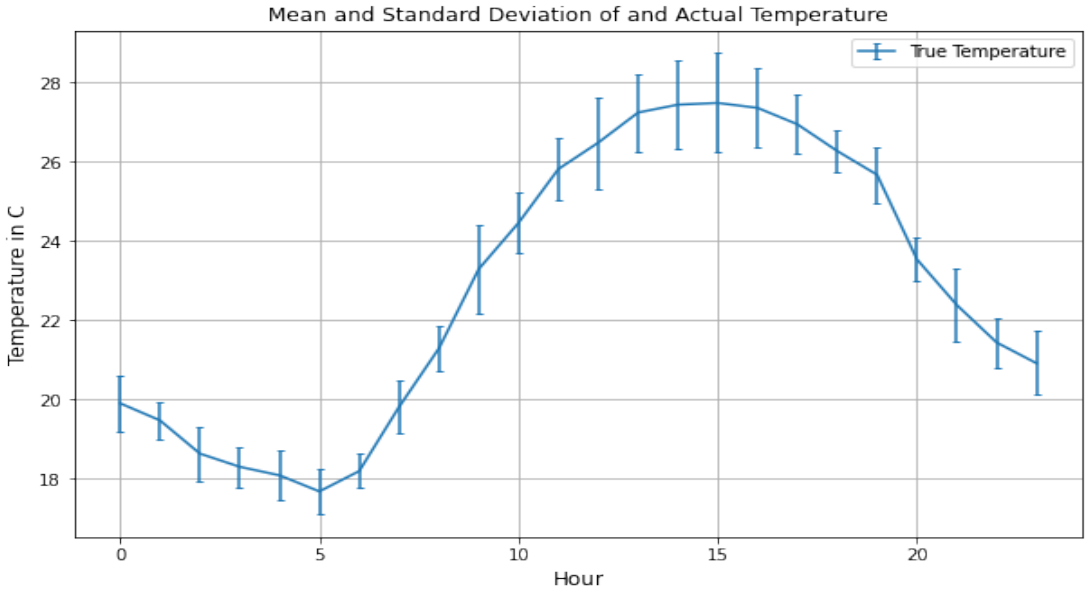

Figure 4.3: Hourly Deviation in True Temperature.

From Figure 4.2 it is evident that daytime hours have a higher error score on average. To confirm that this is not a modeling issue, we plotted the standard deviation in temperature for each hour as in Figure 4.3, which validates that the true temperature for the Madison region itself has a higher deviation in peak daytime hours.

4.2 Random Forest

In this step, we proceed with further experimentation on Random Forest. It is evident from the findings presented by Han Wang *et al.* [26] that tree-based models such as Random Forest work well with spatial interpolation. As expected we get a much better RMSE value as shown in 4.4. If we look at the feature importance plot in Figure 4.5, we can see that features such as Impervious fraction, tree fraction, and closest station temperature have a higher impact on prediction values as expected.

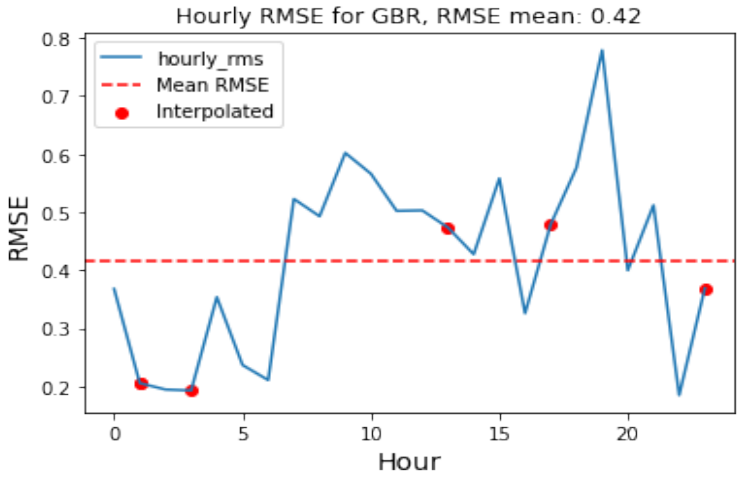

Figure 4.4: Random Forest: Hourly Root Mean Square Error.

If we look at the feature importance plot in Figure 4.5, we can see that features such as Impervious fraction, tree fraction, and closest station temperature have a higher impact on prediction values as expected. Similarly, as expected from our Residual Temperature design, hour as a feature has a weaker impact as discussed in Section 3.2. However, the random forest predictions fail to incorporate the spatial patterns of temperature when we look at the generated domain map in Figure 4.6. Ideally during the night, the relative temperature of water bodies is supposed to be warmer compared to the landmass, which is not observed in this case.

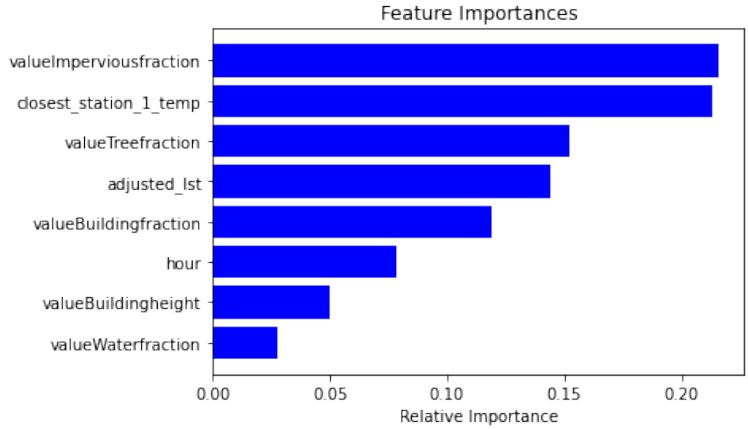

Figure 4.5: Random Forest: Feature Importance Diagram.

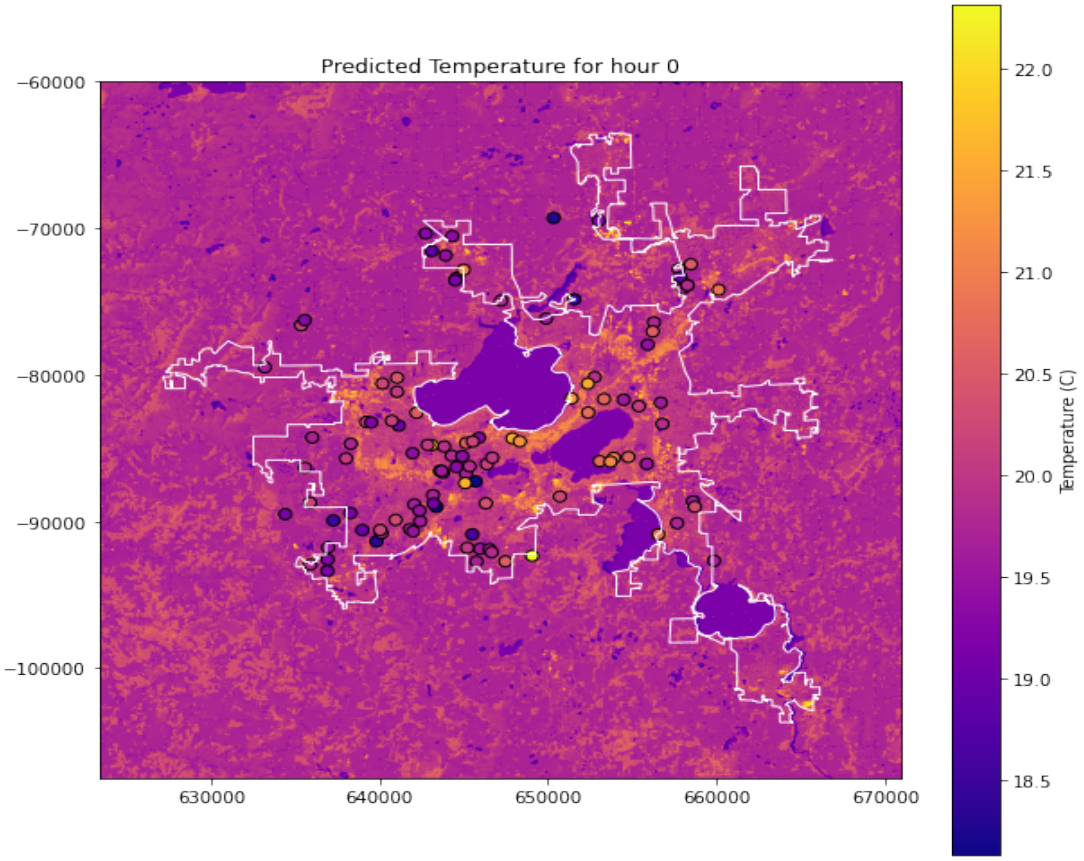

Figure 4.6: Random Forest: Output Map.

4.3 Gradient Boosting Regression

Gradient Boosting is a type of ensemble learning method that combines the predictions of several weak learners, typically decision trees, to create a strong learner. Unlike Random Forest, this model builds upon the previous trees and corrects the errors at each step, enabling the models to capture complex patterns in the data.

The overall RMSE, feature importance values, and output map are shown in Figure 4.7, 4.8, and 4.9 respectively.

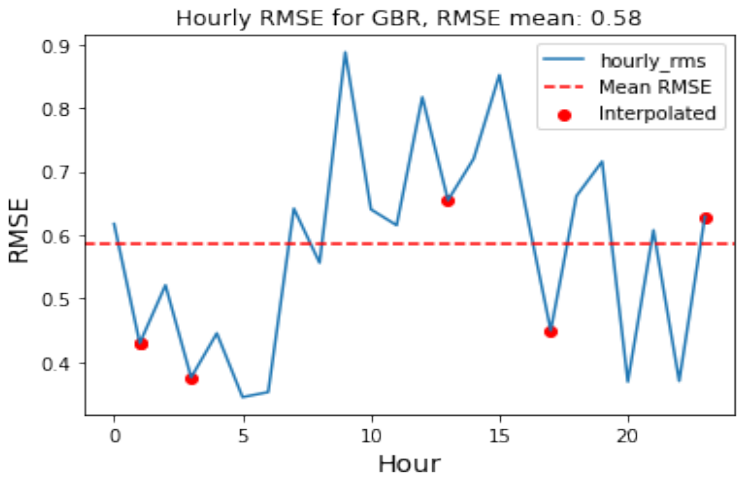

Figure 4.7: Gradient Boosting: Hourly Root Mean Square Error.

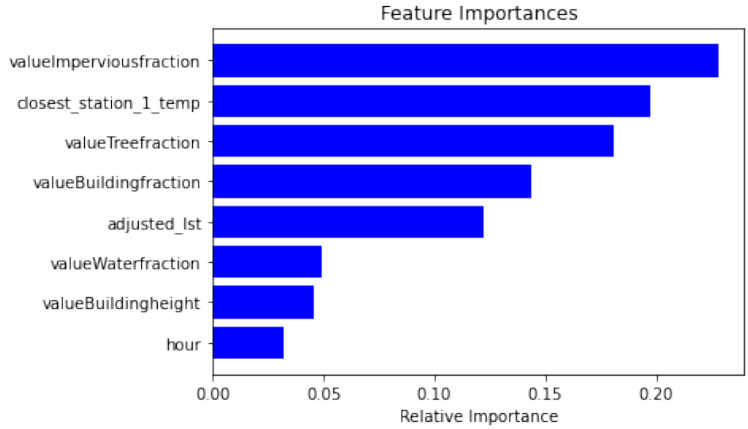

Figure 4.8: Gradient Boosting: Feature Importance Diagram.

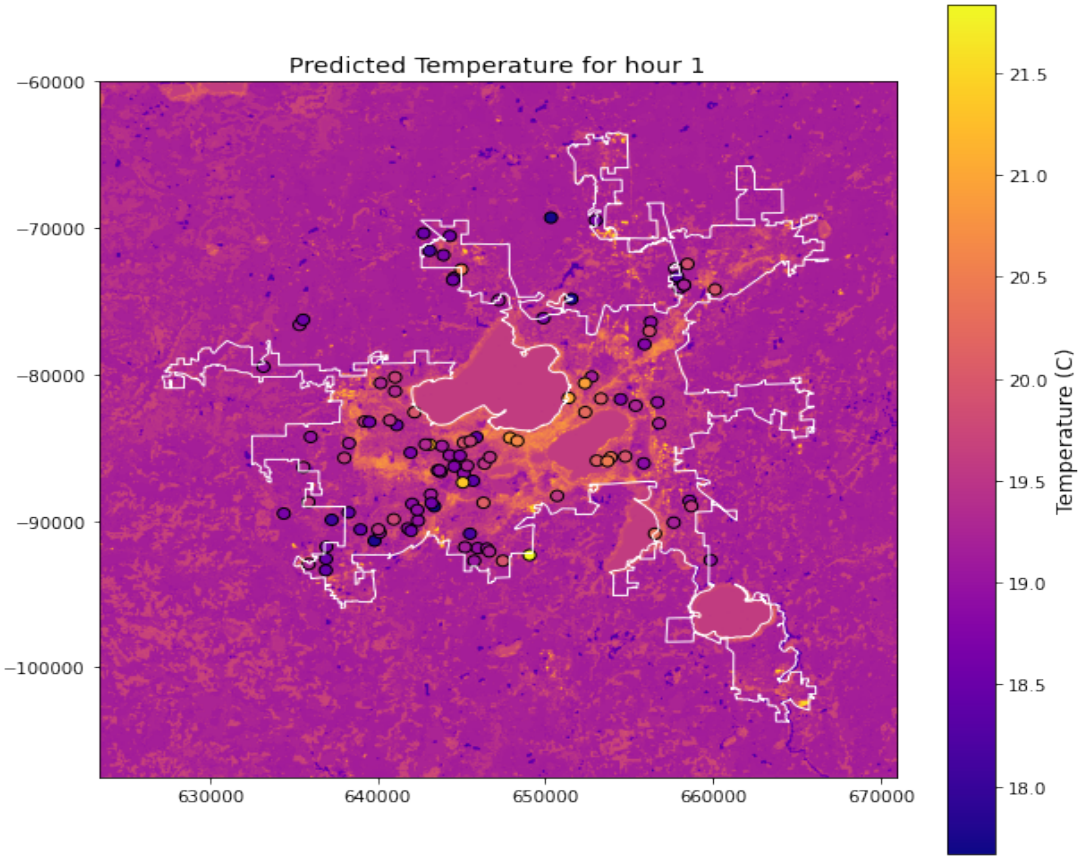

Figure 4.9: Gradient Boosting: Output Map - 1 am.

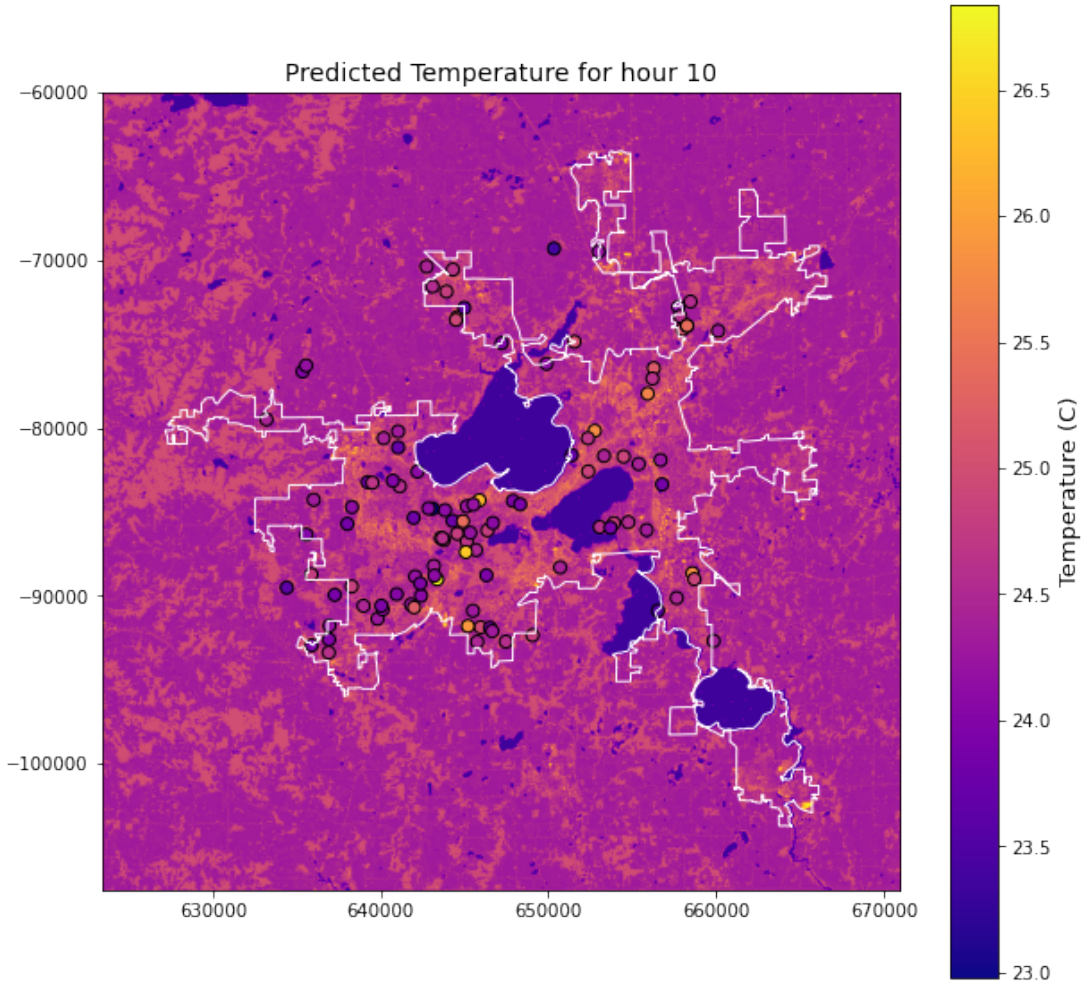

Figure 4.10: Gradient Boosting: Output Map - 10 am.

If we analyze the pictures in Figure 4.9 and 4.10, we can see that although the land-water temperature contrast is perfectly captured for Gradient boosting, certain hours around 8-11 am have a lot of hotspots in rural areas. This marks possible areas for improvement and alternative architectures as proposed in the following sections.

4.4 XG Boost

XGBoost, short for Extreme Gradient Boosting, is a highly efficient and scalable implementation of gradient boosting. It builds upon the principles of gradient boosting by employing a more regularized model and enhancing its performance through parallel computing.

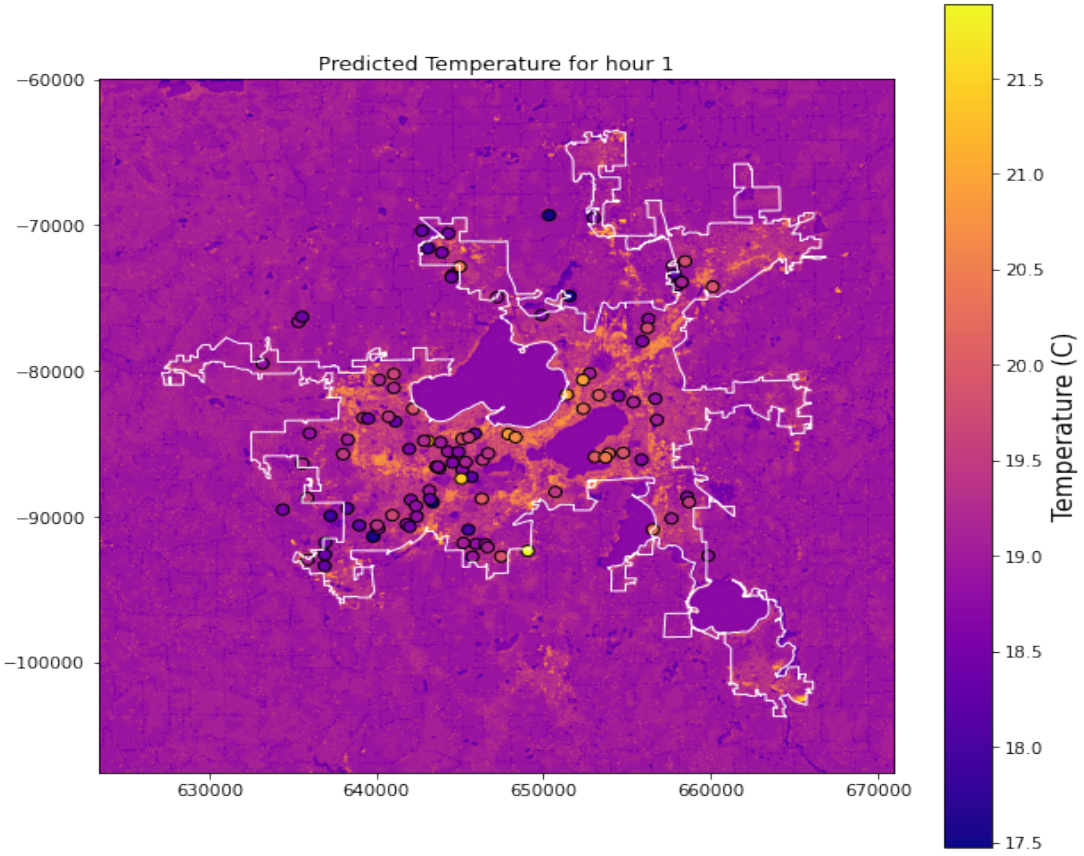

Figure 4.11: XGB: Output Map - 1 am.

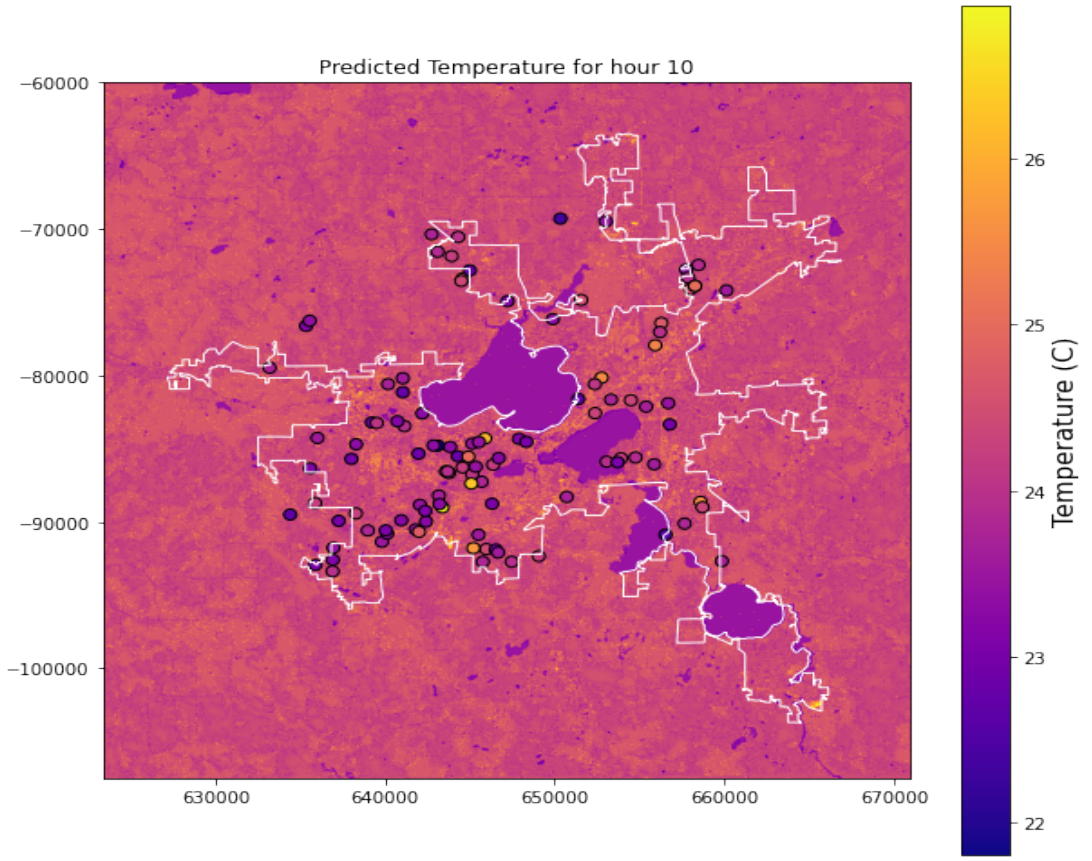

Figure 4.12: XGB: Output Map - 10 am.

Similar to Gradient Boosting, XGBoost iteratively improves the performance of weak learners to create a robust predictive model. The hourly RMSE errors for XGBoost are comparable to Gradient Boosting. However, from the generated domain maps in Figure 4.11, and Figure 4.12, we can see that this algorithm is regularizing the pixels with higher temperatures to remove the hotspots. However, this causes an undue influence on the relative temperature difference of landmass and water bodies which is specifically prominent during night hours.

4.5 Hybrid Architecture

From multiple experiments on GB and XGB variations, it was evident that a single model doesn't work perfectly for all 24 hours. It was also evident in the findings of Hjort *et al.* [10] for spatial predictions that the same tree models performed differently based on the period. Given the nature of the results, it is easy to infer that the models are either being highly regularized, resulting in loss of warm signals in the domain map such as Figure 4.11, or overfitting and resulting in significant rural hotspots as seen in Figure 4.10. Overestimation of low temperatures and underestimation of high temperatures have also been a known issue with tree-based models as reported in several studies [32]. Similarly, the different data filtering mechanisms during our ingestion and processing layer also act as external regularizers, further explaining the possible reasons why the XGB models are underestimating warmer surfaces.

Taking all of this into account, we created a new architecture as shown in Figure 4.13. We propose this architecture based on the assumption that the regularized Gradient Boosting (with filtered data) will penalize the hotspots. In contrast, the unregularized Gradient Boosting will accurately map the water surface temperature. As a result, the average model should perform well. We actively refrain from using more than two models as the ensemble approach that averages multiple base models unintentionally removes sharp signals [26]. This vastly improves the scalability of the entire pipeline and enables us to easily utilize this architecture for different regions and different periods shown in the Appendix.

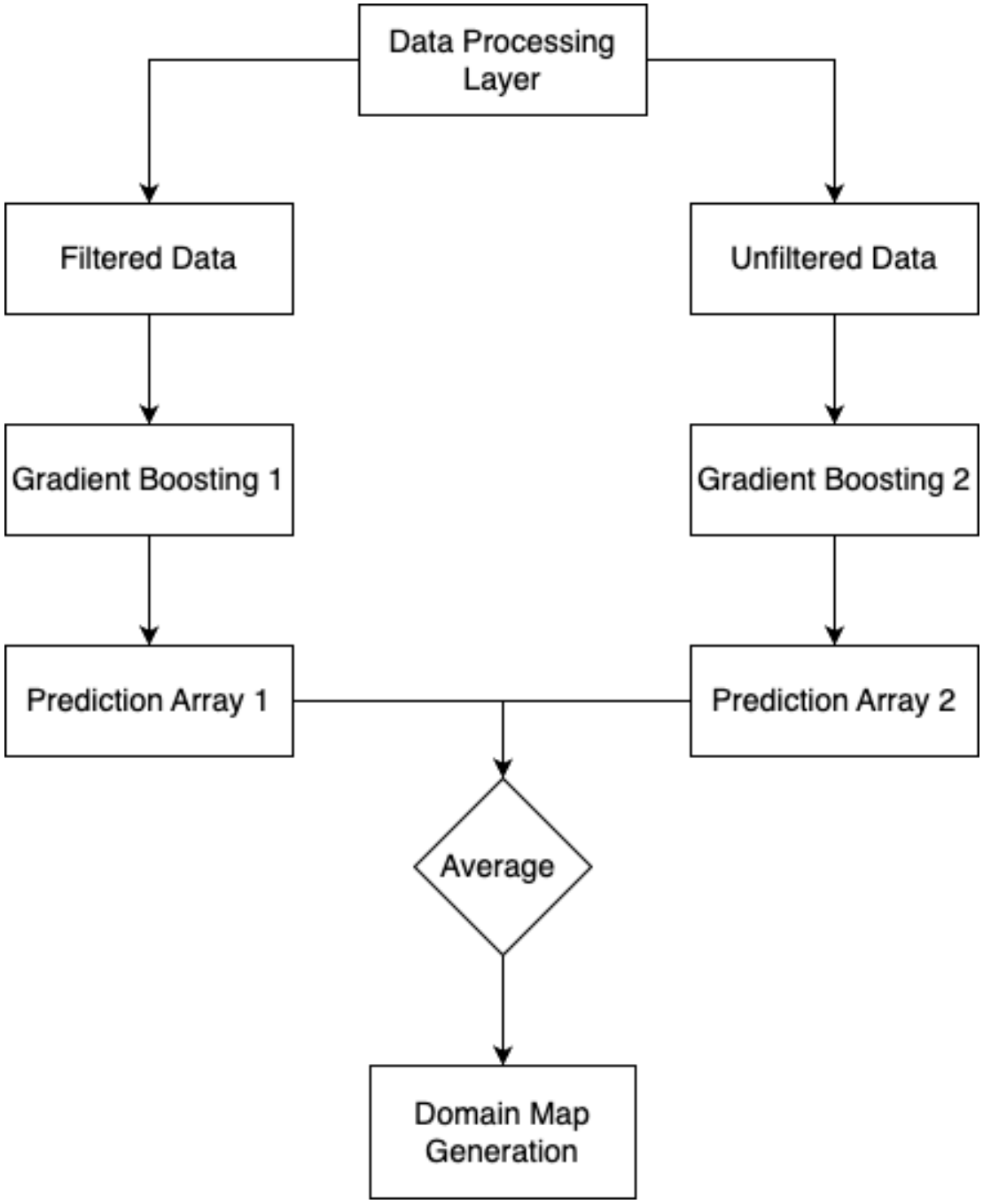

Figure 4.13: Hybrid Training Architecture.

The overall results of this novel approach can be summarized with the overall RMSE plot and domain map, which are shown in Figure 4.14 and the corresponding domain plots.

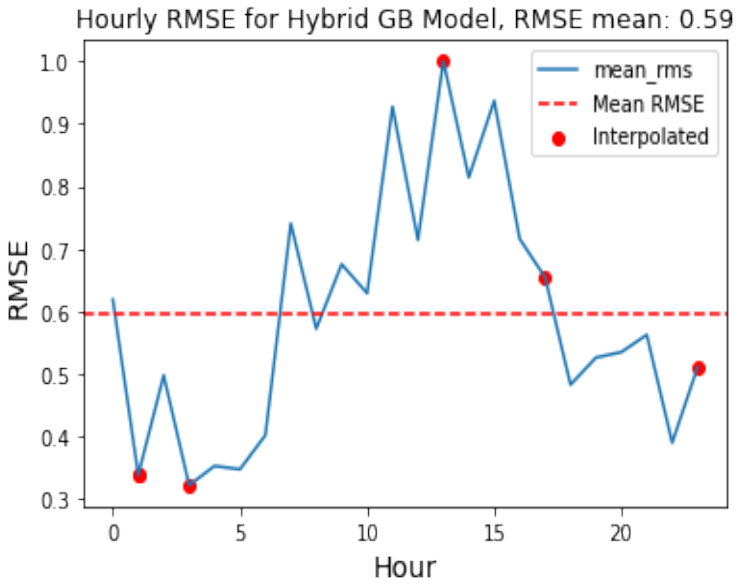

Figure 4.14: Hybrid GB: Hourly Root Mean Square Error.

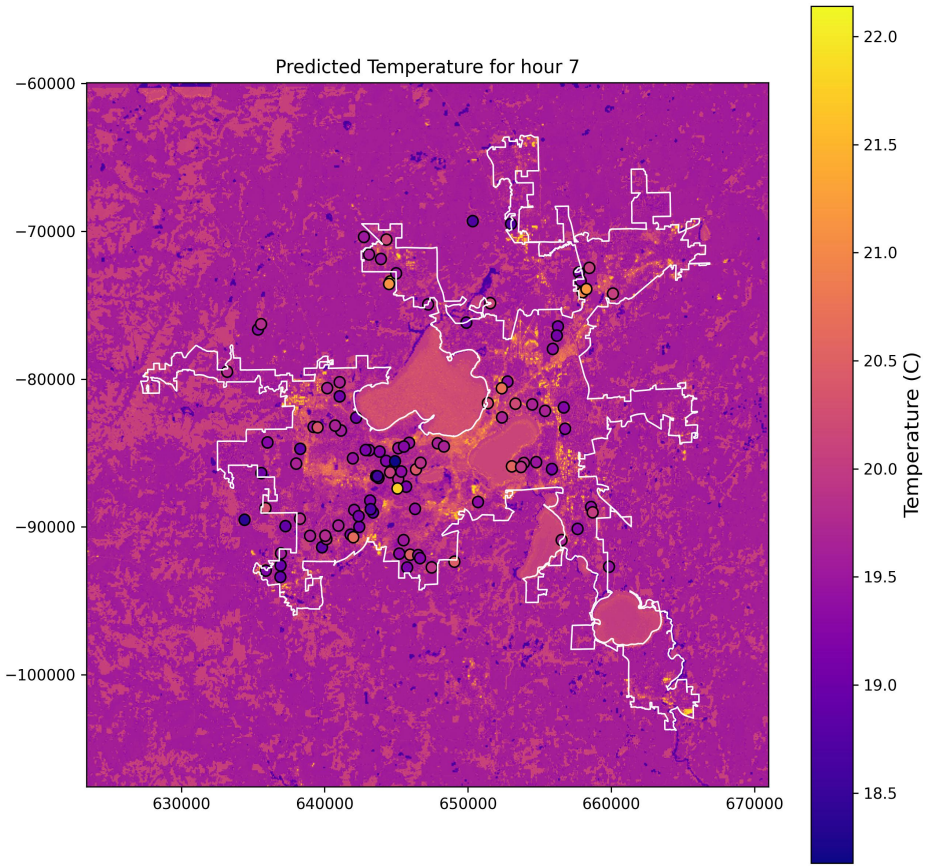

Figure 4.15: Hybrid: Output Map - 7 am.

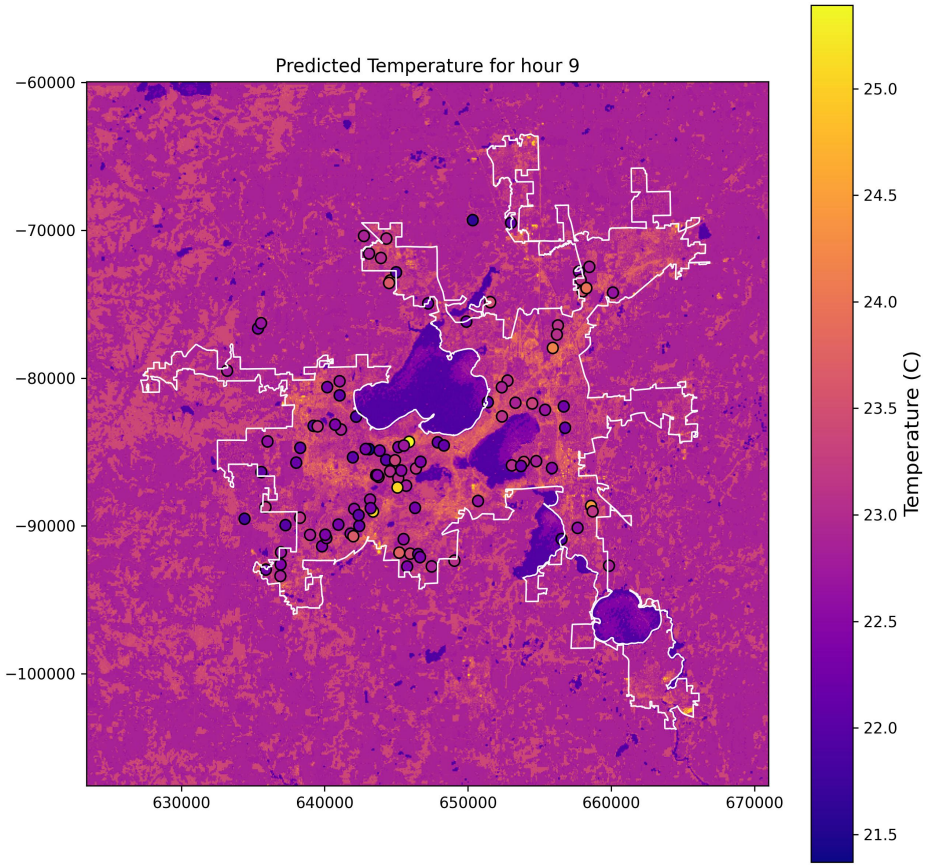

Figure 4.16: Hybrid: Output Map - 9 am.

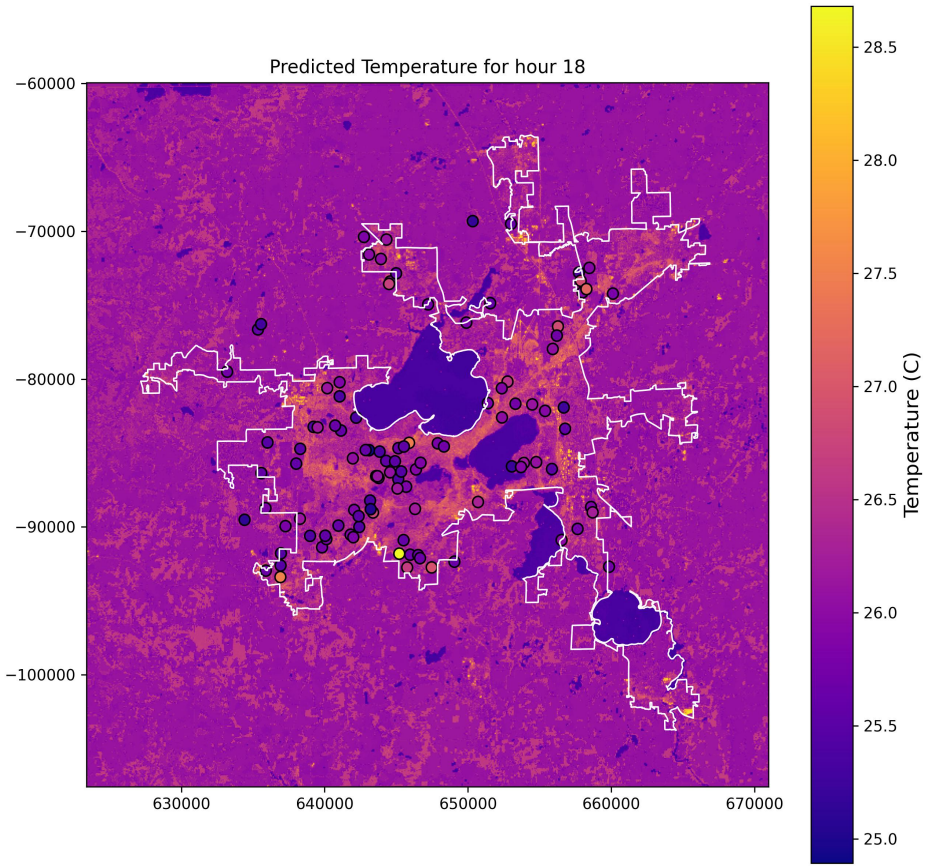

Figure 4.17: Hybrid: Output Map - 6 pm.

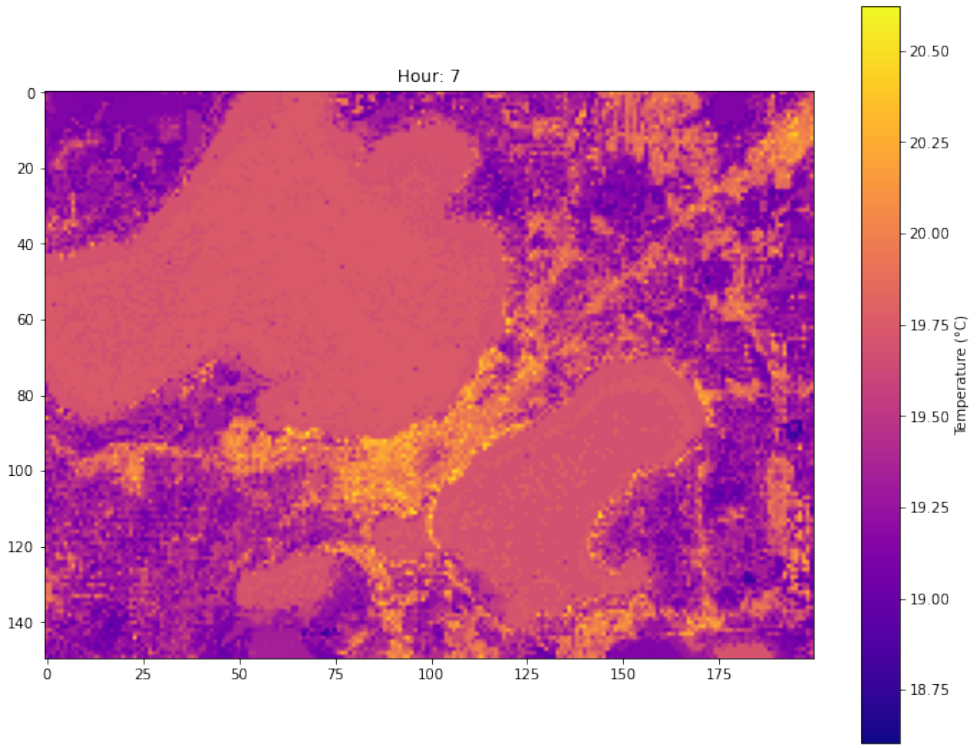

Figure 4.18: Hybrid: Zoomed Map - 7 am.

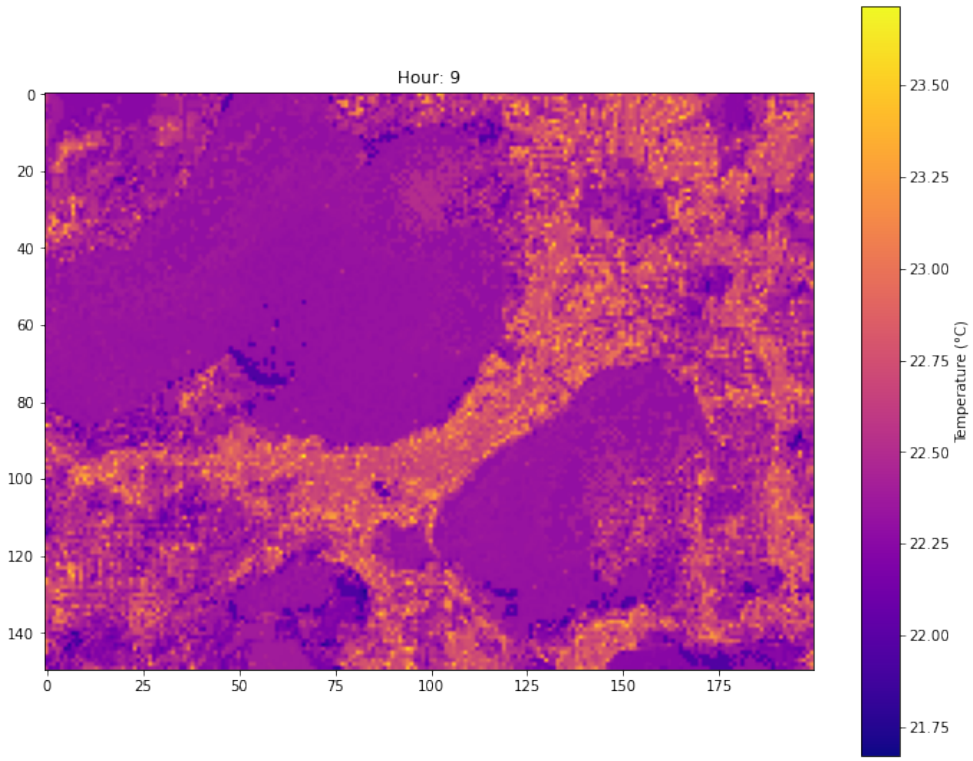

Figure 4.19: Hybrid: Zoomed Map - 10 am.

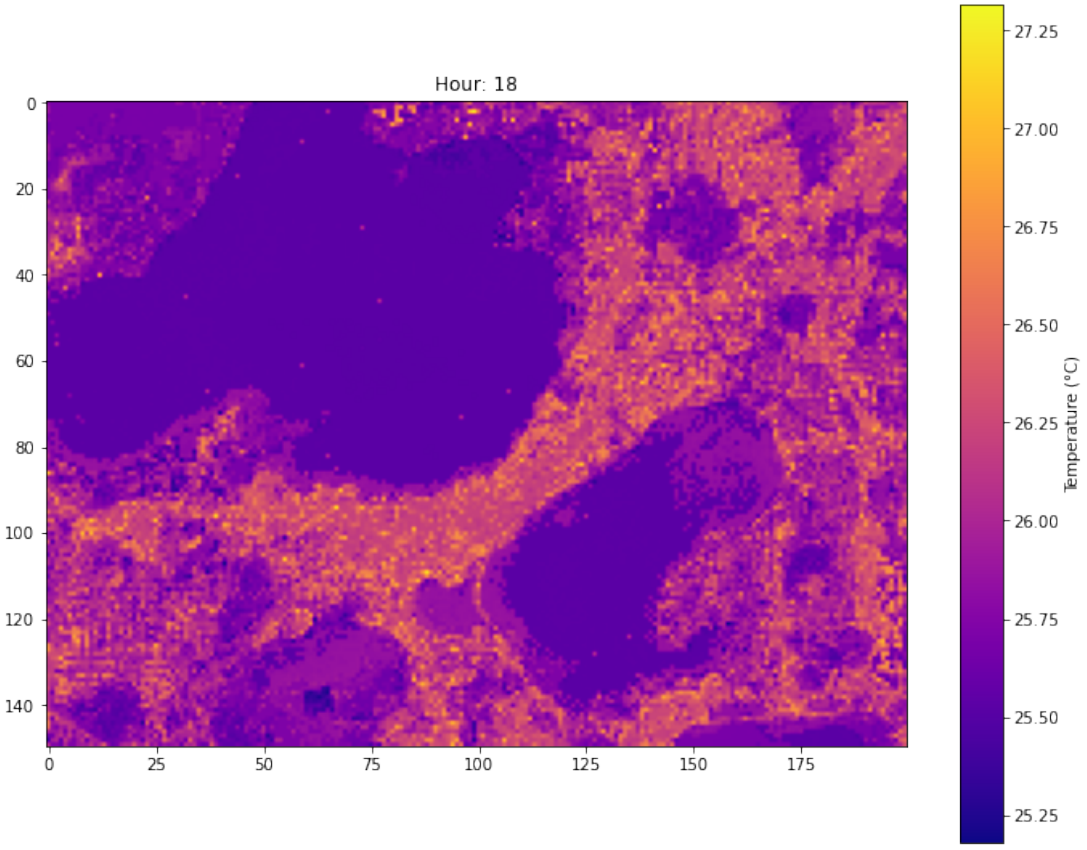

Figure 4.20: Hybrid: Zoomed Map - 3 pm.

A comprehensive comparison of RMSE values using this hybrid approach for the Years 2021 and 2022, for the Las Vegas and Madison Region is shown in Figure 4.1. Further results for the Las Vegas region and other months of Madison are available in the Appendix Section.

Table 4.1: Result Summary.

Location	**Year**	**Month**	**RMSE Mean**	**RMSE Max**	**RMSE Min**
Madison	2021	June	0.632641	1.20058	0.367665
Madison	2021	July	0.495472	0.874031	0.312791
Madison	2021	August	0.594227	1.014628	0.350095
Madison	2022	June	0.627906	1.103264	0.33696
Madison	2022	July	0.509939	0.898639	0.35196
Madison	2022	August	0.590122	0.987345	0.328213
LasVegas	2021	June	0.911951	1.523782	0.59236
LasVegas	2021	July	0.864852	1.471548	0.468812
LasVegas	2021	August	0.853442	1.386684	0.597742
LasVegas	2022	June	0.978681	1.335064	0.671556
LasVegas	2022	July	0.940272	1.426065	0.621941
LasVegas	2022	August	0.846949	1.337896	0.560355

In the summary paper by Wang et al. [26], it was seen that the average mean RMSE at hourly scale is 1.75°C, with the mean RMSE being 1.20°C for all models for 10-100m resolution level. From Table 4.1, we can see that our mean RMSE is always less than 1°C for all of the regions across multiple years and months. This proves that the architecture we developed is scalable across regions and beats the benchmark.

4.6 Artificial Neural Networks

At first, GridSearch and Bayesian Hyperparameter search strategies were completed to find the optimal number of hidden layers and neurons per layer, resulting in a Deep Neural Network of size [20,30,35,15]. The overall RMSE values were comparable to the previous results as shown in Figure 4.21. However, the domain raster images for daytime hours lacked a lot of visible clarity achieved by previous methods and the nighttime temperature patterns for the water surface also showed contradicting patterns as shown in Figure 4.22. Additionally, the overall difficulty of searching for optimal hyperparameters compared to other methods adds an extra layer of complexity to generating output images. The complexity can be inferred from the fact that a good RMSE doesn't equate to a good domain image, based on our previous experiments. Hence, we need to sample output domain images at multiple iterations within the solution space for each hour, which makes it complicated. Hence, the hybrid Gradient boosting approach can be established as the best-performing model by comparison. Previous

studies also point to the fact that tree-based models are highly favored in spatial predictions compared to neural networks [26].

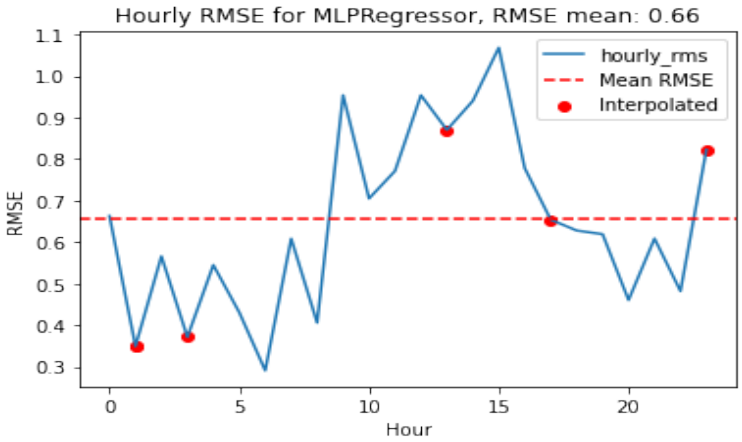

Figure 4.21: RMSE values for Neural Networks.

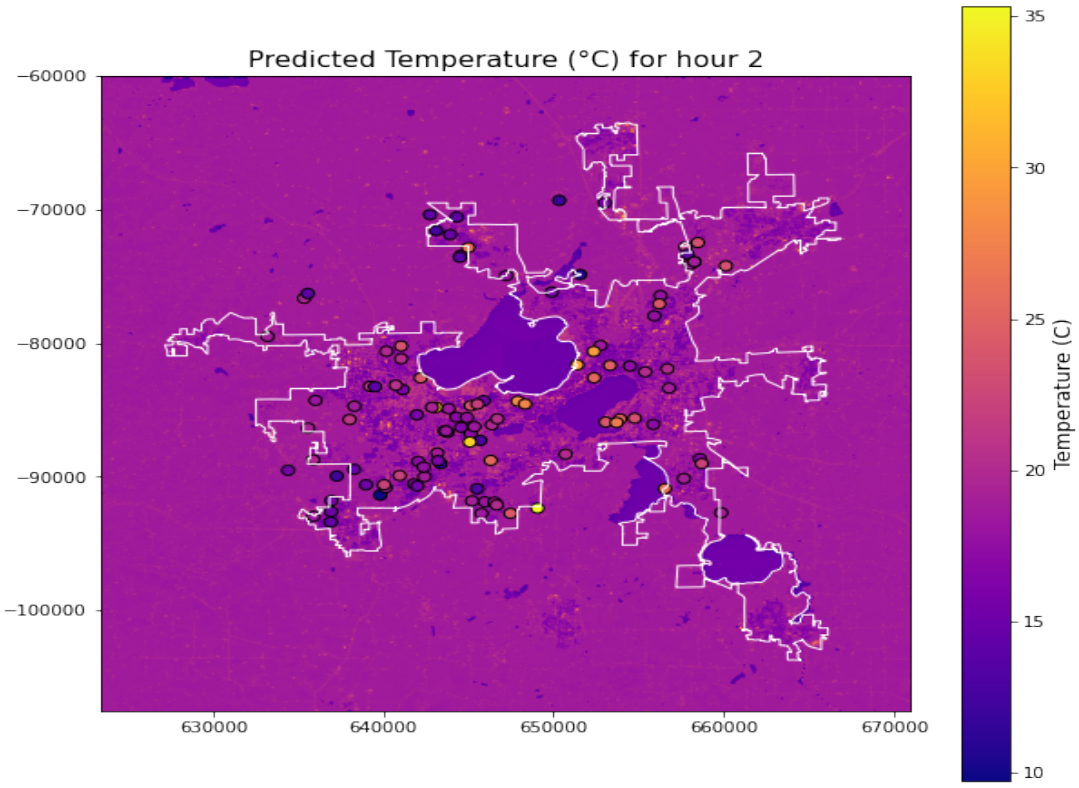

Figure 4.22: Neural Network: Output Map - 2 am.

Chapter 5. Conclusions and Future Work

The research presented in this thesis demonstrates the feasibility and effectiveness of using a comprehensive, multi-source data integration approach for ambient temperature prediction for the diurnal map in urban environments for 70-meter resolution levels at monthly levels for June, July, and August across multiple years and multiple US cities (*i.e.*, Madison and Las Vegas). The findings indicate that the proposed data architecture, coupled with anomaly removal and machine learning models, significantly improves the accuracy of temperature predictions compared to previous studies. We compared different ML algorithms and found Gradient Boosting architecture to be the best model with a mean RMSE of 0.58°C for the Madison region and 1.89°C for the Las Vegas Region, which is significantly better than the previous studies for similar resolution levels. This improvement in error scores can be attributed to our extensive data cleaning mechanism, choice of ECOSTRESS-based LST observations, and the hybrid model architecture that we use for modeling. Additionally, the faster computational speed of tree-based models presents a strong case for their favorability across different regions, compared to neural networks which are computationally more expensive. The feature importances of urban surface properties such as impervious and tree-fraction were relatively higher, indicating the possibil-

ity of experimenting with more spatial properties as input features. LST was lower in the feature ranking as expected, verifying the underlying assumption that it is not the strongest contributor to ambient temperature as mentioned in the previous studies. Additionally, the higher ranking of our feature-engineered variable - "Closest Station Temperature" indicates that there is a good possibility of improving further by including more geospatially relevant features. There are also additional opportunities to optimize the overall method by combining other sources of data and further research on geospatial methods such as Geospatial Regression, Geospatial weighted Neural Networks, and Deep Learning algorithms. We briefly explored the possibility of implementing Geospatial Neural Networks, however, the lack of well-tested packages in Python hindered further progress given the timeline, which can be another significant research contribution in this domain.

The study's success in diverse regions *i.e.*, Madison and Las Vegas underscores the potential applicability across various urban settings, offering valuable insights for urban climate studies, heatwave prediction, and urban planning strategies. This research paves the way for more precise and reliable urban temperature modeling and provides a scalable framework that can be adapted to diverse regions. Future extensions could include combining the results of this study with other spatial predictions such as pollution, moisture, heat index, etc. to get a complete picture of urban climate conditions. The high-resolution domain maps could be used in conjunction with socio-economic data for urban regions to ex-

pand further into the domain of environmental justice studies and understand how vulnerable communities are affected by extreme climatic conditions.

References

[1] NLCD 2019 Land Cover (CONUS) — Multi-Resolution Land Characteristics (MRLC) Consortium — mrlc.gov. `https://www.mrlc.gov/data/nlcd-2019-land-cover-conus`. [Accessed 09-02-2024].

[2] Federico Amato, Fabian Guignard, Sylvain Robert, and Mikhail Kanevski. A novel framework for spatio-temporal prediction of environmental data using deep learning. 2020.

[3] A J Arnfield. Two decades of urban canopy meteorology: A review of progress and its implications for designers. *Building and Environment*, 38(6):659–689, 2003.

[4] Carlos Bartesaghi-Koc, Paul Osmond, and Alan Peters. Innovative use of spatial regression models to predict the effects of green infrastructure on land surface temperatures. *Energy Build.*, 254(111564):111564, January 2022.

[5] M A Brown and T R Oke. The urban boundary layer over a low-rise suburban landscape. *Quarterly Journal of the Royal Meteorological Society*, 116(491):1079–1107, 1990.

[6] A M Coutts, N J Tapper, J Beringer, C P Loughner, and P C Livesley. Street tree shade and street canyon air temperature: Evidence from field measurements in melbourne, australia. *Landscape and Urban Planning*, 132:256–262, 2014.

[7] Daniel Fenner, Benjamin Bechtel, Matthias Demuzere, and Fred Meier. Crowdqc+—a quality-control for crowdsourced air-temperature observations enabling world-wide urban climate applications. *Frontiers in Environmental Science*, 2021.

[8] Daniel Fenner, Fred Meier, Dieter Scherer, and Albert Polze. Spatial and temporal air temperature variability in berlin, germany, during the years 2001–2010. *Urban Clim.*, 10:308–331, December 2014.

[9] Elizabeth Jane Good. An in situ-based analysis of the relationship between land surface "skin" and screen-level air temperatures. *J. Geophys. Res.*, 121(15):8801–8819, August 2016.

[10] Jan Hjort, Juuso Suomi, and Jukka Käyhkö. Spatial prediction of urban–rural temperatures using statistical methods. *Theor. Appl. Climatol.*, 106(1-2):139–152, November 2011.

[11] Dr. Simon Hook. ecostress.jpl.nasa.gov. `https://ecostress.jpl.nasa.gov/`. [Accessed 09-02-2024].

[12] Hiroyuki Kusaka and Fujio Kimura. Coupling a single-layer urban canopy model with a simple atmospheric model: Impact on urban heat island simulation for an idealized case. *J. Meteorol. Soc. Japan*, 82(1):67–80, 2004.

[13] Qi Li, Jiachuan Yang, and Long Yang. Impact of urban roughness representation on regional hydrometeorology: An idealized study. *J. Geophys. Res.*, 126(4), February 2021.

[14] Ruth Lorenz, Zélie Stalhandske, and Erich M Fischer. Detection of a climate change signal in extreme heat, heat stress, and cold in europe from observations. *Geophys. Res. Lett.*, 46(14):8363–8374, July 2019.

[15] T R Oke. The energetic basis of the urban heat island. *Q. J. R. Meteorol. Soc.*, 108(455):1–24, January 1982.

[16] M A Oliver and R Webster. Kriging: a method of interpolation for geographical information systems. *Int. J. Geogr. Inf. Syst.*, 4(3):313–332, July 1990.

[17] Eva Ostertagová. Modelling using polynomial regression. *Procedia Eng.*, 48:500–506, 2012.

[18] David Parastatidis, Zina Mitraka, Nektrarios Chrysoulakis, and Michael Abrams. Online global land surface temperature estimation from landsat. *Remote Sens. (Basel)*, 9(12):1208, November 2017.

[19] J R Quinlan. Induction of decision trees. *Mach. Learn.*, 1(1):81–106, March 1986.

[20] S J Rey. Mathematical models in geography. In *International Encyclopedia of the Social & Behavioral Sciences*, pages 9393–9399. Elsevier, 2001.

[21] Jochen Seidel, Gunnar Ketzler, Benjamin Bechtel, Boris Thies, Andreas Philipp, Jürgen Böhner, Sebastian Egli, Micha Eisele, Felix Herma, Thomas Langkamp, Erik Petersen, Timo Sachsen, Dirk Schlabing, and Christoph Schneider. Mobile measurement techniques for local and micro-scale studies in urban and topo-climatology. *Journal of the Geographical Society of Berlin*, 147(1), 2016.

[22] Vivek Shandas, Jackson Voelkel, Joseph Williams, and Jeremy Hoffman. Integrating satellite and ground measurements for predicting locations of extreme urban heat. *Climate*, 7(1):5, January 2019.

[23] Chris W Strother, Marguerite Madden, Thomas R Jordan, and Andrea Presotto. Applications paper: Lidar detection of the ten tallest trees in the tennessee portion of the great smoky mountains national park. *Photogramm. Eng. Remote Sensing*, 81(5):407–413, May 2015.

[24] Zander S Venter, Oscar Brousse, Igor Esau, and Fred Meier. Hyperlocal mapping of urban air temperature using remote sensing and crowdsourced weather data. *Remote Sens. Environ.*, 242(111791):111791, June 2020.

[25] Stenka Vulova, Fred Meier, Daniel Fenner, Hamideh Nouri, and Birgit Kleinschmit. Summer nights in berlin, germany: Modeling air temperature spatially with remote sensing, crowdsourced weather data, and machine learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, 13:5074–5087, 2020.

[26] Han Wang, Jiachuan Yang, Guangzhao Chen, Chao Ren, and Jize Zhang. Machine learning applications on air temperature prediction in the urban canopy layer: A critical review of 2011–2022. *Urban Climate*, 2022.

[27] Q Weng. Remote sensing-based estimation of impervious surfaces in the united states. *Remote Sensing*, 6(1):1000–1019, 2014.

[28] David C Wheeler and Antonio Páez. Geographically weighted regression. In *Handbook of Applied Spatial Analysis*, pages 461–486. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[29] Zhaowu Yu, Gaoyuan Yang, Shudi Zuo, Gertrud Jørgensen, Motoya Koga, and Henrik Vejre. Critical review on the cooling effect of urban blue-green space: A threshold-size perspective. *Urban For. Urban Greening*, 49(126630):126630, March 2020.

[30] Guoyi Zhang and Yan Lu. Bias-corrected random forests in regression. *J. Appl. Stat.*, 39(1):151–160, January 2012.

[31] Marius Zumwald, Benedikt Knüsel, David N Bresch, and Reto Knutti. Mapping urban temperature using crowd-sensing data and machine learning. *Urban Clim.*, 35(100739):100739, January 2021.

[32] Marius Zumwald, Benedikt Knüsel, David N. Bresch, and Reto Knutti. Mapping urban temperature using crowd-sensing data and machine learning. *Urban Climate*, 35:97–111, 2021.

Appendix A. Hourly Results for LasVegas: June

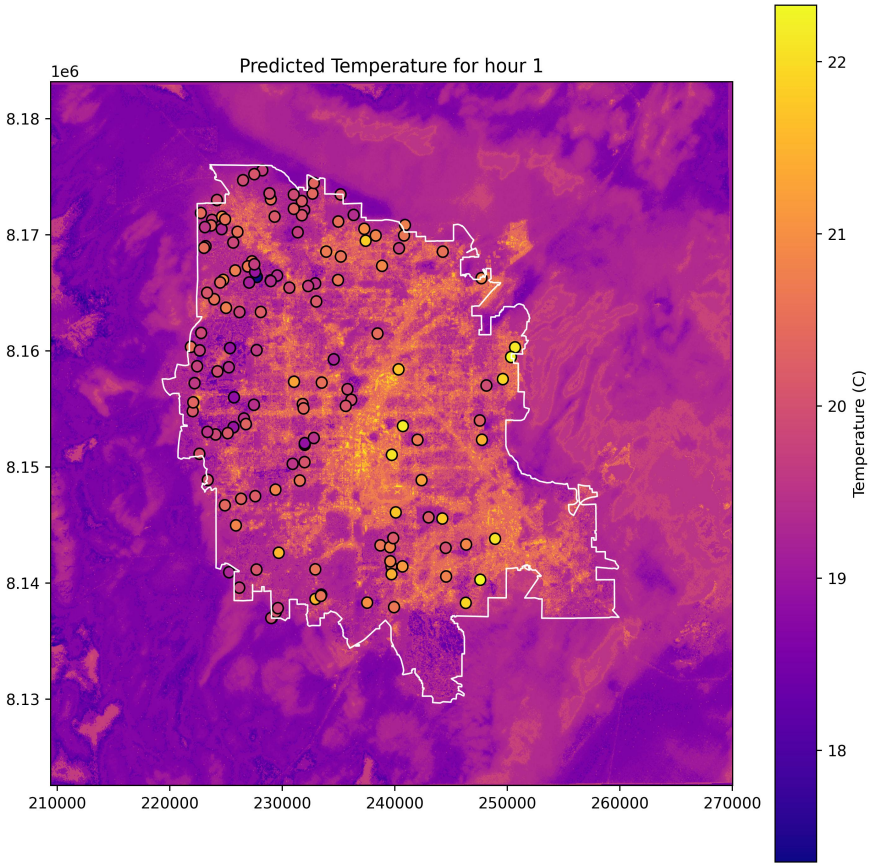

Figure A.1: Hybrid: Output Map - 01 am.

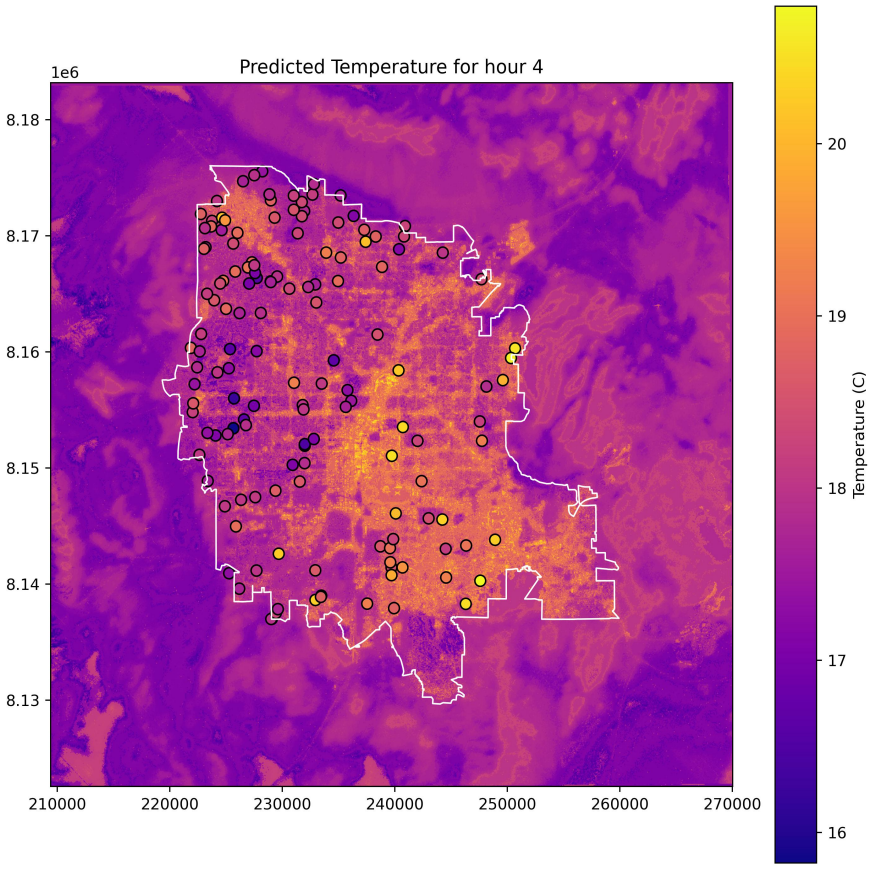

Figure A.2: Hybrid: Output Map - 04 am.

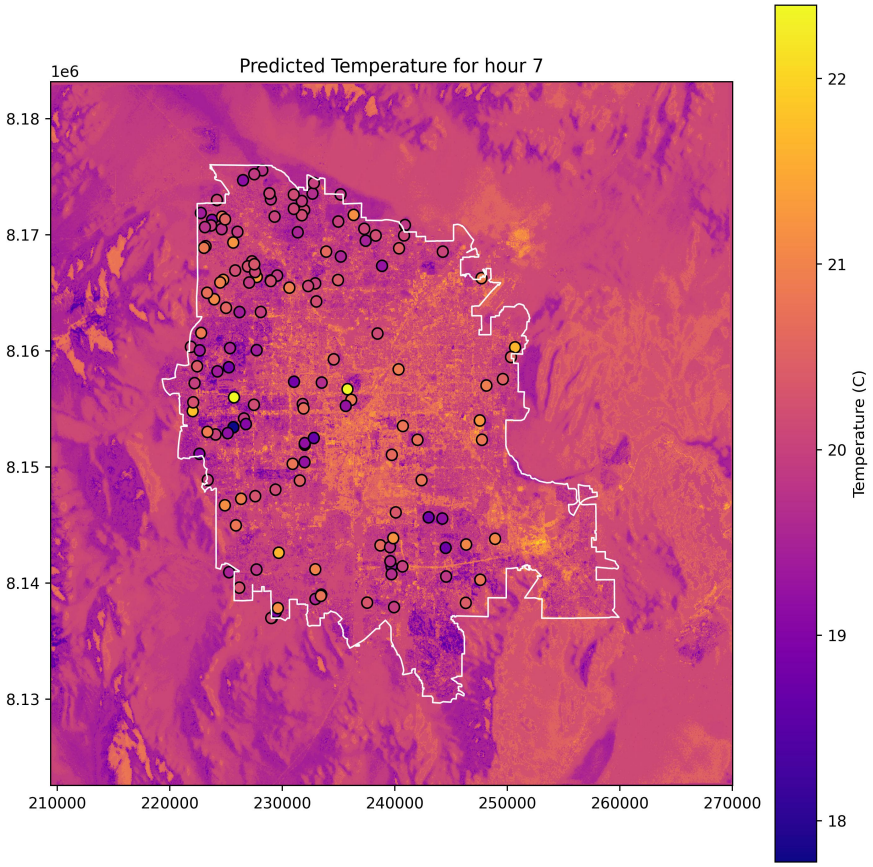

Figure A.3: Hybrid: Output Map - 07 am.

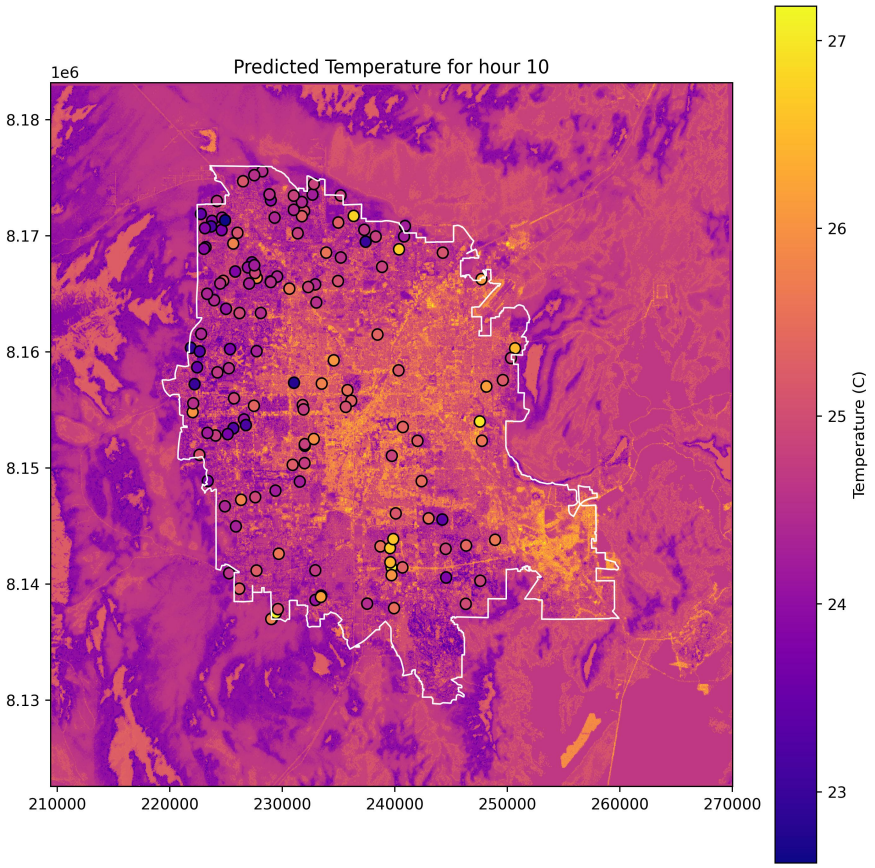

Figure A.4: Hybrid: Output Map - 10 am.

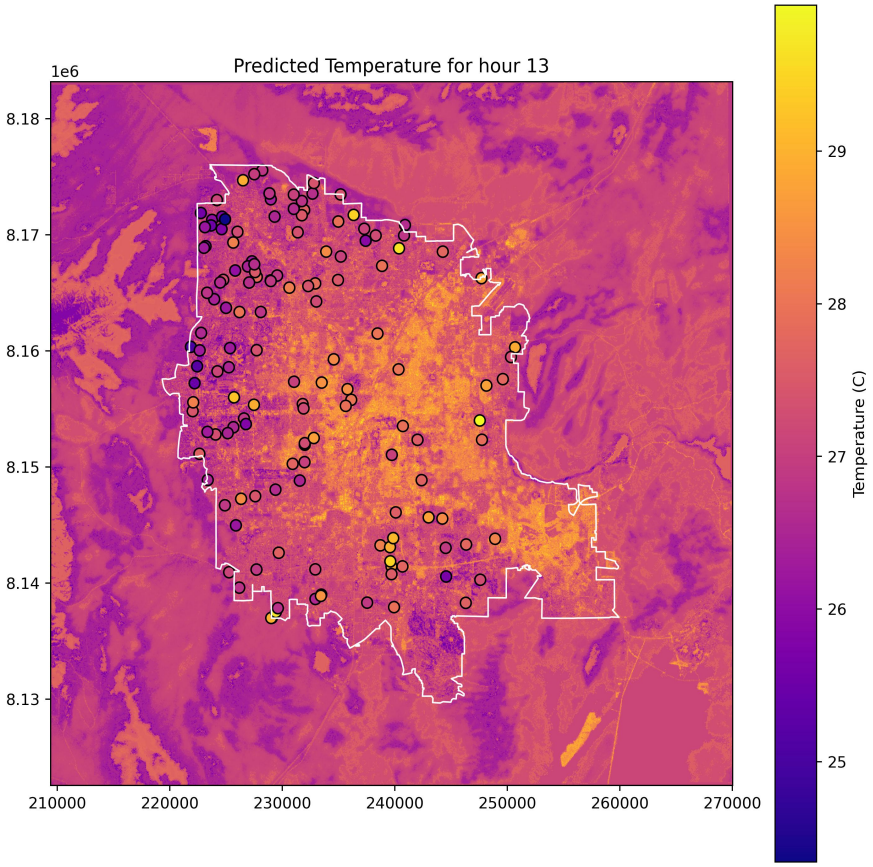

Figure A.5: Hybrid: Output Map - 1 pm.

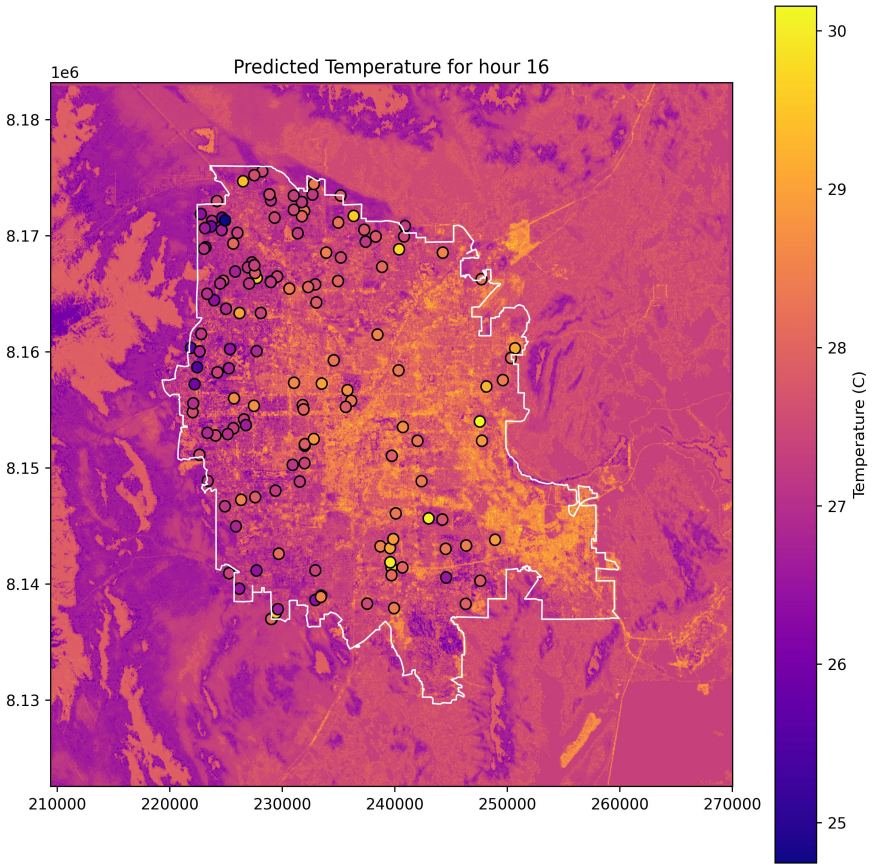

Figure A.6: Hybrid: Output Map - 4 pm.

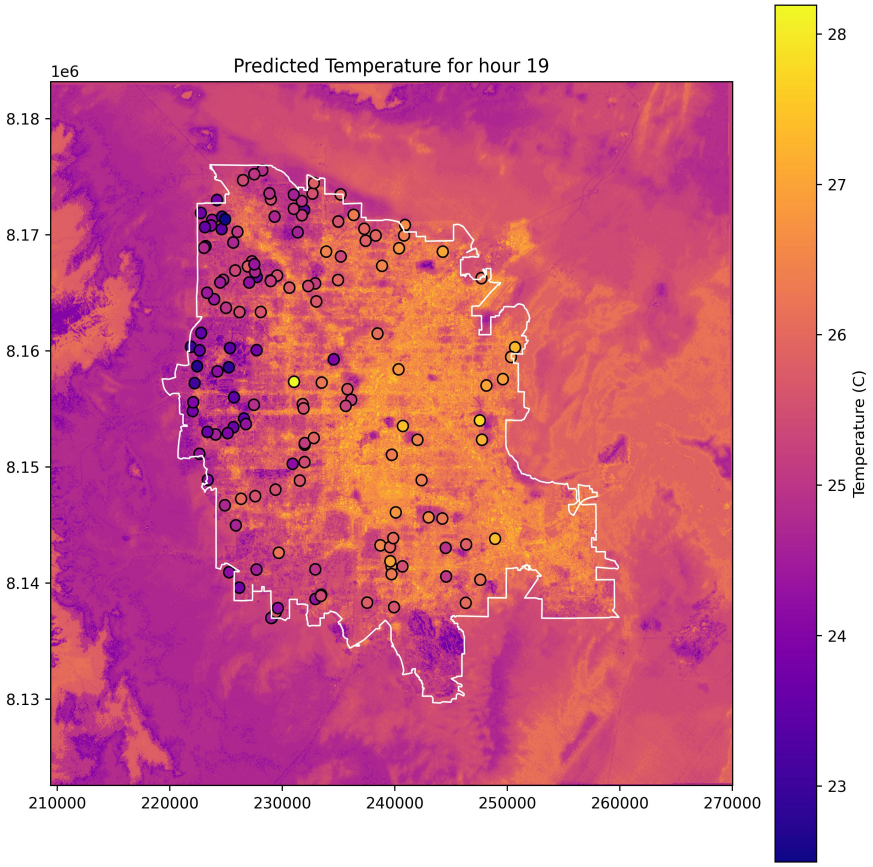

Figure A.7: Hybrid: Output Map - 7 pm.

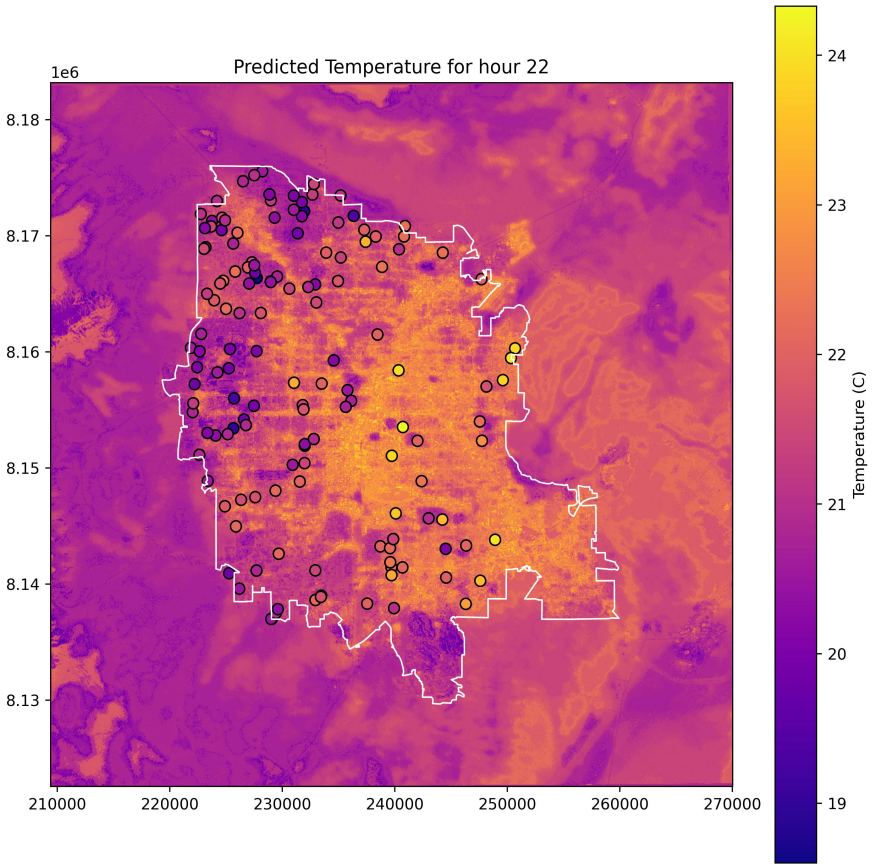

Figure A.8: Hybrid: Output Map - 10 pm.