TITTLE: FINE-SCALED IOT TEMPERATURE FILLING AND URBAN HEAT PREDICTIONS WITH DEEP LEARNING

by

Jingchao Yang
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Geography and Geoinformation Science

Committee:

_____	Dr. Chaowei Yang, Dissertation Director
_____	Dr. Andreas Zufle, Committee Member
_____	Dr. Olga Gkountouna, Committee Member
_____	Dr. Ruixin Yang, Committee Member
_____	Dr. Manzhu Yu, Committee Member
_____	Dr. Dieter Pfoser, Department Chairperson
_____	Dr. Donna M. Fox, Associate Dean, Office of Student Affairs & Special Programs, College of Science
_____	Dr. Fernando R. Miralles-Wilhelm, Dean, College of Science

Date: _____ Summer Semester 2021
George Mason University
Fairfax, VA

TITTLE: Fine-scaled IoT Temperature Filling and Urban Heat Predictions with Deep Learning

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

by

Jingchao Yang
Bachelor of Sciences
Eastern Michigan University, 2016

Director: Chaowei Yang, Professor
Department of Geography and Geoinformation Science

Summer Semester 2021
George Mason University
Fairfax, VA

Copyright 2018 Jingchao Yang
All Rights Reserved

DEDICATION

This is dedicated to my loving parents.

ACKNOWLEDGEMENTS

I would like to thank all people who have helped and inspired me during my doctorate study. I especially want to thank my advisor, Dr. Chaowei Yang, and committee members, Drs. Andreas Zufle, Olga Gkountouna, Ruixin Yang, Manzhu Yu, for their guidance during my research and study at George Mason University. I am grateful for all the supports from Drs. Chaowei Yang, Manzhu Yu, they have been my mentors along the 5-year Ph.D. research life.

I give many thanks to my collaborators at NSF Spatiotemporal Innovation Center for the great job they did and continue to do. It was a pleasure to work with such a group of talented and diligent people. My deepest gratitude goes to my family for their unflagging love and support throughout my life; this dissertation is simply impossible without them. Finally, thanks go out to the Fenwick Library for providing a clean, quiet, and well-equipped repository in which to work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

LIST OF ABBREVIATIONS

Missing At Random……...…...MAR
Missing Completely At Random…….…...MCAR
Not Missing At Random……...…NMAR
Generative Adversarial Imputation Nets…….…… GAIN
National Centers for Environmental Information…….……... NCEI
Local Climatological Data…….……LCD
Environmental Protection Agency…….……EPA
k-Nearest Neighbors …….…….……kNN
Internet of Things……...… IoT
Long short-term memory……...…… LSTM
Data Collection Location …….…….……DCL
Root Mean Square Error…….……...…… RMSE
Mean Square Error…….…….…… MSE
Inverse Distance Weighted …….…….…….……IDW
Random Forest…….…..... RF
Temporal Missing Block…….…….…..... TMB
Spatial Missing Block…….…….…….... SMB
Generative Adversarial Network…….…….…….…..... GAN
Spatio-temporal Multiview-based learning…….……....ST-MVL

ABSTRACT

TITTLE: FINE-SCALED IOT TEMPERATURE FILLING AND URBAN HEAT PREDICTIONS WITH DEEP LEARNING

Jingchao Yang, Ph.D.

George Mason University, 2021

Dissertation Director: Dr. Chaowei Yang

Rising temperature is a major concern of urban livelihood and has become more severe with rapid urbanization. The complexity of built-up urban fabrics and the unevenly distributed anthropogenic heat release has led to urban heat variation. In response to the increasing greenhouse effect in recent years, the demand for understanding the heat variation in the U.S. has risen dramatically. The global warming trend deteriorates the variation by increasing the already high temperatures in heated areas. Many concerns have been brought up related to urban heat variability, primarily in energy and health fields. To address these concerns, many studies have been conducted for urban temperature observations and predictions.

Missing data in observation is inevitable, which makes continuous high-resolution measurements challenging to acquire. Different discriminative and generative models established for sensor missing data filling often show their limitations (e.g., accuracy, stability, efficiency) when fitting into different datasets. Existing research methods for temperature prediction are mainly divided into deterministic methods and statistical methods. Deterministic methods require very informative observations that are difficult

to obtain in practice. In addition, various types of parameters need to be determined, but since these parameters are usually estimated based on experience, the accuracy is limited. Statistical methods, on the other hand, often fail to effectively integrate and analyze multi-source heterogeneous data, which has a considerable impact on temperature. The machine learning (ML) and deep learning (DL) methods proposed in recent years can learn to effectively present features from a large amount of input data. However, to carry out full-coverage high-resolution forecasts, there are high demands to integrate surface weather data and air temperature observations. Data scarcity also brought limitations to many current well-performed ML/DL methods. Another challenge expected to be solved is to transfer and reapply patterns learned in one city to another, as models do not naturally perform well across different regions.

Regarding the missing data challenge, different algorithms (i.e., Kriging, MissForest, GAIN) were selected for comparison. All models built upon these algorithms are tested to fill the missing data at the rate of less than 10%, 20%, 40%, 60%, and 80%. Testing data are selected using either different seasons, or randomly draws from the entire dataset, to measure the stability of these models. Experiments were conducted to shows their performance in data filling accuracy and consistency across different missing data settings. Computational efficiency was considered to provide a complete dataset in real-time. Results demonstrated that each model has its strength and limitations. Ensemble models should be expected to integrate their respective superiorities in computational speed, imputation accuracy, and adaptability to different data missing situations.

Regarding fine-scaled temperature prediction and data scarcity, a framework was proposed to: 1) provide a fast data fusion technique, integrating measurements from the Internet of Things (IoT) of a high spatiotemporal resolution with observations from weather stations; 2) utilize a Long Short-Term Memory network to predict surface temperature from the fusion dataset for four major cities in the U.S.; 3) adopt transfer learning, leveraging the pre-trained model from regions with a higher number of observation stations to predict regions with data scarcity. With the proposed framework, multi-step predictions with low RMSEs were achieved. The transferable model also greatly improved the prediction accuracy for regions with data scarcity up to 26%.

This dissertation makes innovative contribution for the following reasons:

1) The comparison of data filling methods suggests an optimal way to complete hourly IoT temperature measurements in Los Angeles by testing different angles (i.e., computational speed, imputation accuracy, and adaptability to different data missing situations).
2) The DL-based prediction framework provides high-resolution results with up to a 39.6% MAE decrease. It supports data for near future heat-related decision-making in study areas including Los Angeles, New York City, and Atlanta.
3) The transfer learning utilizes well-established models trained by the DL-based prediction framework to minimize the prediction error for regions with data scarcity problems. It improves the predicting MAE up to 25.7%.

CHAPTER ONE. INTRODUCTION

According to the United Nations World Population Prospects, 68% of the world's population is projected to live in urban areas by 2050 (UN DESA, 2018). Temperature is one of the major concerns of urban livability, and rapid urbanization tends to intensify urban issues, such as Urban Heat Island (UHI; Zhong et al., 2017). The global warming trend accentuates the temperature imbalance by increasing already high temperatures in urban areas (Luber and McGeehin, 2008; McCarthy et al., 2010; Harlan et al., 2014). Since 1895, the average annual temperature of the contiguous United States has increased by 0.07 °C per decade. In 2010, it was 12.1 °C, 0.6 °C above normal (NOAA National Climate Data Center, 2010). Lee et al. (2017) hypothesized that global warming would be accountable for up to 71% of the temperature increase in existing urban areas in the 2030s, even with the adoption of high-temperature mitigation strategies. In response to the increase of greenhouse effect in recent years, the demand for understanding the heat variation in the U.S. has risen dramatically (Wu and Li, 2013).

The causes of fine scale urban heat variation can be introduced by the complexity of built-up urban fabrics and different levels of urban canyons (Zhou et al., 2017). The unevenly distributed anthropogenic heat release from transportation and the temperature control of buildings further exacerbate this heat variation. Moreover, differences in

population density, built-up density, and vegetation fractions can also directly or indirectly contribute to the formation of urban heat variation.

A series of health and energy concerns are related to temperature. For example, during 2004–2018, an average of 702 heat-related deaths occurred annually in the U.S. (Vaidyanathan et al., 2020). The total number of deaths during the 1995 Chicago heatwave revealed that the heatwave resulted in 700 more deaths than expected (National Research Council, 2011). Energy consumption is another concern, as the extra cooling energy demand associated with urban overheating for all types of buildings can increase by an average of 12% (Santamouris, 2020). Countries where most buildings have air conditioning (e.g., U.S.) displayed an above-average increase in electricity consumption. The ability to monitor and predict the hourly temperature at the hyper-local level is an asset in managing the risk to human health and energy consumption.

Temperature observations are mostly through satellite imageries and weather stations. The former does not directly measure ambient air temperature with coarse spatiotemporal resolution, while the sparsely distributed weather stations are incapable of providing good spatial continuous observation at a hyper-local level (Holdaway, 1996). On top of that, missing values exist as a critical challenge for both techniques. Missing data filling, either by discriminative or generative models, is essential for fine-scale urban heat variation detection (Yoon et al., 2018).

Existing research methods for prediction are divided into multiple methods, including deterministic, statistical, and machine learning. Deterministic methods are based on aerodynamic theory and physicochemical processes using mathematical

algorithms to establish a numerical model (e.g., The Numerical Weather Prediction; NWP). As statistical methods, the most common approaches are multiple linear regression (MLR; Menon et al., 2017), autoregressive moving average (ARMA; Lydia et al., 2016), and support vector regression (SVR; Kaneda and Mineno, 2016).

Sensor Missing Data Filling

With the high accessibility of in-situ sensors, many have been deployed in the physical world, collecting massive environmental observations. With the Internet of Things (IoT), different IoT-based networks are developed for environmental monitoring, serving as a valuable source for weather observation data fusion (Rawat et al., 2014; Sah and Koli, 2019). Well-established IoT networks achieve near-real-time street-level air temperature measurements, unlike satellite observations that often require downscaling for appreciable spatiotemporal resolution (Ebrahimy and Azadbakht, 2019). However, due to affordability issues, equipment for in-situ measurements is not always built to last, which can result in loss of data quality. Sensor readings are usually lost at various unexpected moments because of sensor or communication errors, e.g., when a sensor loses network access, or when a sensor is powered off. Missed recordings can affect real-time monitoring and compromise the performance of data modeling (Yi et al., 2016; Jaques et al., 2017).

There are three general explanations for the missingness (Rubin, 1987): Missing At Random (MAR), Missing Completely At Random (MCAR) and Missing Not At Random (MNAR; Rubin 1987). When MAR, the population may be represented by the available data, while MCAR is a special case of MAR and occurs when the missing data

is independent from both observable and unobservable factors (Schafer and Graham, 2002). MNAR indicates the missing data depend on other missing values, i.e., one or more factors are impossible to quantify and identify. Since the exact moment when one device stops functioning is generally unknown, assuming recurrent problems are not identified, the missing temperature filling can rely on the MAR mechanism and predict, interpolate, or impute using observed sensor readings (Cheng et al., 2009; Henn et al., 2013; Shtiliyanova et al., 2017).

Different missing data filling techniques have been studied, including Random Forest based imputation (MissForest), K-Nearest Neighbor (KNN), inverse-distance weighting (IDW), and geo-statistical Kriging for handling missing values on sensor networks. Yi et al. (2016) indicated two major challenges behind missing data filling in the spatiotemporal domain: 1) readings can be absent at arbitrary sensors and timestamps; and 2) sensor readings change over location and time significantly and non-linearly, affected by many factors. Temperature data series contain gaps ranging from several hours to several days. The aim of this study is to assess different methodologies to fill missing data for hourly IoT temperature series.

Temperature Prediction and Transfer Learning Framework

Although the numerical model performs as a standard on the temperature forecasting, these models require detailed site-specific information that is difficult to obtain (Chapman and Thornes, 2005). Numerical models also require massive computational power to solve complex equations that predict atmospheric conditions (Hewage et al., 2020). Statistical approaches fail to effectively integrate and analyze non-

linear multi-source heterogeneous data (e.g., traffic flow, meteorological conditions, land use), which can compromise the utility of the temperature predictions (Yang et al., 2020a). The machine (ML) and deep learning (DL) methods acquire effective feature representation from a large amount of input data, providing new approaches for addressing the shortcomings mentioned above.

Temperature interreacts with many other factors, including wind speed, pressure, dew point, and humidity (Anjali et al., 2019; Hossain et al., 2015). One single data point is insufficient to serve as the metric for accurate prediction. To achieve high-resolution predictions, heterogeneous weather observation data fusion is required. As introduced, IoT networks are ideal candidates for such tasks. The shortcoming of IoT networks is that while they provide adequate coverage for areas with high sensor density; the remaining areas are characterized as being data scarce.

Temperature prediction models need to be adapted to different applicable environments. Some models are developed for large-scale temperature forecasting, while others are for smaller region adjusted for more specific environment with higher spatiotemporal resolution. Those are adjusted to fit for specified settings often struggle with adaptability when applying to different regions, which results in a higher model training cost if each region needs to train from scratch. Due to the potential IoT data scarcity problem, those models should also be limited by the region data sufficiency while training. Regions with different environment characters or data scarcity can greatly benefit from transfer learning, where a well-trained model can be reapplied to other settings (regions).

Objectives

In the field of temperature research and GIScience, outreaching to computational science and geographic science, the focus of this dissertation research lies in:

1) Comparing IoT urban temperature missing data filling models. Results from different data filling models vary substantially despite independent model validations, indicating that the performance of data filling models varies from dataset to dataset. Model comparison experiments are conducted to define their strengths and limitations. Comparison should help suggest best utilizations of data filling models for different missing data scenarios.
2) Building high-resolution multivariate temperature predictions. There are a variety of meteorological parameters can affect the temperature conditions. Data fusion provides an integrated high-resolution multivariate dataset for model input to optimize prediction accuracy. Multi-step predictions help to identify the temperature conditions from the next few hours to the next few days, greatly benefiting individuals and decision makers with fast response.
3) Enabling transfer learning. Data scarcity exists, particularly for underdeveloped regions with limited access to data collecting sensors. The lack of observations leads to poor spatiotemporal information coverage, as well as insufficient training data for model learning. Allowing a well-trained model to be reapplied to regions with data scarcity can enhance the prediction accuracy.

Contribution

The main contribution of this dissertation are as follows:

1) The comparison of data filling methods suggests optimal way to complete hourly IoT temperature measurements.
2) The DL-based prediction framework provides high-resolution results and support data for near future heat-related decision making.
3) The transfer learning achieves high prediction accuracy for areas with data scarcity.

Dissertation Organization

The rest of this dissertation is organized as follows. Chapter 2 reviews the literature of related research, in terms of in-situ weather data sources and products, missing data filling techniques, and machine learning in weather study. Chapter 3 compares popular data filling models in different aspects and inspects their adaptability to various IoT temperature missing observation problems. Chapter 4 integrates data from multiple data sources and establishes a multivariate temperature prediction framework with transfer learning capability. Chapter 5 concludes the dissertation and proposes potential future works.

An overall architecture is established to show that the major components of this dissertation can contribute to urban temperature monitoring and analytics science communities and the public (Figure 1).

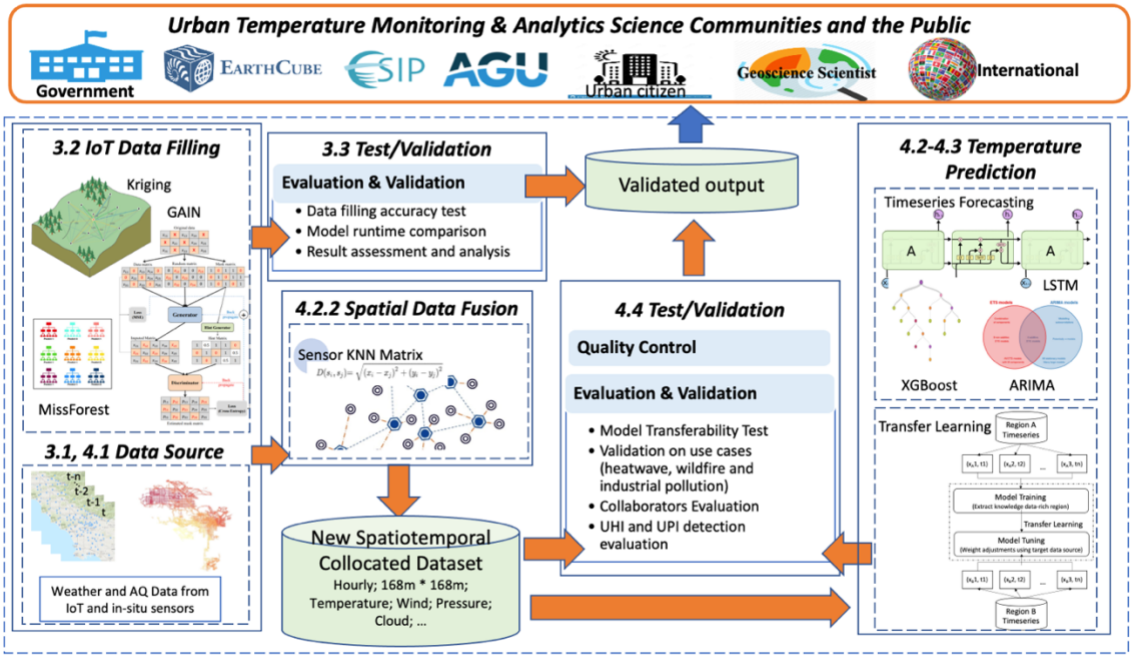


Figure 1 Missing Data Filling and Prediction Architecture for Realtime Temperature

CHAPTER TWO. LITERATURE REVIEW

In-situ Weather Data Sources and Products

Weather forecasting takes atmospheric data observed by different techniques. One is using satellite imageries, which measures land surface temperature continuously in both space and time but is often low in spatial resolution. Satellite observations are also less direct to how human sense environment (ambient temperature), which will not be discussed further in this study. On the contrary, in-situ sensors deployed at weather stations record the ambient temperature at approximately 2 meters above the ground, which is a key variable used to assess local weather change and human-heat interactions. This section will review how two different in-situ data sources (i.e., station data, IoT data) have currently been adopted in research and their limitations.

Station Data

Ground-based station observations, unlike satellite datasets, provide direct measurements. These stations have been built for decades and been contributing data to weather research starting from early stage (Crowe et al., 1978). The increasing number of stations enabled better continuous weather observations. Observations, depend on the sensor setup, include temperature, dew point, relative humidity, precipitation, wind speed and direction, visibility, atmospheric pressure, and types of weather occurrences such as hail, fog, and thunder. National Centers for Environmental Information (NCEI) provides a broad level of service associated with these observations including data collection, quality control, archive, and removal of biases associated with factors such as

urbanization and changes in instrumentation through time. Local Climatological Data (LCD) consists of hourly, daily, and monthly summaries for approximately 1,600 U.S. locations (Boissonnade et al., 2002).

Weather Underground (WU) is a global community of people connecting data from environmental sensors like weather stations monitors so it can provide rich, hyperlocal data (e.g., temperature, pressure, humidity, dew point). There are more than 250,000 personal weather stations, making it the largest of its kind and it provides one of the most local forecasts based on actual weather data points. A recent study based on WU collected 40,025 time series data from year 2012 to year 2016 at Hang Nadim Airport, Indonesia (Salman et al., 2018). Coupling with a DL algorithm, the authors proved the usability of WU for weather variable forecasting with high accuracy.

IoT Data

With the blooming of the internet of things (IoTs), high-resolution time series data collected from densely distributed local sensors became more accessible (Rawat et al., 2014). Different IoT-based networks can be specifically developed for weather prediction or environment monitoring (Sah and Koli, 2019). Different studies across multiple research domains based on IoT datasets and ML/DL models have proved the usability of IoT data with profound results (Widiasari, Nugroho, 2017; Ayele and Mehta, 2018; Chammas et al., 2019). With their high spatiotemporal resolution, these data help lightweight ML/DL models to best achieve their advantage in rapid forecasting. A fast forecast can then lead to fast decision making and response, as recent research found the

intelligence of IoTs can help improve the quality of people's lives (Paul and Saraswathi, 2017).

One of the applications using IoT data is to install smart sensors for real-time environment monitoring and prediction for natural disasters, which enables prevention and fast response. One research established a straightforward system to demonstrate this usability (Yawut and Kilaso, 2011). Weather station networks combined with sensor nodes and a coordinating server were adopted in the research. Different sensor nodes were used to collect temperature, humidity, light, and pressure. Together with decision tree models running on the server end, disaster alert systems were set up to potentially prevent enormous damage from natural disasters. Due to the low power consumption natural of these sensors in the network, the proposed system can be installed in locations that are difficult to hardwire or have no access to electricity.

More studies have demonstrated promising results in using the IoT for weather hazard forecasting, as IoT provides optimal results in obtaining time series data (Widiasari and Nugroho, 2017). Experiments are designed to use Multi-Layer Perceptron Neural Networks (MLPNN) for flood event predictions based on rainfall data, and water levels in the weir. In the MLPNN, two sensors are mounted with one on upstream and one on downstream for monitoring. Data transfer to the server wirelessly, and then forecasting algorithms can be applied to the centralized dataset.

One major limitation that can be found from these studies is the size of the sensor networks. Schatz and Kucharik (2014) reviewed some of the urban climate sensor networks, with at least 10 sensors operating for at least one year, from different studies in

the past 20 years. Overall, the networks are relatively small when considering fine-scale observation, despite the study claimed that the network applied (4 sensors/km2) is one of the most spatially dense and extensive ever deployed.

In-situ Sensor Missing Data Filling

As introduced, in-situ observations are typically limited by the spatially arrayed weather stations distributions, which in turn may often report time series with missing values in both space and time (Holdaway, 1996). Algorithms commonly used in the analysis of such large-scale data often depend on a complete set. Missing value filling methods offer a solution to this problem and can be categorized as either discriminative or generative (Yoon et al., 2018).

Discriminative Filling

The IDW (inverse distance weighted) interpolation is one of the most used interpolation methods and is directly based on the surrounding measured values with weights of the distance to those measurements (Lu and Wong, 2008). It considers spatial correlations between values to interpolate missing values from existing spatial distribution. This method assumes that the variable being mapped decreases in influence with distance from its sampled location, as the first law in geography (Tobler, 1970). This approach is intuitive, efficient, and works well with evenly distributed points. Chen and Liu (2012) applied this method to estimate the rainfall distribution in the middle of Taiwan gained a high correlation coefficient values of over 0.95, proved the usability in their task. IDW was also used to evaluate the impact of pollution, using the measurements from remote stations (De, 2013; Qiao et al., 2018). However, the best

results from IDW are obtained when sampling is sufficiently dense regarding the local variation that are attempting to simulate. If the sampling of input points is sparse or uneven, the results may not sufficiently represent the desired surface (Watson and Philip 1985). Furthermore, IDW is unable to make predictions outside of the maximum and minimum values of the existing cluster.

Due to their popularity, variations and IDW enhancements have also been well explored. Spatio-temporal Multiview-based learning (ST-MVL) ensembles IDW, Simple Exponential Smoothing (SES), User-based Collaborative filtering (UCF), and Item-based Collaborative Filtering (ICF) to impute highly accurate missing data values (Yi et al., 2016). Each of these four empirical statistical models is then put through the linear least square equation to generate a final, holistic value. Despite the success of this approach with 26% accuracy enhancement, it was tested on a small dataset of 16 meteorological sensors; it is unsure how replicable this approach would be towards larger datasets, especially when considering that each model is trained to each individual sensor, and the question remains of how much computational power and time is needed for larger datasets over a larger period. Barbulescu et al. (2020) have recently proposed the optimization of finding the IDW parameter using a nature-inspired metaheuristic, which could potentially help IDW to achieve better performance.

Like IDW, geostatistical kriging weighs the surrounding measured values to derive a prediction for an unmeasured location. Previous studies have demonstrated the advancement of kriging with high accuracy and low bias compared to other methods (Li et al., 2005, Mahdian et al., 2009, Yang et al., 2004). Unlike IDW, the weights in kriging

are based not only on the distance between the measured points and the prediction location but also on the overall spatial autocorrelation of the measured points represented in a variogram (Holdaway, 1996; Aalto et al., 2013). Linear combination of weights is determined by the spatial variation structure (Hattis et al., 2012).

Wu and Li (2013) utilized variables of latitude, longitude, and elevation as residual kriging model input to interpolate the average monthly temperature. The study indicates that adding an elevation factor can enhance predicting performance. Though this proposed model is capable to capture the spatial variability of temperature, it is sensitive to the seasons. In contrast to applying kriging to spatial interpolation, Shtiliyanova et al. (2017) applied a kriging in the temporal dimension to fill in data gaps in time-series of air temperatures. Results show that the method can predict missing temperatures with acceptable accuracy with hourly resolution and for non-high elevation sites. One of the most recent studies adopted space–time regression-kriging to predict monthly air temperature (Li et al., 2020). A time series decomposition was applied for each station, and a multiple linear regression model was used to fit the spatiotemporal trends. A valid nonseparable spatiotemporal variogram function was utilized to describe similarities of the residuals in space–time. A space–time kriging was later applied to predict monthly air temperature, with a highest adjusted R-squared of 0.78.

The k Nearest Neighbor (kNN) technique is the most computationally efficient because it is the lazy learning (Ding et al., 2020). It estimates missing values in data vectors by comparing available values and those of a data set with complete characteristics (Tabassian, 2016). It also provides flexibility to impute both continuous

data and discrete data (Batista and Monard, 2002). However, the choice of tuning parameters without prior knowledge is difficult and might have a dramatic effect on a method's performance (Stekhoven and Bühlmann, 2012).

Generative Filling

Traditionally, machine learning and deep learning technics applied to missing data filling require complete data during training, resulting in a lack of training data when having large missing data rates.

Random forest is one of the most well developed and adopted ensemble learning algorithms for regression tasks. By averaging over many unpruned regression trees, it intrinsically constitutes a multiple imputation scheme. Stekhoven and Bühlmann (2012) proposed an iterative imputation method (missForest) based on a random forest and evaluated using biological data with missing value ranging from 10% to 30%. The built-in out-of-bag error estimates of random forest enables imputation error estimation without the need of a test set. Therefore, it does not require fully observed datasets for training. Random forest can process mixed-type data and is able to create both linear and nonlinear boundaries (Breiman, 2001). This allows MissForest to well handle different types of variables and fill the missing values simultaneously. MissForest has been shown to outperform well-known methods such as k-nearest neighbors (KNN) and parametric MICE (multivariate imputation using chained equation; Waljee et al., 2013; Tang and Ishwaran, 2017).

Despite the statement made by its developers regarding attractive computational efficiency and applicability with high-dimensional data, the dataset adopted in their

experiments is small. Time series datasets can have continuous large data samples and will be explored in our study. Yoon et al. (2018) also found that MissForest as a discriminative model that cannot be adopted as easily when the number of feature dimensions is small, yielding large imputation error when allying to dataset with higher rate of missing values.

Yoon et al. (2018) proposed Generative Adversarial Imputation Nets (GAIN) which adopts the well-known Generative Adversarial Nets (GAN) framework with an additional hint matrix to avoid acquiring complete model training datasets. GAN trains two models simultaneously, a Generator to capture the data distribution, and a Discriminator to estimate the probability that a sample came from the training data rather than generative model (Goodfellow et al., 2014). The Generator is trained to maximize the Discriminator's misclassification rate, while the Discriminator is trained to best classify between observed data and imputed data. Like GAN, the Generator's goal in GAIN is to accurately impute missing data, and the Discriminator's goal is to distinguish between observed and imputed components. A hint matrix containing the index information of the missing data was added to the model, which enabled Discriminator loss to be generated from the hint instead of requesting observations, and the model can thus be trained on incomplete data sets. GAIN outperformed MissForest and Auto-encoder GAIN and demonstrated its usability when handling dataset with different percentage of missing values. The higher the missing rate, the lower the imputation accuracy. Larger training data sample with higher feature dimensions improves accuracy. Similarly, MisGAN was developed using GAN to impute missing data with incomplete

training dataset (Li et al., 2019). MisGAN outperforms GAIN under high missing rates, while GAIN training is quite unstable for the block missingness. Despite the success, MisGAN was developed for the MCAR case, which is different from the nature of temperature missing data (MAR).

Unlike MissForest, both GAIN and MisGAN were established for univariate prediction, where temperature as an environmental variable can be easily affected by other meteorological conditions (e.g., wind speed, humidity, pressure). Implementing multivariate prediction can potentially help the model to be more accurate and robust. Moreover, the proposed GAIN model uses random state to initialize training which can be enhanced by adding extra data preprocessing, allowing temporal pattern integration from temperature time series data.

IoT data, compared to conventional weather station data, has its uniqueness. The advantage of IoT data is its remarkably high spatiotemporal resolution. This advantage generates high dimensional dataset, which is generally a challenge for ML/DL studies. Due to the increasing popularity in IoT data applications, more recent studies start focusing specifically on the IoT missing data issue. Ding et al. (2020) provided a comprehensive review on estimating missing values in IoT time series data using different interpolation algorithm including Radial Basis Functions, Moving Least Squares (MLS), and Adaptive Inverse Distance Weighted, and using standard kNN estimator as a benchmark. The study suggests large differences in computational runtime and accuracy when applying different algorithms to different datasets. The cause of the differences varies depends on the data characteristics.

ML in Weather Study

Due to the limitation of NWP models and the maturity of ML/DL in many research domains, recent weather studies attempt to utilize these models to advance forecasting results. Hippert et al. (2000) used a hybrid system for hourly temperature forecasting by integrating the Autoregressive Integrated Moving Average (ARIMA) model and the Multi-Layer Perceptron Neural Networks (MLPNN). The ARIMA is a class of models that explain a given time series based on its past values (i.e., its own lags and the lagged forecast errors so the equation forecasts future values. The ensembled model solves the drawback of using only one artificial neural network (ANN) or linear prediction framework for forecasting. This model achieves a 1-step (1-hour) prediction for one weather station in Rio with a mean absolute percentage error (MAPE) of 2.7%. Maqsood et al. (2004) developed an ensemble model from MLPNN, Radial Basis Function Network (RBFN), Elman Recurrent Neural Network (ERNN), and Hopfield model (HFM) for different weather forecasting, including temperature, wind speed, and relative humidity. The model's 24-step weather predictions for different weather forecasting for all seasons outperformed each of the standalone models. The success of wind guest predictions using a decision tree model allows local weather monitoring in vineyards, orchards, and different fruit and vegetable crops across 30 locations in different countries (Sallis et al., 2011).

More recently, the Long Short-Term Memory (LSTM) from the DL community is widely used for sequence prediction, including language translation (Luong and Manning, 2015; Huang et al., 2018), speech recognition (Graves et al., 2013; Soltau et al.,

2016), and time series prediction (Hua et al., 2019; Karevan and Suykens, 2020). Within the field of weather prediction, LSTM networks provide short-term prediction of various weather variables, including surface temperature, surface precipitation, wind speed, and solar radiance. Utilizing LSTM, a recent study predicts the occurrence of rapid intensification of tropical cyclones (Li et al., 2017). Another study based on weather balloon data proposes a weighted graph convolutional LSTM for the U.S. country-level temperature forecasting (Wilson et al., 2018). Despite the success of this model (outperformed all baseline models), only 67 stations across the U.S. were used with two weather measurements per day. A newly proposed approach that combines the LSTM with the Empirical Mode Decomposition (EMD) technique can predict El Niño one year in advance, allowing preparation for El Niño-related extreme weather events (Wang et al., 2021).

The fruitful achievements in using ML/DL for hourly temperature forecasting for the past 20 years are reviewed (Cifuentes et al., 2020), including ML/DL techniques, notably MLPNN, Support Vector Machines (SVM), Autoregressive (AR), ARIMA, RBFN, LSTM, and ensemble modeling. The comparison reveals the LSTM to be superior at the single-step prediction at a regional scale with simulation data. Among those utilized real weather observation data, the best predictions achieved Mean Absolute Errors (MAEs) of 0.27°C for 1-step (one hour for each step, AR + MLPNN; Jallal et al., 2019), 1.20°C for 4-step (SVM; Chevalier et al., 2011), 1.62°C for 8-step (Ward MLPNN; Smith et al., 2009), and 1.87°C for 12-step predictions (Ward MLPNN; Smith et al., 2009). The best MAEs for different multi-step predictions serve as references for

our model evaluations. The XGBoost (eXtreme Gradient Boosting), a popular time series prediction algorithm, is implemented recently for outdoor air temperature prediction using weather station data (Ma et al., 2020) with a RMSE of 6.3°C for 3-step predictions. The XGBoost is an ensemble tree (decision-tree-based) algorithm that uses a gradient boosting framework. Gradient boosting is a supervised learning algorithm, which accurately predicts a target variable by combining the estimates of a set of weaker models.

Weather research is beyond the comparison among ML/DL models and extends the capability of ML-based weather predictions compared with the well-recognized NWPs. Du (2018) compared the MAPEs of ANN (4.06%), SVR (3.87%), and GP (3.81%) based methods with NWP (4.57%) for 3-step wind forecasting, showing all ML-based algorithms outperforming the NWP. Hewage et al. (2020) propose a novel lightweight weather prediction model using LSTM and Temporal Convolutional Networks (TCN), which run on a standalone PC for a better short-term prediction than the well-recognized Weather Research and Forecasting (WRF) model for up to 12 hours. Despite the success, the author argues that local weather station data would be highly beneficial to the established model. The WRF data used in the article requires an additional 3 hours of data access time, delaying forecasts.

IoT Weather Forecasting

IoT sensor networks are increasingly used in environmental monitoring, offering real-time measurements with high spatial resolutions (Ballari et al., 2012). Weather studies across multiple research domains based on IoT datasets and ML/DL models prove

IoT data usability with promising results (Ayele and Mehta, 2018; Chammas et al., 2019). With its high spatiotemporal resolution, these data help lightweight ML/DL models best achieve their fast-forecasting advantages compared to numerical models. A fast forecast leads to accelerated decision making and response, as recent research found the intelligence and the autonomous of IoTs can improve the quality of people's lives (Paul and Saraswathi, 2017).

Yawut and Kilaso (2011) adopt weather station networks combined with IoT sensors demonstrating the usability of IoT data for real-time natural disasters monitoring and prediction and fostering fast response and life-loss prevention. Different sensors collect temperature, humidity, light, and pressure, and with decision tree models, establish disaster alert systems to prevent damage from natural disasters. Due to the sensors' low power consumption, the proposed system is installed in locations challenging to hardwire or use solar energy for flexibility. Similarly, Martinez et al. (2017) utilizes low-power IoT sensors for different glacier stick-slip motion investigations and expected to enable monitoring for a broader range of areas. Another promising result in using the IoT for weather hazard forecasting is demonstrated by Widiasari and Nugroho (2017): their study concludes that IoT provides optimal results in obtaining time series data. Experiments using MLPNN for flood event predictions based on rainfall data and water levels in a weir with two sensors upstream and downstream for monitoring. Data are transferred to the server wirelessly with forecasting algorithms applied to the centralized dataset.

The size of the sensor networks remains a shortcoming. Schatz and Kucharik (2014) reviewed urban climate sensor networks, with at least ten sensors operating for at least one year from different studies in the past 20 years. The networks are small for high spatiotemporal observations, despite the study's claim that the network they applied (4 sensors/km2) is one of the most spatially dense and extensive. There is a lack of using densely deployed sensors to fully achieve the potential of IoT networks in weather forecasting. By contrast, hundreds of IoT sensors for each study region are utilized for data fusion in the proposed framework. The forecasting model based on this fusion result leads to high-resolution temperature prediction.

Transfer Learning

In any neural network, models are trained to find the 'perfect' weights for prediction. The idea of transfer learning is applicable since the weights in the first few layers of a DL model are learning low-level features (like edges and corners in computer vision). In this case, it is not necessary to learn the same features from scratch while training on similar data. A pre-trained model on a source dataset can then be transferred and fine-tuned on a target dataset without having to modify the hidden layers of the network (Fawaz et al., 2019).

LSTM has been widely adopted for the time series prediction task; however, the algorithm relies on the assumption that there are sufficient training and testing data coming from the same distribution (Ye and Dai, 2018). Transfer learning can provide the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned from old data when the new dataset does not have

enough records (Olivas et al., 2009). It uses pre-trained models as the starting point to also save the vast compute and time resources required to develop neural network models on related problems (Yosinski et al., 2014). One limitation is that the transferability gap grows as the distance between tasks increases, particularly when transferring higher layers. But even features transferred from distant tasks are better than training from scratch with random weights.

There are many cases where the size of newly collected data is small, resulting in relatively small amount of fresh training data, while abundant old data can be obtained. Since time series data usually vary with time, samples over a long-time span differ widely from each other commonly. Hence, applying old data directly to prediction process is normally considered inadvisable. A hybrid algorithm called TrEnOS-ELMK was developed for time series prediction while adopting transfer learning to make knowledge transferred from old data possible (Ye and Dai, 2018).

In the weather study, a trial was proposed for transferring information via training a DL model on data-rich wind farms and then finely tuned with data from newly built farms (Hu et al., 2016). As to the newly built wind farms, sufficient historical data is not available for training an accurate model, while some older wind farms may have long-term wind speed records. The experimental results show that prediction errors are significantly reduced when using the proposed technique. In this dissertation, transfer learning is explored, as the values of temperature observations vary from region to region but the datasets themselves are highly similar. Specifically, even all study areas selected are considered as major cities, LA and NYC have the most stations while Chicago and

Atlanta have much less. Stations were also built at different times and have different lengths of historical data, such as the wind farm problem. Proposed transferable models are expected to solve the data imbalance problem.

CHAPTER THREE. IOT TEMPERATURE MISSING DATA FILLING MODEL COMPARISON

IoT Temperature Data and Study Area

IoT Temperature Data

One-year vehicle-based IoT ambient air temperatures are provided by wireless GeoTab's GO devices. GeoTab provides a truck-mounted sensor dataset, with observations at each predefined geohash grid during certain hours only for trucks in passing. Each geohash covers an area of 153m*153m. Since it is a vehicle-based data collection, geohash grids does not provide full data coverage, e.g., there are grids that have never collected data within our data collection. In this dissertation, we treat geohash grids as points and define them as weather data collection locations (DCLs, like stations) with time series temperature measurements. DCLs may have missing values even when they are on the truck route. This may cause by sensor functionality issues or simply no trucks pass through at the timestamp. GeoTab does not provide historical data to general users, and the near-real-time temperature data used in this dissertation were harvested daily using Python and SQL from Google Cloud Big Table API. Data are hourly temperature records from 04/29/2019 8:00 am to 05/01/2020 6:00 am and are preprocessed by GeoTab to remove anomalies.

Table 1 Data Sources for IoT Temperature

Dataset	Org. & Data Source	Spatial Resolution	Temporal Resolution	Time Coverage	Related Product/Variable

GeoTab IoT	GeoTab, Inc. https://data.geotab.com/weather/temperature	153m x 153m; 23,300 DCLs (LA)	hourly	2019-2020	Temperature

GeoTab IoT contains various data fields, including the sensor metadata and different parameter readings at different timestamps (Table 2). Temperature in Celsius is utilized through the dissertation for easy comparison to other research literatures.

Table 2 Filed information of GeoTab IoT data

Field	Type	Description
Geohash	STRING	Geohash at the 7 character level (153m x 153m)
GeohashBounds	GEOGRAPHY	Polygon object of the geohash bounds
Latitude_SW	FLOAT	Latitude of the southwest corner of the geohash
Longitude_SW	FLOAT	Longitude of the southwest corner of the geohash
Latitude_NE	FLOAT	Latitude of the northeast corner of the geohash
Longitude_NE	FLOAT	Longitude of the northeast corner of the geohash
City	STRING	City (or municipality) within which the geohash resides (U.S., Canada, and Mexico only)
County	STRING	County within which the geohash resides (U.S. and Mexico only)
State	STRING	State within which the geohash resides (U.S., Canada, and Mexico only)
Country	STRING	Country (or territory) within which the geohash resides (English common name)
ISO_3166_2	STRING	ISO-3166-2 codes for country and subdivision
TimezoneName	STRING	Name of the time zone in which the geohash resides
LocalDate	DATE	Local date when the temperature was recorded

LocalHour	STRING	Local hour (24 hr) when the temperature was recorded
UTC_Date	DATE	UTC date when the temperature was recorded
UTC_Hour	STRING	UTC hour (24 hr) when the temperature was recorded
Temperature_C	FLOAT	Average temperature within the geohash (in ℃)
Stdev_C	FLOAT	Standard deviation of the temperature readings within the geohash during the hour in which temperature was recorded (in ℃)
Temperature_F	FLOAT	Average temperature within the geohash (in ℉)
Stdev_F	FLOAT	Standard deviation of the temperature readings within the geohash during the hour in which temperature was recorded (in ℉)
Version	STRING	Version number of the dataset

Study Area

LA was selected for this study due to its large in-city heat variations, particularly in downtown area, making it more significant to have continuous high resolution temperature measurements. Heat variation patterns differ depending on the time of day due to the urban fabric and human activities (Figure 2).

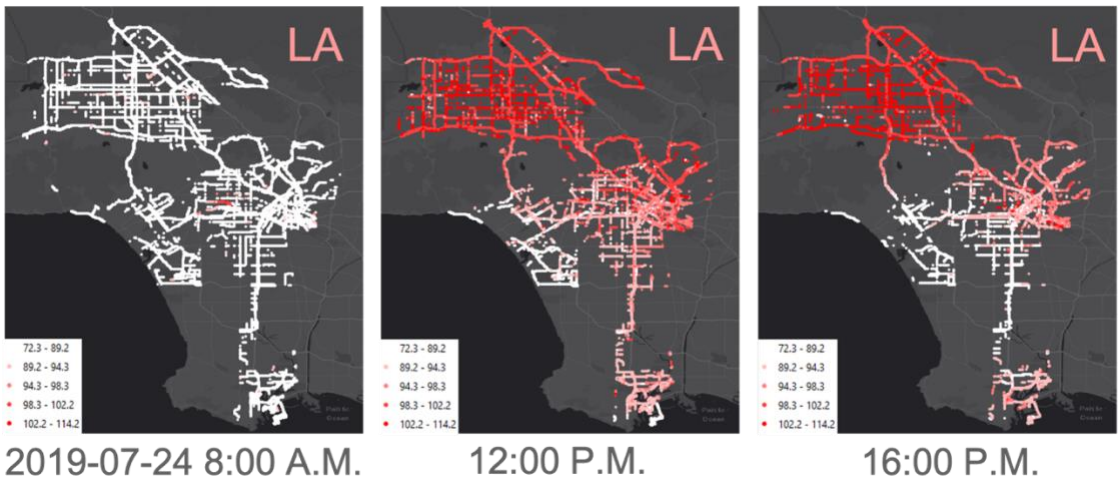


Figure 2 LA Heat Variation from GeoTab IoT Temperature Observations

Data missing rate controls DCL coverage and distribution of a city, and the DCLs in downtown LA are selected for data filling comparison (Figure 3). Distributions demonstrate DCLs with missing rate less than 20%, 40%, 60% and 80%. Figures with higher missing rate include the DCLs from those with lower missing rate (e.g., Figure 3b, c, and d include the DCLs from Figure 3a). Due to the nature of vehicle based IoT, only a small portion of DCLs have good data completeness with missing rate < 20% located on major road with high traffic flow. The ability to fill missing data for DCLs with higher missing rate allows better city area coverage and provides higher temperature resolution.

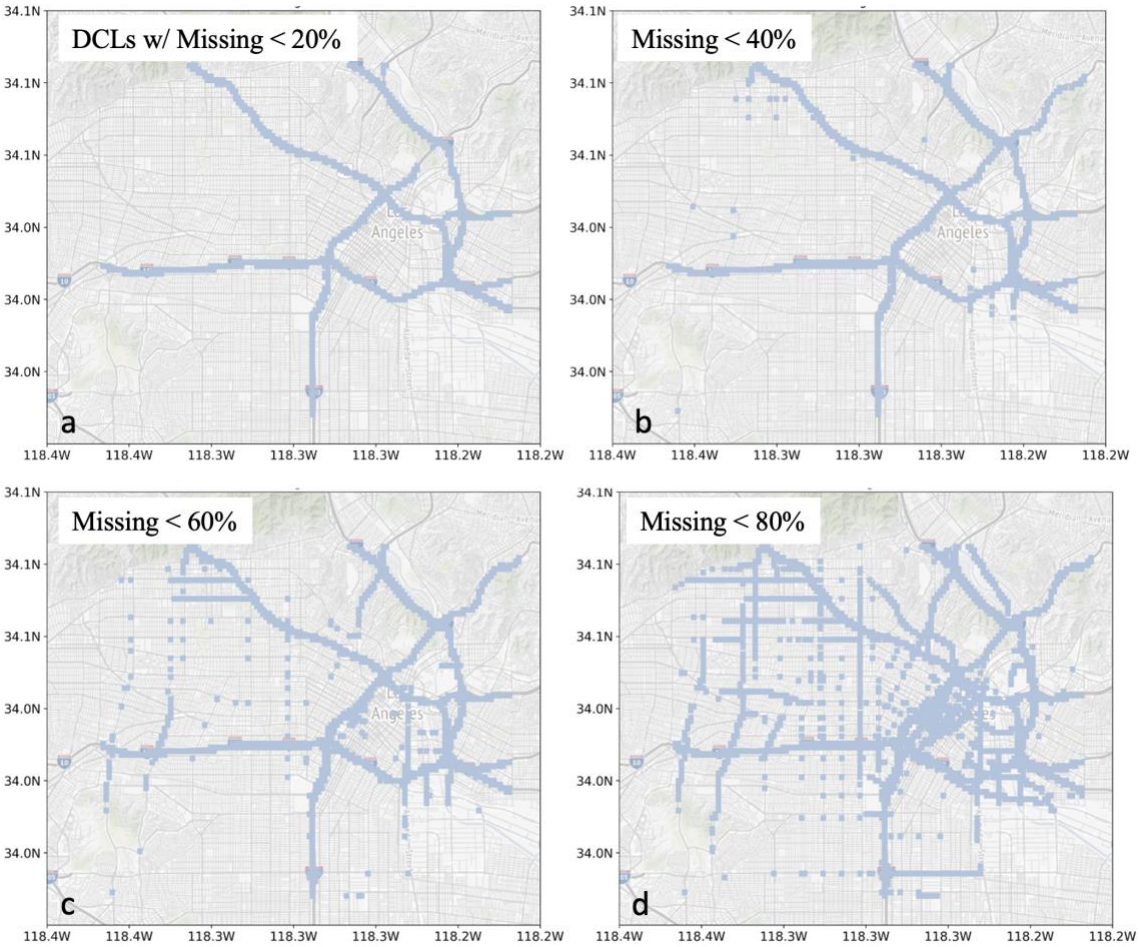


Figure 3 The Distributions for DCLs with Missing Data Less Than 20%, 40%, 60% and 80%

Popular Data Filling Techniques

As introduced in literature review, there are many different data filling techniques. Different datasets have their own uniqueness and often require comprehensive testing to find the optimized solution. MissForest was first proposed and tested in biological fields, while GAIN was applied to general ML datasets. To our best knowledge, no data filling research has been conducted for this high-resolution vehicle

based IoT temperature dataset. This section will closely examine three different popular data filling algorithms, where each represents their superiority in their domain.

Kriging

Data interpolation will be performed using kriging, as the temperature observations are assumed to be continuous in space, and the IoT dataset is dense enough to provide results with high accuracy. Kriging is an advanced geostatistical procedure that generates an estimated surface from a scattered set of points with temperature. It weighs the surrounding measured values to derive a prediction for an unmeasured location. The formula is formed as a weighted sum of the data (Equation 1).

Equation 1 Kriging

$$Z_{(S_0)} = \sum_{i=1}^{N} \lambda_i Z_{(S_i)}$$

, where $Z_{(S_i)}$ is the measured value at the i location; λ_i is an unknown weight for the measured value at the i location; S_0 is the prediction location; and N is the number of measured values. The weights (λ_i) are based not only on the distance between the measured points and the prediction location but also on the overall spatial arrangement of the measured points (Figure 3). The result from kriging will expand the raw observations from each timestamp to a full coverage (size of the sensor network in ideal situation as explained).

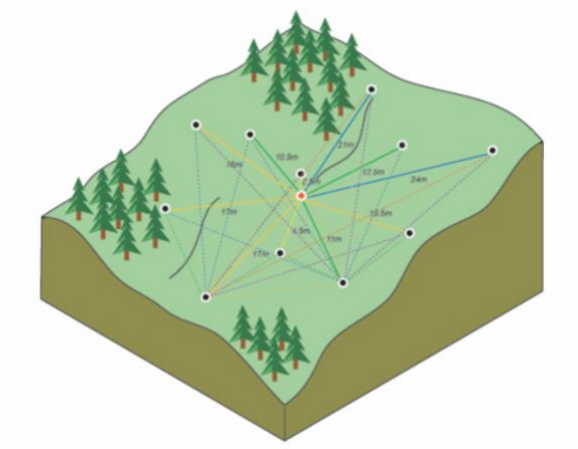


Figure 4 Kriging Interpolation Demonstration

The red point is the interpolation target (S_0) in the study area with all black point indicate locations with observations. The Euclidean distance is adopted in this scenario and the spatial arrangement is generated based on the distance and the distribution of observations.

To apply Kriging to the time series data for spatial interpolation, algorithm will be applied to each timestamp that has at least one reading from one sensor.

MissForest

When the structure of MissForest is decomposed, it becomes apparent that Random Forest (RF) is essential to this algorithm. The RF is a classification algorithm consisting of many decision trees (Breiman, 2001). RF takes advantage of the bagging mechanism by allowing each individual tree to randomly sample from the dataset with replacement, resulting in several different trees. Each tree in a RF only picks from a random subset of features. This forces even more variation amongst the trees in the model and ultimately results in a lower correlation across trees and more diversification.

MissForest inherited these advancements from RF (with bagging and feature randomness) to address the missing data problem. Data are imputed by regressing each variable in turn against all other variables and then predicting missing data for the dependent variable using the fitted forest (Stekhoven and Bühlmann, 2012).

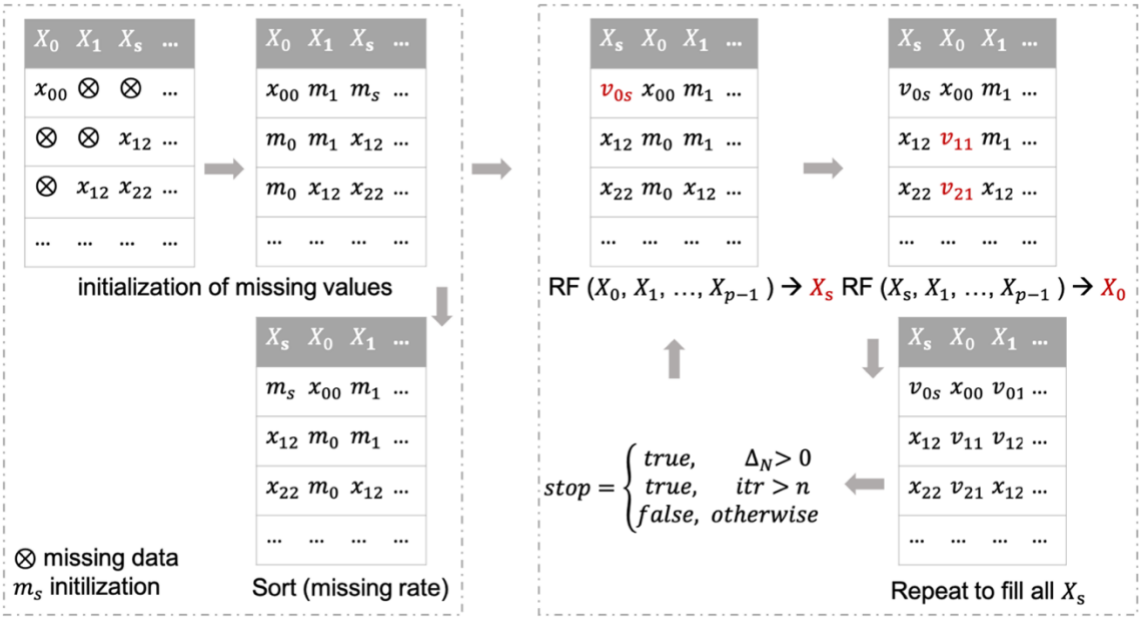


Figure 5 Overview of the MissForest Algorithm

To fit this model for the IoT temperature data, I assume the dataset $X = (X_0, X_1, X_2, \ldots, X_{p-1})$ has $n * p$ dimensions, where n is the total timestamps and p is the number of sensors. Each sensor X_s has missing values at a timestamp $i_{mis}^{(s)} \subseteq \{0, 1, 2, \ldots, n-1\}$. The dataset can be separated into four parts:

- The missing values of variable X_s, denoted by $y_{mis}^{(s)}$
- The observed values of variable X_s, denoted by $y_{obs}^{(s)}$

- The variables other than X_s with observations $i_{obs}^{(s)}$denoted by $X_{obs}^{(s)}$
- The variables other than $X_s$with observations $i_{mis}^{(s)}$ denoted by $X_{mis}^{(s)}$

Note that $X_{obs}^{(s)}$ is typically not completely observed since the index $i_{obs}^{(s)}$ corresponds to the observed values of the sensor X_s. Likewise, $X_{mis}^{(s)}$ is typically not completely missing.

This proposed MissForest algorithm directly predicts the missing values using an RF trained on the observed parts of the dataset. The model makes an initial guess for the missing values in X using a chosen imputation method (e.g., mean imputation) to begin with (Figure 5). All stations are then sorted to according to ascending order by missing rate. For each variable X_s, the missing values are imputed by first fitting an RF with response $y_{obs}^{(s)}$ and predictors $X_{obs}^{(s)}$; then, predicting the missing values $y_{mis}^{(s)}$ by applying the trained RF to $X_{mis}^{(s)}$. Updating the update imputed matrix using predicted $y_{mis}^{(s)}$. The imputation procedure is repeated until the difference Δ_N between the newly imputed data matrix and the previous one increases for the first time with respect to both variable types.

Equation 2 MissForest Stopping Criteria

$$\Delta_N = \frac{\sum_{j \in N}(X_{new}^{imp} - X_{old}^{imp})^2}{\sum_{j \in N}(X_{new}^{imp})^2}$$

, where N is a set of variables.

GAIN

GAN is widely used for editing or generating images, security purposes, and in many other areas, outperforming most of the neural network architectures. As introduced in Section 2.2.2, the nature of generative model allows it to generate new data instances from zero. However, using GAN for production level missing data imputation is still challenging. GAIN was proposed by Yoon et al. (2018) and is one of the most popular GAN architectures for missing data filling. It is realized in this research as comparable with Kriging and MissForest to handle IoT temperature missing data filling. The idea behind it is straightforward: the Generator takes the vector of real data which has some missing values and imputes them accordingly. The imputed data is fed back to the Discriminator whose job is to figure out which data was originally missing. Unlike in a standard GAN where the output of the Generator is either completely real or completely fake, in this setting the output is comprised of some components that are real and some that are fake.

In the model structure, there are three matrices used as model input (Figure 5). Data matrix (X) is the original IoT observations. Mask matrix (M) is pre-determined by the dataset, where 1 means observation exist, and 0 means the value is missing. A random matrix is randomly generated based on M, to assign initial values for the missing data spots. Similarly, we assume the dataset $X = (X_0, X_1, X_2, \dots, X_{p-1})$ has $n * (p-1)$ dimensions, where n is the total timestamps and p is the number of DCLs. $M = (M_0, M_1, M_2, \dots, M_{p-1})$, taking values in $\{0,1\}^{p-1}$. Based on X and M, we define a random matrix ($\tilde{X}$, Equation 3).

Equation 3 GAIN Random Matrix Construction

$$\tilde{X} = \begin{cases} x_i, & if\ m_i = 1 \\ *, & otherwise \end{cases}$$

During the training, M tells the Generator which values are missing, or which values are present. $\tilde{X}$ adds randomness to initialize the Generator to impute. $\bar{X}$ is produced from the Generator for data imputation.

Equation 4 GAIN Imputation Matrix

$$\bar{X} = G\left(\tilde{X}, M, (1 - M) \odot Z\right)$$

Equation 5 GAIN Complete Matrix

$$\hat{X} = M \odot \tilde{X} + (1 - M) \odot \bar{X}$$

, where Z is a noise variable, independent from other variables. $\odot$ denotes element-wise multiplication. $\hat{X}$ indicates the complete data matrix.

Rather than identifying whether an entire vector is real or fake, the Discriminator attempts to distinguish which components are real (observed) or fake (imputed). The highlight of GAIN is the introduction of Hint matrix (H), which is supplied to the Discriminator to support fake data identification. The improvement of using hint matrix produces smaller Discriminator loss, thus ensure the Generator to learn. The initialization of H is predefined to determine the amount of information contained in H about M.

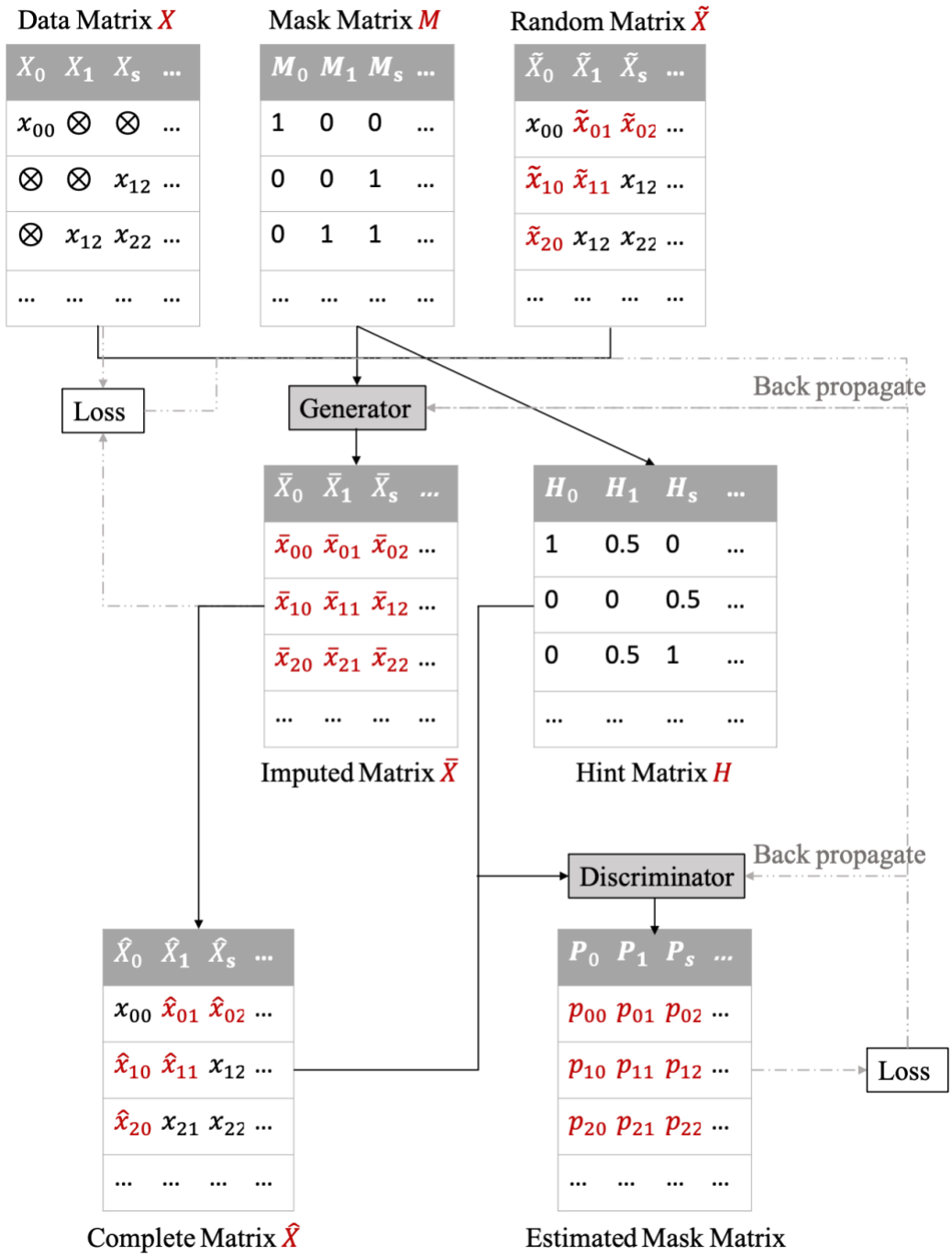


Figure 6 GAIN Model Structure

Essentially, GAIN is used to train Discriminator to maximize the probability of correctly predicting M, while the Generator is trained to minimize the probability of the Discriminator predicting M.

IoT Temperature Data Filling

The uniqueness of this vehicle based IoT data is that DCLs in low traffic roads can have continuous missing data (temporal missing block, TMB), causing extremely high missing data rate for certain DCLs. The temporal block missing increases the possibility of spatial block missing (i.e., no DCLs has data at certain timestamp). Kriging utilizes values from neighboring DCLs to fill missing data at certain location would not work when spatial missing block (SMB, e.g., late at night when no vehicle is driving and collecting data). In this case, most of the experiments are conducted after removing timestamps with spatial block missing.

Different algorithms may have various error estimation techniques. MissForest utilizes out-of-bag error for model performance estimation, while GAIN produces the whole matrix and applies MSE for those have original observations with predicted values. To fairly compare the performance of the three algorithms, 10% of the data is randomly selected and removed before data filling and then used for accuracy calculation. For those that requires model training before data filling, the data is split into different seasons, and one of the four seasons is determined as the testing set once the model has been trained.

Data Filling with Default Settings

To start the experiments, three different algorithms are applied with parameters set to default. In addition, a Baseline is added into the initial comparison. Essentially, the Baseline imputation compute for each missing data by utilizing the sum of the overall mean of the whole matrix with the difference of the row average from the overall mean and the difference of the column average from the overall mean.

Fall and summer seasons were selected to calculate data filling accuracy respectively (Figure 7), meaning for models (i.e., GAIN) that require training, all other three seasons will be the training data. As introduced, SMB exits when there are no readings for all DCLs at certain timestamp. SMBs are removed as part of the data preprocessing to fairly compare all three different models, since kriging does not apply to SMB situation. Compared to using mean absolute error (MAE), RMSE has the benefit of penalizing large errors more and is preferred for accuracy assessment (Equation 6).

Equation 6 RMSE

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(P_i - O_i)^2}$$

Kriging outperforms the other two through all missing rate settings, when SMBs are ignored (Figure 7a, b). Shaded area indicates the RMSE range from the smallest of a testing DCL to the largest, while the line plot shows the mean. In contrary to how Kriging and MissForest demonstrate steady and low increase of RMSE with the missing rate get larger, GAIN gets dramatic increase when missing rate is large. In the experiment settings, the significant difference first shows when including all DCLs has missing data

smaller than 60%. The maximum RMSE or GAIN at missing data smaller than 80% tripled than that of Kriging or MissForest. The pattern exists for when the summer season or the fall season are the testing seasons. All three selected techniques demonstrated their usability for IoT temperature missing data filling compares to the Baseline. Although the Baseline imputation handles missing data better than GAIN (with slightly larger RMSE than that of MissForest), it does not demonstrate advantages over Kriging when neither can apply to SMBs. Due to the lack of significant benefits of utilizing the Baseline, only the three selected techniques are included in the rest of the experiments.

The only study case conducted in this chapter that utilizes all data point is when testing the performance between only MissForest and GAIN (Figure 7c, d). Despite the still higher average RMSE across the experiment settings, when fitting all data for data filling, the GAIN model has a lower maximum RMSE tested in summer season. It falls back into the same pattern for high missing rate data filling with drastically increased prediction errors.

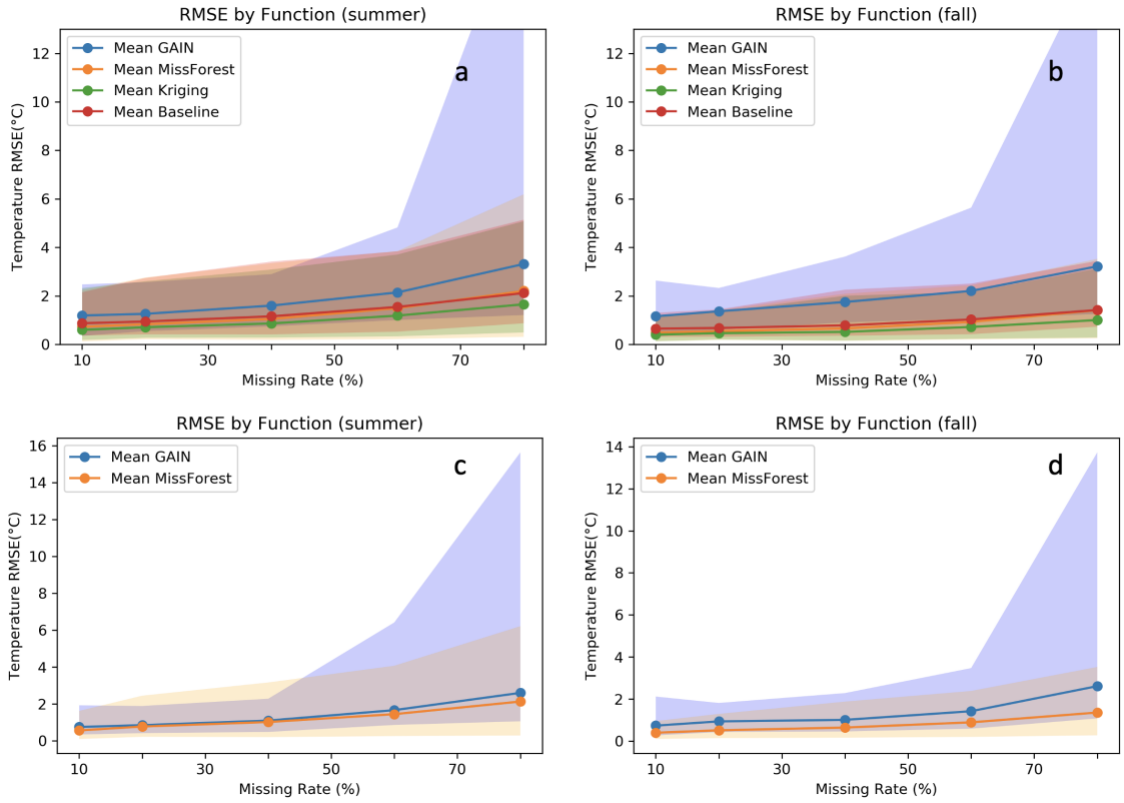


Figure 7 Data Filling Accuracy Comparison with Default Settings. 7a, b for SMB removed; 7c, d for all data

To further examine the spatial pattern of model performance, the RMSE for each DCLs is plotted (Figure 8). The darker the red, the higher the RMSE. DCLs with lower missing rate remain low RMSEs when data filling is applied to larger datasets include DCLs with higher missing rates. The spatial distribution of DCLs explains the reason for Kriging to perform well on DCLs with large missing rates. The idea behind the Kriging introduced in Section 3.2.1 is that it weighs its surrounding observations for the missing locations. Most of the ones that have large missing rate are in downtown areas

surrounded by DCLs with small missing values (Figure 3). The spatial accessibility of the close by readings enables Kriging to make accurate predictions.

Figure 8 Missing Data Filling RMSE Spatial Distribution

Runtime Comparison

The purpose of missing data filling for IoT temperature is to better support real-time heat variation detection, which benefit urban livings by making fast heat related response. Therefore, the computational time is another critical indication for different algorithms. In the experiments, all models are tested under the same Google Colab Pro environment with following hardware configurations:

- CPU: Intel(R) Xeon(R) CPU @ 2.30GHz (* 4)
- GPU 0: Tesla P100-PCIE-16GB
- Memory Total: 26 GB

Data filling for different missing data rate are compared. MissForest has a much longer runtime compared to the other two, making it not suitable for real-time (hourly) data filling and prediction. GAIN performs the fastest with less than 6 minutes to fill all DCLs with less than 80% missing data, and 86 seconds for the 10% setting.

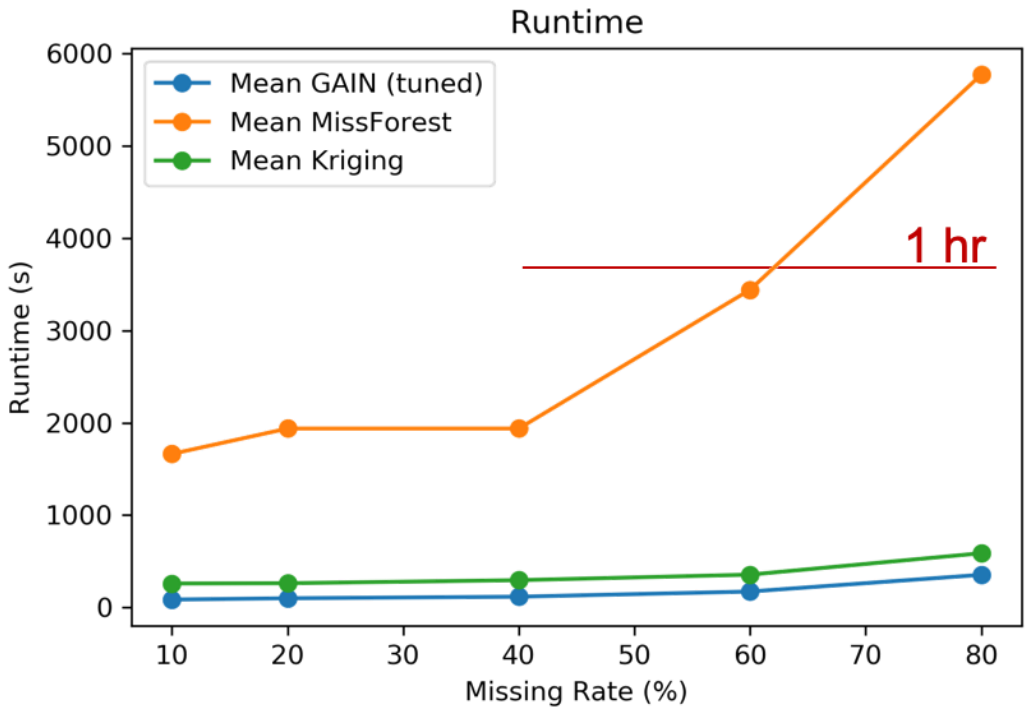


Figure 9 Data Filling Algorithm Runtime Comparison

GAIN Tuning

ML and DL algorithms in general requires hyperparameter tuning to allow optimal performance when fitting different datasets. Due to the large runtime of MissForest and how it fails to complete the IoT missing temperature filling tasks within a reasonable time range, the tuning experiments are designed for GAIN. There are three major parameters controls the accuracy of GAIN predictions (i.e., Batch size, Hint rate, and Alpha). Batch size decides the selected sample size (time stamps) utilized for each training epoch. Hint rate determines the amount of information of mask matrix passes to Discriminator (a higher hint rate helps Discriminator to identify imputed data instance

and lowers its loss). Alpha is for predefined hyper-parameter to help update Generator using stochastic gradient descent (SGD; Yoon et al., 2018).

The grid search provided by GridSearchCV exhaustively generates candidates from a grid of parameter values specified with the param_grid parameter. For instance, to tune hint rate and alpha, batch size and epoch number should be set to the same value (Figure 10). The parameters (i.e., p_hint and alpha) input to the mode are updated with a for loop going through the param_grid list.

```
param_grid = [
    {'mb_size': 100, 'p_hint': 0.9, 'alpha': 2, 'epoch': 300},
    {'mb_size': 100, 'p_hint': 0.9, 'alpha': 4, 'epoch': 300},
    {'mb_size': 100, 'p_hint': 0.7, 'alpha': 8, 'epoch': 300},
    {'mb_size': 100, 'p_hint': 0.7, 'alpha': 10, 'epoch': 300},
    {'mb_size': 100, 'p_hint': 0.5, 'alpha': 2, 'epoch': 300},
    {'mb_size': 100, 'p_hint': 0.5, 'alpha': 8, 'epoch': 300},
    {'mb_size': 100, 'p_hint': 0.1, 'alpha': 10, 'epoch': 300}
]
```

Figure 10 Grid Search Hyperparameter Setting Example

From the initial comparison, GAIN performs worse on the fall season dataset with a missing rate of less than 60% (Figure 7a, b). This dataset is adopted for model tuning. Batch size is tested at 50, 100, 500, and 1000 (the default was 128). It stopped at 1000 based on the assumption that a larger batch size reduces the stochasticity of the gradient descent and can cause model overfitting. The hint rate is put to the range of 0.1 to 0.9, with a set increase of 0.2. Alpha is set from 2 to 20, with a set increase of 2. Results are

plotted into grids with different color stand for averaged RMSEs from DCLs. This comparison suggests a best configuration among experimented settings with:

- Batch 1000
- Hint Rate 0.7
- Alpha 20

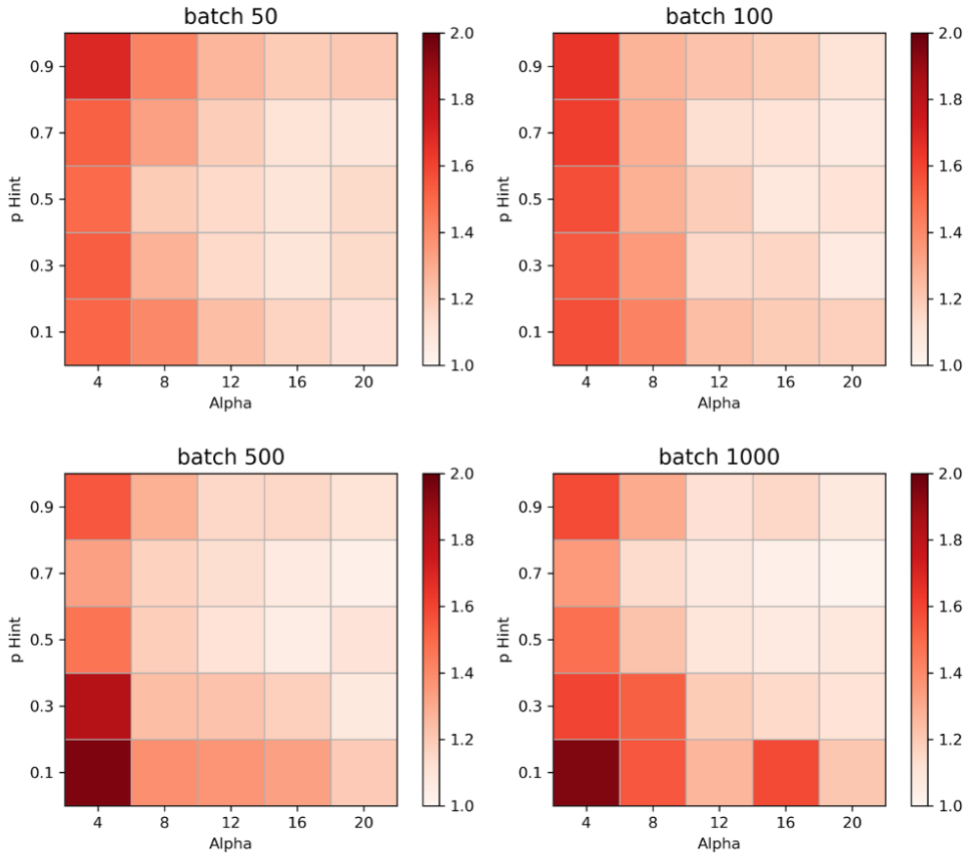


Figure 11 GAIN Hyperparameter Tuning Grid Search Result

Different from the idea of conventional machine learning, one epoch in GAIN means that one batch with randomly selected samples has had an opportunity to update the internal model parameters (instead of working through the entire training dataset). Still, number of epochs decides the times a model gets updated from learning. Using the parameter setting defined from previous grid search, model performances are tested at the epoch size of 200, 400, 600, …, 2000 in the same Google Colab Pro environment (i.e., Intel(R) Xeon(R) CPU @ 2.30GHz (* 4); Tesla GPU P100-PCIE-16GB; Memory 26 GB). The results are averaged in five repeated experiments, and it is revealed that the "ultimate" GAIN model requires 1600 epochs, with a batch size of 1000, a hint rate of 0.7, and an alpha of 20 (Figure 12).

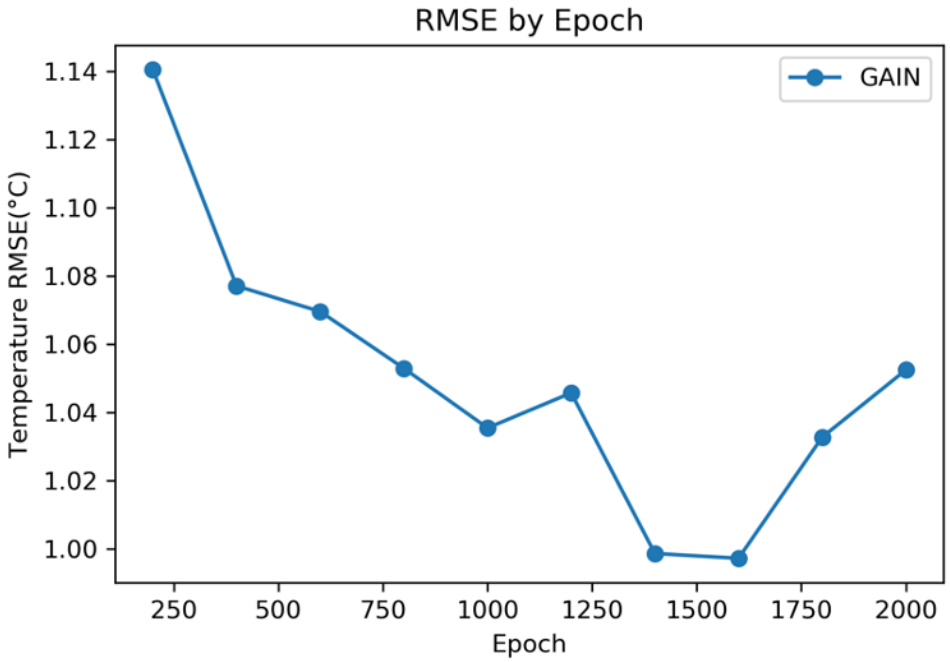


Figure 12 Epochs for an "Ultimate" GAIN

Tunned model parameter settings are adopted for the rest of this chapter for more performance comparisons. Using fall data, tunned GAIN shows great improvement at missing rate set to less than 60% (Figure 13). Both the mean RMSE and maximum match closely to Kriging and MissForest, but with a much faster data filling speed. The RMSE also dropped at a missing rate less than 80%.

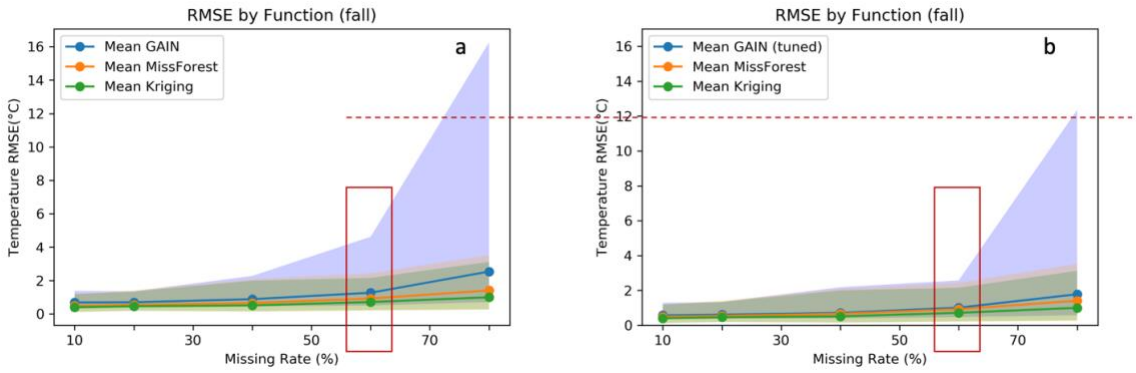


Figure 13 Performance of GAIN After Tuning Comparison

Looking at the RMSE spatial distribution at 60% (Figure 14), the overall RMSE has dropped with overall lighter color marks throughout all DCLs. Like the still high maximum RMSE demonstrated on Figure 13b at 80% missing rate, though smaller with ~25% decrease, there are few dark red spots located in the downtown LA. This experiment explains how model tuning can be effective on the tuning setting (i.e., data filling for fall season at missing rates < 60%) and expand to other settings (e.g., larger

missing rate at 80%). However, extreme cases (the dark reds on Figure 14d) for different settings should be further adjusted for potential better fits.

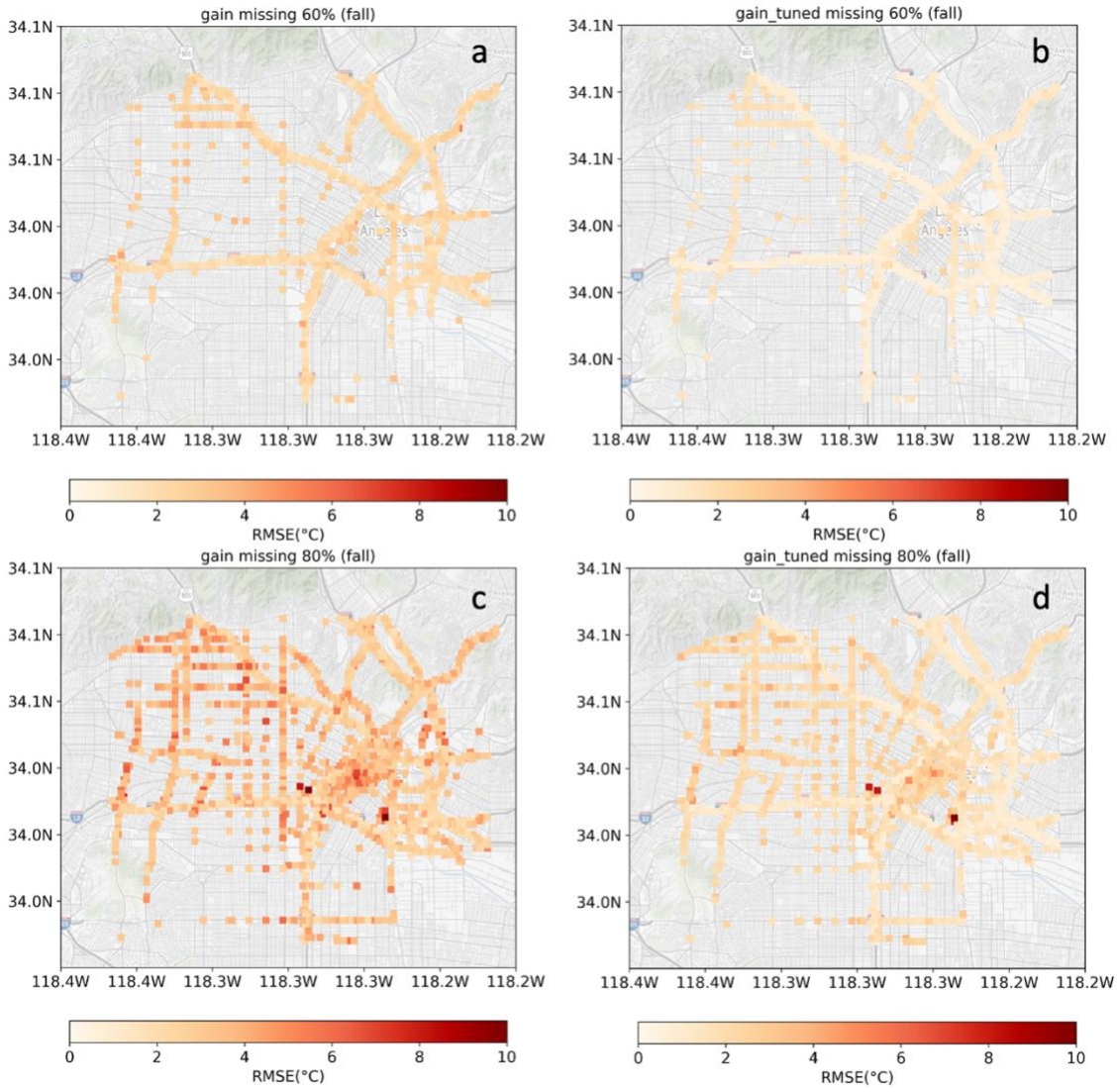


Figure 14 RMSE Spatial Distribution of GAIN After Tuning Comparison

Seasonal Data Filling

Experiments composed in this section are based on the GAIN model with tuned parameters due to its effectiveness. All three models are compared under all missing rate settings for all four seasons. GAIN provides competitive accuracy (average RMSE) for all four seasons compares to Kriging and MissForest across all settings (Figure 15). Like introduced, GAIN does not handle well for edge cases at 80% missing rate, generating a high RMSE at these DCLs and leading to significantly large maximum RMSEs in shaded areas for all seasons.

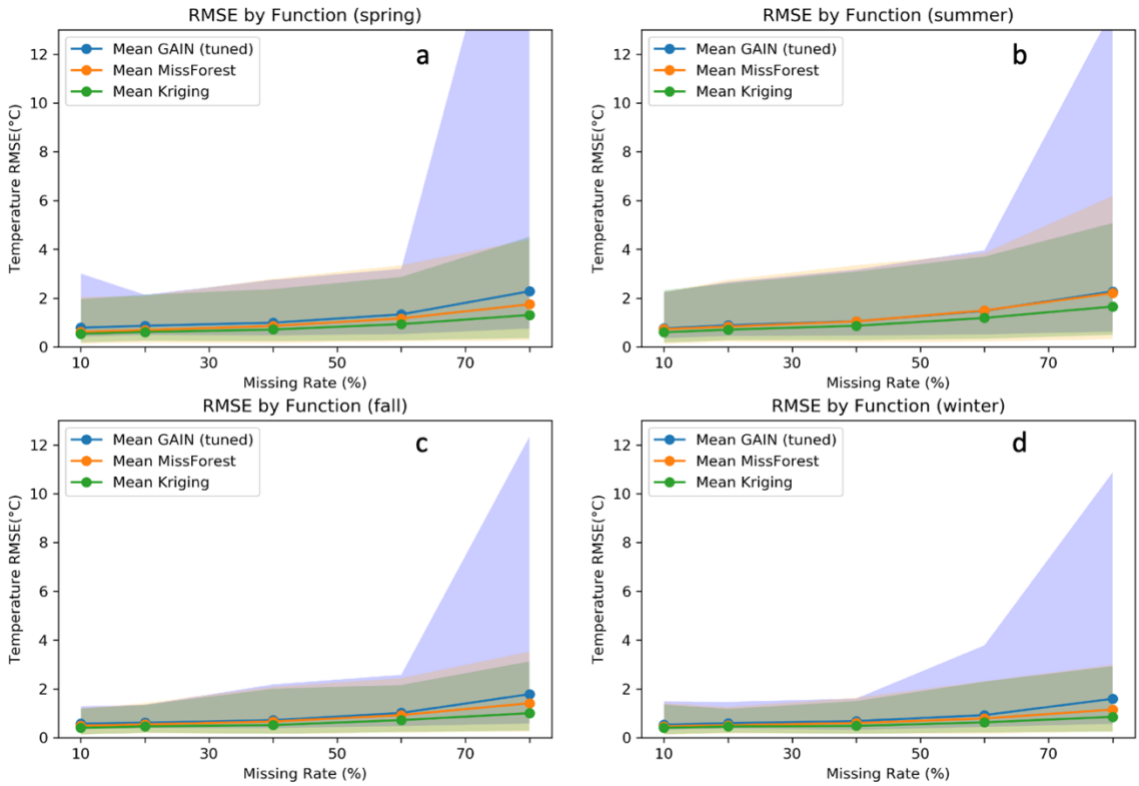


Figure 15 Seasonal Data Filling RMSE Comparison

To look closely at the numbers, at a missing rate < 60%, GAIN gives average Normalized RMSEs (NRMSE) close to or smaller than MissForest with lower standard deviations (Table 3). A low standard deviation indicates RMSEs are clustered around the mean, providing more stable predictions across all DCLs. The differences can be explained by the model algorithm. The GAIN model selects a subset of data randomly from a separate training dataset during each training epoch, enabling it to be more robust when handling edge cases (after tuned). On the contrary, MissForest performs only on a given test dataset with mean imputation initialization, making it less adaptive to certain edge cases when not enough observations during random forest model fitting. Kriging offers the lowest RMSE with minimum standard deviation, making it the optimal algorithm for IoT missing temperature filling at all settings without SMBs. The outstanding performance of Kriging is majorly due to the spatial density of the IoT dataset since this geostatistical method is heavily reliant on surrounding observations.

Table 3 Data Filling Performance in Terms of RMSE (Average ± Std of RMSE)

Missing < 60%	**Temp (℃)**	**Kriging**	**MissForest**	**GAIN**
Spring	16.99 ± 4.40	0.93 ± 0.52	1.17 ± 0.64	1.33 ± 0.53
Summer	22.86 ± 4.78	1.19 ± 0.69	1.49 ± 0.90	1.48 ± 0.77
Fall	20.77 ± 5.90	0.72 ± 0.39	0.93 ± 0.47	1.01 ± 0.42
Winter	14.64 ± 4.39	0.63 ± 0.29	0.78 ± 0.33	0.92 ± 0.37

Unlike GAIN that trains and tests on separate data, Kriging and MissForest only perform on given testing sets to fill the missing values. The IoT temperature data

collection only covers one year, meaning GAIN could be trained on datasets with missing patterns that exist in the testing set. To avoid the bias towards Kriging and MissForest, random data selection for model benchmarking is performed (Figure 16). The filling accuracy shows GAIN outperforms MissForest at most settings (missing rate < 20%, 40%, and 60%), and Kriging still gives minimum RMSEs.

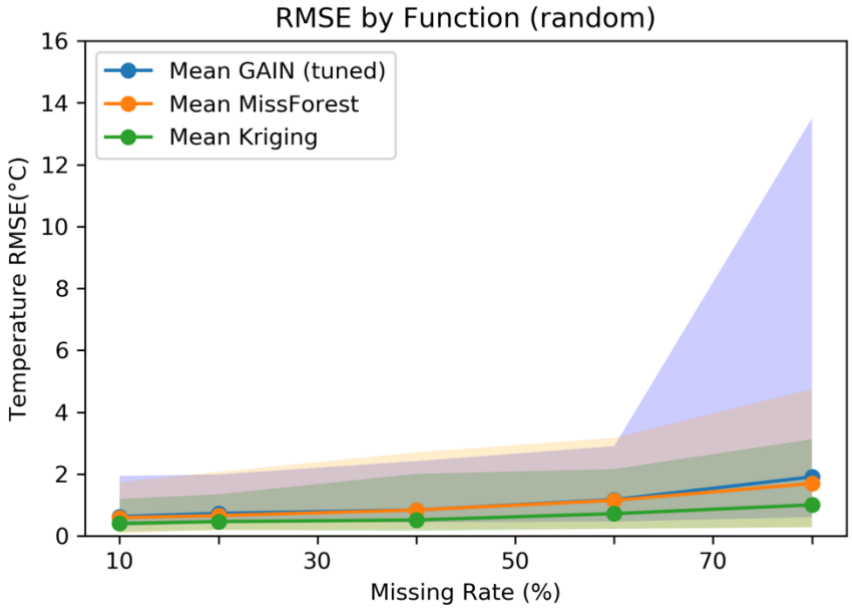


Figure 16 Data Filling RMSE on Random Testing Sets

Time series data patterns are extracted for certain DCLs, since low prediction accuracy appears repeatedly for different seasonal tests when we examine the RMSE spatial distribution. Two DCLs (i.e., '9q5csmk' and '9q5csmp') are displayed here as examples. Illogical observation data appeared in late July 2019 with sudden drops. Temperature declined over 70°C within one hour then jumped back. The lowest temperature recorded for LA in 2019 is

3°C on January 02, 2019, making these recordings unreliable from the IoT data collection. To better perform data filling, extra data correction and cleaning is necessary, despite the GeoTab claims that anomalous data and outliers have been removed. Data correction remains challenging due to the lack of other ground truth observations that can provide such high spatiotemporal resolution, and a future study on the topic is crucial for better missing data filling.

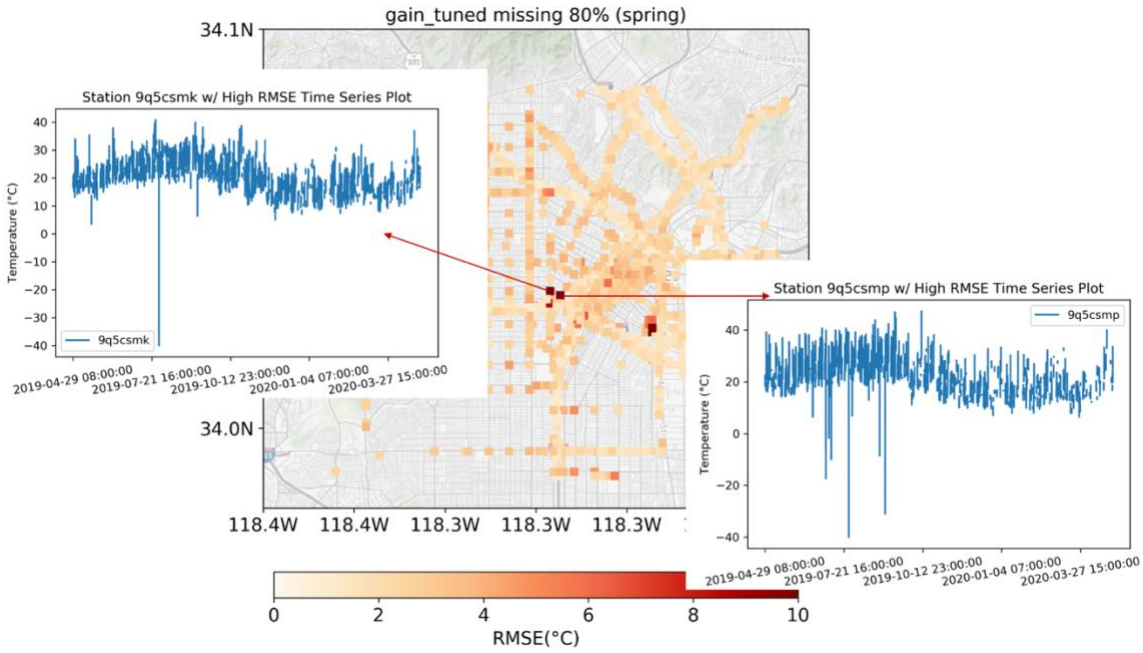


Figure 17 Time Series Plots for DCLs with High Data Filling RMSE

All models compared in this research have their limitations, either compromised on the accuracy at high missing rate DCLs, or large runtime. Kriging outperforms MissForest and GAIN but is incapable of filling values within SMBs. To optimize temperature missing data filling, a hybrid missing data filling scheme is expected to

integrate the fast computation speed from GAIN, high accuracy from Kriging, and all data filling capability from GAIN or MissForest.

CHAPTER FOUR. TEMPERATURE PREDICTION AND TRANSFER LEARNING FRAMEWORK

Data Description

Compares to the missing data filling (Chapter 3), more regions are selected to complete temperature prediction study, i.e., Los Angeles (LA), New York City (NYC), Atlanta (AT), and Chicago (CHI). GeoTab IoT temperature data are collected in the same way as introduced in Section 3.1.1 for all additional three regions aside of LA. From the largest to the smallest sensor network, the four study regions have DCLs of 38,175 (NYC), 23,300 (LA), 24,662 (CHI), and 11,377 (AT). Like previously introduced, temperature is recorded vehicularly, the number of DCLs is ideal when all DCLs provide data (i.e., having vehicles driven continuously). Due to the missing temperature filling still has uncertainty, only DCLs with < 5% missing data are used to optimize model training (e.g., 927 DCLs in LA out of 23, 300) in this study. As sensors are truck-mounted, data coverage is best in the major cities with a higher density of road networks and traffic. The maps (Figure 18) of GeoTab DCLs in the four regions show the selected DCLs located on major roads (high traffic volume for full data coverage). As an example, most of the selected DCLs in LA are located over the state freeway (e.g., Santa Ana Fwy, Golden State Fwy). The same applies to NYC, AT, and CHI.

Meteorological data are also collected for the four regions from Weather Underground (WU). Data details for the four regions are listed in Table 4. The resolution of ground-based observations varies by location, depending on the coverage of IoT DCLs or WU stations in different regions. Each data collection has different attributes (e.g.,

resolution, coverage), and these are presented with expected data processing techniques in the following sub-sections.

Table 4 Data Sources for IoT Temperature and WU Meteorological Observations

Dataset	***Org. & Data Source***	***Spatial Resolution***	***Temporal Resolution***	***Time Coverage***	***Variable***	***Role***
GeoTab IoT	GeoTab, Inc. https://data.geotab.com/weather/temperature	927 DCLs (LA) 1045 DCLs (NYC) 508 DCLs (AT) 269 DCLs (CHI)	hourly	2019-2020	Temperature	Data fusion. Both predictors and prediction target.
Weather Underground (WU)	IBM Weather Underground https://www.wunderground.com/about/data	100 stations (LA) 136 stations (NYC) 27 stations (AT) 2 stations (CHI)	hourly	2019-2020	Temperature, humidity, pressure, wind speed, UV index, etc.	Data fusion. Predictors.

Weather Underground (WU) Meteorological Data

The WU comprises individual (250,000 globally) collected measurements from the environment using personal weather stations (PWS), providing hyperlocal data. These observation stations are more sparsely distributed as compared to IoTs (Figure 18). And like IoT data sources, station observations are ground-based.

Since temperature is often affected by other meteorological variables, one advantage of utilizing WU is the concurrent acquisition of additional weather metrics (e.g., temperature, humidity, wind speed, pressure, cloud coverage). Observations from WU enrich the model training features after data fusion and improve the accuracy of predictions (Table 5).

Table 5 Filed information of WU PWS data

Field	Type	Description
Time	DATE	UTC date when the data was recorded
summary	STRING	Weather description like “clear”
Icon	STRING	Weather description icon display on map like “clear-day icon”
precipIntensity	FLOAT	Precipitation Intensity
precipProbability	FLOAT	Precipitation Probability
precipType	FLOAT	Precipitation Type
temperature	FLOAT	Temperature (in °F)
apparentTemperature	FLOAT	Apparent Temperature (in °F)
dewPoint	FLOAT	Dew Point (in °F)
humidity	FLOAT	Humidity
pressure	FLOAT	Pressure
windSpeed	FLOAT	Wind Speed
windGust	FLOAT	Wind Gust
windBearing	FLOAT	Wind Bearing
cloudCover	FLOAT	Cloud Cover
uvIndex	FLOAT	UV Index
visibility	FLOAT	Visibility
Ozone	FLOAT	Ozone
local_datetime	DATE	Local date when the data was recorded
Lon	FLOAT	Longitude
Lat	FLOAT	Latitude
precipAccumulation	FLOAT	Precipitation Accumulation

Study Area

As stated, four major cities in the U.S. are selected to build and test temperature prediction models (Figure 18). Cities have presents different DCL patterns due to differences in the road network density and traffic flow. On the maps, gray dots represent all DCLs, and blue triangles are selected DCLs. Red squares are WU locations.

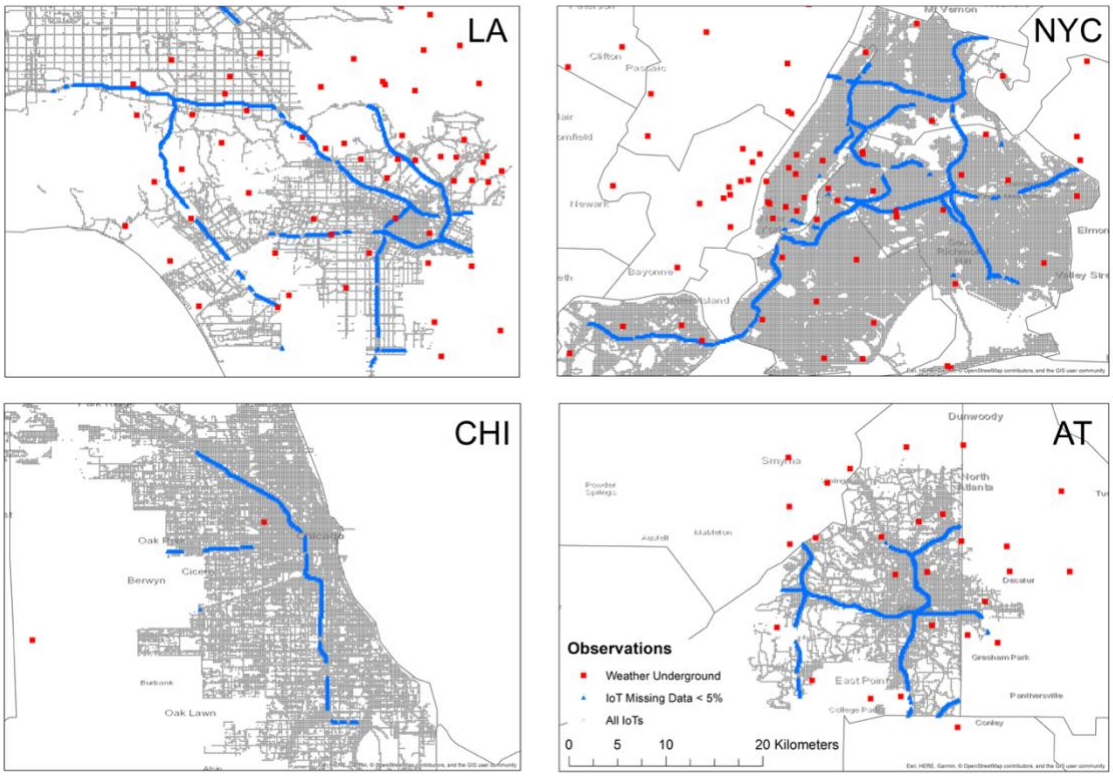


Figure 18 IoT and WU Data Distribution

High-resolution Multivariate Temperature Predictions

Before starting the major task of temperature prediction, we structured a framework that presents as a workflow to compose all features required to be established. Data fusion is crucial for multivariate prediction, and a parallel matrix manipulation allows fast nearest neighbor paring is introduced. The structure of LSTM is detailed to explain the reason for it to outperform others and selected as the core model of this prediction framework. Training procedures are documented to help audience replicate our experiments with datasets of their choice.

Framework

The framework for temperature prediction (Figure 19) consists of four modules, starting with the acquisition of the temperature metrics/predictors streamed with the target temperature through the data fusion module. From this module, data from different sources are integrated into spatiotemporal dimensions (Yang et al., 2020b). This fusion dataset enables the later multivariate prediction with accuracy improvements. LSTM construction as a second module is customized to best fit the integrated dataset for temperature prediction optimization. Train-test data split allows models to learn better with less overfitting and more robust. Training dataset feed to the LSTM, and model performance is evaluated using different prediction scenarios at the following module. Prediction model as a module is interchangeable, different models are tested and compared to generate optimal results. Transfer learning is the last step to further enhance the multi-step prediction accuracy, using a well-established pre-trained model, for regions with data scarcity and high initial prediction errors.

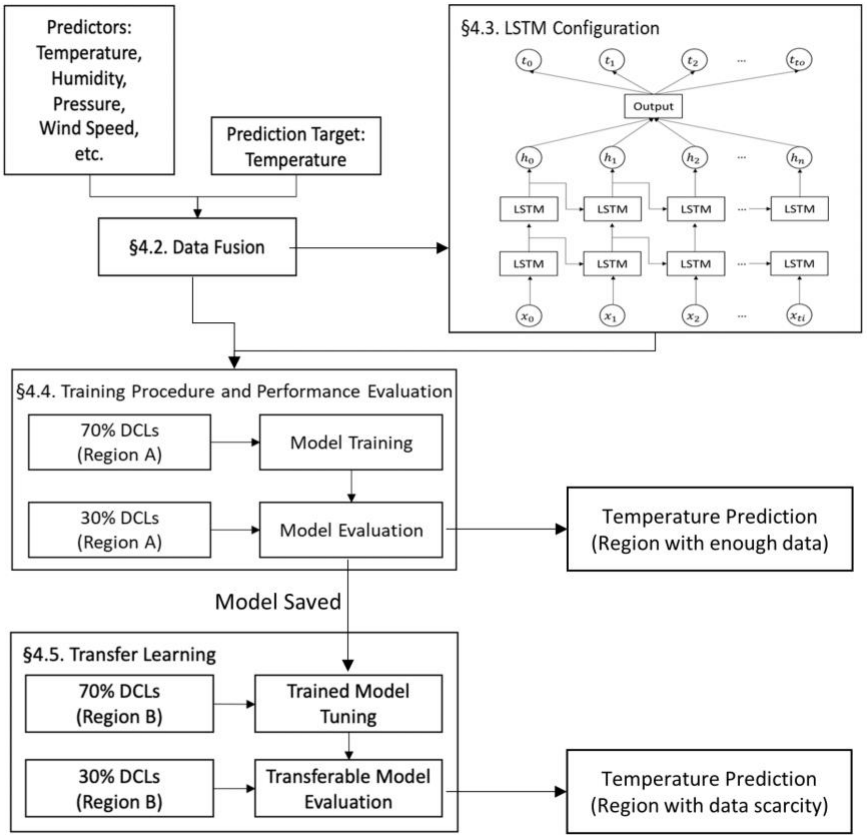


Figure 19 Framework for Temperature Prediction

Data fusion

Data fusion integrates the extra weather observations (i.e., humidity, pressure, wind speed, dew point, precipitation probability, cloud cover, and UV index) from WU stations with IoT DCLs and generates a single fused dataset (Figure 20).

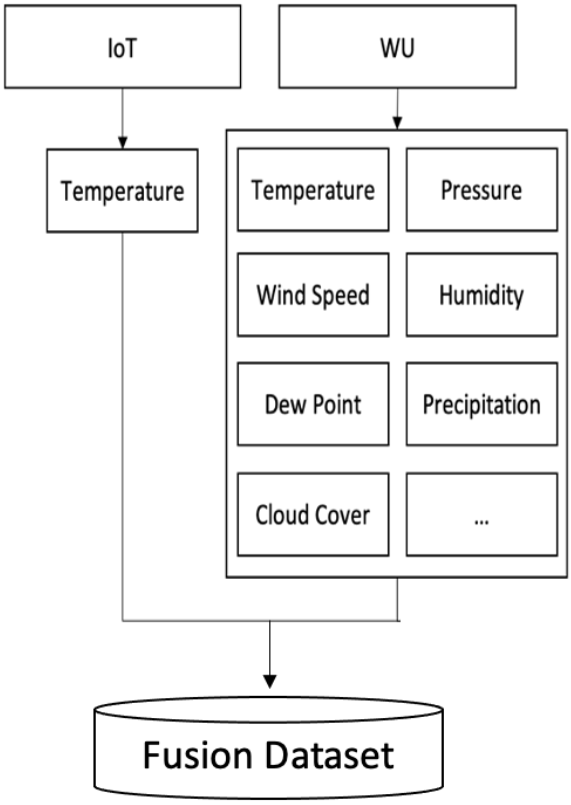


Figure 20 Data Fusion

One set of Sensor Distance Matrices is constructed for each region. The Sensor Distance Matrix stores the Euclidean Distance from WU stations to IoT DCLs in spatiotemporal dimensions (Yang et al., 2020b), after which each IoT DCL is paired with its nearest WU station. This fusion dataset remains the set of DCLs from IoTs but with additional weather observations from WU integrated. Distances are calculated using matrix construction and vectorization to achieve fast parallel computation (Equations 7 - 9).

Equation 7 Matrix Outer Addition

$$P = WU[x_0, \dots, x_k, y_0, \dots, y_k]^2 + IoT[x_0, \dots, x_i, y_0, \dots, y_i]^2$$

$$= \begin{bmatrix} p_{[0,0]} = WU[x_0]^2 + IoT[x_0]^2 + WU[y_0]^2 + IoT[y_0]^2 & \cdots & p_{[k,0]} \\ \vdots & \ddots & \vdots \\ p_{[0,i]} & \cdots & p_{[k,i]} \end{bmatrix}$$

Equation 8 Dot Product of Two Matrices

$$M = \begin{bmatrix} IoT[x_0] & IoT[y_0] \\ \vdots & \vdots \\ IoT[x_i] & IoT[y_i] \end{bmatrix} \cdot \begin{bmatrix} WU[x_0] & \cdots & WU[x_k] \\ WU[y_0] & \cdots & WU[y_k] \end{bmatrix}$$

$$= \begin{bmatrix} m_{[0,0]} = WU[x_0] * IoT[x_0] + WU[y_0] * IoT[y_0] & \cdots & m_{[k,0]} \\ \vdots & \ddots & \vdots \\ m_{[0,i]} & \cdots & m_{[k,i]} \end{bmatrix}$$

Equation 9 Euclidean Distance Matrix

$$D = \sqrt{(P - 2M)}$$

, where i for $i \in I$ is each DCLs from IoT dataset, and k for $k \in K$ is each station from WU. x, y are longitude and latitude respectively. The P and M are two matrices constructed for each region, and as inputs for distance matrix (D) calculation.

A model input for each training has a size of a selected time-window (T) multiplied by the number of variables (N, N*T). The weights for each parameter are updated within each learning process, stored in trained models, and fine-tuned when applied to a different region during transfer learning.

LSTM

The LSTM is a recurrent neural network (RNN), and like all other RNNs have a chain-like structure (Figure 21). The key to LSTMs is the cell state, which allows the LSTM to remove or add information through different gates. The three gates (i.e., forget gate, input gate, output gate) are composed of a sigmoid neural network layer and a

pointwise multiplication operation. The sigmoid layer outputs numbers between 0 (discard all information) and 1 (keep all information). The forget gate (f_t, Equation 10) determines what information is eliminated from the cell state, the input gate (i_t, Equation 11) determines what new information is stored in the cell state, and the output gate (o_t, Equation 12) determines the output.

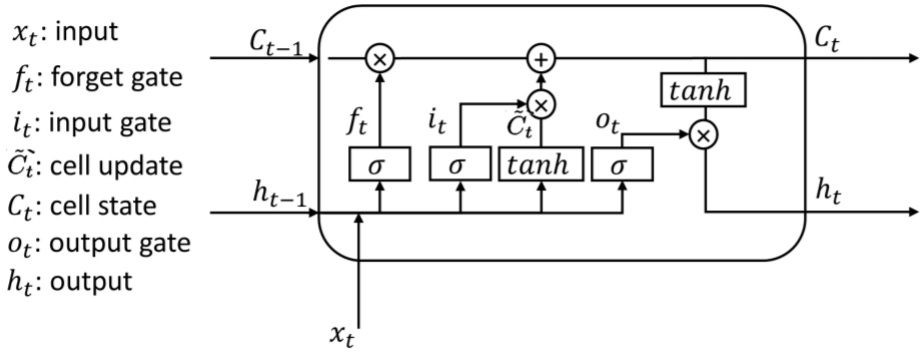


Figure 21 LSTM Model Structure

In Figure 21, boxes are sigmoid functions, and circles are multiplication operations. From left to right, the cell state ($\boldsymbol{C_{t-1}}$) and output ($\boldsymbol{H_{t-1}}$) from the last module with new input ($\boldsymbol{x_t}$) are going through different sigmoid functions and multiplication operations. Cell state ($\boldsymbol{C_t}$) and output ($\boldsymbol{H_t}$) for the current module are updated accordingly and passed to the next module.

The associated equations with the three gates are the following:

Equation 10 LSTM Forget Gate

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f)$$

Equation 11 LSTM Input Gate

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i)$$

Equation 12 LSTM Output Gate

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o)$$

, where w_x and b_x are the weights and biases, respectively for each gate.

The LSTM allows multiple input variables, a significant benefit in time series prediction, as classical linear methods are difficult to adapt to multivariate prediction problems. In this study, the fusion dataset is formatted into three dimensions when feeding into LSTM and includes samples, timesteps, and multivariate. The LSTM seamlessly supports continuous multi-step prediction, producing multiple neurons representing the incremental timestamp (N).

Training Procedure and Performance Evaluation

The framework generalizes multi-step temperature prediction. During the prediction within each region, 70% of the total DCLs from the fusion dataset are used for training (Figure 22). The model gains generalization by training on many stations across the region. Generalization from each model trained in different regions varies on the DCL distributions, assuming the more sparsely distributed sites (e.g., LA) are better generalized. The remainder (30%) of the DCLs are used for the evaluation of the local model generalization. Within the 70% training DCLs, 70% recordings are used for training (30% for validation). A model is considered as well-generalized when showing accurate prediction on the testing DCLs. Only well-generalized models are used for predictions.

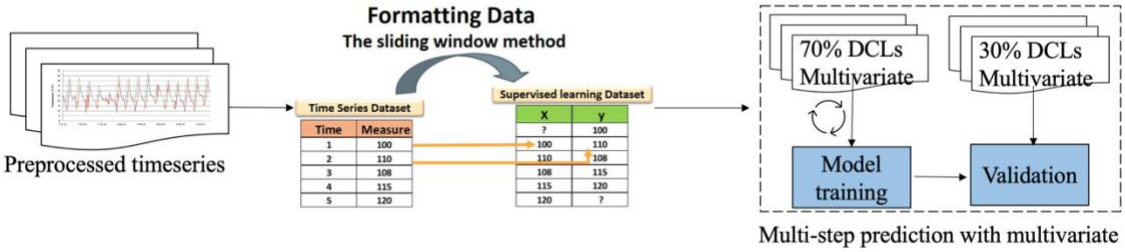


Figure 22 Temperature Prediction Training Data Processing

The full potential of LSTM is revealed using hyperparameter tuning, which optimizes prediction accuracy while avoiding model overfitting. Parameters include the number of LSTM layers, number of nodes in each LSTM layer, learning rate, dropout rate, number of epochs, and different optimizers. Dropout is a regularization method in which input and recurrent connections to LSTM units are probabilistically excluded from activation and weight updates while training a network. This reduces overfitting and improves model performance. Early stopping stops model iteration when errors increase. The effect of each parameter is examined while keeping the other parameters fixed. The model is coded in Python (PyTorch GPU version). The eight weather features from data fusion are model inputs.

Results are evaluated using Root Mean Square Error (RMSE, Equation 6) and R-squared. The Mean Absolute Error (MAE, Equation 13) is adopted as an additional measurement for comparison to other research in the literature.

Equation 13 MAE

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|P_i - O_i|$$

, where O_i is the observed air temperature, P_i is the predicted air temperature, and n is the number of test samples.

Transfer Learning

Transfer learning is to improve learning in a new task by transferring knowledge from a related task that has already been learned. In any neural network, models are trained to find the optimized weights for prediction. Transfer learning is applicable since the first few layers of a DL model capture low-level features, and it is unnecessary to learn features de novo on similar data. Although the number of temperature observations varies by region, the relationship between temperature and the other weather parameters is similar. A pre-trained model on a source dataset from one region is fine-tuned on a target dataset from another region without modifications of the hidden layers of the network (Fawaz et al., 2019). In this study, all models trained in regions with sufficient data (e.g., LA model was trained using LA local dataset before transferring) are applied to the region with data scarcity for transferability tests (i.e., CHI). Transfer learning follows the workflow (Figure 23).

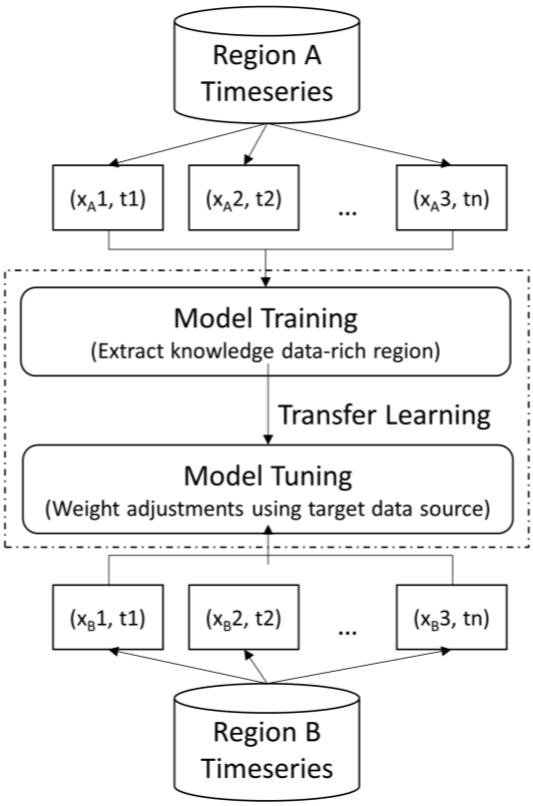


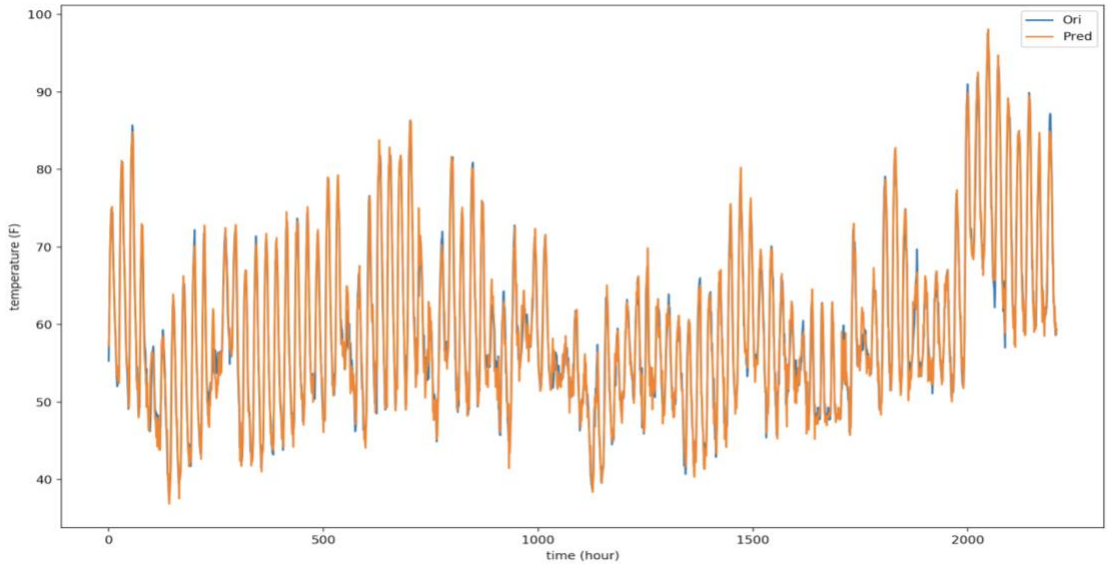
Figure 23 Transfer Learning Workflow

Assuming Region A (source region) has sufficient temperature recordings for model training, a model is saved after being trained and directly loaded for tuning in Region B, the target region with data scarcity. The tuned model is used later for target region temperature prediction.

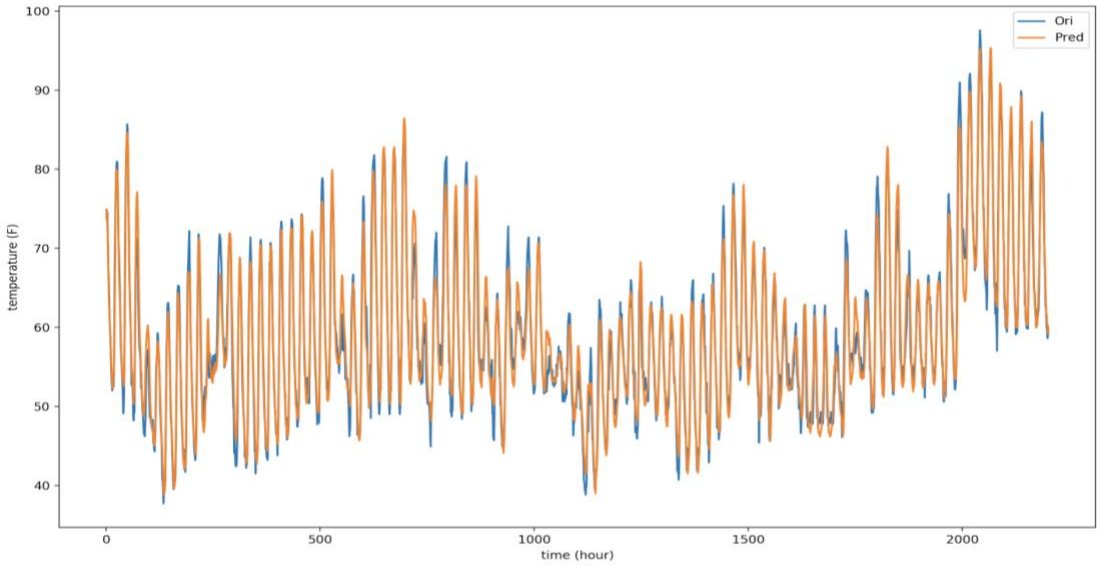
Experiments and Results

Model Performance Comparison

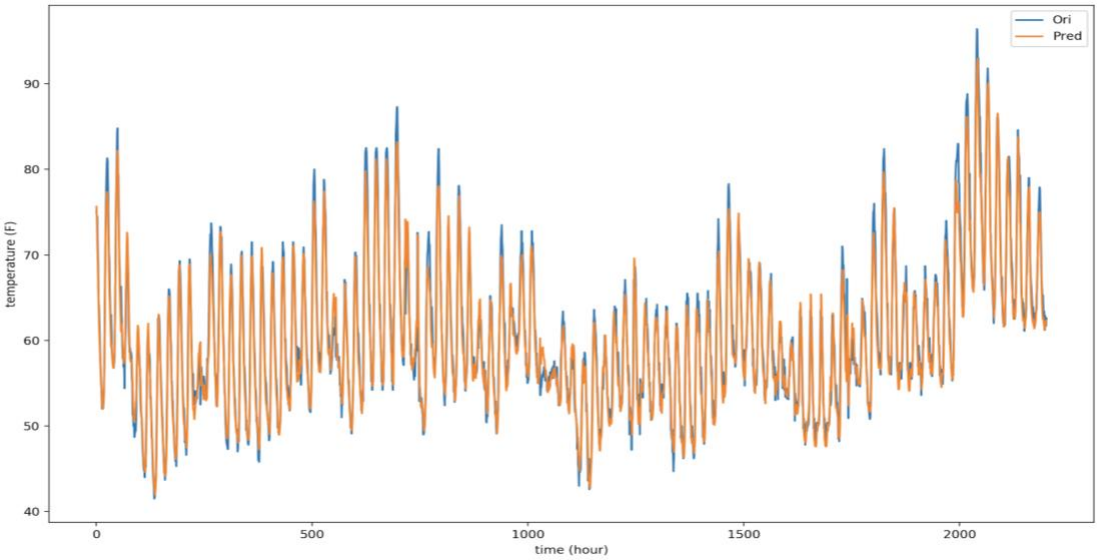
Two widely adopted ML models (i.e., ARIMA, XGBoost) are built for model comparison, starting with next-hour (1-step) temperature prediction for a selected DCL. To examine the model fitness on IoT temperature data, temperature is the only variable utilized as model input. RMSE and R^2 are calculated on testing data after model trainings for fitness evaluation. Predictions are plotted along with observations to display how accurate does each model predict (Figure 24).



ARIMA
RMSE: 1.94
R^2 = 0.97



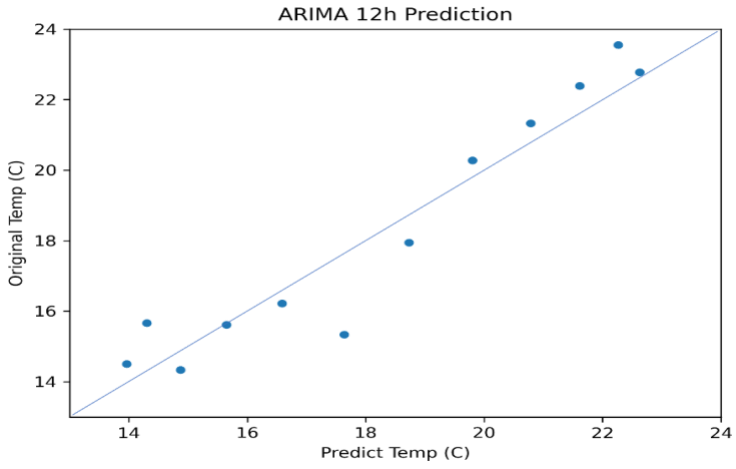
XGBoost
RMSE: 2.35
R^2 = 0.95



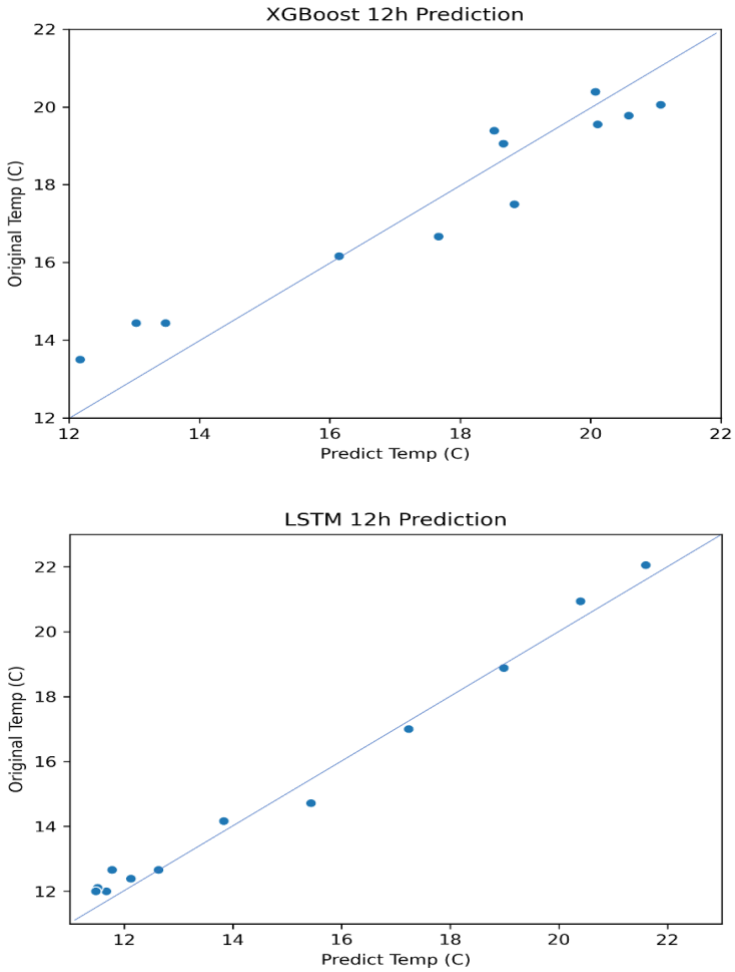
LSTM
RMSE: 1.85
R^2 = 0.98

Figure 24 Model Performance Comparison at Single Station (Univariate)

The model performance comparison at one selected DCL using only temperature observation suggests the superiority of LSTM. To extend the comparison of the selected models, experiments are developed for 12-step continuous temperature prediction using the past 24-hour multivariate weather observations (fusion data). The minimums (Min) indicate the best fitting DCLs and the maximums (Max) from the most unfitted DCLs. The LSTM yielded the best prediction results with the lowest RMSE (1.43) and highest R^2 (0.97) (Figure 25). Unlike LSTM and XGBoost, ARIMA does not support multivariate as input and only relies on the historical temperature measurements for prediction, resulting in the lowest accuracy among all models. In contrast to the high accuracy during univariate experiment (outperformed XGBoost), this indicates the advantage of data fusion. Although XGBoost allows multivariate input, the continuous prediction does not leverage the time dependency and produces less consistent results.



ARIMA
R^2 = 0.89
Max RMSE = 5.51
Min RMSE = 4.25



XGBoost
R^2 = 0.94
Max RMSE = 2.13
Min RMSE = 1.83

LSTM
R^2 = 0.97
Max RMSE = 1.94
Min RMSE = 1.43

Figure 25 Model Performance Comparison for 12-step Prediction with 24-hour Input (Multivariate)

Evaluation of the Localized Temperature Prediction

Model result validation and error assessment are essential for model evaluation. Models are only evaluated on their corresponding training regions. The evaluations using different lengths of data input for different prediction lengths show the average RMSE from all testing DCLs for each city (Figure 26a). The shaded area indicates the RMSEs expansion from the best fitted (Min RMSE) to the worst DCL (Max RMSE). The change

in R^2 across all multi-step prediction scenarios by city (Figure 26b) indicates that the longer the prediction length, the less accuracy in the model (higher RMSE and lower R^2). Starting from the beginning with significantly low prediction errors (e.g., using the previous 24-hour temperature to predict the next 1-hour), RMSE increases with the enlarged prediction length. Among all cities, the framework performed best in LA with the lowest RMSEs and the worst in CHI with multiple sudden increases in RMSE. Given that the number of adopted DCLs is only half that of New York, the AT has surprisingly low RMSE and high R^2.

Since the R^2 is a relative measure of fit and RMSE is an absolute measure, most prediction scenarios across all study regions have more distinguishable RMSEs with smaller differences in R^2. Cities with more data entry for model training perform better, mainly when predicting a more extended time range. The sudden drop in R^2 for CHI indicates that input weather measurements do not account for much of the temperature variation when predicting over the long term, due to the lack of training data. Together with the highest standard error, CHI is selected as the target region for transfer learning.

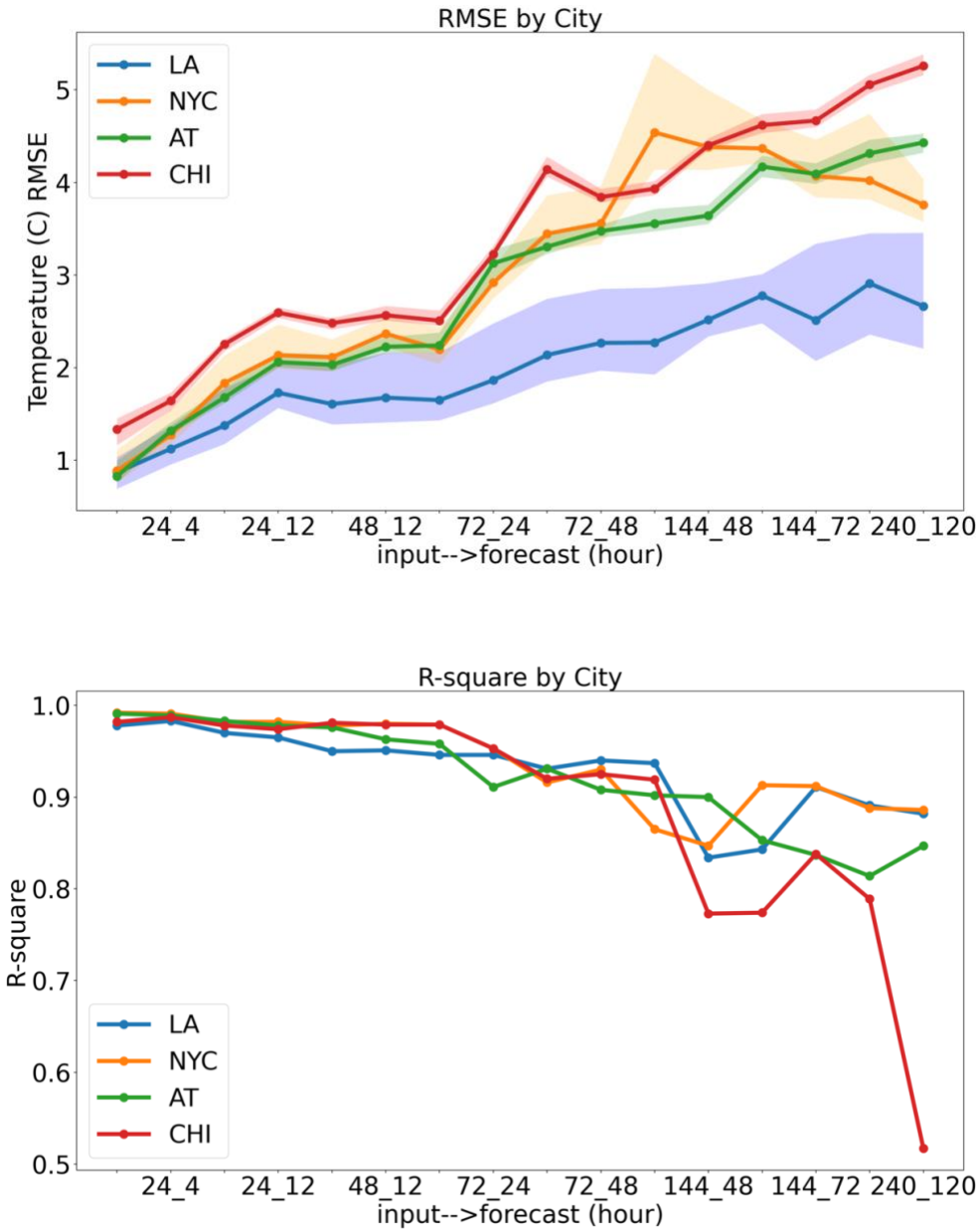


Figure 26 Multi-step Temperature Prediction Evaluation Using RMSE (a) and R2 (b)

The color-shaded area in (a) indicates the RMSEs expanded from the Min RMSE to the Max RMSE. Evaluations are plotted in Blue, Orange, Green, and Red for LA, NYC, AT, and CHI, respectively.

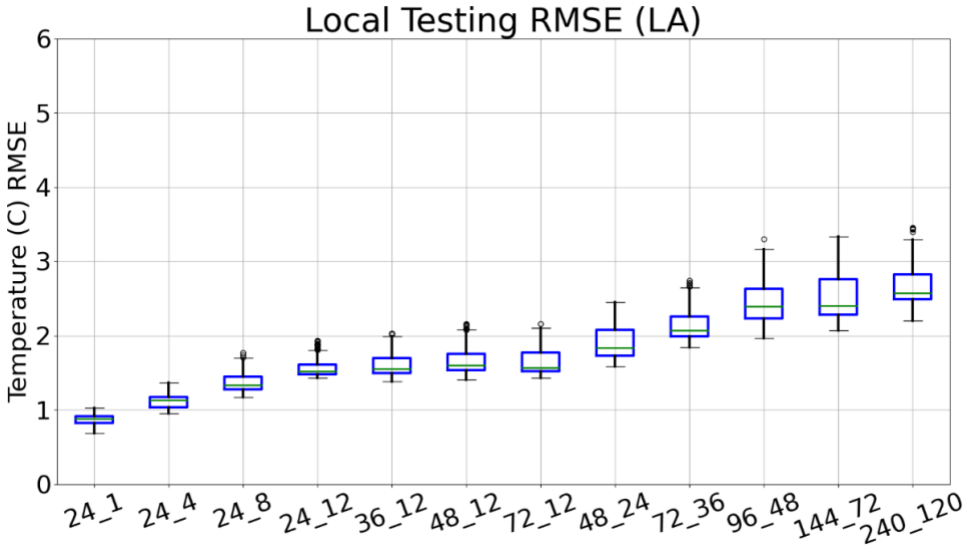
Compared with previous studies, the proposed framework shows 30-40% accuracy increases (Table 6). For the most extreme case (forecasting for the next 120 hours), the MAE from the best fitting DCL in LA (1.66°C) is ~10% smaller than that of other studies' next 12-hour prediction (1.87°C; Table 6).

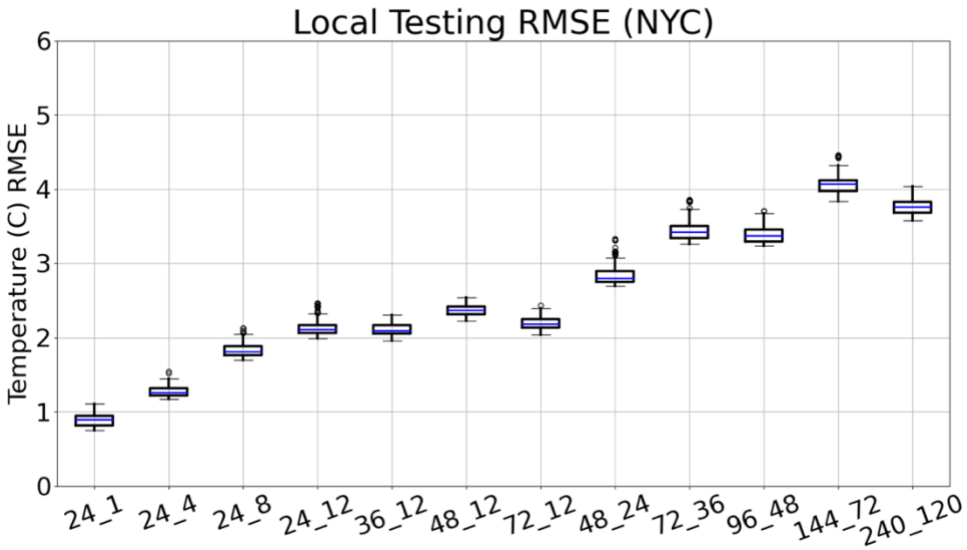
Table 6 Proposed Multi-step Predictions in Comparison to the Best Results Reviewed by Cifuentes et al. (2020)

MAE (°C)	Previous Studies	Proposed Framework (Study region)	Max Accuracy Change
4-step	1.20 SVM; Chevalier et al., 2011	0.91 (LA) 0.88 (NYC) 0.96 (AT)	-26.7%
8-step	1.62 Ward MLPNN; Smith et al., 2009	0.98 (LA) 1.37 (NYC) 1.22 (AT)	-39.5%
12-step	1.87 Ward MLPNN; Smith et al., 2009	1.13 (LA) 1.45 (NYC) 1.58 (AT)	-39.6%

The box plots illustrate the increasing dispersion of RMSEs with increasing prediction length (Figure 27). The experiments compare different multi-step prediction scenarios, where 24_4, 24_8, and 24_12 is conducted to compare with previous studies (Table 6). Experiments of 36_12, 48_12, and 72_12 show how enlarged input length can affect the 12-step continuous prediction accuracy (compares to 24_12). The comparison

shows that twice the input length (24_12) balances the overall prediction accuracy and the RMSE variance among DCLs (smaller box size). Therefore, the remaining experiments are performed using 2X input for 1X output. With the increasing RMSE, box size and the length of the whiskers in the LA chart increase as outliers emerge. The expanding box size and whisker length demonstrate the struggle of the model when predicting certain DCLs, which is expected as the DCL in LA expands through a larger region. Outliers also explain how the model balances accuracy and generalization when fitting into more considerable weather variations in a larger region. Smaller boxes and shorter whiskers with fewer outliers in the other three regions (NYC, AT, and CHI) support this as their data expands through smaller regions (Figure 18). Boxes in all regions show positive skewness, indicating more than half of the DCLs have lower RMSEs than the average. It is proposed that these models fit well for most DCLs, and only a small group of DCLs show low accuracy.


Local Testing RMSE (LA)
Temperature (C) RMSE
0
1
2
3
4
5
6
24_1
24_4
24_8
24_12
36_12
48_12
72_12
48_24
72_36
96_48
144_72
240_120


Local Testing RMSE (NYC)
Temperature (C) RMSE
0
1
2
3
4
5
6
24_1
24_4
24_8
24_12
36_12
48_12
72_12
48_24
72_36
96_48
144_72
240_120

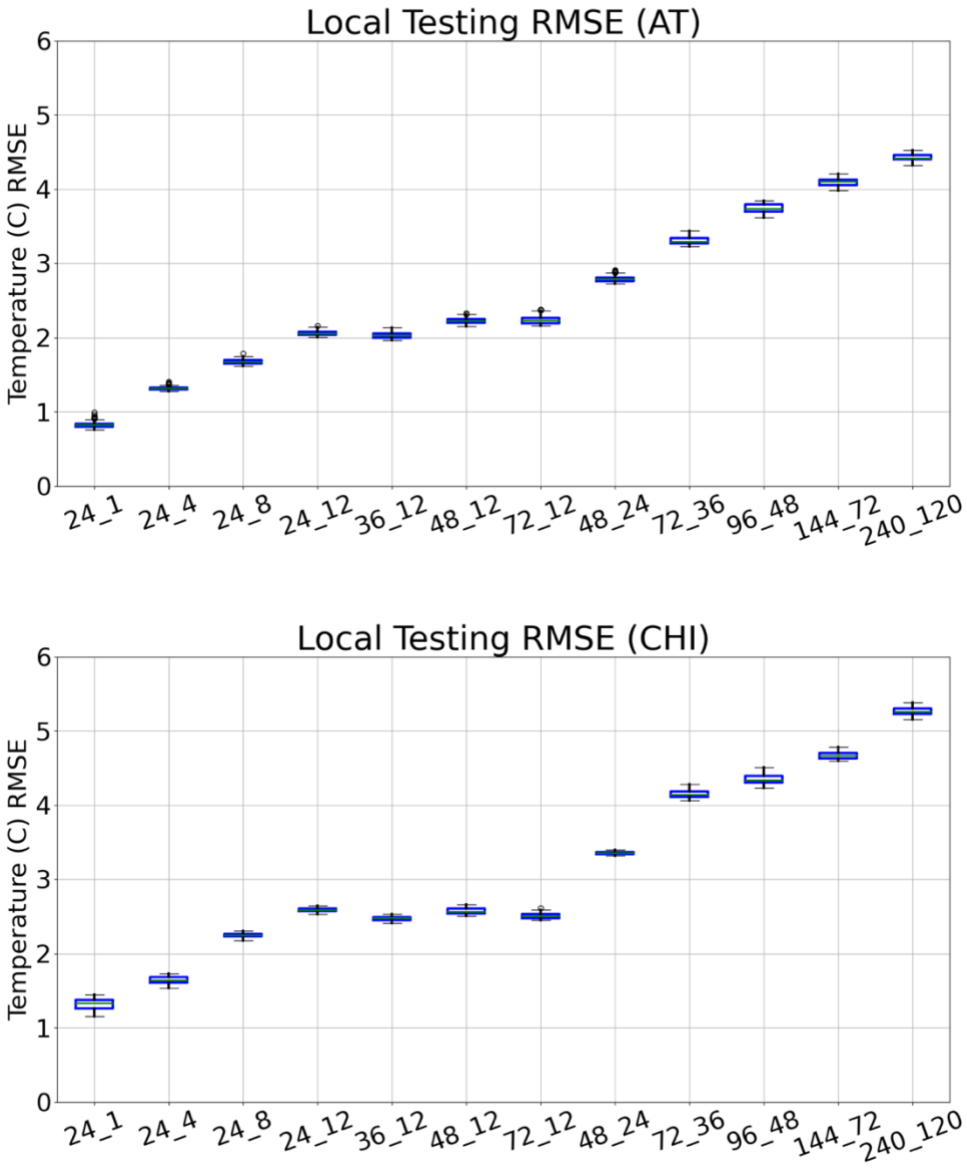


Figure 27 Multi-step LSTM with Multivariate Prediction Result Evaluation

Though increasing prediction length leads to rising RMSEs, introducing a longer input time range reduces the prediction error. For instance, when comparing the box plot for 24h→ 12h and 36h→ 12h, the latter yields better results. However, this is not always

the case: when evaluating the result from 36h→ 12h and 48h→ 12h, the increased input provides lower accuracy.

Transferability Evaluation

Chicago (CHI) is used as the target region to evaluate transfer learning as it has the lowest number of DCLs and yields the least satisfying accuracy during model testing (Figure 26). Models trained in the other three regions (LA, NYC, and AT) are directly loaded and fitted to the CHI dataset for model tuning (Section 4.3). Fine-tuned models are later used for CHI temperature prediction. Matrices are built for comparing the model performance during transfer learning (Table 7). Accuracy is compared using the same set of equations (RMSEs, R^2). Improvements are obtained for all, particularly when the prediction length (up to 25.7% enhancement) is enlarged. This is useful since the CHI locally trained models perform well when predicting shorter periods but struggle for longer period predictions due to data scarcity. The increased R^2 indicates how the transferable models better fit the CHI dataset than the CHI locally trained model.

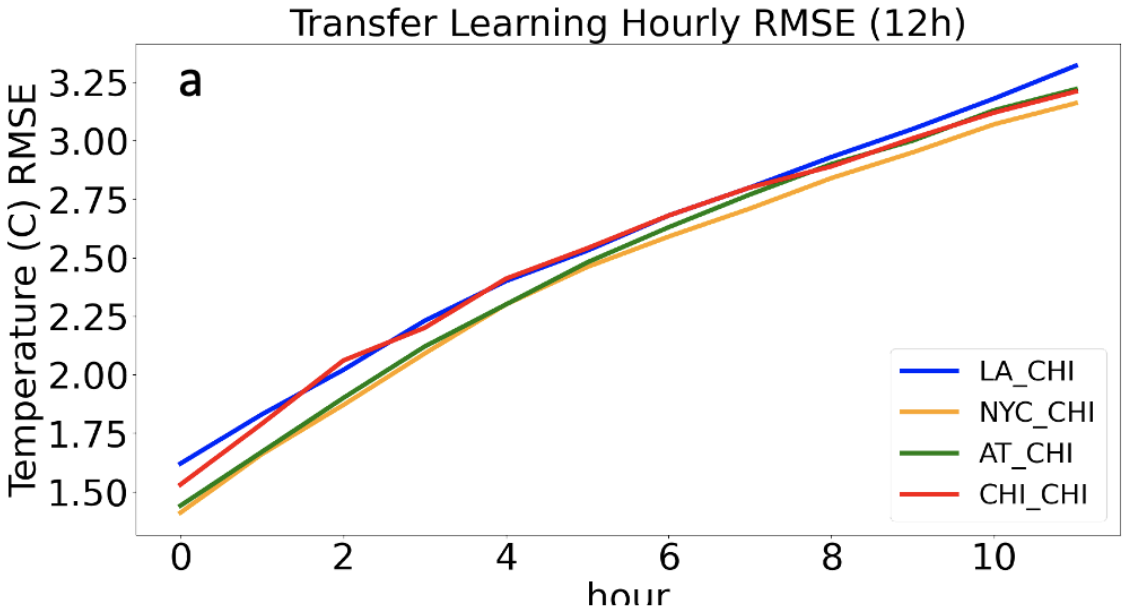
Table 7 Transferable Prediction Evaluation Matrix

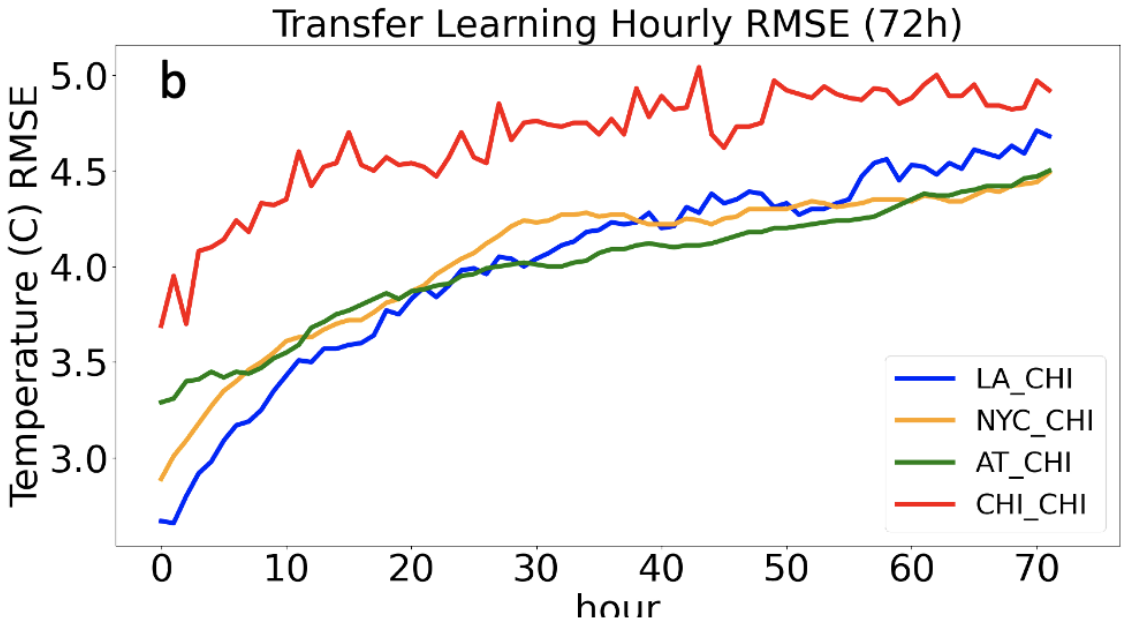
Accuracy Change	**CHI (baseline)**	**LA predicting CHI**	**NYC predicting CHI**	**AT predicting CHI**
24h→12h	R^2: 0.96 - -	R^2: 0.97 -0.7% RMSE -2.9% MAE	R^2: 0.98 -5.8% RMSE -7.8% MAE	R^2: 0.98 -7.7% RMSE -9.3% MAE
144h→72h	R^2: 0.70 - -	R^2: 0.92 -24.9% RMSE -24.2% MAE	R^2: 0.91 -24.3% RMSE -24.4% MAE	R^2: 0.87 -25.2% RMSE -24.2% MAE

240h→120h	R^2: 0.51 - -	R^2: 0.83 -13.9% RMSE -16.1% MAE	R^2: 0.87 -21.0% RMSE -25.7% MAE	R^2: 0.88 -16.5% RMSE -17.9% MAE

The CHI (baseline) is the prediction results from the CHI locally trained model. All accuracy changes are measured using Max RMSE (or MAE) change when applying transfer learning. Negative numbers indicate the RMSE (or MAE) decreases (i.e., better accuracy than baseline).

Hourly RMSEs are plotted to further assess how the adopted transfer learning reduces the prediction error due to data scarcity (Figure 28). The transfer models perform similarly to the CHI's local trained model when predicting short period temperatures (Figure 28a), where the RMSE increases with lead time. By increasing the prediction lead time, and the transferred models start to show better performance with steady growth in low standard errors (Figure 28b). In a more extreme case, the CHI model struggles initially, giving an overall much higher RMSEs (Figure 28c). The NYC model outperforms both the LA and AT models, displaying a near-flat RMSE across the 120-hour prediction.


Transfer Learning Hourly RMSE (12h)
a
Temperature (C) RMSE
3.25
3.00
2.75
2.50
2.25
2.00
1.75
1.50
0
2
4
6
8
10
hour
LA_CHI
NYC_CHI
AT_CHI
CHI_CHI


Transfer Learning Hourly RMSE (72h)
b
Temperature (C) RMSE
5.0
4.5
4.0
3.5
3.0
0
10
20
30
40
50
60
70
hour
LA_CHI
NYC_CHI
AT_CHI
CHI_CHI

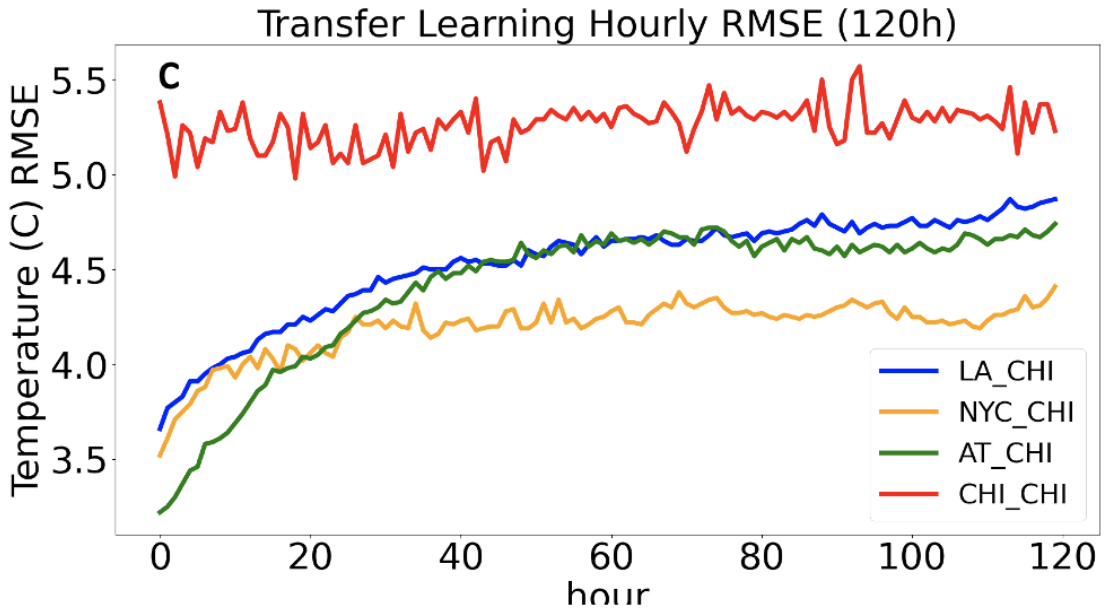


Figure 28 **Comparison of the Transfer Model Hourly RMSE.**
(a) shows prediction RMSE of each step (hour) when using 24-hour data input for 12-step continuous prediction. (b) shows 144-hour for 72-step. (c) shows 240-hour for 120-step. Evaluations are plotted in Blue, Orange, and Green for transfer models from LA (LA_CHI), NYC (NYC_CHI), and AT (AT_CHI), respectively. CHI locally trained model (CHI_CHI) as baselines is colored in Red.

Transfer learning is proven to be useful in cases of data scarcity. The effectiveness of these transfer learning models is due to the high spatial correlation (mostly major roads) and high density distributed of the IoT DCLs. The variation in performance is caused by the similarity differences shared among sources and target cities. The ideal density of DCLs for transferable model training is a future research initiative.

CHAPTER FIVE. CONCLUSION AND FUTURE WORK

Temperature is one of the major concerns of urban livability, and intensified global warming accentuates the temperature imbalance by increasing the already high temperatures in urban areas. A series of health and energy concerns are related to temperature. To better assist fine-scaled temperature observation and predictions for fast heat-related responses, this dissertation utilized vehicle-based IoT as a main data source. Well-established IoT networks achieve near-real-time street-level air temperature measurements, unlike satellite observations that often require downscaling for appreciable spatiotemporal resolution. The missing data issue on IoT datasets is inevitable, and therefore targeted missing temperature observation filling was studied by comparing different state-of-art data filling algorithms. Then a multivariate temperature prediction framework is proposed that adopts IoT observations and integrates meteorological observations from external data resources (WU). Lastly, transfer learning is successfully applied, enabling well-trained models to continuously predict highly accurate temperatures for tested regions with data scarcity.

Conclusion

The rich selection for missing data filling models granted possibility for better utilizing IoT datasets. The missing of IoT temperature observation is treated as the MAR mechanism, since the exact moment when one device stops functioning is generally unknown, and can be predicted, interpolated, or imputed using observed sensor readings. Selected algorithms (i.e., Kriging, MissForest, and GAIN) are considered for comparison

as state-of-art from their domain in statistics, machine learning and deep learning. Different algorithms demonstrated their advantages and weakness under different testing scenarios. All models built upon these algorithms are tested to fill the missing data at rate of less than 10%, 20%, 40%, 60%, and 80%. Testing data are then selected using either different seasons, or randomly draw from the entire dataset, to measure the stability of these models.

We defined spatial missing block (SMB) to introduce the situation when all IoT sensors do not have readings at a timestamp. Kriging outperforms MissForest and GAIN in data filling accuracy with lowest average RMSE and minimum error standard deviation, making it the most stable solution to fill datasets without SMBs. The nature of Kriging only allows it to fill missing data when close by observations are available. MissForest gives out competitive results across all test settings with RMSEs close to the Kriging results. However, the long model runtime makes it incapable of real-time data filling (over an hour to produce results). The tuned GAIN model as a deep learning model offers the most balanced performance when filling data with missing the rate < 60% but not on edge cases with the missing rate < 80%. The results are indistinguishable from MissForest but slightly less accurate than Kriging, with the fastest runtime. It takes less than 6 minutes to produce a filled dataset from training (Figure 9). Under the test with SMB, where Kriging was eliminated by default, and GAIN outperforms MissForest on most miss rate settings.

Data filling experiments are comprehensively studied; however, the result can be biased due to spatiotemporal information not being fairly integrated into all models.

Kriging uses spatial autocorrelation only and fills the missing data timestamp by timestamp, meaning it does not include any temporal information into consideration. For GAIN model, such information is partially participated, since each batch is composed as a random selection from the entire training data. The larger the batch, the higher the possibility that timestamps with similar patterns get selected. MissForest has the best spatiotemporal information integration, since it takes the whole time series in (or as long as it needs to be), and all the DCLs (spatial information) as attributes to train the model to the best. This could help explain the longer runtime for MissForest. Kriging adopts variogram that for each point, only nearest observations are used while calculating for each missing point, where MissForest must take all observations into consideration by default.

To accurately predict high spatiotemporal resolution urban temperature, a framework is proposed using IoT data fusion and deep learning. A data fusion technique is achieved by parallel matrix computation for fast heterogeneous data integration to enrich IoT data features. A fusion dataset is utilized for multivariate LSTM prediction support. Different multi-step prediction scenarios are tested. The LSTM as a DL algorithm is proven to offer advanced prediction capability for multi-step temperature prediction with multivariate on our fusion dataset compared to ARIMA and XGBoost. The generalized prediction model performs well on most of the local testing DCLs. The proposed framework enables effective and consistent predictions across different study regions and gives competitive accuracy compared to other methodologies. Temperature predictions completed on major roads (missing data <5%) were considered as sufficient

to represent the study areas since the distribution of these selected sensors was widely expanded. Even for the most extreme case tested (120-step prediction), the proposed framework outperforms the best model reviewed for 12-step prediction. Despite the encouraging results, a greater prediction length leads to a larger error, and the optimized input-output combination for different multi-step predictions needs to be explored. It is convincing that these models can be applied to the rest of DCLs if there are long enough consecutive recordings for model input, which can be achieved by properly integrating missing data filling algorithms.

The development of model transfer learning is derived from the reusability of trained models from regions with sufficient temperature observations. Transfer learning minimize the prediction error for regions with data scarcity problem and improves the predicting MAE up to 25.7%. Enhanced prediction accuracy from transferable models conquers the data scarcity problem and allows the proposed framework to be more widely adopted. The framework can be implemented in other weather parameter predictions (e.g., humidity, pressure) and is expected to assist city planning and management (Murphy, 1993).

Limitations of this temperature prediction framework are severalfold and warrant discussion below. Despite the high resolution of the IoT dataset, especially when comparing with traditional weather station measurements, this dataset does not offer complete coverage of uniformly distributed surface temperature observations as achieved using satellite images. The number of DCLs varies mainly from region to region, and the proposed transfer learning does not predict sites absent of DCLs. One solution is

spatiotemporal data interpolation to fill the area as grids. However, popular algorithms like IDW only achieve accurate results if the sampling of input points is dense or the results do not represent the desired surface (Watson and Philip 1985). Transfer learning showed different improvement levels and needed further exploration to understand the reasons to explain why some transferable models are better than others. For instance, the LA model has the highest local prediction accuracy but does not provide the best results when transferring to CHI.

For both studies, only one year of data is collected due to the limited access to the historical IoT dataset. A more prolonged time coverage for model training is expected to improve prediction by integrating yearly trends. The influence of climatic patterns should be considered for better SMB missing data fillings and long-term predictions.

Future Works

Based on the results from the dissertation, there are more to explore for potential enhancements:

1) IoT temperature data correction is necessary for better missing data filling and predictions. Error readings can cause a model to learn from inaccurate information, producing incorrect results. However, not many data resources are available at such high resolution for IoT data correction. Credible social media data that collects real-time information, followed by human mobility in the urban area, has the potential to help detect the false IoT sensor readings (Yang et al., 2019).

2) Explorations should be initialized for a better data filling scheme targeting spatiotemporal IoT data specifically, given the explosive growth of IoT earth observations. Different data filling mechanisms have their limitations. An ensemble missing data filling model is expected. Multivariate data filling models exist and should also be explored (Little and Schluchter, 1985). For instance, the diversifications of street blocks can be calculated and adopted as a parameter since differences in buildings and activities along the road network indicate that there are unevenly distributed anthropogenic heat releases (Teng et al., 2019).
3) Integrating a properly designed missing data filling model into temperature prediction can significantly improve the prediction coverage. This can be completed once the first two works are conducted and can provide reliable datasets. For a larger prediction coverage, a data storage and retrieval mechanism redesign should enhance the processing speed for missing data filling and time series prediction (Hu et al., 2018; Xu et al., 2021).
4) The established temperature prediction framework yields high accuracy and precision; however, it still struggles with longer-term forecasting. Climate change and temperature-related spatiotemporal event detection should step in to emphasize the long-term temperature pattern (Yu et al., 2020). Despite the computational efficiency of the proposed models and the framework, enabling cloud computing could allow larger-scale data manipulation (Li et al., 2020).

REFERENCES

Aalto, J., Pirinen, P., Heikkinen, J. and Venäläinen, A., 2013. Spatial interpolation of monthly climate data for Finland: comparing the performance of kriging and generalized additive models. Theoretical and applied climatology, 112(1), pp.99-111.

Anjali, T., Chandini, K., Anoop, K. and Lajish, V.L., 2019, July. Temperature Prediction using Machine Learning Approaches. In 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT) (Vol. 1, pp. 1264-1268). IEEE.

Ayele, T.W. and Mehta, R., 2018, April. Air pollution monitoring and prediction using IoT. In 2018 second international conference on inventive communication and computational technologies (ICICCT) (pp. 1741-1745). IEEE.

Batista, G.E. and Monard, M.C., 2002. A study of K-nearest neighbour as an imputation method. His, 87(251-260), p.48.

Barbulescu, A., Bautu, A. and Bautu, E., 2020. Optimizing inverse distance weighting with particle swarm optimization. Applied Sciences, 10(6), p.2054.

Breiman, L., 2001. Random forests. Machine learning, 45(1), pp.5-32.

Chammas, M., Makhoul, A. and Demerjian, J., 2019. An efficient data model for energy prediction using wireless sensors. Computers & Electrical Engineering, 76, pp.249-257.

Chen, F.W. and Liu, C.W., 2012. Estimation of the spatial rainfall distribution using inverse distance weighting (IDW) in the middle of Taiwan. Paddy and Water Environment, 10(3), pp.209-222.

Cheng, D., Zhu, D., Broadwater, R.P. and Lee, S., 2009. A graph trace based reliability analysis of electric power systems with time-varying loads and dependent failures. Electric power systems research, 79(9), pp.1321-1328.

Cifuentes, J., Marulanda, G., Bello, A. and Reneses, J., 2020. Air Temperature Forecasting Using Machine Learning Techniques: A Review. Energies, 13(16), p.4215.

Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L. and Muller, P.A., 2019. Deep learning for time series classification: a review. Data Mining and Knowledge Discovery, 33(4), pp.917-963.

Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial networks. arXiv preprint arXiv:1406.2661.

Harlan, S.L., Chowell, G., Yang, S., Petitti, D.B., Morales Butler, E.J., Ruddell, B.L. and Ruddell, D.M., 2014. Heat-related deaths in hot cities: estimates of human tolerance to high temperature thresholds. International journal of environmental research and public health, 11(3), pp.3304-3326.

Hattis, D., Ogneva-Himmelberger, Y. and Ratick, S., 2012. The spatial variability of heat-related mortality in Massachusetts. Applied Geography, 33, pp.45-52.

Henn, B., Raleigh, M.S., Fisher, A. and Lundquist, J.D., 2013. A comparison of methods for filling gaps in hourly near-surface air temperature data. Journal of Hydrometeorology, 14(3), pp.929-945.

Hewage, P., Trovati, M., Pereira, E. and Behera, A., 2020. Deep learning-based effective fine-grained weather forecasting model. Pattern Analysis and Applications, pp.1-24.

Hippert, H.S., Pedreira, C.E. and Souza, R.C., 2000, July. Combining neural networks and ARIMA models for hourly temperature forecast. In Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium (Vol. 4, pp. 414-419). IEEE.

Holdaway, M.R., 1996. Spatial modeling and interpolation of monthly temperature using kriging. Climate Research, 6(3), pp.215-225.

Hossain, M., Rekabdar, B., Louis, S.J. and Dascalu, S., 2015, July. Forecasting the weather of Nevada: A deep learning approach. In 2015 international joint conference on neural networks (IJCNN) (pp. 1-6). IEEE.

Hu, F., Xu, M., Yang, J., Liang, Y., Cui, K., Little, M.M., Lynnes, C.S., Duffy, D.Q. and Yang, C., 2018. Evaluating the open source data containers for handling big geospatial raster data. ISPRS International Journal of Geo-Information, 7(4), p.144.

Hu, Q., Zhang, R. and Zhou, Y., 2016. Transfer learning for short-term wind speed prediction with deep neural networks. Renewable Energy, 85, pp.83-95.

Jallal, M.A., Chabaa, S., El Yassini, A., Zeroual, A. and Ibnyaich, S., 2019, April. Air temperature forecasting using artificial neural networks with delayed exogenous input. In 2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS) (pp. 1-6). IEEE.

Jaques, N., Taylor, S., Sano, A. and Picard, R., 2017, October. Multimodal autoencoder: A deep learning approach to filling in missing sensor data and enabling better mood prediction. In 2017

Kaneda, Y. and Mineno, H., 2016. Sliding window-based support vector regression for predicting micrometeorological data. Expert Systems with Applications, 59, pp.217-225.

Lee, H.S., Trihamdani, A.R., Kubota, T., Iizuka, S. and Phuong, T.T.T., 2017. Impacts of land use changes from the Hanoi Master Plan 2030 on urban heat islands: Part 2. Influence of global warming. Sustainable Cities and Society, 31, pp.95-108.

Li, X., Cheng, G. and Lu, L., 2005. Spatial analysis of air temperature in the Qinghai-Tibet Plateau. Arctic, Antarctic, and Alpine Research, 37(2), pp.246-252.

Li, S., Griffith, D.A. and Shu, H., 2020. Temperature prediction based on a space–time regression-kriging model. Journal of Applied Statistics, 47(7), pp.1168-1190.

Li, Y., Yu, M., Xu, M., Yang, J., Sha, D., Liu, Q. and Yang, C., 2020. Big data and cloud computing. In Manual of Digital Earth (pp. 325-355). Springer, Singapore.

Lu, G.Y. and Wong, D.W., 2008. An adaptive inverse-distance weighting spatial interpolation technique. Computers & geosciences, 34(9), pp.1044-1055.

Luber, G. and McGeehin, M., 2008. Climate change and extreme heat events. American journal of preventive medicine, 35(5), pp.429-435.

Lydia, M., Kumar, S.S., Selvakumar, A.I. and Kumar, G.E.P., 2016. Linear and non-linear autoregressive models for short-term wind speed forecasting. Energy Conversion and Management, 112, pp.115-124.

Ma, J., Ding, Y., Cheng, J.C., Jiang, F. and Wan, Z., 2019. A temporal-spatial interpolation and extrapolation method based on geographic Long Short-Term Memory neural network for PM2. 5. Journal of Cleaner Production, 237, p.117729.

Mahdian, M.H., Bandarabady, S.R., Sokouti, R. and Banis, Y.N., 2009. Appraisal of the geostatistical methods to estimate monthly and annual temperature. Journal of Applied Sciences, 9(1), pp.128-134.

Maqsood, I., Khan, M.R. and Abraham, A., 2004. An ensemble of neural networks for weather forecasting. Neural Computing & Applications, 13(2), pp.112-122.

McCarthy, M.P., Best, M.J. and Betts, R.A., 2010. Climate change in cities due to global warming and urban effects. Geophysical research letters, 37(9).

Menon, S.P., Bharadwaj, R., Shetty, P., Sanu, P. and Nagendra, S., 2017, December. Prediction of temperature using linear regression. In 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT) (pp. 1-6). IEEE.

National Research Council, 2011. Climate stabilization targets: emissions, concentrations, and impacts over decades to millennia. National Academies Press.

NOAA National Climatic Data Center., 2010. State of the climate: national overview for annual 2010. Available online: http://www.ncdc.noaa.gov/sotc/national/2010/13 (accessed on 27 June 2021).

Olivas, E.S., Guerrero, J.D.M., Martinez-Sober, M., Magdalena-Benedito, J.R. and Serrano, L. eds., 2009. Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques: Algorithms, Methods, and Techniques. IGI Global.

Paul, P.V. and Saraswathi, R., 2017, March. The Internet of Things—A comprehensive survey. In 2017 International Conference on Computation of Power, Energy Information and Commuincation (ICCPEIC) (pp. 421-426). IEEE.

Qiao, P., Lei, M., Yang, S., Yang, J., Guo, G. and Zhou, X., 2018. Comparing ordinary kriging and inverse distance weighting for soil as pollution in Beijing. Environmental Science and Pollution Research, 25(16), pp.15597-15608.

Rawat, P., Singh, K.D., Chaouchi, H. and Bonnin, J.M., 2014. Wireless sensor networks: a survey on recent developments and potential synergies. The Journal of supercomputing, 68(1), pp.1-48.

Rubin, D.B., 1987. Multiple imputation for nonresponse in surveys (Vol. 81). John Wiley & Sons;

Sah, H.K. and Koli, S.M., 2019. WEATHER PREDICTION USING MULTIPLE IoT BASED WIRELESS SENSORS. Acta Technica Corviniensis-Bulletin of Engineering, 12(4), pp.123-127.

Santamouris, M., 2020. Recent progress on urban overheating and heat island research. Integrated assessment of the energy, environmental, vulnerability and health impact. Synergies with the global climate change. Energy and Buildings, 207, p.109482.

Salman, A.G., Heryadi, Y., Abdurahman, E. and Suparta, W., 2018. Single layer & multi-layer long short-term memory (LSTM) model with intermediate variables for weather forecasting. Procedia Computer Science, 135, pp.89-98.

Schafer, J.L. and Graham, J.W., 2002. Missing data: our view of the state of the art. Psychological methods, 7(2), p.147

Schatz, J. and Kucharik, C.J., 2014. Seasonality of the urban heat island effect in Madison, Wisconsin. Journal of Applied Meteorology and Climatology, 53(10), pp.2371-2386.

Shtiliyanova, A., Bellocchi, G., Borras, D., Eza, U., Martin, R. and Carrère, P., 2017. Kriging-based approach to predict missing air temperature data. Computers and Electronics in Agriculture, 142, pp.440-449.

Stekhoven, D.J. and Bühlmann, P., 2012. MissForest—non-parametric missing value imputation for mixed-type data. Bioinformatics, 28(1), pp.112-118.

Tabassian, M., Alessandrini, M., Jasaityte, R., De Marchi, L., Masetti, G. and D'hooge, J., 2016, September. Handling missing strain (rate) curves using K-nearest neighbor imputation. In 2016 IEEE International Ultrasonics Symposium (IUS) (pp. 1-4). IEEE.

Tang, F. and Ishwaran, H., 2017. Random forest missing data algorithms. Statistical Analysis and Data Mining: The ASA Data Science Journal, 10(6), pp.363-377.

Teng, X., Yang, J., Kim, J.S., Trajcevski, G., Züfle, A. and Nascimento, M.A., 2019, August. Fine-grained diversification of proximity constrained queries on road networks. In Proceedings of the 16th International Symposium on Spatial and Temporal Databases (pp. 51-60).

Tobler, W.R., 1970. A computer movie simulating urban growth in the Detroit region. Economic geography, 46(sup1), pp.234-240.

UN Department of Economic and Social Affairs (UN DESA). 2018 Revision of World Urbanization Prospects. Available online: https://www.un.org/development/desa/publications/2018-revision-of-world-urbanization-prospects.html (accessed on 6 November 2020).

Vaidyanathan, A., Malilay, J., Schramm, P. and Saha, S., 2020. Heat-Related Deaths—United States, 2004–2018. Morbidity and Mortality Weekly Report, 69(24), p.729.

Waljee, A.K., Mukherjee, A., Singal, A.G., Zhang, Y., Warren, J., Balis, U., Marrero, J., Zhu, J. and Higgins, P.D., 2013. Comparison of imputation methods for missing laboratory data in medicine. BMJ open, 3(8), p.e002847.

Watson, D.F. and Philip, G.M., 1985. A refinement of inverse distance weighted interpolation. Geo-processing, 2(4), pp.315-327.

Widiasari, I.R. and Nugroho, L.E., 2017, November. Deep learning multilayer perceptron (MLP) for flood prediction model using wireless sensor network based hydrology time series data mining. In 2017 International Conference on Innovative and Creative Information Technology (ICITech) (pp. 1-5). IEEE.

Wilson, T., Tan, P.N. and Luo, L., 2018, November. A Low Rank Weighted Graph Convolutional Approach to Weather Prediction. In 2018 IEEE International Conference on Data Mining (ICDM) (pp. 627-636). IEEE.

Wu, T. and Li, Y., 2013. Spatial interpolation of temperature in the United States using residual kriging. Applied Geography, 44, pp.112-120.

Xu, M., Liu, Q., Sha, D., Yu, M., Duffy, D.Q., Putman, W.M., Carroll, M., Lee, T. and Yang, C., 2020. PreciPatch: A Dictionary-based Precipitation Downscaling Method. Remote Sensing, 12(6), p.1030.

Xu, M., Zhao, L., Yang, R., Yang, J., Sha, D. and Yang, C., 2021. Integrating memory-mapping and N-dimensional hash function for fast and efficient grid-based climate data query. Annals of GIS, 27(1), pp.57-69.

Yang, J.S., Wang, Y.Q. and August, P.V., 2004. Estimation of land surface temperature using spatial interpolation and satellite-derived surface emissivity. Journal of Environmental Informatics, 4(1), pp.37-44.

Yang, J., Yu, M., Qin, H., Lu, M. and Yang, C., 2019. A twitter data credibility framework—Hurricane Harvey as a use case. ISPRS International Journal of Geo-Information, 8(3), p.111.

Yawut, C. and Kilaso, S., 2011, May. A wireless sensor network for weather and disaster alarm systems. In International Conference on Information and Electronics Engineering, IPCSIT (Vol. 6, pp. 155-159).

Ye, R. and Dai, Q., 2018. A novel transfer learning framework for time series forecasting. Knowledge-Based Systems, 156, pp.74-99.

Yi, X., Zheng, Y., Zhang, J. and Li, T., 2016. ST-MVL: filling missing values in geo-sensory time series data.

Yosinski, J., Clune, J., Bengio, Y. and Lipson, H., 2014. How transferable are features in deep neural networks?. In Advances in neural information processing systems (pp. 3320-3328).

Yoon, J., Jordon, J. and Schaar, M., 2018, July. Gain: Missing data imputation using generative adversarial nets. In International Conference on Machine Learning (pp. 5689-5698). PMLR.)

Yu, M., Bambacus, M., Cervone, G., Clarke, K., Duffy, D., Huang, Q., Li, J., Li, W., Li, Z., Liu, Q. and Resch, B., 2020. Spatiotemporal event detection: a review. International Journal of Digital Earth, 13(12), pp.1339-1365.

Zhou, B., Rybski, D. and Kropp, J.P., 2017. The role of city size and urban form in the surface urban heat island. Scientific reports, 7(1), pp.1-9.

BIOGRAPHY

Jingchao Yang received his bachelor's degree majored in Computer Science and minored in Geoinformation Science from both Eastern Michigan University and Central China Normal University in 2016. He joined the NSF Spatiotemporal Innovation Center, George Mason University, in 2016. Since then, he's involved in different NSF/NASA funded projects, including "Micro-scale Urban Heat Island Spatiotemporal Analytics and Prediction Framework" from 2016-2021. His current research involves machine- and deep- learning time-series forecasting with spatial analytics. He contributed to papers published in top journals, and also participated as major authors in several book chapters.