IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Using long short-term memory (LSTM) and Internet of Things (IoT) for localized surface temperature forecasting in an urban environment

MANZHU YU[1], FANGCAO XU[1], WEIMING HU[1], JIAN SUN[1], AND GUIDO CERVONE[1,2]

[1]Department of Geography and Institute for Computational and Data Science, The Pennsylvania State University, University Park, PA, USA (e-mail: mqy5198@psu.edu, xfangcao@psu.edu, wuh20@psu.edu, jbs6371@psu.edu, guc18@psu.edu)

[2]Research Application Laboratory (RAL), National Center for Atmospheric Research (NCAR), Boulder, CO, USA

Corresponding author: Manzhu Yu (e-mail: mqy5198@psu.edu).

This work was supported by the Penn State Institutes of Energy and the Environment Seed Grant.

ABSTRACT The rising temperature is one of the key indicators of a warming climate, capable of causing extensive stress to biological systems as well as built structures.Ambient temperature collected at ground level can have higher variability than regional weather forecasts, which fail to capture local dynamics. There remains a clear need for accurate air temperature prediction at the suburban scale at high temporal and spatial resolutions. This research proposed a framework based on a long short-term memory (LSTM) deep learning network to generate day-ahead hourly temperature forecasts with high spatial resolution. Air temperature observations are collected at a very fine scale (∼150m) along major roads of New York City (NYC) through the Internet of Things (IoT) data for 2019-2020. The network is a stacked two layer LSTM network, which is able to process the measurements from all sensor locations at the same time and is able to produce predictions for multiple future time steps simultaneously. Experiments showed that the LSTM network outperformed other traditional time series forecasting techniques, such as the persistence model, historical average, AutoRegressive Integrated Moving Average (ARIMA), and feedforward neural networks (FNN). In addition, historical weather observations are collected from in situ weather sensors (i.e., Weather Underground, WU) within the region for the past five years. Experiments were conducted to compare the performance of the LSTM network with different training datasets: 1) IoT data alone, or 2) IoT data with the historical five years of WU data. By leveraging the historical air temperature from WU, the LSTM model achieved a generally increased accuracy by being exposed to more historical patterns that might not be present in the IoT observations. Meanwhile, by using IoT observations, the spatial resolution of air temperature predictions is significantly improved.

INDEX TERMS air temperature, Internet of Things (IoT), long short-term memory (LSTM), urban weather

I. INTRODUCTION

ONE of the significant aspects of climate change is the globally rising temperature. According to the National Oceanic and Atmospheric Administration (NOAA) 2020 Global Climate Summary, the land and ocean surface temperature of August 2020 was 0.94 °C (1.69 °F) above average and ranked as the second-highest August temperature since 1880 [1]. Rising temperature and extreme heat events, exacerbated by the urban heat island effect, can produce life-threatening conditions to humans, overheat rivers, and increase risks to plants and wildlife. The urban heat island (UHI) is a phenomenon in which urban areas have higher temperatures (1–7 °F) than the surrounding rural areas [2]. Apart from the overall rising temperature, urban heat islands can be caused by reduced natural landscapes in urban areas, urban material properties that reflect less solar energy, urban geometries that hinder wind flow, and heat generated from human activities. Over 55% of the world's population lives in urban areas, which is predicted to reach 68% by 2050 [3]. Therefore, increasing risks of heat-related deaths and illnesses and growing demands of power exist in urban areas.

It is essential to adequately monitor local temperature

dynamics to mitigate the risks associated with increasing global temperatures. For that purpose, it is necessary to have good spatiotemporal coverage of temperature measurements. Regional weather forecasts provide a spatiotemporally continuous estimate of weather conditions, but such estimates are still limited in their spatial resolution, especially for personal or street-level uses [4]. Localized weather can be quite different from regional weather forecasts. Localized heat forecasting can help identify the regions prone to overheating, target warnings to citizens about potential heatwaves and provide aid to residents in time [5]. There is an important need for accurate hourly air temperature measurements at very high spatial resolution in urban environments.

Although high spatial-resolution weather simulation models can produce local forecasts, the accuracy of the predictions and future mitigation decisions are still heavily influenced by the availability of observations—the ground truth. These mitigation measures can be salting icy roads, turning on public water sprays, or providing shelters to the public in extreme weather situations [6]. The verification of model forecasts also requires high-resolution observations. However, traditional monitoring infrastructures cannot provide such information due to the limited number of discrete stations installed. The increasing availability of Internet of Things (IoT) sensors can provide an excellent complement to traditional in situ observations regarding local uncertainty. For example, the Array of Things network in Chicago has embedded approximately 150 sensors to monitor the urban climate at a community level [7]. The surface temperature has also been estimated using smartphone battery temperatures through crowdsourcing where proper quality control is conducted [8, 9]. The fifth generation (5G) of mobile technologies and their potential impact on the IoT will bring enormous benefits to localized weather observations with higher data transmission speeds and more connected networks [10].

This research proposed a framework by integrating long-term historical in situ observations and IoT observations together to train a long short-term memory (LSTM) network for air temperature prediction within New York City (NYC). By leveraging the historical air temperature data from in situ observations, the LSTM model can be exposed to more historical patterns that might not be present in the IoT observations. Meanwhile, by using IoT observations, the spatial resolution of air temperature predictions is significantly improved.

The paper is organized as follows. The related works are reviewed and discussed in Section 2. In situ and IoT sensor measurements used in this study are introduced in Section 3. The LSTM model adopted in this study is described in Section 4. Experimental results are reported in Section 5, and extreme cases are demonstrated in Section 6, followed by the conclusions and discussion in Section 7.

II. RELATED WORKS

A. IOT FOR URBAN TEMPERATURE MONITORING

An increasing number of cities are implementing urban meteorological monitoring projects of differing size and scales as part of "smart city" initiatives and scientific research projects, including the Birmingham urban climate laboratory [11], the Safe Community Alert Network at Montgomery County [12], the Array of Things network in Chicago [7], and the Smart Santander in Spain [13]. These initiatives and research projects have provided unprecedented new opportunities for high-resolution monitoring of the urban climate. Moreover, monitoring helps the city become smarter by controlling energy demand and reducing transport network disruption.

For temperature studies, increasing the spatiotemporal resolution of urban meteorological monitoring becomes even more crucial. Street-level air temperature has tremendous spatiotemporal variability that impacts vulnerable populations in different ways. For example, the Array of Things network in Chicago installed approximately 150 stationary devices ("nodes"), typically at street intersections, to monitor the city's climate, noise level, and air quality [7]. However, these "nodes" help increase the density of monitoring only to a certain extent, where each "node" covers a community instead of a street. The Smart Santander project is now embedding the city with more than 12,500 sensors [13]. Many sensors are mounted on stationary objects, such as trash containers, streetlights, and parking spaces. In contrast, other sensors are mounted on vehicles such as police cars and taxicabs that monitor air pollution and traffic conditions. The data collected from the larger number of sensors leads to improvements in urban weather monitoring and a better grasp of urban issues.

Utilizing their high spatiotemporal resolution, researchers have explored IoT infrastructures to monitor urban climate and assist in various urban issues. Chapman et al. [14] used the IoT network to measure rail moisture and leaf-fall contamination to achieve a low-cost, real-time, and high spatiotemporal resolution rail monitoring system. Chapman and Bell [6] demonstrated a use case utilizing IoT sensors to obtain high-resolution temperature observations for winter road maintenance. These observations are used in route-based forecasting models to determine which road segments need salting treatments in snowy or icy conditions. Ferranti et al. [15] utilized an IoT network to monitor the temperature rise along railways and analyze the relationship between railway failure and the gradual rise in temperature during the early- or mid-summer season. This relationship could be useful in heat risk management to potentially reduce disruptions and delays in railway services. Kraemer et al. [16] utilized an IoT system with solar power and weather forecasts to predict solar power energy. They selected relevant features from weather forecasts and trained machine learning models that generate predictions with 20% better accuracy than current state-of-the-art predictions. Solar energy predictions can be used for effective energy budget planning. Using IoT hardware, software, and communication technologies, Shapsough et al. [17] developed a cost-effective system for stakeholders to

monitor and efficiently control large-scale solar photovoltaic systems and evaluate the effects of environmental factors on the systems.

This research uses the data collected by a vehicle-based IoT network that collects air temperature information along major roads in and around large cities in the U.S. The collected information is then preprocessed to eliminate outliers and noise and aggregated hourly into approximately 150 m × 150 m grids. The high spatial and moderately high temporal resolution provides us with a great opportunity to monitor air temperature on a suburban scale.

B. MACHINE LEARNING TEMPERATURE PREDICTION

Air temperature prediction is one of the most critical aspects of climate study. Accurate temperature prediction can provide crucial guidance for the decision-making process to address environmental, ecological, or industrial problems. Machine learning techniques have been used in air temperature predictions based on the time series of historical air temperature and possibly other input predictors, such as humidity, wind speed and direction, and surface pressure. Exemplary machine learning methods include support vector machines (SVMs), artificial neural networks (ANNs), and, more recently, convolutional neural networks (CNNs), geographically weighted neural networks, and long short-term memory (LSTM) recurrent neural networks (RNNs). Some works have explored the capability of using machine learning methods to predict global temperature under climate change for future decadal or longer time scales [18, 19, 20]. Most of these studies demonstrated the impacts of CO2 emissions on global temperature increases and compared the global temperature predictions generated by machine learning methods with the Intergovernmental Panel on Climate Change (IPCC) scenarios [18]. However, global temperature models do not provide local or regional forecasts with fine-scaled air temperature variability.

Other works have integrated observations from weather stations into machine learning models for regional or local air temperature forecasting on an hourly or daily basis. For example, Smith et al. [21] used a Ward-style ANN with historical 24-hour air temperatures, wind speed, precipitation, relative humidity, and solar radiation to predict the air temperature at one or multiple future hours in Georgia. Ward-style ANNs are single-layer feedforward neural networks that utilize backpropagation and activation functions to optimize weights and biases. The results showed an increasing error when predicting a longer period in the future, with the mean absolute error (MAE) ranging from 0.516 °C at the one-hour horizon to 1.873 °C at the twelve-hour horizon. Focusing on the same region, Chevalier et al. [22] compared the support vector regression (SVR) and the single-layer ANN to predict sudden changes in air temperature and observed different capabilities of the two models in predicting year-round and winter-only datasets. The results showed that SVR was predicted more accurately for the year-round dataset, whereas ANN generally outperformed SVR using the winter-only dataset. Adding more hidden layers, Hossain et al. [23] trained a three-layer feedforward neural network with historical 24-hour air temperature from weather stations, barometric pressure, humidity, and wind speed to predict the air temperature at a certain future hour. Expanding from the geographically weighted regression (GWR), Du et al [24] integrated GWR with neural networks to account for nonstationary weight metrics that incorporate the spatial distribution of the environmental observations. Similarly, Wu et al. [25] expanded the GWRNN into both spatial and temporal weighted regressions that account for spatiotemporal non-stationary dependencies within the environmental observations to enhance the forecasting. Hewage et al. [26] trained and compared two deep learning models (LSTM and Convolution RNN) with surface temperature, pressure, wind, precipitation, humidity, snow, and soil temperature that are generated from numerical weather prediction models. Both models are used to predict air temperature for a specific future hour (one-step prediction), and both models are composed of five hidden layers.

Most of the works mentioned above focus on one-step prediction instead of multistep prediction (prediction for multiple specific future time steps). These studies have used data collected from weather stations or numerical simulations that generally have a coarse spatial resolution and cannot provide street-level air temperature variability. There have been various studies using IoT networks and machine learning techniques for air quality prediction [27, 28], agricultural frost prediction [29], or building heating and cooling demand prediction [30]. However, to the authors' knowledge, this is one of the first studies integrating traditional in situ observations and IoT observations to improve the multistep prediction capability of deep learning techniques.

III. STUDY AREA AND DATA

In this study, we focus on exploring the air temperature in NYC. Table 1 lists the sources and spatiotemporal resolutions of the datasets used in this study. The IoT data are from the GeoTab data platform, and the data availability ranges from April 29, 2019, to May 1, 2020. For the same period, we also downloaded the air temperature from discrete weather stations in Weather Underground.

TABLE 1. Summary of data collections

Data type	Internet of Things (IoT) sensor measurements	Weather stations
Data source	GeoTab (https://data.geotab.com/)	Weather Underground
Spatial and temporal resolutions	153 m x 153 m grids along the major roads for every 60 minutes	Fixed sensor location for every 60 minutes
Time range	May 1, 2019 – Apr 30, 2020	Jan 1, 2015 – Apr 30, 2019
Number of grids/stations	36,970	130
Value range	[-12.20, 48.00] °C	[-22.55, 36.94] °C
Missing data ratio range	[5.25%, 100%]	0.076%

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116809, IEEE Access

A. GEOTAB

The GeoTab data platform provides ambient air temperature data collected from sensors mounted on vehicles. This research uses data where anomalous data and outliers have already been removed. The measurements are aggregated to the 7-character geohash level (153 m x 153 m) every 60 minutes. GeoTab tracks over 900,000 vehicles and generates temperature data from over 250,000 vehicles per hour throughout North America. This allows GeoTab to accumulate millions of temperature data points per hour near real-time and generate a comprehensive, continuously updating map of road temperatures.

B. WEATHER UNDERGROUND

To build a reliable temperature prediction model, a long-term historical air temperature dataset pertaining to climate patterns is required. However, the availability of GeoTab lasts only one year, indicating that it is not sufficient to build the prediction model with GeoTab alone. Contrary to GeoTab, data collected from weather stations are generally archived for a long period. The dataset we used in this study is the hourly air temperature from Weather Underground (WU) in 2015–2019. WU is a network of weather stations that combines authoritative observation systems and personal weather stations. The authoritative observation systems include the Automated Surface Observation System (ASOS) stations located at airports throughout the country and the Meteorological Assimilation Data Ingest System (MADIS) managed by the National Oceanic and Atmospheric Administration (NOAA). The Federal Aviation Administration maintains ASOS stations, and observations are updated hourly or more frequently when adverse weather affecting aviation occurs (such as low visibility and precipitation). Personal weather stations (PWSs) results are contributed by volunteers who purchase and install weather sensors in and around their houses or workplaces. These PWS stations are put through strict quality controls, and observations are updated as often as every 2.5 seconds. There are 130 stations within the bounding box of our study area, and the missing data rate for all these stations is mostly less than 0.005. We directly filled the data with nan for each station by their missing time points.

To verify and cross-validate that GeoTab and WU both represent the local air temperature in NY city, the correlations between nearby (within 2km) GeoTab grids and WU stations are calculated for the same time period as GeoTab (May 2019-Apr 2020). Among the GeoTab grids and WU stations, 11 pairs of nearby locations from the two sources are identified, with different distances between each other within the pair (Fig. 1). Both the longest distanced pair and the shortest distanced pair show good correlations within each other regarding air temperature (Fig. 1 b and d). The time series comparisons also show good alignments (Fig. 1 c and e).

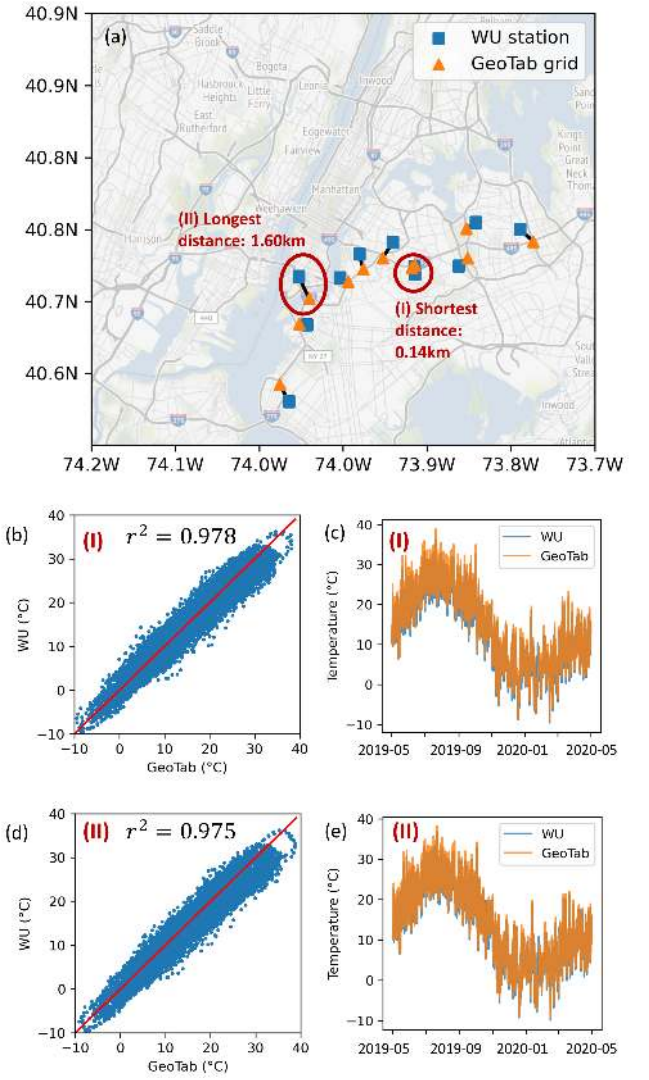

FIGURE 1. (a) GeoTab grids and their nearby WU stations within 2km. (b) and (c) show the value distribution, correlation coefficient, and time series comparison for the pair of GeoTab grid and WU station with the shortest distance among the available pairs. Similarly, (d) and (e) show the pair with the longest distance.

C. MISSING DATA HANDLING

Despite the high spatiotemporal resolution, an obvious disadvantage of vehicle-based measurement is the missing data issue due to the small (or even zero) number of vehicles passing through the same location within that hour. The aggregated air temperature for that spatiotemporal grid will be of less quality or have missing data. Here, we demonstrate the missing data rate of the GeoTab data. (Fig. 2a) shows the spatial distribution of all 36,970 GeoTab grids. Note that a particular grid may not have data since there might not be enough vehicles traveling through that grid in a specific hour. Thus, each grid has a different ratio of missing data, and similarly, each hour has a different spatial distribution based on the data available.

The missing data ratios for some GeoTab grids are rel-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116809, IEEE Access

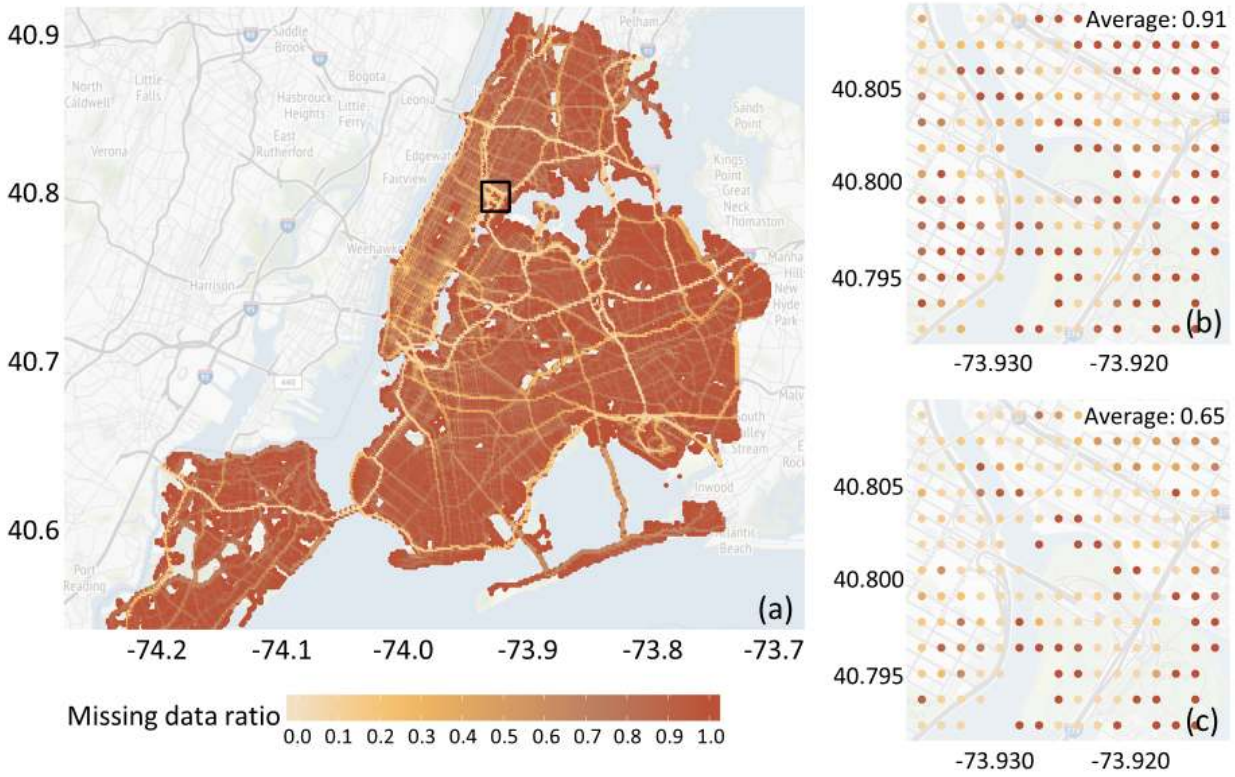

FIGURE 2. (a) Overall missing data ratio for each GeoTab grid, (b) spatial distribution of missing data ratio for the same subregion at 4:00 am local time (having the maximum overall missing ratio), and (c) spatial distribution of missing data ratio for the same subregion at 8:00 am local time (having the minimum overall missing ratio).

atively high compared to the WU dataset. To address this problem, we processed the GeoTab data in two steps: 1) select GeoTab grids along the major roads with a missing data ratio less than a certain threshold (i.e., under 5.5%, 10%, 20%, …, 50%), and 2) linearly interpolate the data of each grid in time and then find the nearest 20 stations for the average. We interpolated the missing data linearly for each grid based on temporal dependencies. These grids are distributed along major roads, which is reasonable since data-collection is vehicle based. There are more vehicles on major roads than on other secondary roads. This data characteristic makes this study more specific, focusing on air temperature prediction along major roads of NYC.

IV. METHODS

A. LSTM

Recurrent neural networks (RNNs) have been used to learn sequential patterns in time series data. Taking the current and the previous status, a hidden state at a time step t of an RNN can take the memory forward to predict the next time step (t+1). LSTM is a typical kind of RNN and can learn for a more extended period than a simple RNN [31]. The hidden state of LSTM can be controlled at the gates to avoid the vanishing gradient and the exploding gradient problems that are usually suffered by RNNs [32, 33].

The key to LSTM is the cell state, and adding or removing information to or from the cell state is achieved by gates, which is composed of a sigmoid neural net layer and a point-wise multiplication operation. The sigmoid layer's output is between 0 and 1, with 0 indicating letting no information passing through, and 1 indicating all information. An LSTM cell contains three gates: input gate, output gate, and forget gate (Fig. 3a). Within each cell, the first step is to select the cell state from the previous time step and retain part of the information into the current time step using the forget gate. The forget gate is described as Equation 1, where the hidden state of the previous time step (h_{t-1}) and the value of the current time step (x_t) are taken into account in the sigmoid function $\sigma(\cdot)$. The second step is to control the inward information into the cell using the input gate. This process is conducted using a sigmoid layer (Equation 2) that determines which values of the cell state to update and a tanh layer (Equation 3) that creates intermediate values ($\tilde{C}_t$) to update the cell state (Equation 4). The last step is to control the outward information from the cell. This process is achieved by a sigmoid layer (Equation 5) that determines which values of the cell state to output and a tanh layer that standardizes the values of the cell state. The sigmoid layer and the tanh layer are then multiplied to calculate the current hidden state (Equation 6).

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116809, IEEE Access

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * tanh(C_t) \tag{6}$$

In Equations 1-6, the sigmoid functions are calculated as $\sigma(x) = 1/(1 + e^{-x})$, the tangent function is calculated as $tanh(x) = (e^x - e^{-x})/(e^x + e^{-x}))$, W_f, W_i, W_C, W_o are the weight matrices, b_f, b_i, b_C, b_o are the bias vectors, x_t is the current input, h_{t-1} and h_t are the hidden states of the previous time step and the current time step.

LSTM models are a natural fit for our problem due to the following two reasons. First, LSTM is capable of handling long sequential data processing because the design of gates allows intact memory propagation, shown as the state passing, which avoids, to some extent, the gradient vanishing and exploding issues. Second, comparing to conventional RNN, LSTM is relatively insensitive to the "gap" length, i.e., the 'interval' between two adjacent cells. Temperature data have a similar characteristic because extreme weather may break the internal pattern existed in temperature. The gates design is extremely useful to eliminate the outlier data when finding the interior pattern of the time-series data. For example, the forget gate has the capacity to fully block the cell state memory passed from the previous time step.

The LSTM architecture used in this study is a conventional LSTM neural network for temperature prediction, which consists of N number of LSTM layers and one fully connected (FC) layer (Fig. 3b). The input data is feature vector of surface temperature observed for the past time series. The input data is fed into the stacked N layers of LSTM cells, where N is a tunable hyperparameter. The output of LSTM cells can be stacked into a matrix as input of the next layer. An LSTM layer is comprised of a set of M hidden nodes, where M is another tunable hyperparameter. When a single sequence of length sl_in is passed into the network, each individual element of the sequence is passed through each and every hidden node. Each hidden node gives a single output for each input it sees, which results in an overall output from the hidden layer of shape (sl_in, M). After the set of LSTM layers, a FC layer is added to the network for final output. The input size of the final FC layer is equal to the number of hidden nodes in the LSTM layer that precedes it. The output of this final FC layer is dependent on the output sequence length, sl_out, that the model will predict. For the multi-step temperature prediction, we used the Mean Squared Error (MSE) loss and the Adam optimizer.

B. PERFORMANCE EVALUATION

To evaluate the model performance, we used the root mean square error (RMSE) and bias error (bias). RMSE and bias are on the same scale as the data and are calculated as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\tilde{y_i} - y_i)^2} \tag{7}$$

$$Bias = \tilde{y_i} - y_i \tag{8}$$

where $\tilde{y_i}$ is the prediction and y_i is the ground truth for data sample i. Note that each data sample i contains a vector with a length of 24. RMSE and bias are calculated for each station and each test sample, and the predicted sequence is evaluated as a single entity. Specifically, bias is calculated as the mean bias for the predicted sequence.

C. TRAINING PROCEDURE

To demonstrate the capability of the LSTM architecture, the model is trained using 90% and validated using 10% GeoTab dataset and WU dataset. The model is tested using 27 randomly selected time series of continuous 72 (previous 48 + targeting 24) for GeoTab stations with up to 5.5% missing data ratio. The 27 different days within the time range of the GeoTab observations are between May 1, 2019 and April 30, 2020 under different weather conditions. In the selected testing data, the temperature difference within the targeting 24 hours ranges from 5 to 20 °C, which well represents the whole data. In addition, we ensure that the training and testing data are across all available months so that there is no bias or difference in the difficulty of predicting the test data. All experiments conducted use the same testing days for a fair comparison. The GeoTab grids used in the testing data have up to 5.5% missing data, where the values are the least interpolated.

D. HYPERPARAMETER TUNING

The effect of different combinations of the numbers of LSTM layers, hidden nodes, and other hyperparameters on the prediction accuracy is investigated by changing the LSTM layers from 1 to 3, hidden nodes from 24 to 72, and the learning rate from 0.1 to 0.005. The RMSEs and training time are recorded in (Fig. 4). While tuning the hyperparameters, we observed the following characteristics:

- When the learning rate decreases from 0.05 to 0.005, the training process takes longer but persists for more training epochs before the model overfits. The model trained with a learning rate of 0.05 has the largest RMSE and increasing the learning rate to 0.1 results in a 3% decrease in RMSE, while decreasing the learning rate to 0.005 results in a 10% decrease in RMSE.
- The LSTM architecture with three layers outperforms the one with two layers by 20%, given that the hidden node size is set to 24.
- Increasing the hidden size from 24 to 48 using the two-layer LSTM architecture can achieve similar accuracy to

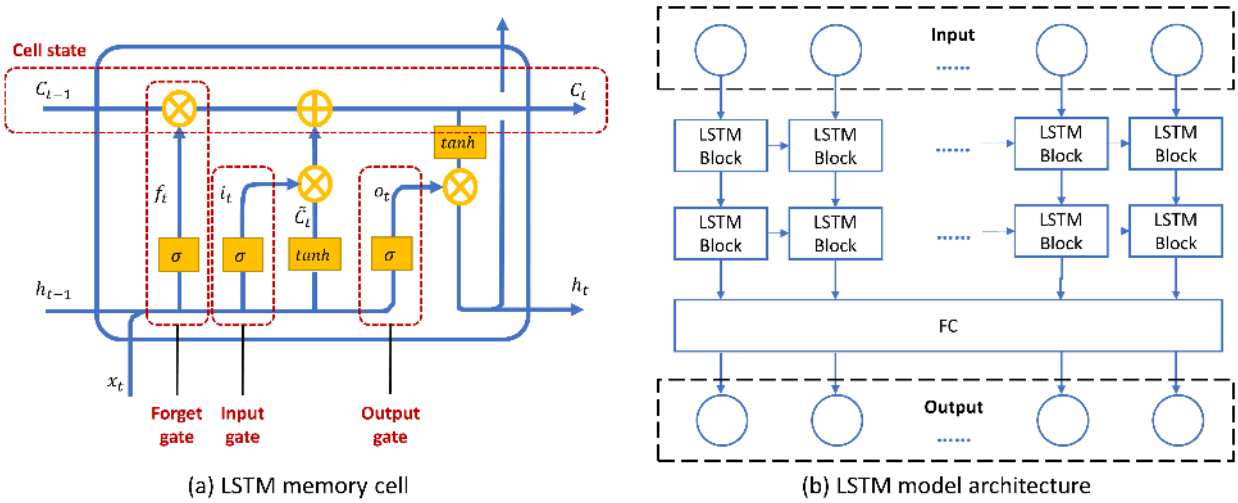

FIGURE 3. The architecture of a LSTM cell.

three layers but 24 nodes, but the two-layer architecture is more efficient than the three-layer architecture.

- Continuously increasing the number of hidden nodes from 48 to 72 and the number of hidden layers from 2 to 3 does not significantly improve accuracy. Nevertheless, the training time is three to five times longer than that with 48 nodes and two layers.

Based on the characteristics discussed above, the hyper-parameters used in the experiments are listed below. The number of hidden layers is selected as 2. The number of hidden nodes is selected as 48, with a learning rate of 0.005 to balance model accuracy and training efficiency.

The number of epochs is determined by initiating a sufficiently large best loss and updating it at each epoch and applying the early stopping technique to avoid overfitting. Comparing each epoch's running loss with the best loss recorded, if the running loss is ten times larger than the best loss, the training process will be terminated. We also found that the number of batches between 5 and 10 for each station can avoid overfitting too fast in very few epochs. Thus, the batch size is set as 500 and 5,000 for the GeoTab grid and WU station, respectively. These batch sizes correspond proportionally to the number of training samples for each GeoTab grid, and the WU station is approximately 4,000 and 40,000.

V. EXPERIMENT RESULTS

To demonstrate the prediction capability of the proposed approach, we compared the model performance with other commonly used time series forecasting methods. The lengths of the input sequence and output sequence are 48 hours and 24 hours, respectively, in the experiment. To understand the impact of missing data on our proposed approach's predictability, we also conducted a sensitivity experiment by changing the missing data ratio from 5.5% to 50%. To understand the impact of adding historical WU data to the training, a comparison experiment was conducted to examine whether and how the historical WU data can improve local weather predictions. Models were implemented using PyTorch 1.5, and experiments were conducted on a 64-bit Dell desktop with an NVIDIA GeForce RTX 2070, 32 GB RAM and Intel (R) Core i9–9900 CPU.

A. OVERALL PERFORMANCE OF THE PREDICTORS IN COMPARISON

Performance is compared between Persistence Model, Historical Average, AutoRegressive Integrated Moving Average (ARIMA), Feedforward Neural Network (FNN), LSTM (GeoTab), and LSTM (GeoTab+WU). We selected the baseline models that are widely accepted by literature such as Hyndman and Athanasopoulos [34], Du et al. [35], and Lyu et al. [36], and they each represent a different type of time series forecasting model. All models are tested on the same GeoTab grids with a missing data ratio of up to 5.5% to predict the 24-hour surface temperature for the predetermined 27 testing days, given 48-hour previous observations.

- **The persistence model** is one of the simplest methods for predicting the future behavior of a time series. Persistence implies that future values of the time series are calculated on the assumption that conditions remain unchanged between "current" time and future time t + TH [37].
- **The historical average** is calculated as the average value of the previous two days.
- **ARIMA** is a class of models that explains data using time series data on its past values and uses linear regression to make predictions. Assuming that data have an autoregression relationship with their past values, the ARIMA model uses the dependent relationship between the current value and the past values within the time

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116809, IEEE Access

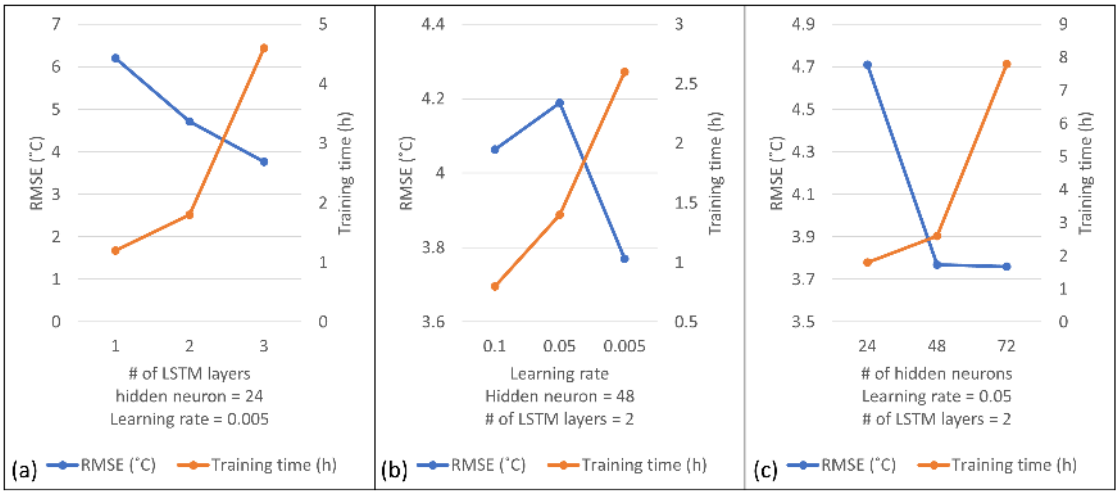

FIGURE 4. RMSEs and training time (a) changing # of LSTM layers from 1 to 3 when fixing hidden neuron = 24 and learning rate = 0.005 (b) changing learning rate from 0.1 to 0.005 when fixing hidden neuron = 48 and # of LSTM layers = 2 and (c) changing # of hidden neuron from 24 to 72 when fixing learning rate = 0.005 and # of LSTM layers = 2

series. ARIMA is also a moving average model, where the model's forecast depends linearly on its past values. To achieve the best performance of the ARIMA model, we used Auto ARIMA without tuning the required parameters of the ARIMA. The Auto ARIMA model generates the optimal p, d, and q values suitable for the dataset to provide better forecasting.

- **FNN** is a multilayer perceptron with additional hidden nodes between the input and output layers. In this network, data move in the only forward direction without any cycles or loops [38]. This research aims to produce multistep time-series predictions, so it is essential to design an FNN with multiple outputs. Each neuron in the output layer focuses on the prediction of the considered variable at a different time step. The main issue with this architecture is that it does not take into account that the outputs are sequential (i.e., the same variable at different time steps). In fact, the model would act in the same way if the outputs were to predict different system variables simultaneously.
- The **LSTM (GeoTab)** model was trained using only GeoTab data, and the **LSTM (GeoTab +WU)** model was trained using both GeoTab and WU data. The two LSTM models used same-size networks to predict on the testing dataset.

Table 2 shows that our proposed method - LSTM (GeoTab+WU) model - achieves the best performance in RMSE. Historical Average achieves the second-best performance out of all considered models, especially better than LSTM (GeoTab). There is the possibility that the short-term availability of GeoTab dataset limited the capability of the LSTM architecture. All the considered models achieve a mean RMSE within the range of 3-4°C, indicating that using the previous 48 hours to predict the future 24 hours on fine-scale GeoTab stations is not an easy task.

TABLE 2. Performance comparison

	RMSE (min/mean/max)	Bias (min/mean/max)
Persistence model	3.24 / 3.80 / 4.20	1.56 / 2.26 / 2.65
Historical average	3.36 / 3.71 / 4.03	0.4 / 0.49 / 0.61
ARIMA	3.29 / 3.98 / 4.56	0.34 / 0.92 / 1.52
FNN	3.59 / 3.86 / 4.13	0.67 / 0.95 / 1.2
LSTM (GeoTab)	3.49 / 3.77 / 4.02	0.22 / 0.55 / 1.1
LSTM (GeoTab + WU)	2.71 / 2.99 / 3.31	0.19 / 0.57 / 0.97

The RMSE and bias scores for each predicting hour during testing are reported in (Fig. 5). The Persistence Model, Historical Average, and ARIMA showed similar sine wave patterns due to the diurnal change of surface temperature, where the largest overestimating error occurs at the 11th predicting hour (i.e., local noontime), and the largest underestimating error occurs at the 19th predicting hour (i.e., local 8 pm). In contrast, the performance of the three neural networks—FNN, LSTM (GeoTab), and LSTM (GeoTab+WU)—decays with the prediction of future hours. These three neural architectures are able to learn and predict diurnal changes successfully and seem to show relatively similar predictive power, with the FNN predictor providing the most unsatisfactory performance. The biases of FNN after hour 13 are lower than the ones of LSTM (GeoTab+WU) by less than 1 °C, and the RMSEs of FNN are more than 1 °C higher than LSTM (GeoTab+WU) and become increasingly higher starting from hour 8. One possibility of the poor performance of FNN is that the FNN architecture explicitly used in this study is a multioutput network, with each output neuron irrelevant to other output neurons. In contrast, the LSTM models, including LSTM (GeoTab) and LSTM (GeoTab+WU), propagate the information through time, remembering past inputs and reproducing the nonlinear function. The results show that training the LSTM predictors

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116809, IEEE Access

strongly mitigates this issue, supporting our hypothesis that this training method allows proper information flow over subsequent time steps. The LSTM (GeoTab+WU) model showed a similar overall trend over the predicting hours to the LSTM (GeoTab) model, with progressively decreasing RMSE errors.

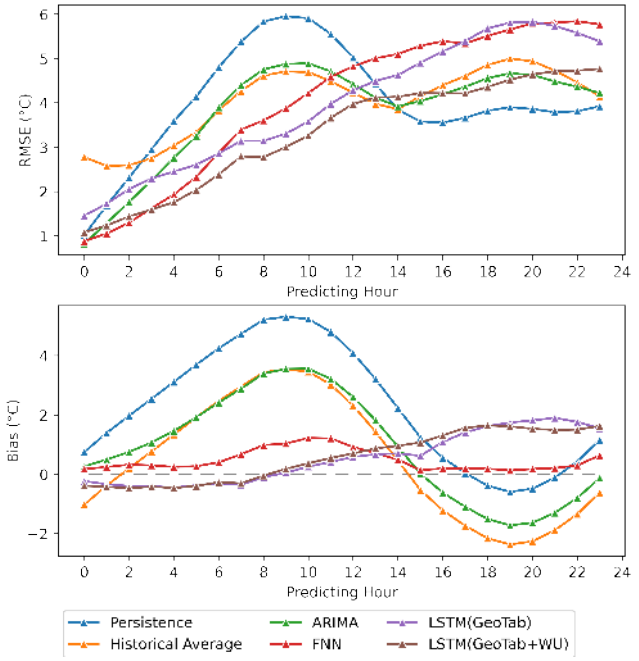

FIGURE 5. RMSE and bias scores obtained with the six predictors. Performance computed on the test dataset.

B. SENSITIVITY OF GEOTAB MISSING DATA RATIO

To understand the impact of missing data on our proposed approach's predictability, a sensitivity experiment was conducted by training the model with GeoTab grids having a missing data ratio up to between 5.5% and 50%, with 5% as the increment (Fig. 6). There are 455 GeoTab grids with 5.5% missing data, 1650 GeoTab grids with less than 10% missing data, and 4689 GeoTab grids with less than 50% missing data. Statistics of GeoTab grids with different missing data ratio were displayed in in Fig. 6, including the count of GeoTab grids in Fig. 6c, and the minimum, maximum, and median values of daily differences in air temperature in Fig. 6d. With a higher missing data ratio, daily temperature differences remain similar for the minimum, maximum, and median values, indicating that the data with 5.5% missing data ratio generally represents the whole data, with an increasing number of outlying high daily temperature differences. For the model trained by GeoTab only and the model trained by GeoTab+WU, integrating GeoTab grids with more missing data, the testing errors generally increase with noticeable fluctuations. The errors are primarily caused by spatiotemporal interpolation while estimating the values for inconsistent missing data. The fluctuating pattern might impact cutting off GeoTab grids using the missing data ratio without considering the spatial continuity of GeoTab grids.

In addition, models trained by GeoTab showed less fluctuation than those trained by GeoTab+WU in RMSE errors, indicating that models are less sensitive to the increasing missing data ratio. In contrast, models trained by GeoTab+WU showed more obvious fluctuations and larger increases in RMSE errors, indicating that models are more sensitive to the increasing missing data ratio. One possible reason is that the WU dataset is not interpolated, and it represents real-world observations. By adding more GeoTab grids with a higher interpolation rate whose observations are smoothed, the model started to negotiate with both WU and IoT to learn the smoothed pattern, which leads to greater fluctuations. Another possible reason for the different impacts on the two types of models is that GeoTab and WU have different value ranges of surface temperature, and GeoTab shows a larger spatiotemporal variability of surface temperature. When training the models with historical WU data and GeoTab, the models are impacted by the historical WU data more than GeoTab data, where WU had a longer period and a more constrained spatiotemporal variability. Involving GeoTab grids with more missing data downgraded the performance of the models trained primarily by WU.

C. IMPACT OF ADDING HISTORICAL WU IN TRAINING

To understand the impact of adding historical WU data to the training, a comparison experiment was conducted to examine whether the historical data can improve local weather predictions. Fig. 7 shows the performance of models trained on GeoTab and GeoTab+WU. It is clear that when the WU dataset is added to the training, the RMSE is improved ∼20% (mean value: 3.1 vs. 3.8 °C), and the overestimating bias is reduced for most of the experiments except for the one trained using 5.5% missing data. To investigate the spatial distribution of performance, another comparison experiment was conducted to train and test the LSTM model with GeoTab or GeoTab+WU on GeoTab grids with different missing data ratios. Fig. 7 shows that LSTM (GeoTab) and LSTM (GeoTab+WU) have similar spatial patterns of mean RMSE, whereas adding historical WU data in training significantly reduced the RMSEs for most of the GeoTab grids, except for the isolating GeoTab grids. Note that a few GeoTab grids along the river showed worse results for the 15% set when using GeoTab +WU. This effect occurs because these GeoTab grids are located next to the river, far from other GeoTab grids, and are very sparsely distributed in space. This problem is solved when we increase the GeoTab grids to 20%; when more GeoTab grids along the river in those regions are selected for training, the network learned their patterns successfully.

VI. DISCUSSIONS

A. LIMITATION

One limitation of this study is that it can only predict a certain location where past measurements of air temperature

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116809, IEEE Access

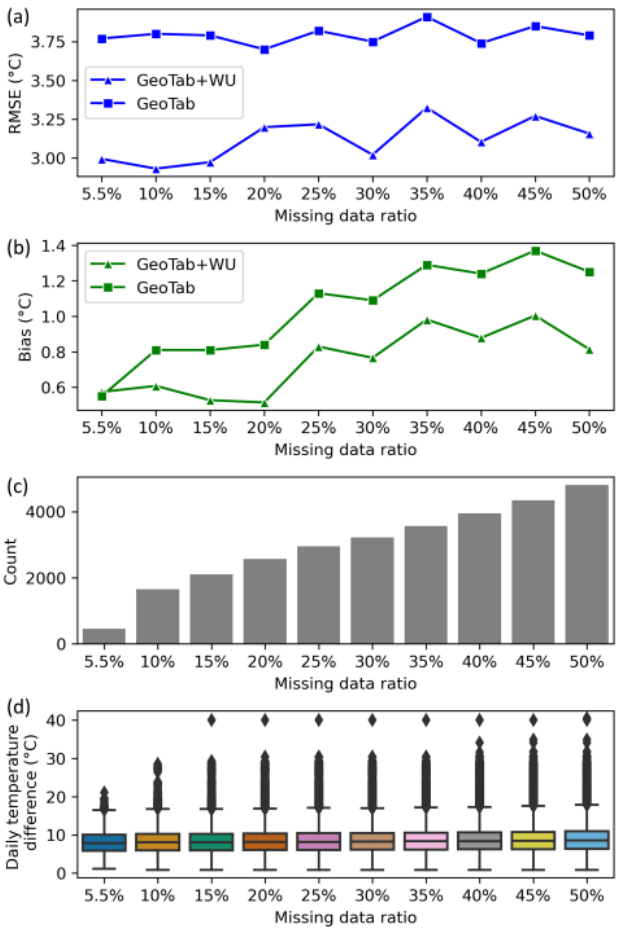

FIGURE 6. Mean (a) RMSE and (b) bias errors for the same testing data generated by models trained using GeoTab grids with different missing data ratios. Errors are also shown for models trained using the GeoTab grids and WU historical data. (c) GeoTab grid counts varying with missing data ratio. (d) Daily temperature difference varying with missing data ratio.

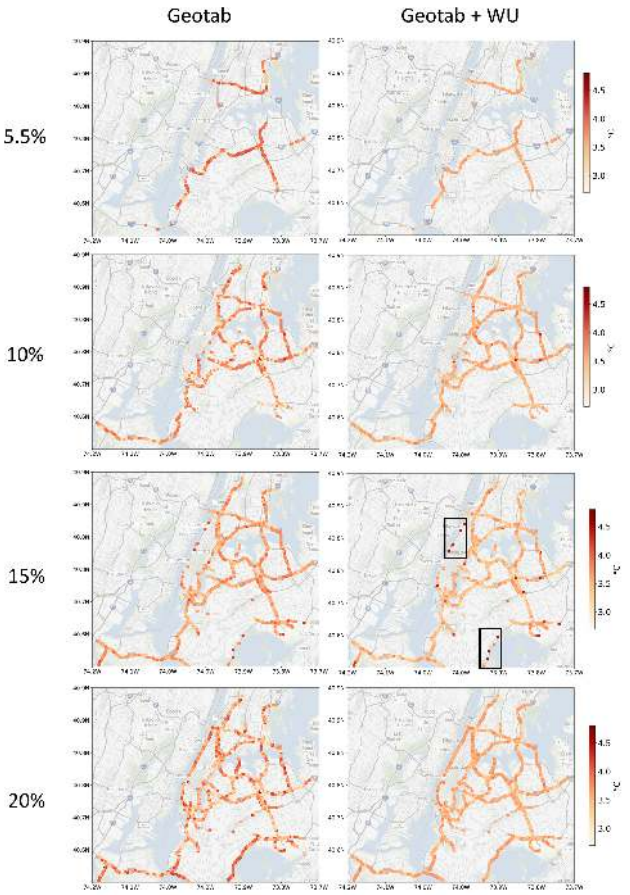

FIGURE 7. Spatial distributions of RMSE averaged for all testing days for each testing station. Comparison of models trained by GeoTab only and GeoTab + WU.

are available. Compared to WU, the GeoTab air temperature data are densely distributed grids along major roads, providing better spatial coverage and a higher spatial resolution. Experimental results showed that combining two observational datasets with a longer training period provides a better prediction accuracy. Therefore, the trained model can be used to retrain for the same area given that there are more new sensors deployed and more of the latest measurements retrieved.

B. COMPARISON WITH HRRR

To compare the air temperature predictions between the proposed approach and the numerical simulation, we downloaded the high-resolution rapid refresh (HRRR) predictions. The HRRR model is a 3 km resolution, hourly updated atmospheric model. Radar data are assimilated in the HRRR every 15 minutes over a 1-hour period. The dataset used in this study is downloaded from the University of Utah HRRR archive [39]. Fig. 8a shows the spatial locations of the HRRR grid points, and each HRRR grid point is compared with a GeoTab grid within 2 km. Each pair of HRRR and GeoTab grids are linked with a black line. The RMSEs are color-coded in Fig. 8a, with the RMSEs of HRRR predictions visualized on each HRRR grid point and the RMSEs of LSTM (GeoTab+WU) predictions visualized on each GeoTab grid point. The RMSEs shown in this figure are the mean RMSE throughout the testing days. A remarkable observation is that LSTM (GeoTab+WU) predicted an approximately 30% lower RMSE than HRRR (mean RMSE: 3.64 vs. 4.61 °C).

The performance is also decomposed into the 24 prediction hours in Fig. 8b and Fig. 8c. The HRRR predictions showed the diurnal change of predicting error in a sine wave pattern, and the amplitude of the sine wave during the local afternoon time was larger than that in the local morning, partly due to the propagating error of the prediction (Fig. 8b). The bias values of HRRR predictions showed overestimations during the morning and underestimations during the afternoons. The LSTM (GeoTab+WU) predictions showed an increasing RMSE over the 24 prediction hours, but the increase stopped around hour 18. In addition, LSTM (GeoTab+WU) predictions generally overestimate the surface temperature, but the overestimation is generally lower or approximately 1 °C (Fig.

8c).

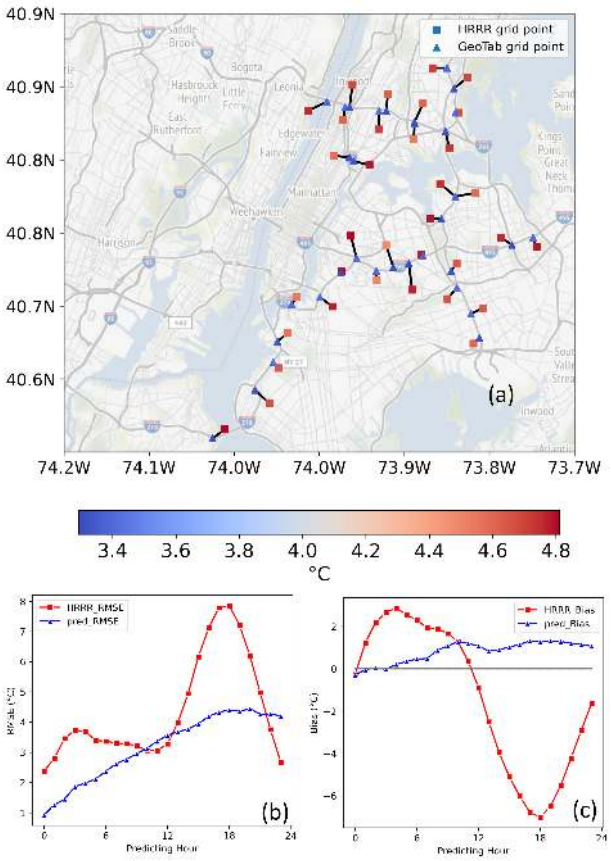

FIGURE 8. Comparing RMSEs between HRRR predictions and LSTM (GeoTab+WU) predictions at adjacent GeoTab grids.

C. EXTREME CASES

We further investigated the model performance of LSTM (GeoTab+WU) over different testing days. For the 27 testing days, the overall average RMSE is 2.99 °C. However, the RMSEs of some selected days are much higher than others. Fig. 9a shows the histogram of the average RMSE over 1,650 GeoTab grids for each selected day. The trained model performed well for most days with an average RMSE under 4, and Fig. 9 7b represents one example. There are three selected days, May 15, 2019, November 1, 2019, and January 13, 2020, when the model failed to predict well, and these days were confirmed to have rapidly changing weather based on the records from the weather stations (Fig. 9c, Fig. 9d, and Fig. 9e). These three days have temperature patterns distinct from those of the previous two days. Note that the performance is highly related to the lengths of the time series used as previous values and target values, i.e., 48 and 24 hours. Possible improvements can be made by integrating regional weather forecasts for long-term weather projections.

VII. CONCLUSIONS

In this paper, we proposed a framework by integrating long-term historical in situ observations and IoT observations to

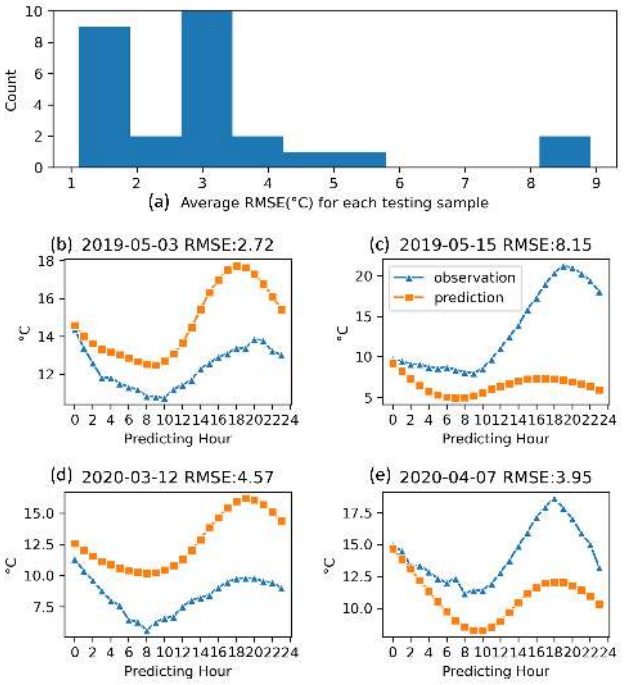

FIGURE 9. Histogram of RMSEs for each testing sample and demonstration of extreme weather cases.

train a LSTM network for air temperature prediction within the city of New York. We compared the proposed framework with other time series prediction methods, specifically the persistence model, historical average, ARIMA, and FNN. The LSTM network was trained in two different ways: 1) LSTM (GeoTab), which used the IoT observations alone, and 2) LSTM (GeoTab+WU), which used the IoT observations and the historical records from weather stations. The results showed that our proposed framework of integrating historical weather observations significantly improved the predictive performance of the LSTM network and outperformed the other statistical and deep learning-based time series prediction methods. By leveraging the historical air temperature data from in situ observations, the LSTM model can be exposed to more historical patterns that might not be present in the IoT observations. Meanwhile, by using IoT observations, the spatial resolution of air temperature predictions is significantly improved.

References

[1] R. J. Dunn et al. “Global climate”. In: *Bulletin of the American Meteorological Society* 101.101 (8) (2020), S9–S127.

[2] T. R. Oke. “City size and the urban heat island”. In: *Atmospheric Environment (1967)* 7.8 (1973), pp. 769–779.

[3] U. Nations. *World Urbanization Prospects: The 2018 Revision (ST/ESA/SER.A/420)*. 2019. URL: https://population.un.org/wup/Publications/Files/WUP2018-Report.pdf.

[4] R. Pelta and A. A. Chudnovsky. "Spatiotemporal estimation of air temperature patterns at the street level using high resolution satellite imagery". In: *Science of the Total Environment* 579 (2017), pp. 675–684.

[5] Y. Shi, L. Katzschner, and E. Ng. "Modelling the fine-scale spatiotemporal pattern of urban heat island effect using land use regression approach in a megacity". In: *Science of the Total Environment* 618 (2018), pp. 891–904.

[6] L. Chapman and S. J. Bell. "High-resolution monitoring of weather impacts on infrastructure networks using the Internet of Things". In: *Bulletin of the American Meteorological Society* 99.6 (2018), pp. 1147–1154.

[7] C. E. Catlett et al. "Array of things: a scientific research instrument in the public way: platform design and early lessons learned". In: *Proceedings of the 2nd international workshop on science of smart city operations and platforms engineering*. 2017, pp. 26–33.

[8] A. Overeem et al. "Crowdsourcing urban air temperatures from smartphone battery temperatures". In: *Geophysical Research Letters* 40.15 (2013), pp. 4081–4085.

[9] C. Muller et al. "Crowdsourcing for climate and atmospheric sciences: current status and future potential". In: *International Journal of Climatology* 35.11 (2015), pp. 3185–3203.

[10] I. Alawe et al. "Improving traffic forecasting for 5G core network scalability: A Machine Learning approach". In: *IEEE Network* 32.6 (2018), pp. 42–49.

[11] L. Chapman et al. "The Birmingham urban climate laboratory: An open meteorological test bed and challenges of the smart city". In: *Bulletin of the American Meteorological Society* 96.9 (2015), pp. 1545–1560.

[12] K. Benson et al. "SCALE: Safe community awareness and alerting leveraging the internet of things". In: *IEEE Communications Magazine* 53.12 (2015), pp. 27–34.

[13] P. Sotres et al. "Practical lessons from the deployment and management of a smart city Internet-of-Things infrastructure: The SmartSantander testbed case". In: *IEEE Access* 5 (2017), pp. 14309–14322.

[14] L. Chapman, E. Warren, and V. Chapman. "Using the internet of things to monitor low adhesion on railways". In: *Proceedings of the Institution of Civil Engineers-Transport*. Vol. 169. Thomas Telford Ltd. 2016, pp. 321–329.

[15] E. Ferranti et al. "Heat-related failures on southeast England's railway network: Insights and implications for heat risk management". In: *Weather, Climate, and Society* 8.2 (2016), pp. 177–191.

[16] F. A. Kraemer et al. "Operationalizing Solar Energy Predictions for Sustainable, Autonomous IoT Device Management". In: *IEEE Internet of Things Journal* (2020).

[17] S. Shapsough et al. "Using IoT and smart monitoring devices to optimize the efficiency of large-scale distributed solar farms". In: *Wireless Networks* (2018), pp. 1–17.

[18] R. Fildes and N. Kourentzes. "Validation and forecasting accuracy in models of climate change". In: *International Journal of Forecasting* 27.4 (2011), pp. 968–995.

[19] A. Abubakar et al. "Utilising key climate element variability for the prediction of future climate change using a support vector machine model". In: *International Journal of Global Warming* 9.2 (2016), pp. 129–151.

[20] H. Hassani et al. "Predicting global temperature anomaly: A definitive investigation using an ensemble of twelve competing forecasting models". In: *Physica A: Statistical Mechanics and its Applications* 509 (2018), pp. 121–139.

[21] B. A. Smith, G. Hoogenboom, and R. W. McClendon. "Artificial neural networks for automated year-round temperature prediction". In: *Computers and Electronics in Agriculture* 68.1 (2009), pp. 52–61.

[22] R. F. Chevalier et al. "Support vector regression with reduced training sets for air temperature prediction: a comparison with artificial neural networks". In: *Neural Computing and Applications* 20.1 (2011), pp. 151–159.

[23] M. Hossain et al. "Forecasting the weather of Nevada: A deep learning approach". In: *2015 international joint conference on neural networks (IJCNN)*. IEEE. 2015, pp. 1–6.

[24] Z. Du et al. "Geographically neural network weighted regression for the accurate estimation of spatial non-stationarity". In: *International Journal of Geographical Information Science* 34.7 (2020), pp. 1353–1377.

[25] S. Wu et al. "Geographically and temporally neural network weighted regression for modeling spatiotemporal non-stationary relationships". In: *International Journal of Geographical Information Science* 35.3 (2021), pp. 582–608.

[26] P. Hewage et al. "Deep learning-based effective fine-grained weather forecasting model". In: *Pattern Analysis and Applications* (2020), pp. 1–24.

[27] İ. Kök, M. U. Şimşek, and S. Özdemir. "A deep learning model for air quality prediction in smart cities". In: *2017 IEEE International Conference on Big Data (Big Data)*. IEEE. 2017, pp. 1983–1990.

[28] H.-Y. Jin, E.-S. Jung, and D. Lee. "High-performance IoT streaming data prediction system using Spark: a case study of air pollution". In: *Neural Computing and Applications* (2019), pp. 1–8.

[29] K. Brun-Laguna et al. "A demo of the PEACH IoT-based frost event prediction system for precision agriculture". In: *2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE. 2016, pp. 1–3.

[30] X. Luo et al. "Development of an IoT-based big data platform for day-ahead prediction of building heating and cooling demands". In: *Advanced Engineering Informatics* 41 (2019), p. 100926.

[31] S. Hochreiter and J. Schmidhuber. "LSTM can solve hard long time lag problems". In: *Advances in neural information processing systems* 9 (1996), pp. 473–479.

[32] F. A. Gers, J. Schmidhuber, and F. Cummins. "Learning to forget: Continual prediction with LSTM". In: *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*. IEEE. 1999.

[33] A. Graves et al. "A novel connectionist system for unconstrained handwriting recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 31.5 (2008), pp. 855–868.

[34] R. J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[35] S. Du et al. "Multivariate time series forecasting via attention-based encoder–decoder framework". In: *Neurocomputing* 388 (2020), pp. 269–279.

[36] P. Lyu et al. "LSTM based encoder-decoder for short-term predictions of gas concentration using multi-sensor fusion". In: *Process Safety and Environmental Protection* 137 (2020), pp. 93–105.

[37] R. H. Inman, H. T. Pedro, and C. F. Coimbra. "Solar forecasting methods for renewable energy integration". In: *Progress in energy and combustion science* 39.6 (2013), pp. 535–576.

[38] A. Urso et al. "Data mining: Classification and prediction". In: *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics* 384 (2018).

[39] B. K. Blaylock, J. D. Horel, and S. T. Liston. "Cloud archiving and data mining of High-Resolution Rapid Refresh forecast model output". In: *Computers & Geosciences* 109 (2017), pp. 43–50.