GeoHealth

RESEARCH ARTICLE

10.1029/2023GH000784

Forecasting West Nile Virus With Graph Neural Networks: Harnessing Spatial Dependence in Irregularly Sampled Geospatial Data

Adam Tonks[1], **Trevor Harris**[2], **Bo Li**[1], **William Brown**[3], **and Rebecca Smith**[3]

[1]Department of Statistics, University of Illinois at Urbana-Champaign, Champaign, IL, USA, [2]Department of Statistics, Texas A&M University, College Station, TX, USA, [3]Department of Pathobiology, University of Illinois at Urbana-Champaign, Champaign, IL, USA

Key Points:

- Many machine learning methods applied to environmental problems do not account for spatial dependence
- We apply a spatially aware graph neural network model to forecast West Nile virus
- Graph neural networks applied to irregularly sampled geospatial data can outperform a range of baseline methods

Correspondence to:

A. Tonks,
aytonks2@illinois.edu

Citation:

Tonks, A., Harris, T., Li, B., Brown, W., & Smith, R. (2024). Forecasting West Nile virus with graph neural networks: Harnessing spatial dependence in irregularly sampled geospatial data. *GeoHealth*, *8*, e2023GH000784. https://doi.org/10.1029/2023GH000784

Received 9 JAN 2023
Accepted 7 JUN 2024

Author Contributions:

Conceptualization: Adam Tonks, Trevor Harris, Bo Li, William Brown, Rebecca Smith
Data curation: Adam Tonks, William Brown, Rebecca Smith
Formal analysis: Adam Tonks, William Brown
Funding acquisition: Bo Li, Rebecca Smith
Investigation: Adam Tonks, Trevor Harris, Bo Li, Rebecca Smith
Methodology: Adam Tonks, Trevor Harris, Bo Li, Rebecca Smith
Project administration: Trevor Harris, Bo Li, William Brown, Rebecca Smith
Resources: Trevor Harris, William Brown
Software: Adam Tonks, Trevor Harris, William Brown
Supervision: Trevor Harris, Bo Li, Rebecca Smith

Abstract Machine learning methods have seen increased application to geospatial environmental problems, such as precipitation nowcasting, haze forecasting, and crop yield prediction. However, many of the machine learning methods applied to mosquito population and disease forecasting do not inherently take into account the underlying spatial structure of the given data. In our work, we apply a spatially aware graph neural network model consisting of GraphSAGE layers to forecast the presence of West Nile virus in Illinois, to aid mosquito surveillance and abatement efforts within the state. More generally, we show that graph neural networks applied to irregularly sampled geospatial data can exceed the performance of a range of baseline methods including logistic regression, XGBoost, and fully-connected neural networks.

Plain Language Summary Many machine learning methods have been applied to geospatial data, that is data that has a spatial dimension and corresponds to particular regions on Earth. However, many of these methods do not account for the fact that data recorded at nearby locations will be correlated. We apply a deep learning method called a graph neural network to overcome this issue. Our application to West Nile virus forecasting in Illinois could aid mosquito surveillance and abatement efforts within the state. This shows that graph neural networks could be a good option for other geospatial data, since they outperform a range of baseline methods in our particular problem.

1. Introduction

1.1. West Nile Virus

West Nile virus (WNV) is an arbovirus that causes the disease West Nile fever. The virus was first identified in Uganda in 1937, and subsequently in other central African countries shortly thereafter. The first appearance of WNV in the United States occurred in 1999, and the disease has since become endemic in much of North America. The death rate of those with serious disease is about 10%. The number of annual deaths across the United States alone has occasionally reached the hundreds, highlighting the urgent need for improved measures to curb WNV transmission (McDonald et al., 2021).

Birds serve as the main reservoir for WNV, while certain species of mosquitoes, primarily *Culex*, serve as vectors to transmit WNV from birds to humans. The circulation of WNV in nature is thus maintained via a cycle of mosquitoes and animal hosts. Within the United States, the American Robin is thought to be the dominant host species. House sparrows and crows are also susceptible to WNV, though they comprise a minor percentage of mosquito blood meals. Infected birds sometimes show symptoms of WNV leading to death, though are often asymptomatic. Mosquito species apart from *Culex*, such as *Aedes*, are also able to infect humans with WNV. Over 65 species of mosquitoes have the ability to be infected with WNV (Colpitts et al., 2012).

The burden of WNV on human and animal populations in the United States is high (Ronca et al., 2021). Massive die-offs of American cows have occurred as a result of WNV. Even closely monitored captive animals, such as exotic birds housed at the Bronx Zoo, have not been spared from encephalitis caused by the virus (Centers for Disease Control and Prevention, 1999). Outbreaks in human populations have also resulted in significant psychological burden, due to the virus's ability to spread via blood donations, organ transplants, and breast milk (Hadler et al., 2015). One study has estimated that the cumulative financial cost of WNV from 1999 to 2012 in the United States equals approximately US$778 million (Staples et al., 2014).

Validation: Adam Tonks, Trevor Harris, Bo Li, William Brown, Rebecca Smith
Visualization: Adam Tonks
Writing – original draft: Adam Tonks
Writing – review & editing: Adam Tonks, Trevor Harris, Bo Li, William Brown, Rebecca Smith

Curbing transmission of WNV via mosquito control remains challenging (Fernandes et al., 2018), in part due to the difficulty of effective targeting of mosquito eradication techniques such as insecticide, larvacide and bug traps. Such eradication techniques can be highly resource intensive, and judicious selection of deployment locations is one way to improve their effectiveness (Joshi & Miller, 2021). Therefore, accurate short-term forecasting of mosquito populations and WNV disease can aid overall mosquito control efforts. Notably, although the problem of forecasting mosquito populations and disease from trap and weather data has seen extensive application of machine learning methods, no such previous work has emphasized the spatial dependence inherent to the problem (Joshi & Miller, 2021).

Machine learning methods applied within this domain are mostly limited to those which do not account for spatial or temporal structure, such as fully-connected neural networks and support vector machines (Joshi & Miller, 2021). For example, one study of mosquito abundance in South Korea employed a shallow fully-connected network and compared its performance to linear regression (Lee et al., 2016). They found the performance of the models to be equivalent, which suggests that the ability of their neural network to fit any continuous function within a specific range (Hornik et al., 1989) is futile without regularization methods specific to the studied problem. In other words, machine learning methods are most powerful when they are tailored toward the type of input data. For example, convolutional neural networks are typically used for image recognition due to their desirable property of shift-invariance. This means that translated images are treated similarly by such models, which avoids overfitting to data points that are similar to each other. For our data, the relative spatial position of data points contains important information, so it is more appropriate to use spatially-derived input graphs to recognize spatial correlations between nearby data points. We did not find any applications of GNNs to vector-borne disease modeling in the literature.

Biologically, we know that the bird-mosquito transmission cycle will result in spatial clusters of infection, such that viral levels in a surrounding area can be predictive of presence in a particular trapped population. In theory, this should mean that diffusion of the virus (through mosquito or bird movement) should be as important to viral presence at a particular location as local maintenance of existing infection due to weather and mosquito ecology.

1.2. Deep Learning for Geospatial Data

Over the past decade, deep learning methods have been increasingly used for scientific modeling in areas such as particle physics (Bourilkov, 2019), drug discovery (Jiménez-Luna et al., 2020), and environmental forecasting (Shi et al., 2015). Although a range of machine learning methods, including deep learning methods, have seen applications to forecasting with geospatial data, much of the previous work has either concentrated on object classification in image-like data collected using remote sensing (Gopal, 2016; Soliman & Terstriep, 2019), or employed models that do not inherently account for the spatial dependence of geospatial data sets (Joshi & Miller, 2021). A notable exception to this is the use of neural networks employing convolution layers, namely convolutional neural networks (CNNs) and graph neural networks (GNNs), to generate nowcasts or forecasts using geospatial data (Fan et al., 2021; Jiang & Zhang, 2019; Lee et al., 2017; Shi et al., 2015).

CNNs have been shown to perform well on gridded spatiotemporal data (Fan et al., 2021; Shi et al., 2015), for example, popular climate model output data set products such as CPC Merged Analysis of Precipitation (CMAP). Previous work has shown that CNNs using such data can outperform other statistical models in the nowcasting and forecasting of precipitation and haze, especially when their architectures are further specialized to the problem domain, as is the case with HazeNet (C. Wang, 2021) and ConvLSTM (Shi et al., 2015). However, CNNs are not applicable to many geospatial problems where spatial processes are irregularly sampled and cannot be directly encoded into a simple lattice. Although it is possible to interpolate observations onto a grid so that existing CNN approaches can be employed, this could lead to propagating errors if the interpolation were inaccurate.

1.3. Related Work

Previous efforts to forecast mosquito populations and disease can be roughly divided into three different approaches: those using mechanistic models, those using conventional statistical methods, and those using machine learning methods. A recent review examined 221 papers that employed machine learning methods, and summarized a selection of 120 of these (Joshi & Miller, 2021). The most popular diseases studied were dengue and malaria, while the most popular mosquito genera studied was *Aedes*, which appeared in 109 of the papers. *Culex* was studied in 53 of the papers (some of the selected papers studied multiple genera). Half of these papers utilized geospatial data for their predictions, while the remaining papers analyzed vision, audio, text, and citizen science data. Of these geospatial papers that directly forecasted the spread of mosquito borne diseases, support vector machines, decision trees and artificial neural networks were utilized with varying degrees of success.

The use of artificial neural networks to forecast vector-borne diseases dates back to at least 2006 (Kiang et al., 2006). This work on malaria predating the era of more sophisticated neural network architectures understandably uses a rather rudimentary fully-connected neural network with just one hidden layer. In later years, neural networks with two hidden layers were explored for predicting malaria in Brazil (da Cunha et al., 2010). The same approach was revisited again in more recent years (Kalipe et al., 2018). However, more sophisticated neural network models were employed in two studies forecasting dengue in Brazil (Mussumeci & Codeço Coelho, 2020) and Kuala Lumpur (Pham et al., 2018). These two studies used a long short-term memory (LSTM) model, which is a type of recurrent neural network (RNN) that processes sequential data, such as time series data. The Brazil study was notable in using hierarchical clustering as a pre-processing step to determine variables associated with nearby cities to use as features for generating city-by-city forecasts. This is in contrast to other studies that generally did not account for spatial dependence in mosquito abundance or disease. Furthermore, transfer learning has also been explored in conjunction with LSTM models for dengue forecasting (Xu et al., 2020). Therefore, it appears that machine learning methods that exploit the temporal aspect of vector-borne disease prediction problems has been a popular area of study.

All surveyed work exploring artificial neural networks applied to geospatial data used either fully-connected networks or RNNs. A variety of data and model tuning approaches were also identified. For example, some studies utilized synthetic rather than real data (Bogado et al., 2020; Kinney et al., 2021). Another study that applied a fully-connected artificial neural network to predict mosquito abundance in urban areas of South Korea varied the learning rate and number of neurons in the hidden layer to find the model with minimum mean square error (Lee et al., 2016). In contrast to machine learning based work regarding the temporal aspects of vector borne disease forecasting, research into methods that exploit the spatial aspects appears somewhat underdeveloped.

1.4. Graph Neural Networks and GraphSAGE

Irregularly sampled geospatial data can, however, be represented as a spatial graph, which encodes spatial observations as nodes and optionally uses some measure of distance as edge weights. Therefore, GNNs offer an attractive substitute over CNNs, since they can operate directly on arbitrary graphs, such as spatial graphs (Wu et al., 2021; Zhou et al., 2020). GNNs have been previously applied to geospatial data for prediction of place characteristics (Zhu et al., 2020), metro station area vibrancy (Xiao et al., 2021), and social media check-ins (Zhu et al., 2021), as well as for forecasting of crop yield (Fan et al., 2021) and traffic (Jiang & Luo, 2021). Often, we can choose an input spatial graph that clearly reflects the actual locations of observations; this is the case with crop yield data for polygonal regions, where region adjacencies can be translated directly to graph adjacencies, and with traffic data, where the literature already suggests methods for constructing such graphs (most typically using road links for edges) (Jiang & Luo, 2021).

When there is no clear choice of an input spatial graph, we may always encode observations as a complete graph. However, this may not be the best alternative. We may construct a hierarchy of spatial subgraphs for linking spatial point events, ranging from the complete graph to mutually nearest neighbors (Fortin & Dale, 2005). Outside of this hierarchy, the representation of spatial information into graphs that can be used with GNNs remains an area of active research (Danel et al., 2020; Klemmer et al., 2021; Zhang & Zhao, 2021). To the best of our knowledge, there is a lack of previous work in exploring the real-world application of GNNs to forecasting problems using irregularly sampled geospatial data encoded as such spatial subgraphs. To this end, we demonstrate an application of GNNs to the forecasting of WNV-positive mosquitoes in Illinois, and show that our model outperforms existing state-of-the-art methods.

GNNs can be seen as generalizations of CNNs, with convolutional operations compatible not only with lattice graphs, but arbitrary ones. Furthermore, these operations are permutationally invariant, meaning that the output of such operations is the same for isomorphic graphs, regardless of the permutation of input nodes (Hamilton et al., 2017). As a rough example, a properly defined graph convolution would produce the same output regardless of which adjacency matrix is used as input. In contrast, convolutions used in a CNN are not permutationally invariant; reordering image pixels while maintaining the same lattice graph, such as by rotating or mirroring the image, leads to different convolution operation outputs.

GNNs are typically considered within a "message-passing" framework, where the neighborhood of each node defines a computational graph from which "messages" generated by nearby nodes are received and then "aggregated" to compute node embeddings which can be used for prediction (Gilmer et al., 2017). Most popular GNN methods, such as GraphSAGE and Geom-GCN (Pei et al., 2020), can be viewed within this framework. Multiple graph convolution layers can be stacked within a GNN to increase the depth of the computational graph, thus increasing the maximum distance to nodes whose features are considered in generating node embeddings and predictions.

GraphSAGE is a framework proposed in 2017 for efficiently generating node embeddings using node feature information (Hamilton et al., 2017). In contrast to many other approaches for generating node embeddings, GraphSAGE can operate on previously unseen nodes, including completely unseen graphs. This makes GraphSAGE ideal for our purposes, since input graphs for which we wish to produce predictions correspond to days not included in the training data.

As considered within the message-passing framework, at each layer, the GraphSAGE operator with the mean aggregator applied to node i is defined as

$$\mathbf{h}_\mathbf{i}' = \mathbf{W}_\mathbf{1}\mathbf{h}_\mathbf{i} + \mathbf{W}_\mathbf{2}\frac{\sum_{j \in \mathrm{N}(i)}\mathbf{h}_\mathbf{j}}{|\mathrm{N}(i)|},$$

where $\mathbf{h}_\mathbf{i}$ is the current node embedding, $\mathbf{h}_\mathbf{j}$ are the node embeddings of neighboring nodes, and $\mathbf{W}_\mathbf{1}$ and $\mathbf{W}_\mathbf{2}$ are matrices of trainable weights. In effect, nodes are no longer treated as independent observations using only the $\mathbf{W}_\mathbf{1}\mathbf{h}_\mathbf{i}$ term as the layer output, as would be the case in a fully-connected neural network. Instead, the GraphSAGE operator combines this term with $\mathbf{W}_\mathbf{2}\frac{\sum_{j \in \mathrm{N}(i)}\mathbf{h}_\mathbf{j}}{|\mathrm{N}(i)|}$, which is a weight matrix applied to the mean of node embeddings from nodes adjacent to node i.

Although the use of k-nearest neighbors in combination with GNNs has been mentioned in the literature (Zhu et al., 2021), such an approach has not yet seen widespread real-world applications. In particular, none of the surveyed mosquito abundance and disease prediction papers used machine learning approaches that accounted for the spatial dependence of the data.

2. Data

2.1. Data Set

The data set provided by the Illinois Department of Public Health (IDPH) contains 133,867 observations from 2008 to 2021. Each observation records whether a given trap on a given day is positive for WNV or not. Approximately 98% of the traps in the data set are gravid traps that collect Culex mosquitoes and, rarely, *Aedes albopticus*. The remaining 2% of traps include aspirator traps, New Jersey light traps and miniature light traps with dry ice. The mean pool size is 38. The bait used differs by mosquito abatement district, but typically includes a mixture of rabbit pellets, grass and whey protein. Data set variables include trap latitude and longitude, sample collection day, number of mosquitoes in sample, and test result (positive or negative). In our analysis, the date associated with each observation is the sample collection day, rather than the test day. Figure 1 shows the trap locations of a particular week in 2019. County populations and Interstate Highways are indicated on the map in Figure 1. Most trap locations are concentrated around Chicago and other population centers. 94 of Illinois' 102 counties are represented in the data set.

Since a large number of traps were tested infrequently, sometimes just a few times over the 13-year span of the data set, trap locations with a testing frequency less than the 90th percentile (128 tests) are excluded from our analysis. This corresponds approximately to a minimum of 1 test each week during the summer mosquito season, which spans primarily from June to August. The traps that are tested more frequently than this are located primarily in the Chicago metropolitan area. This reduces the total number of observations in the data set from 133,867 to 81,481, and unique trap locations from 3,332 to 333.

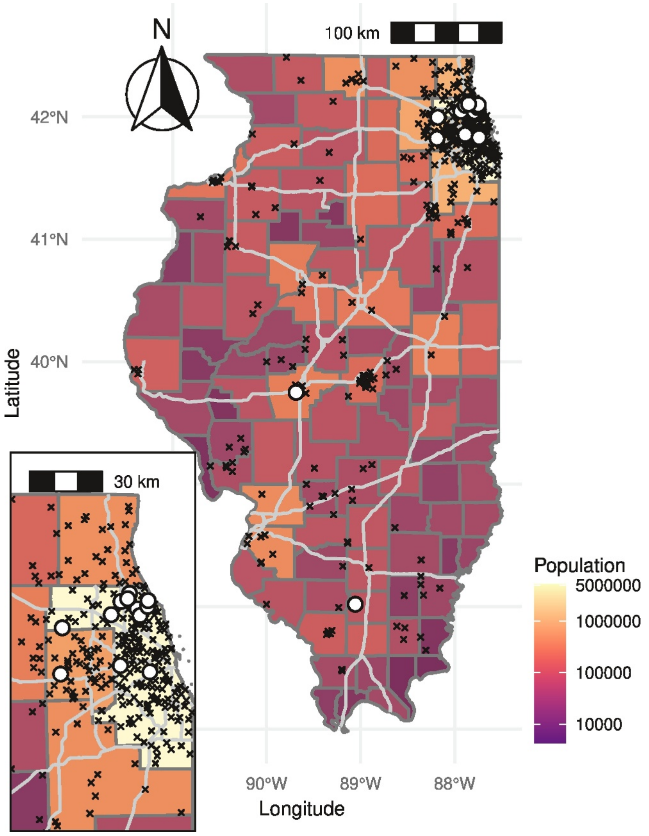

Figure 1. Location of traps tested during the week of Monday July 15 2019 to Sunday July 21 2019 in Illinois, with Chicago metropolitan area inset (negative tests indicated as crosses and positive tests as circles).

2.2. Formulation of WNV Forecasting Problem

Let $P_i(t)$ denote the test result for trap i on day t. We let $P_i(t) = 1$ if trap i tested positive on day t, and $P_i(t) = 0$ if trap i tested negative, or was not tested at all, on day t.

For each trap i_0, we use neighboring trap and weather data collected on day t_0, or prior, to forecast whether trap i_0 will be positive on any day in the lth week ahead. We consider forecast lead times of $l \in \{1, \ldots, 7\}$ weeks, where lead time l indicates the number of weeks ahead of the data available on day t_0. That is, if we are considering $P_{i_0}(t_0)$, we wish to predict whether any of $P_{i_0}(t_0 + 7l), P_{i_0}(t_0 + [7l + 1]), \ldots, P_{i_0}(t_0 + [7l + 6])$ equal 1. In other words, we wish to predict the variable $A_{i_0}(t_0 + 7l) = \max(P_{i_0}(t_0 + 7l), P_{i_0}(t_0 + [7l + 1]), \ldots, P_{i_0}(t_0 + [7l + 6]))$ for lead times $l \in \{1, \ldots, 7\}$.

For example, if we are generating forecasts for trap 5 on day 100, we make 7 total forecasts: whether or not a positive test will be observed at that trap within the week spanning days 107–113, within the week spanning days 114–120, and so on, up to the week spanning days 149–155. The features used may include weather and neighboring trap data from day 100 and any preceding days.

By considering a function of 7 consecutive days, instead of just a day at a time, we gain a better idea of whether WNV-positive mosquitoes are present in the trap vicinity. This is because for any particular day, a trap only captures a small, biased sample of the mosquito population. Hence considering mosquitoes captured across several days allows for a larger, more representative sample that is still relevant to the time point of interest. Although we could have taken the mean of $P_{i_0}(t_0 + 7l), P_{i_0}(t_0 + [7l + 1]), \ldots, P_{i_0}(t_0 + [7l + 6])$ to achieve smoothing similar to that of kernel density estimation, taking the maximum instead allows us to interpret $A_{i_0}(t_0 + 7l)$ simply as the positivity of a mosquito pool aggregated from pools collected on days $t_0 + 7l$ to $t_0 + [7l + 6]$.

Our formulation of this forecasting problem is similar to that seen previously in the literature, where a similar smoothing scheme was conducted using a 3-week moving window to reduce variation in the data resulting from environmental factors that were not of interest, such as moonlight and wind speed (Ripoche et al., 2019).

3. Methods

3.1. Model

We use a 4-layer GNN model using GraphSAGE layers with the mean aggregator and no L^2 normalization. L^2 normalization standardizes the norm of vectors to the unit norm while preserving direction. Although L^2 normalization is often applied to the node embeddings generated by GraphSAGE layers, we found that it was completely detrimental to model performance for the given problem, leading to a model that could only make the same prediction for every observation. This indicates that both the norm and direction of the node embedding vectors is informative, rather than only the direction. The model contains a total of 361 trainable parameters. The first 3 layers generate 8-dimensional node embeddings, which are then passed through a ReLU activation function. The last layer generates 1-dimensional node embeddings that are used as inputs to a sigmoid function to predict the probability of a positive case in the lth week $\hat{P}(A_{i_0}(t_0 + 7l) = 1)$. In summary, our model utilizes 4 consecutive GraphSAGE layers, each of which takes the previous layer's output as input. The exception is the first layer, which takes the graph node features as input. Each of the first 3 layers outputs 8-dimensional node embedding vectors, while the final layer outputs a log-odds score. The log-odds score is passed to a softmax sigmoid function, which generates probability values in the range (0, 1).

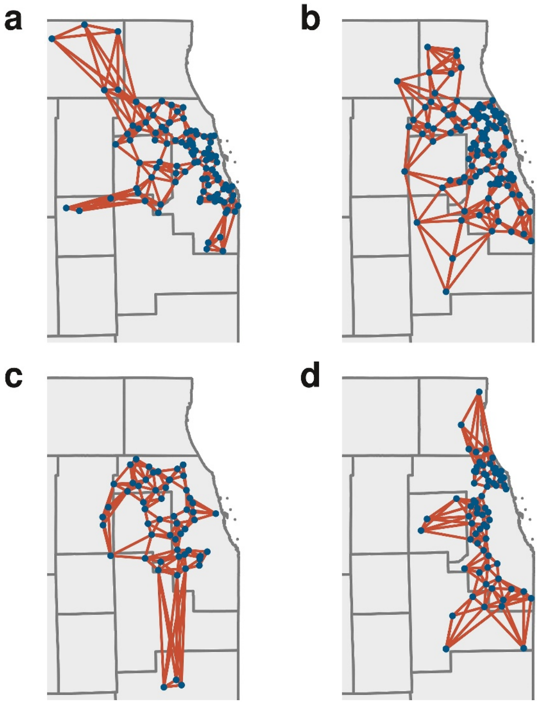

Figure 2. GNN input graphs created for (a) Tuesday July 16 2019 (b) July 17 (c) July 18 (d) July 19 using k-nearest neighbors with $k = 5$, zoomed into the Chicago metropolitan area.

3.2. Graph Creation and Node Features

A graph is created for each day in the data set. Each node in the graph represents a trap location, and edges between nodes are determined by k-nearest neighbors, thus creating a k-nearest neighbor graph (k-NNG). In other words, for each trap location, a graph edge is drawn between it and the k nearest other trap locations, which allows our GNN model to pool information across connected traps. Features, such as past trap positivity and weather, are associated with each node. Since the number and location of tested traps differ each day, the graphs also differ by day. The graphs are undirected and unweighted, and may be disconnected.

By encoding traps into k-NNGs, GraphSAGE can learn to aggregate information across connected sites. This gives our model a level of geospatial awareness because the graphs encode the relative location of each trap. Geographically near traps have a short path between them in our k-NNGs, indicating a high degree of information sharing, while geographically distant traps either have a long path between them or are completely disconnected, indicating a low degree of or nonexistent information sharing. We found that further including latitude and longitude as covariate features did not improve model performance metrics, which shows that our graphs capture a sufficient level of spatial information for prediction.

Edges found using k-nearest neighbors that connect two traps at a distance greater than 100 km are excluded from the graphs, to limit the range of the spatial dependence accounted for. We present results using graphs for $k = 1$, 2, 5, 10, 20. An example of generated graphs with $k = 5$ is depicted in Figure 2. Distances between trap locations are calculated from their latitudes and longitudes using the Haversine formula.

Recent trap and weather data are associated directly with the nodes as features. For the node corresponding to location i_0 and day t_0, the feature $A_{i_0}(t_0 - 6) = \max(P_{i_0}(t_0 - 6), P_{i_0}(t_0 - 5), \ldots, P_{i_0}(t_0))$ encodes trap positivity from the past week. The weather features $X_j(t_0)$ encode data for weather variables $j \in \{1, 2, 3\}$, corresponding to cooling degree days, heating degree days and precipitation at Chicago O'Hare International Airport on day t_0. This data is taken from NOAA's *Daily Summaries Station Details* data set, which contains temperature, wind and precipitation station observations from 1946 to present. The data set is updated weekly. These weather variables were found to be statistically significant in related work on WNV and mosquito abundance forecasting (J. Wang et al., 2011).

3.3. Model Tuning, Training and Testing

Models are trained in PyTorch Geometric (Fey & Lenssen, 2019) on a server running CentOS Linux 7 (Core) utilizing a NVIDIA Quadro GV100 (32GB). Training of each GNN model takes approximately 30s.

The data set is split into training (years 2008–2016), validation (years 2017–2018) and test (years 2019–2021) sets, covering approximately 8.5, 2, and 2.5 years of data respectively. The data set is split in such a way in order to be representative of the associated real-world application: a public health department possesses data on consecutive historical trap observations (the "training and validation sets"), and wishes to make accurate forecasts on future observations (the "test" set). Shuffling the observations before dividing them into training, validation and test sets would not be reflective of this scenario, since future data would be used to predict past observations. Essentially, the statistical problem would no longer be one of extrapolation, but of interpolation, which would artificially improve the out of sample performance metrics. In addition, since we use lagged data as features, the true values of forecasted observations would be contained in the features if we were to split the data set in this manner.

The Adam optimizer is used with a learning rate of 0.001 to minimize cross-entropy loss using a batch size of 100. Models are trained until no improvement in validation loss is seen in 10 consecutive epochs, up to a maximum of 1,000 epochs.

Tuning of the model hyperparameters is conducted using the validation set without reference to the test set.

All code and the data set provided by the IDPH have been archived online. Further details may be found in the Availability statement.

4. Experiments

4.1. Baseline Models

The performance of the GNN model is compared to that of several standard baseline machine learning and statistical models. These are logistic regression, a fully-connected neural network, and XGboost. Logistic regression is a commonly used statistical model that models the log-odds of a binary event as a linear combination of a set of features. Furthermore, since we are using the mean aggregator with GraphSAGE, it is equivalent to a 1-layer version of our GNN model. Fully-connected neural networks are the most basic type of neural network, upon which all other architectures are built. We use a 4-layer network with 8 neurons per layer and ReLU activation functions, except for the output layer. Finally, XGboost is a popular, state-of-the-art implementation of a decision tree boosting algorithm and is commonly used in machine learning competitions due to its excellent predictive performance, even with the default hyperparameters that we employ (Mitchell & Frank, 2017; xgboost developers, 2022). We choose these three baseline methods to represent the three distinct categories of models currently in use for mosquito-borne disease modeling, outside of mechanistic models: standard regression-based statistical models (Cianci et al., 2015; Mishra et al., 2019; Sass et al., 2022), basic neural network models (Lee et al., 2016; Moore, 1991; Rubio-Solis et al., 2019) and tree-based machine learning algorithms (Cianci et al., 2015; Ding et al., 2018; Mudele et al., 2021). Although these approaches lack the fundamental spatial awareness of graph neural networks (Bronstein et al., 2021), we can imbue them with some limited spatial information by including spatially lagged covariates. Thus we take these methods as our point of comparison to demonstrate the importance of spatial information and to demonstrate why natively spatial algorithms should be preferred over non-spatial algorithms with improvized spatial covariates.

These models are trained in a similar manner as the GNN model, using early stopping to prevent overfitting. Features are identical, but there is no consideration of the graph structure imposed on the data. The observations are therefore assumed to be independent.

Though this is a spatiotemporal forecasting problem, our focus is on its spatial aspects. Therefore, since the temporal context in our GNN model is simply represented by feature values at previous timesteps, we do not include any baseline models that account for the temporal context in a more sophisticated manner, such as specialized autoregressive models.

4.2. Performance Metrics

The performance metrics primarily considered are accuracy, Brier score, and AUC. Since the models output probability predictions $\hat{P}(A_{i_0}(t_0 + 7l) = 1)$, accuracy is computed by considering the predicted class as that with the larger predicted probability. That is, $\hat{A}_{i_0}(t_0 + 7l) = 1$ if $\hat{P}(A_{i_0}(t_0 + 7l) = 1) > p_t$, and 0 otherwise, where $p_t = 0.5$.

Since accuracy only considers the performance of a model at a particular selected trade-off between sensitivity and specificity, for example, that selected here by letting $p_t = 0.5$, we primarily consider AUC in assessing model performance. The need to consider performance across a range of trade-offs between sensitivity and specificity is particularly relevant in the context of epidemiology and medicine, and thus has seen widespread usage in these fields. AUC has a number of appealing properties, including insensitivity toward label distributions and costs (Yang et al., 2022). That AUC is superior to accuracy when judged on a number of formal criteria has also been established in the machine learning literature (Huang & Ling, 2005). For instance, by considering the sensitivity and specificity across a range of threshold probabilities p_t, AUC effectively accounts for the "confidence" of a class prediction, while accuracy disregards the magnitude of predicted probabilities and considers only on which side of p_t they lie. Finally, AUC has a simple interpretation: given a randomly chosen positive observation and a randomly chosen negative observation, AUC is the probability that the predicted probability of the positive observation being in the positive class is greater than that of the negative observation.

We provide the Brier score, a proper scoring rule, as an additional metric to assess the models. The Brier score is calculated as the mean squared error between predicted probabilities for the positive class and actual class labels. Smaller values indicate better performance.

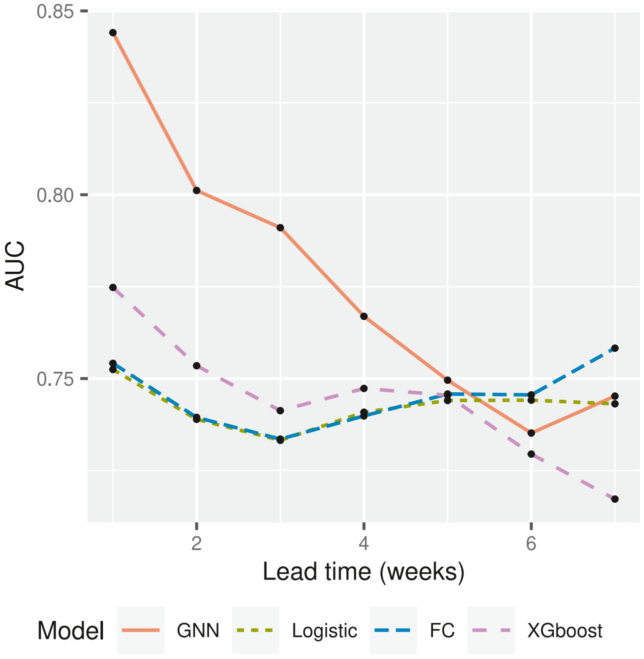

Figure 3. AUC values for GNN ($k = 5$) and baseline models at increasing lead times.

4.3. Determining k for Input Graphs

Graphs are created using k-nearest neighbors for $k = 1, 2, 5, 10, 20$. Graph nodes are associated with both trap and weather data and used as input into the GNN model. The AUC values of using these graphs for various lead times is shown in Figure 4, and are used to select an appropriate value of k for later model comparisons.

While no single value of k results in model performance greater than other values across all lead times, the AUC values corresponding to $k = 5$ and $=10$ are relatively high across all lead times. The sum of AUC across all lead times is also highest for $k = 5$ and $k = 10$. Therefore, we use GNN models with graphs corresponding to $k = 5$ for subsequent comparisons, since doing so offers similar performance but faster computation than $k = 10$.

4.4. Forecasting Results

Figure 3 and Table 1 indicate that at lead times of 1–5 weeks, the GNN model outperforms all other baseline models in terms of AUC and Brier score. The associated ROC curve at a lead time of 1 week is illustrated in Figure 5. The performance difference is most substantial at the smallest lead times and decreases with increasing lead time. At lead times of 6 and 7 weeks, the GNN model outperforms XGboost but is outperformed by logistic regression and the fully-connected neural network model, though all models perform very similarly with the exception of XGboost.

Confusion matrix values are additionally provided in Table 2. The discrepancies between the values in Tables 1 and 2 are due to the use of multiple model training runs and the stochastic nature of the Adam optimizer. However, these discrepancies are small, as indicated by the "AUC SD" column of Table 1. The values in Table 2 are computed by letting $p_t = 0.5$ and therefore represent only a single scenario of a trade-off between sensitivity and specificity. For example, if higher sensitivity were desired, the value of p_t could be made smaller. This would increase the counts of true positives and false positives, and decrease the counts of true negatives and false negatives. To illustrate this trade-off explicitly, we have included metrics at additional values of p_t in Table 3.

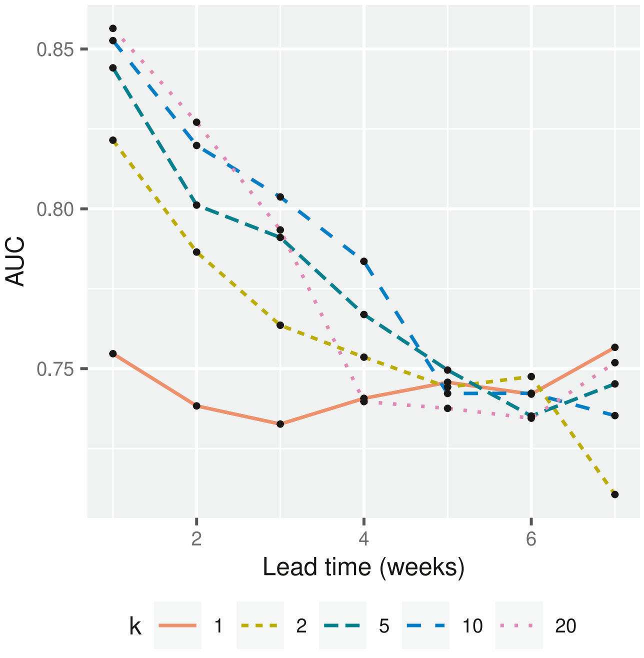

Figure 4. AUC values for GNN models using various input graphs at increasing lead times.

At a lead time of 1 week, AUC is 8.944% greater for the GNN model than for XGboost, the next-best model. At a lead time of 2 weeks, AUC is 6.330% greater. At the largest lead times of 6–7 weeks, the fully-connected neural network appears to be the best model in terms of AUC and Brier score, but with an AUC value greater than the GNN model by only 1.415% and 1.758% for lead times of 6 and 7 weeks, respectively.

Trap data is more informative of the class labels at the smallest lead times, while weather data is more informative at the greatest lead times. Therefore, we hypothesize that at the smallest lead times, the GNN model is able to capitalize on the spatial dependence of the trap data, unlike the baseline models. However, at the largest lead times, such an advantage no longer exists, and hence all models have roughly similar forecasting performance while relying on the weather data. We investigate this hypothesis further in the following subsection.

We also note that although model accuracies for lead times greater than approximately 4 weeks do not deviate substantially from those found using a ZeroR model, where all predictions are simply the data set's majority class, all AUC values are significantly greater than 0.5. This indicates that these models perform better than one that simply predicts "no WNV" for all observations, at least at particular trade-offs between sensitivity and specificity. In addition, the class imbalance further renders AUC as a better performance

Table 1
Metrics for Logistic Regression, Fully-Connected Neural Network, XGboost and GNN (k = 5) Models at Various Lead Times (in Weeks)

Lead time	Model type	Accuracy	Brier score	AUC	AUC SD
1	Log.	0.8686	0.1058	0.7525	0.0001
	FC	0.8720	0.1062	0.7542	0.0012
	XGb	0.8710	0.1056	0.7748	0.0000
	GNN	**0.8776**	**0.0926**	**0.8441**	0.0024
2	Log.	0.8600	0.1124	0.7389	0.0000
	FC	0.8596	0.1127	0.7394	0.0007
	XGb	0.8605	0.1155	0.7535	0.0000
	GNN	**0.8634**	**0.1058**	**0.8012**	0.0030
3	Log.	0.8439	0.1171	0.7332	0.0001
	FC	0.8464	0.1178	0.7335	0.0070
	XGb	0.8491	0.1210	0.7413	0.0000
	GNN	**0.8494**	**0.1117**	**0.7910**	0.0033
4	Log.	0.8464	0.1206	0.7408	0.0001
	FC	0.8444	0.1210	0.7398	0.0024
	XGb	**0.8497**	0.1213	0.7473	0.0000
	GNN	0.8457	**0.1178**	**0.7669**	0.0066
5	Log.	**0.8464**	0.1206	0.7440	0.0002
	FC	**0.8464**	0.1203	0.7458	0.0060
	XGb	0.8344	0.1246	0.7455	0.0000
	GNN	**0.8464**	**0.1199**	**0.7496**	0.0138
6	Log.	**0.8464**	0.1196	0.7441	0.0559
	FC	**0.8464**	**0.1194**	**0.7456**	0.0398
	XGb	**0.8464**	0.1258	0.7295	0.0000
	GNN	**0.8464**	0.1201	0.7352	0.0067
7	Log.	**0.8464**	0.1195	0.7431	0.0370
	FC	**0.8464**	**0.1180**	**0.7583**	0.0445
	XGb	0.8362	0.1278	0.7172	0.0000
	GNN	**0.8464**	0.1191	0.7452	0.0474

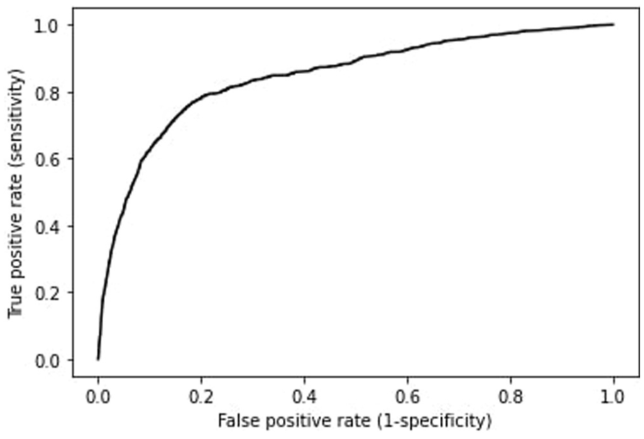

Figure 5. ROC curve for GNN ($k = 5$) model at lead time of 1 week.

measure than accuracy in this scenario. In more formal terms, class imbalance causes AUC to be more "discriminating" than accuracy, in the sense that AUC can often correctly distinguish between the performance of two models when accuracy cannot (Huang & Ling, 2005).

4.5. Use of Trap Versus Weather Data

We hypothesize that the greater relative performance of the GNN model at smaller lead times is related to its ability to capitalize on spatial information. To investigate this hypothesis, we assess the performance of our model with and without trap data as input. We know that trap data is more informative for forecasting at smaller lead times, since there is high correlation in trap positivity for observations close to each other in time. On the other hand, weather data is more informative for forecasting at larger lead times, since weather conditions have a large effect on the development of mosquito larvae, but a small effect on the adult mosquito populations that are actually collected in the traps. Therefore, since we intend for the GNN model to harness the spatial information contained only within the trap data, we expect that the lead times where the GNN model outperforms the baseline models are precisely those where the trap data is most informative.

Figure 6 shows the AUC values of the GNN model using trap and weather data, trap data only, and weather data only at increasing lead times. Results from the GNN model using trap and PRISM weather data are also illustrated, which will be discussed in a later section. At the smallest lead times, the AUC values are substantially greater using trap data only rather than using weather data only. At a lead time of 1 week, the AUC value is 0.8568 using trap data only and 0.5972 using weather data only, which is a 30.30% decrease.

On the other hand, the performance difference is greater at the largest lead times, but the relationship is reversed. At a lead time of 7 weeks, the AUC value is 0.7651 using weather data only and 0.3790 using trap data only, which is a 50.46% decrease. Notably, the performance of the model using trap data only in this case is below that of a no-skill model with AUC equal to 0.5. This implies that the trap-only model could simply be equivalent to a no-skill model on the time period covered by the test set, but variance in the test set leads to an AUC value somewhat close, but not equal to 0.5.

It is not wholly surprising that the trap data $A_{i_0}(t_0 - 6)$ is best at forecasting $A_{i_0}(t_0 + 7l)$ for small lead times l, because it is expected that two observations tend to be more correlated when they are close in time and space. That weather data plays a stronger role in prediction with larger lead time could be a signal of a cumulative and delayed effect of weather on mosquito populations and disease. The average *Culex* mosquito life cycle is approximately 6 weeks, so the effects of current weather on mosquito reproduction and development manifest in adult populations approximately 6 weeks into the future (Sim et al., 2015).

Somewhat surprisingly, at both the smallest and largest lead times, it is not the model using both trap and weather data that performs the best, but rather the trap-only and weather-only models, respectively. At the smallest lead times, we speculate that the variance in the labels explained by the weather data is already almost fully explained by the trap data, so adding weather data to the trap-only model only adversely increases model complexity and hence overfitting on the training set. A similar explanation applies to the largest lead times.

Table 2
Confusion Matrix Metrics for Logistic Regression, Fully-Connected Neural Network, XGboost and GNN (k = 5) Models at Various Lead Times (in Weeks) Computed by Letting $p_t = 0.5$

Lead time	Model type	True pos.	False neg.	False pos.	True neg.
1	Log.	800	1,352	485	11,371
	FC	805	1,347	492	11,364
	XGb	**833**	**1,319**	488	11,368
	GNN	830	1,322	**404**	**11,452**
2	Log.	537	1,615	**346**	**11,510**
	FC	635	1,517	431	11,425
	XGb	622	1,530	424	11,432
	GNN	**660**	**1,492**	443	11,413
3	Log.	62	2,090	95	11,761
	FC	81	2,071	104	11,752
	XGb	**160**	**1,992**	122	11,734
	GNN	114	2,038	**85**	**11,771**
4	Log.	5	2,147	11	11,845
	FC	0	2,152	**0**	**11,856**
	XGb	**179**	**1,973**	132	11,724
	GNN	43	2,109	69	11,787
5	Log.	0	2,152	**0**	**11,856**
	FC	0	2,152	**0**	**11,856**
	XGb	**53**	**2,099**	221	11,635
	GNN	0	2,152	**0**	**11,856**
6	Log.	0	2,152	**0**	**11,856**
	FC	0	2,152	**0**	**11,856**
	XGb	**2**	**2,150**	2	11,854
	GNN	0	2,152	**0**	**11,856**
7	Log.	**0**	**2,152**	**0**	**11,856**
	FC	**0**	**2,152**	**0**	**11,856**
	XGb	**0**	**2,152**	143	11,713
	GNN	**0**	**2,152**	**0**	**11,856**

At the intermediate lead times of 3–5 weeks, the AUC for the model using both trap and weather data is marginally greater than that for either model using trap or weather data alone.

4.6. Further Experiments With Weather Data

Although we had already trialed other combinations of weather variables before settling on those that were best-supported by the literature, namely cooling degree days, heating degree days and precipitation, the high performance of the weather-only model prompted further attempts to harness the information contained in the weather data.

Our first set of experiments involved using the existing setup, but with inclusion of the same data at additional lags. That is, instead of using only weather and trap data at the current day t_0, we tested our model using multiple lags with data from days t_0, $t_0 - 7$, data from days t_0, $t_0 - 7$, $t_0 - 14$, and so on. We observed increases in AUC over our existing model of up to a few percent across all lead times, though they were most substantial at the intermediate lead times. These results are illustrated in Figure 7.

In our next set of experiments, we used a similar overall setup as above, but utilized weekly, rather than daily data for the same weather variables. Trap data was included only for day t_0. Furthermore, southern traps in the region with Cfa (humid subtropical) Köppen climate type were associated with weather data from St. Louis Lambert International Airport, rather than Chicago O’Hare International Airport. Again, results were equivocal as performance varied across weather data lags and lead times, although some improvement to model performance could be seen at the greater lead times.

Finally, we attempted to harness weather data available at a higher resolution in widely-used and publicly available gridded data products. The Parameter-elevation Regressions on Independent Slopes Model (PRISM) data set from the PRISM Climate Group at Oregon State University is modeled using climatologically-aided interpolation (CAI), which applies interpolation to daily data using long-term historical trends to estimate regional spatial patterns of climate (PRISM Climate Group, 2014). We used the 4 km resolution data set and employed the “Data Explorer” tool to acquire daily temperature and precipitation data for each of the trap locations. As with the original model, cooling degree days, heating degree days and precipitation were used as model features alongside trap data. The AUC values at increasing lead times are illustrated in Figure 6, labeled “all (PRISM).”

The results using the trap-specific PRISM interpolation data showed an overall performance decrease when compared to the original model using only Chicago O’Hare International Airport station data. Though this is not an unreasonable finding, it is somewhat unexpected, thus additional steps were undertaken to ensure that no errors in data processing had occurred. A number of randomly selected traps and days were selected to check for correlation between the PRISM and Chicago O’Hare International Airport processed weather data. This included verification of the lagged weather data associated with each trap collection day. Furthermore, measurement units, rounding and data type (float vs. integer) were checked for differences.

We note that at the greater lead times, where trap data becomes uninformative, the model using trap and PRISM data still substantially outperforms the model using trap data only. In addition, further investigation showed that a model using only PRISM data is still informative, though underperforms the weather-only model using the Chicago O’Hare International Airport data.

There are several plausible reasons for which trap-specific weather data may decrease, rather than increase, model performance. The increased complexity of the features may make it more difficult for the GNN model to learn generalizable patterns, especially given the small number of observations used relative to that typical in machine

Table 3
Metrics for GNN (k = 5) Model at Various Probability Thresholds p_t

Forecast distance (weeks)	Probability threshold	Precision	Recall (sensitivity)	Specificity
1	0.50	0.6894	0.3857	0.9685
	0.33	0.5606	0.5869	0.9165
	0.17	0.3904	0.7821	0.7783
2	0.50	0.6013	0.3090	0.9628
	0.33	0.5210	0.5135	0.9143
	0.17	0.2885	0.7932	0.6450
3	0.50	0.4067	0.0283	0.9925
	0.33	0.4933	0.3936	0.9266
	0.17	0.2375	0.8606	0.4984
4	0.50	0.3000	0.0028	0.9988
	0.33	0.4100	0.3113	0.9187
	0.17	0.2294	0.9308	0.4324
5	0.50	0.0000	0.0000	1.0000
	0.33	0.3489	0.2500	0.9153
	0.17	0.2159	0.9382	0.3815
6	0.50	0.0000	0.0000	1.0000
	0.33	0.2996	0.0386	0.9836
	0.17	0.2230	0.9270	0.4135
7	0.50	0.0000	0.0000	1.0000
	0.33	0.0000	0.0000	1.0000
	0.17	0.2318	0.9507	0.4282

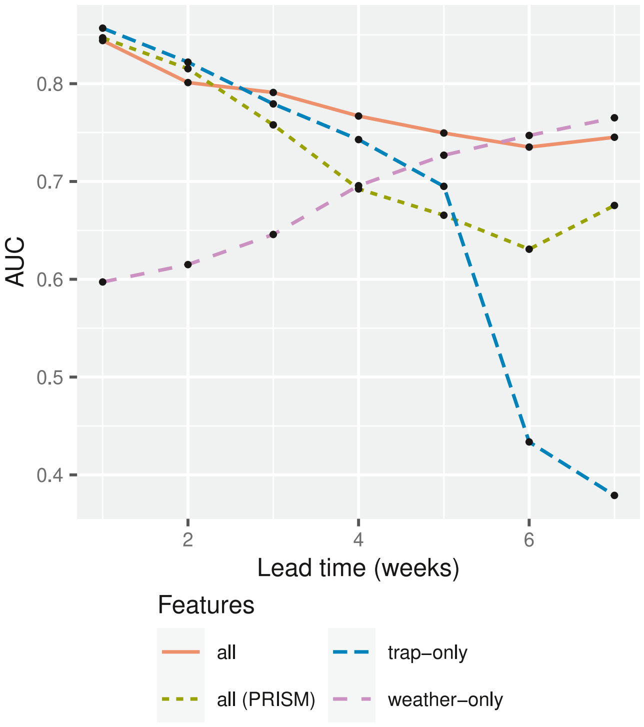

Figure 6. AUC values for GNN (k = 5) model using various features at increasing lead times.

learning models. Keeping in mind that the GNN model is a black-box model focused on prediction rather than inference (Breiman, 2001), we note that although the model architecture remained unchanged, the effective number of features did indeed increase by an order of magnitude. This is because the Chicago O'Hare International Airport data was associated individually with each node and there are several hundred unique trap locations for which PRISM data was acquired. We may also consider a hypothetical scenario where we first used the PRISM data, then conducted feature engineering by daily averaging of all weather values in the same climate region. Increased model performance in such a scenario may be less surprising to our intuition.

From a physical and biological perspective, it is wholly possible that location-specific weather data is not more informative than regional airport weather data. Location-specific data are not a truly accurate reflection of the local weather, since they are interpolated and not measured; a variety of statistical methods are available for the assessment of such differences (Harris et al., 2021). On the other hand, weather data from airports must necessarily be accurate and reliable for airport operations. Furthermore, large airports such as Chicago O'Hare International Airport and St. Louis Lambert International Airport are surrounded by areas of low elevation and located on the periphery of cities, while traps are typically located near residential areas within the interior of cities. Therefore, actual weather conditions at trap locations and regional airports will differ; trap locations are more likely to experience the effects of urban microclimates (Barry & Blanken, 2016), while regional airports are more likely to experience weather conditions more closely resembling the broader surrounding region. In-depth study of the

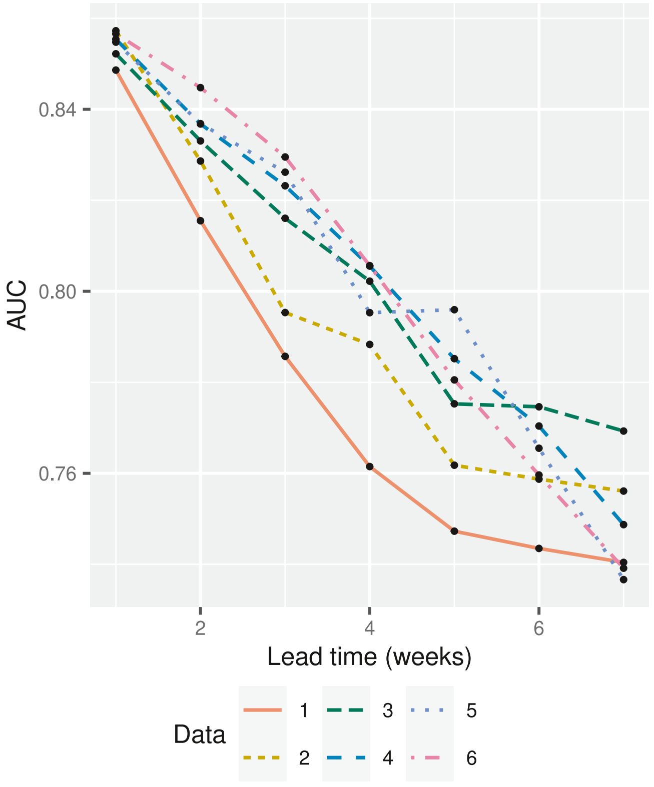

Figure 7. AUC values for model using additional data at increasing lead times, with additional data at extended lags (in weeks).

biology underlying vector-borne WNV transmission is beyond the scope of this work and detailed expositions are available elsewhere (Karki et al., 2020), but we do not discount the possibility that they could sometimes depend more strongly on the weather conditions of the broader surrounding region than those of a local microclimate.

There are myriad ways in which weather and trap data can be combined as features input into the model. Although we have explored several of these, further research would be needed to identify the precise combinations of weather variable type, location and lag which result in optimal model performance at various lead times. In particular, an investigation into harnessing gridded weather data products using GNNs may be valuable. Given that gridded weather data is temporal and lies on a lattice spatial graph, GNNs integrated with combined CNN-RNN units, such as one based on convLSTM (Shi et al., 2015), could be promising.

4.7. Discussion on Graph Isomorphism

Our model is geospatially aware in the sense that the relative positions of traps are encoded in the graph structure and uses edges of the graph to aggregate nearby nodes. The exact positions, via latitude and longitude, are not used except to construct the graph.

Indeed, seeing as the graphs are undirected and only preserve distance between different locations but not directions, they may have isomorphism in the sense that the same graph may be encoded by multiple arrangements of traps. Therefore, the absolute spatial location of traps is not even indirectly encoded. However, we believe that the vast majority of information concerning the relative direction of neighboring traps from a given trap is in fact indirectly preserved. Here, we take "relative direction" to mean that the true absolute angle of a neighboring trap from a given trap may not be recovered, but the angle between a neighboring trap, the given trap, and another neighboring trap may be recovered. This is because if the graphs were fully-connected, such a recovery is possible from the distance matrix via an eigenvalue decomposition (Crippen, 1978). Of course, the graphs are *k*-nearest neighbor graphs, not fully-connected graphs, but Figure 2 shows that the graphs tend to be fully-connected in smaller neighborhoods of nodes. We note that it is within these smaller neighborhoods that we expect relevant features to be pooled by the GNN model for prediction.

One consideration from the above is whether or not this information can actually be utilized by our model. We believe that this raises broader questions about the level of abstraction required of our features and neural network model, with consideration for the relatively small volume of data that is being used. For a specific example of this mentioned earlier, when we attempted to explicitly encode geographic locations by including trap latitudes and longitudes, we actually observed decreased model performance. This is likely due to the fact that the relevant aspects of trap locations, namely distances and angles between neighboring traps, cannot be readily inferred from trap latitudes and longitudes. This is one of our motivations for using a graph-based neural network architecture: the explicit trap locations are encoded into graphs from which paths of disease spread may be more readily inferred.

Overall, there exists a trade-off dependent on the level of abstraction inherent to the features and neural network model. At a low level of abstraction, too much irrelevant information is available to a model with high complexity and flexibility. On the other hand, at a high level of abstraction, relevant information may be lost in highly-engineered features utilized by a model with low flexibility. In our work, we show that the abstraction of trap geographic locations into *k*-nearest neighbor spatial graphs is a feasible modeling approach. Though it would be ideal to conduct the abstraction in such a way as to preserve more problem-relevant information, namely directions between neighboring traps, it is not apparent to us even after extended thought how this may done within the GraphSAGE framework. As mentioned earlier, the GraphSAGE framework was proposed in 2017 (Hamilton et al., 2017), and the representation of spatial information into graphs, particularly for input into GNN models, remains an area of active research (Danel et al., 2020; Gastner & Newman, 2006; Klemmer et al., 2021; Zhang &

Zhao, 2021). Therefore, it appears that further feature engineering or modifications to the GraphSAGE framework to explore better encodings of our spatial data may be beyond the scope of our work.

5. Conclusion

Previous approaches to forecasting mosquito populations and disease do not effectively account for the spatial dependence of geospatial data. This motivates our employment of a GNN model to forecast WNV in Illinois by constructing input graphs using *k*-nearest neighbors. Our model outperforms several baseline models across a number of metrics for short-term forecasting at lead times of 1–5 weeks, demonstrating that GNNs are a powerful and practicable approach for short-term WNV forecasting. In addition, our model's performance is similar to that of the baseline models at larger lead times, suggesting that no trade-off with longer-term forecast performance is necessitated by the employment of a GNN model. However, the relative usefulness of this approach is restricted by the availability of trapping data, and so may be limited in rural areas where trap density is lower. Future work could include an implementation of our methods in cooperation with public health departments using real-time data to further realize the potential social impact of improved WNV forecasting in mosquito abatement efforts. Additionally, an application of a GNN model to mosquito disease forecasting in other regions, or to other vector borne diseases, could be explored. Methods of graph creation beyond *k*-nearest neighbors and the marginal benefit of incorporating CNN and RNN units into the model in such contexts may also be of interest.

Conflict of Interest

The authors declare no conflicts of interest relevant to this study.

Data Availability Statement

Availability statement Data used for the training and testing of the models in this paper has been deposited in the Illinois Data Bank, and can be accessed at (Tonks, 2023b). The code for the models themselves has been made public on GitHub. v1.0.0 has been archived using CERN Data Centre's Zenodo at (Tonks, 2023a). These are both licensed under the GNU Affero General Public License. Data processing was conducted in R 4.2.2 (R Core Team, 2022), and model training and testing was conducted in Python 3.8.3 (Van Rossum & Drake Jr, 2009).

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant NSF-DMS-1830312. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. This publication was supported by Cooperative Agreement U01 CK000505, funded by the Centers for Disease Control and Prevention. Its contents are solely the responsibility of the authors and do not necessarily represent the official views of the Centers of Disease Control and Prevention or the Department of Health and Human Services. Data used in this publication was kindly provided by the Illinois Department of Public Health. Its contents are solely the responsibility of the authors and do not necessarily represent the official views of the Illinois Department of Public Health.

References

Barry, R. G., & Blanken, P. D. (2016). Urban climates. In *Microclimate and local climate* (pp. 243–260). Cambridge University Press. https://doi.org/10.1017/CBO9781316535981.013

Bogado, J. V., Stalder, D., Gómez, S., & Schaerer, C. (2020). Deep learning-based dengue cases forecasting with synthetic data. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, *7*(1).

Bourilkov, D. (2019). Machine and deep learning applications in particle physics. *International Journal of Modern Physics A*, *34*(35), 1930019. https://doi.org/10.1142/S0217751X19300199

Breiman, L. (2001). Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science*, *16*(3), 199–231. https://doi.org/10.1214/ss/1009213726

Bronstein, M. M., Bruna, J., Cohen, T., & Veličković, P. (2021). Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*.

Centers for Disease Control and Prevention. (1999). Outbreak of West Nile-like viral encephalitis–New York, 1999. *MMWR. Morbidity and mortality weekly report*, *48*(38), 845–849.

Cianci, D., Hartemink, N., & Ibáñez-Justicia, A. (2015). Modelling the potential spatial distribution of mosquito species using three different techniques. *International Journal of Health Geographics*, *14*(1), 1–10. https://doi.org/10.1186/s12942-015-0001-0

Colpitts, T. M., Conway, M. J., Montgomery, R. R., & Fikrig, E. (2012). West Nile virus: Biology, transmission, and human infection. *Clinical Microbiology Reviews*, *25*(4), 635–648. https://doi.org/10.1128/cmr.00045-12

Crippen, G. (1978). Note rapid calculation of coordinates from distance matrices. *Journal of Computational Physics*, *26*(3), 449–452. https://doi.org/10.1016/0021-9991(78)90081-5

da Cunha, G. B., Luitgards-Moura, J. F., Naves, E. L. M., Andrade, A. O., Pereira, A. A., & Milagre, S. T. (2010). Use of an artificial neural network to predict the incidence of malaria in the city of Canta, state of Roraima. *Revista da Sociedade Brasileira de Medicina Tropical*, *43*(5), 567–570. https://doi.org/10.1590/s0037-86822010000500019

Danel, T., Spurek, P., Tabor, J., Śmieja, M., Struski, Ł., Słowik, A., & Maziarka, Ł. (2020). Spatial graph convolutional networks. In H. Yang, K. Pasupa, A. C.-S. Leung, J. T. Kwok, J. H. Chan, & I. King (Eds.), *Neural information processing* (pp. 668–675). Springer International Publishing.

Ding, F., Fu, J., Jiang, D., Hao, M., & Lin, G. (2018). Mapping the spatial distribution of Aedes aegypti and Aedes albopictus. *Acta Tropica*, *178*, 155–162. https://doi.org/10.1016/j.actatropica.2017.11.020

Fan, J., Bai, J., Li, Z., Ortiz-Bobea, A., & Gomes, C. P. (2021). A GNN-RNN approach for harnessing geospatial and temporal information: Application to crop yield prediction. *arXiv*. https://doi.org/10.48550/ARXIV.2111.08900

Fernandes, J. N., dos Santos, L. M. B., Chouin-Carneiro, T., Pavan, M. G., Garcia, G. A., David, M. R., et al. (2018). Rapid, noninvasive detection of Zika virus in Aedes aegypti mosquitoes by near-infrared spectroscopy. *Science Advances*, *4*(5), eaat0496. https://doi.org/10.1126/sciadv.aat0496

Fey, M., & Lenssen, J. E. (2019). Fast graph representation learning with PyTorch Geometric. arXiv preprint arXiv:1903.02428.

Fortin, M.-J., & Dale, M. R. T. (2005). *Spatial analysis: A guide for ecologists*. Cambridge University Press. https://doi.org/10.1017/CBO9780511542039

Gastner, M. T., & Newman, M. E. (2006). The spatial structure of networks. *European Physical Journal B: Condensed Matter and Complex Systems*, *49*(2), 247–252. https://doi.org/10.1140/epjb/e2006-00046-8

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., & Dahl, G. E. (2017). Neural message passing for quantum chemistry. In *Proceedings of the 34th international conference on machine learning* (Vol. 70, pp. 1263–1272). JMLR.org.

Gopal, S. (2016). Artificial neural networks in geospatial analysis. In *International encyclopedia of geography* (pp. 1–7). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781118786352.wbieg0322

Hadler, J. L., Patel, D., Nasci, R. S., Petersen, L. R., Hughes, J. M., Bradley, K., et al. (2015). Assessment of arbovirus surveillance 13 years after introduction of West Nile virus, United States. *Emerging Infectious Diseases*, *21*(7), 1159–1166. https://doi.org/10.3201/eid2107.140858

Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. In I. Guyon, et al. (Eds.), *Advances in neural information processing systems* (Vol. 30). Curran Associates, Inc. Retrieved from. Retrieved from https://proceedings.neurips.cc/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7ebea9-Paper.pdf

Harris, T., Li, B., Steiger, N. J., Smerdon, J. E., Narisetty, N., & Tucker, J. D. (2021). Evaluating proxy influence in assimilated paleoclimate reconstructions—Testing the exchangeability of two ensembles of spatial processes. *Journal of the American Statistical Association*, *116*(535), 1100–1113. https://doi.org/10.1080/01621459.2020.1799810

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366. https://doi.org/10.1016/0893-6080(89)90020-8

Huang, J., & Ling, C. (2005). Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, *17*(3), 299–310. https://doi.org/10.1109/TKDE.2005.50

Jiang, W., & Luo, J. (2021). Graph neural network for traffic forecasting: A survey. arXiv preprint arXiv:2101.11174.

Jiang, W., & Zhang, L. (2019). Geospatial data to images: A deep-learning framework for traffic forecasting. *Tsinghua Science and Technology*, *24*(1), 52–64. https://doi.org/10.26599/TST.2018.9010033

Jiménez-Luna, J., Grisoni, F., & Schneider, G. (2020). Drug discovery with explainable artificial intelligence. *Nature Machine Intelligence*, *2*(10), 573–584. https://doi.org/10.1038/s42256-020-00236-4

Joshi, A., & Miller, C. (2021). Review of machine learning techniques for mosquito control in urban environments. *Ecological Informatics*, *61*, 101241. https://doi.org/10.1016/j.ecoinf.2021.101241

Kalipe, G., Gautham, V., & Behera, R. K. (2018). Predicting malarial outbreak using machine learning and deep learning approach: A review and analysis. In *2018 international conference on information technology (ICIT)* (pp. 33–38). https://doi.org/10.1109/ICIT.2018.00019

Karki, S., Brown, W. M., Uelmen, J., Ruiz, M. O., & Smith, R. L. (2020). The drivers of West Nile virus human illness in the Chicago, Illinois, USA area: Fine scale dynamic effects of weather, mosquito infection, social, and biological conditions. *PLoS One*, *15*(5), 1–19. https://doi.org/10.1371/journal.pone.0227160

Kiang, R., Adimi, F., Soika, V., Nigro, J., Singhasivanon, P., Sirichaisinthop, J., et al. (2006). Meteorological, environmental remote sensing and neural network analysis of the epidemiology of malaria transmission in Thailand. *Geospatial Health*, *1*(1), 71–84. https://doi.org/10.4081/gh.2006.282

Kinney, A. C., Current, S., & Lega, J. (2021). Aedes-AI: Neural network models of mosquito abundance. *PLoS Computational Biology*, *17*(11), 1–20. https://doi.org/10.1371/journal.pcbi.1009467

Klemmer, K., Safir, N., & Neill, D. B. (2021). Positional encoder graph neural networks for geographic data. In *CoRR, abs/2111.10144*. Retrieved from https://arxiv.org/abs/2111.10144

Lee, H. H., Chulakadabba, A., Tonks, A., Yang, Z., & Wang, C. (2017). Predicting the occurrence of haze events in Southeast Asia using machine learning algorithms. In *AGU fall meeting abstracts* (Vol. 2017, p. PA31C-02).

Lee, K. Y., Chung, N., & Hwang, S. (2016). Application of an artificial neural network (ANN) model for predicting mosquito abundances in urban areas. *Ecological Informatics*, *36*, 172–180. https://doi.org/10.1016/j.ecoinf.2015.08.011

McDonald, E., Mathis, S., Martin, S. W., Staples, J. E., Fischer, M., & Lindsey, N. P. (2021). Surveillance for West Nile virus disease—United States, 2009–2018. *American Journal of Transplantation*, *21*(5), 1959–1974. https://doi.org/10.15585/mmwr.ss7001a1

Mishra, V. K., Tiwari, N., & Ajaymon, S. (2019). Dengue disease spread prediction using twofold linear regression. In *2019 IEEE 9th international conference on advanced computing (IACC)* (pp. 182–187). https://doi.org/10.1109/IACC48062.2019.8971567

Mitchell, R., & Frank, E. (2017). Accelerating the XGBoost algorithm using GPU computing. *PeerJ Computer Science*, *3*, e127. https://doi.org/10.7717/peerj-cs.127

Moore, A. (1991). Artificial neural network trained to identify mosquitoes in flight. *Journal of Insect Behavior*, *4*(3), 391–396. https://doi.org/10.1007/bf01048285

Mudele, O., Frery, A. C., Zanandrez, L. F., Eiras, A. E., & Gamba, P. (2021). Modeling dengue vector population with earth observation data and a generalized linear model. *Acta Tropica*, *215*, 105809. https://doi.org/10.1016/j.actatropica.2020.105809

Mussumeci, E., & Codeço Coelho, F. (2020). Large-scale multivariate forecasting models for dengue - LSTM versus random forest regression. *Spatial and Spatio-temporal Epidemiology*, *35*, 100372. https://doi.org/10.1016/j.sste.2020.100372

Pei, H., Wei, B., Chang, K. C.-C., Lei, Y., & Yang, B. (2020). Geom-gcn: Geometric graph convolutional networks. arXiv preprint arXiv:2002.05287.

Pham, D. N., Aziz, T., Kohan, A., Nellis, S., Jamil, J. B. A., Khoo, J. J., et al. (2018). How to efficiently predict dengue incidence in Kuala Lumpur. In *2018 fourth international conference on advances in computing, communication & automation (ICACCA)* (pp. 1–6). https://doi.org/10.1109/ICACCAF.2018.8776790

PRISM Climate Group. (2014). PRISM climate data [Computer software manual]. *PRISM Climate Group*. Retrieved from https://prism.oregonstate.edu

R Core Team. (2022). R: A language and environment for statistical computing [Computer software manual]. *R Core Team*. Retrieved from https://www.R-project.org

Ripoche, M., Campagna, C., Ludwig, A., Ogden, N. H., & Leighton, P. A. (2019). Short-term forecasting of daily abundance of West Nile virus vectors Culex pipiens-restuans (Diptera: Culicidae) and Aedes vexans based on weather conditions in southern Québec (Canada). *Journal of Medical Entomology*, *56*(3), 859–872. https://doi.org/10.1093/jme/tjz002

Ronca, S. E., Ruff, J. C., & Murray, K. O. (2021). A 20-year historical review of West Nile virus since its initial emergence in North America: Has West Nile virus become a neglected tropical disease? *PLoS Neglected Tropical Diseases*, *15*(5), 1–20. https://doi.org/10.1371/journal.pntd.0009190

Rubio-Solis, A., Musah, A., P. Dos Santos, W., Massoni, T., Birjovanu, G., & Kostkova, P. (2019). ZIKA virus: Prediction of Aedes mosquito larvae occurrence in Recife (Brazil) using online extreme learning machine and neural networks. In *Proceedings of the 9th international conference on digital public health* (pp. 101–110). Association for Computing Machinery. https://doi.org/10.1145/3357729.3357738

Sass, D., Li, B., Clifton, M., Harbison, J., Xamplas, C., & Smith, R. (2022). The impact of adulticide on Culex abundance and infection rate in North Shore of Cook County, Illinois. *Journal of the American Mosquito Control Association*, *38*(1), 46–58. https://doi.org/10.2987/21-7036

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., & Woo, W.-C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 28). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper/2015/file/07563a3fe3bbe7e3ba84431ad9d055af-Paper.pdf

Sim, C., Kang, D. S., Kim, S., Bai, X., & Denlinger, D. L. (2015). Identification of FOXO targets that generate diverse features of the diapause phenotype in the mosquito Culex pipiens. *Proceedings of the National Academy of Sciences*, *112*(12), 3811–3816. https://doi.org/10.1073/pnas.1502751112

Soliman, A., & Terstriep, J. (2019). Keras Spatial: Extending deep learning frameworks for preprocessing and on-the-fly augmentation of geospatial data. In *Proceedings of the 3rd ACM SIGSPATIAL international workshop on AI for geographic knowledge discovery* (pp. 69–76). Association for Computing Machinery. https://doi.org/10.1145/3356471.3365240

Staples, J. E., Shankar, M. B., Sejvar, J. J., Meltzer, M. I., & Fischer, M. (2014). Initial and long-term costs of patients hospitalized with West Nile virus disease. *The American Journal of Tropical Medicine and Hygiene*, *90*(3), 402–409. https://doi.org/10.4269/ajtmh.13-0206

Tonks, A. (2023a). adtonks/wnv-gnn: v1.0.0 [software]. *Zenodo*. https://doi.org/10.5281/zenodo.7897830

Tonks, A. (2023b). Data for the paper "Forecasting West Nile Virus with graph neural networks: Harnessing spatial dependence in irregularly sampled geospatial data" [Dataset]. *University of Illinois at Urbana-Champaign*. https://doi.org/10.13012/B2IDB-3628170_V1

Van Rossum, G., & Drake Jr, F. L. (2009). *Python 3 reference manual*. Scotts Valley, CA: CreateSpace.

Wang, C. (2021). Forecasting and identifying the meteorological and hydrological conditions favoring the occurrence of severe hazes in Beijing and Shanghai using deep learning. *Atmospheric Chemistry and Physics*, *21*(17), 13149–13166. https://doi.org/10.5194/acp-21-13149-2021

Wang, J., Ogden, N. H., & Zhu, H. (2011). The impact of weather conditions on Culex pipiens and Culex restuans (Diptera: Culicidae) abundance: A case study in Peel region. *Journal of Medical Entomology*, *48*(2), 468–475. https://doi.org/10.1603/ME10117

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2021). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(1), 4–24. https://doi.org/10.1109/TNNLS.2020.2978386

xgboost developers. (2022). Xgboost parameters (revision 07b2d5a2.). Retrieved from https://xgboost.readthedocs.io/en/latest/parameter.html

Xiao, L., Lo, S., Zhou, J., Liu, J., & Yang, L. (2021). Predicting vibrancy of metro station areas considering spatial relationships through graph convolutional neural networks: The case of Shenzhen, China. *Environment and Planning B: Urban Analytics and City Science*, *48*(8), 2363–2384. https://doi.org/10.1177/2399808320977866

Xu, J., Xu, K., Li, Z., Meng, F., Tu, T., Xu, L., & Liu, Q. (2020). Forecast of dengue cases in 20 Chinese cities based on the deep learning method. *International Journal of Environmental Research and Public Health*, *17*(2), 453. https://doi.org/10.3390/ijerph17020453

Yang, Z., Xu, Q., Bao, S., Cao, X., & Huang, Q. (2022). Learning with multiclass AUC: Theory and algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(11), 7747–7763. https://doi.org/10.1109/TPAMI.2021.3101125

Zhang, Z., & Zhao, L. (2021). Representation learning on spatial networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, & J. W. Vaughan (Eds.), *Advances in neural information processing systems* (Vol. 34, pp. 2303–2318). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper/2021/file/12e35d9186dd72fe62fd039385890b9c-Paper.pdf

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., et al. (2020). Graph neural networks: A review of methods and applications. *AI Open*, *1*, 57–81. https://doi.org/10.1016/j.aiopen.2021.01.001

Zhu, D., Liu, Y., Yao, X., & Fischer, M. M. (2021). Spatial regression graph convolutional neural networks: A deep learning paradigm for spatial multivariate distributions. *GeoInformatica*, *26*(4), 645–676. https://doi.org/10.1007/s10707-021-00454-x

Zhu, D., Zhang, F., Wang, S., Wang, Y., Cheng, X., Huang, Z., & Liu, Y. (2020). Understanding place characteristics in geographic contexts through graph convolutional neural networks. *Annals of the Association of American Geographers*, *110*(2), 408–420. https://doi.org/10.1080/24694452.2019.1694403