Analysis of Light Pollution's Impact on Biodiversity in Hong Kong

A Data-Intensive Science Project

Abstract

This project investigates the ecological impact of light pollution on species richness in Hong Kong by integrating night sky brightness (NSB) and species occurrence data. We developed a comprehensive data processing pipeline for spatiotemporal data, built multivariate regression models achieving $R^2 > 0.99$, and designed a Graph Attention Network (GAT) with Long Short-Term Memory (LSTM) model to capture complex spatiotemporal dynamics. Our results demonstrate a significant negative correlation between light pollution and biodiversity, highlighting the critical need for environmental mitigation strategies.

Submitted by: SU Zhiya (22254706) - Primary Contributor for Data Preprocessing & Statistical Modeling
Teammates: JIA Hansen (22256733), DOU Jiabao (22258248)

Date of Submission: May 2025
University: Hong Kong Baptist University
Course: MATH3836 - Data Mining

GitHub Repository

For full code implementation, please visit: https://github.com/Sophia0514/MATH3836_Project

My Key Contributions

- Led the statistical analysis and data preprocessing pipeline for the project. Spearheaded the integration of heterogeneous spatiotemporal datasets (night sky brightness & species occurrence) using Python, which involved meticulous cleaning, anomaly detection, and feature engineering.
- Designed and implemented the core geospatial data unification strategy by developing a spatial matching algorithm to correlate species data with light pollution metrics and applying Kriging interpolation to transform discrete sensor readings into a continuous spatial field for comprehensive analysis.
- Solely developed and optimized the multivariate regression modeling framework. Systematically advanced the model from simple linear to high-degree polynomial regression, ultimately employing regularization techniques (Ridge/Lasso) to effectively solve overfitting and achieve a predictive accuracy of $R^2 > 0.99$.
- Provided critical analytical support for the deep learning component by ensuring the statistical integrity of the data fed into the hybrid GAT-LSTM architecture and assisting in the interpretation of results to bridge statistical findings with neural network predictions.

Contents

1 Introduction

Light pollution is everywhere in our daily lives, but do you really understand it? Light pollution is a problem that has arisen from the excessive use of lighting systems by humans. The most obvious effect is the disappearance of stars in the urban night sky, which are covered by the lights of numerous buildings. This puts research into the observation of the universe in trouble, and it also causes serious damage to human health and the ecological balance. Light pollution in Hong Kong is an overlooked but far-reaching environmental problem that poses a potential threat to ecosystems and biodiversity, and there is no system to research how the light pollution affects the life of organisms. Therefore, our project tries to study the effect of light pollution on species group number.

2 Data Preprocessing

2.1 Dataset Selection

In our project, we select two datasets.

One is the Global night sky brightness monitoring network supported by National Science Foundation's NOIRLab (https://globeatnight.org/gan-mn/).

The other is the Occurrence Data of Hong Kong Species supported by Hong Kong Open Data Platform
(https://data.gov.hk/sc-data/dataset/hk-afcd-afcdlist-hkspeciesoccurrencecsdi).

2.2 Dataset Interpretation

The following the the interpretation of the table header of two datasets.

2.2.1 Dataset Interpretation (Hong Kong night sky brightness)

id: unique identifier for each row of data
created: the time the data record was created
received_utc: the UTC time the data was received
received_adjusted: the adjusted reception time
sqmle_serial_number: the serial number of the device, usually used to uniquely identify the device
nsb: Night Sky Brightness
sensor_frequency: the operating frequency of the sensor
sensor_period_count: sensor cycle count
sensor_period_second: the number of seconds of the sensor cycle
temperature: the temperature at the time of recording
device_code: device code

2.2.2 Dataset Interpretation (Occurrence Data of Hong Kong Species)

OBJECTID: Unique identifier for each row of data
scientific: The scientific name of the animal
family: The family to which the animal belongs
date: The date the data was recorded
OBJECTID_1: Another identifier
gno: Number of species
Shape_Length: The perimeter of the observed geometric shape
Shape_Area: The area of the observed geometric shape
geometry: The specific observed geometry information including longitude and latitude

2.3 Preprocessing for Hong Kong night sky brightness

Based on the night light intensity data (NSB) collected from 2022 to 2024, through the data analysis, cleaning and consolidation, we reveal the pattern of change of night light intensity in Hong Kong from both temporal and spatial dimensions.

2.3.1 Data sources and preliminary consolidation

The original data come from night light intensity records collected from several photometric sites in Hong Kong. In order to realize effective mapping of the geographic locations of the observation sites, we introduce spatial mapping data location and map _rel, which are used to store the latitude and longitude information corresponding to the device number and to match the device number with the name of the specific observation site, respectively.

A complete dataset containing data for the whole year was constructed by reading and merging data from all months. Subsequently, we extracted temporal features such as year, month, date and hour from timestamps to lay the foundation for subsequent multi-scale analysis.

2.3.2 Anomaly Data Detection and Cleaning

Since there may be missing values, corrupted data, or extreme anomalies in the raw data that affect the accuracy of the analysis results, we performed anomaly detection and and filtering of the data before analysis and modeling.

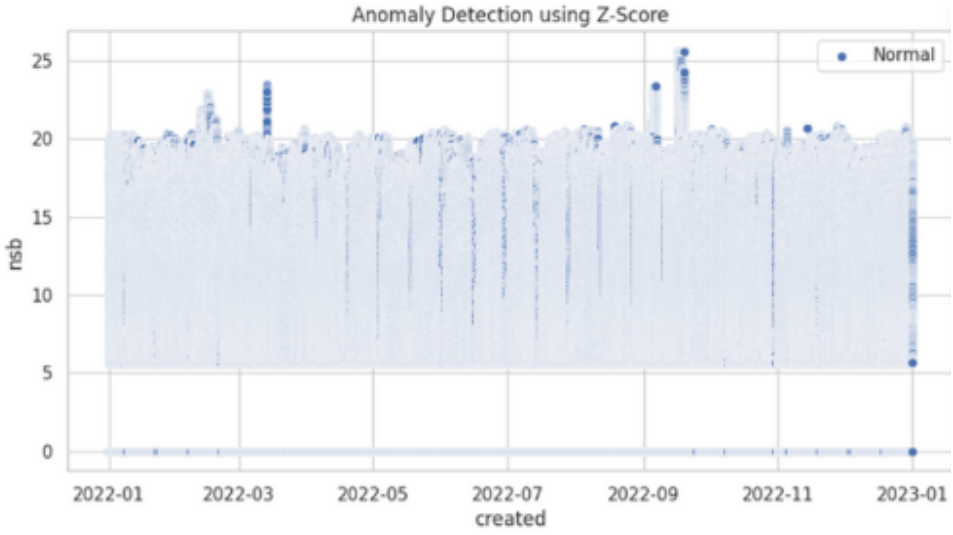
(1) Z-Score Detection of Numerical Anomalies

Z-Score is a commonly used normalization method to identify outliers that deviate from the mean. The formula is as follows:

$$Z = \frac{x - \mu}{\sigma}$$

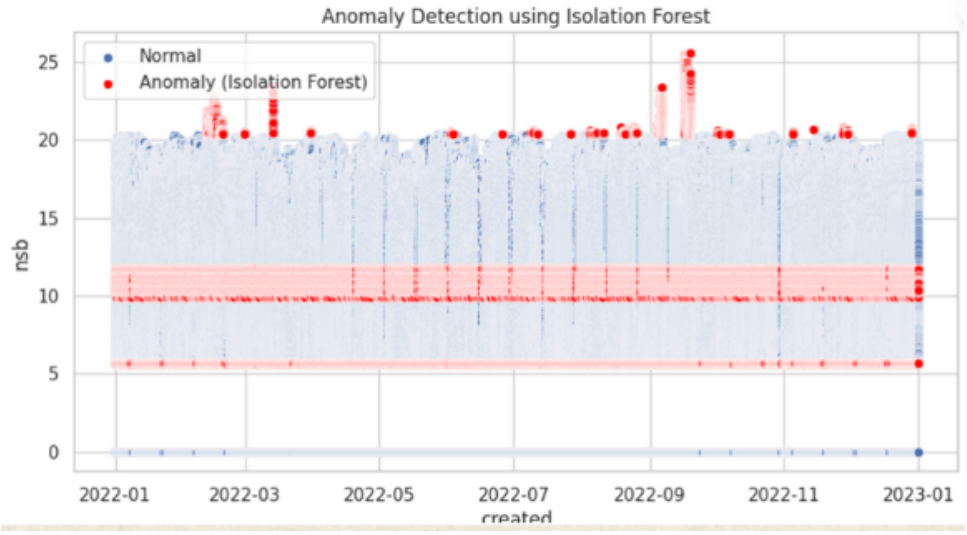
Where x is the sample value, μ is the mean, and σ is the standard deviation. The sample is usually considered an outlier when $|Z| > 3$. We apply the Z-Score method to the

nsb field to identify and exclude light intensity values that significantly deviate from the normal range.



(2) Isolation Forest to detect high-dimensional anomalies

In addition to variate anomaly detection, we use the Isolation Forest algorithm to identify anomalous behaviors in the multidimensional feature space. This algorithm isolates sample points by randomly selecting features and continuously splitting the data; anomalies are usually easier to isolate, i.e., fewer splits are required.



(3) Missing Value and Format Error Handling

In addition, we check the integrity of time fields such as hour and month, and delete the records with missing values. We also cleaned up the redundant commas in the latitude and longitude fields and converted them to floating point format to ensure that they can be used for geo-visualization. After the above steps, an effective training dataset with uniform structure and no anomaly interference is finally obtained.

2.3.3 Temporal feature extraction and statistics

(1) Time dimension feature extraction

The following time features are extracted from the timestamp field created:

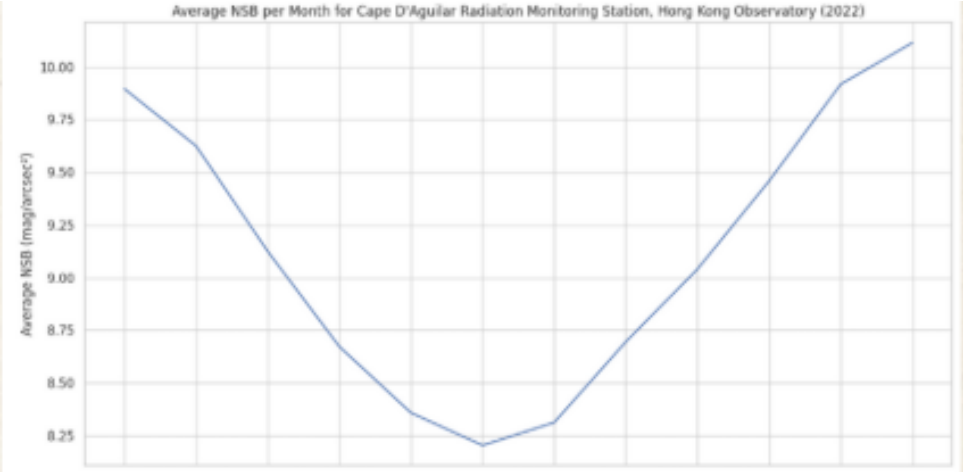
- Year
- Month
- Day
- Hour

(2) Statistical analysis and graphical presentation

In order to reveal the trend of light intensity over time, we have drawn three types of graphs based on monthly, quarterly and hourly dimensions respectively:

• Line graph by months

For each observation point, grouped by month, the average light intensity value is calculated and a line graph is plotted to show the trend of light intensity for the 12 months of the year. For example, some observation points show a significant increase in light intensity in December, which may be related to Christmas lights, and a decrease in light intensity in the summer months due to weather (e.g., rainfall, cloud cover).

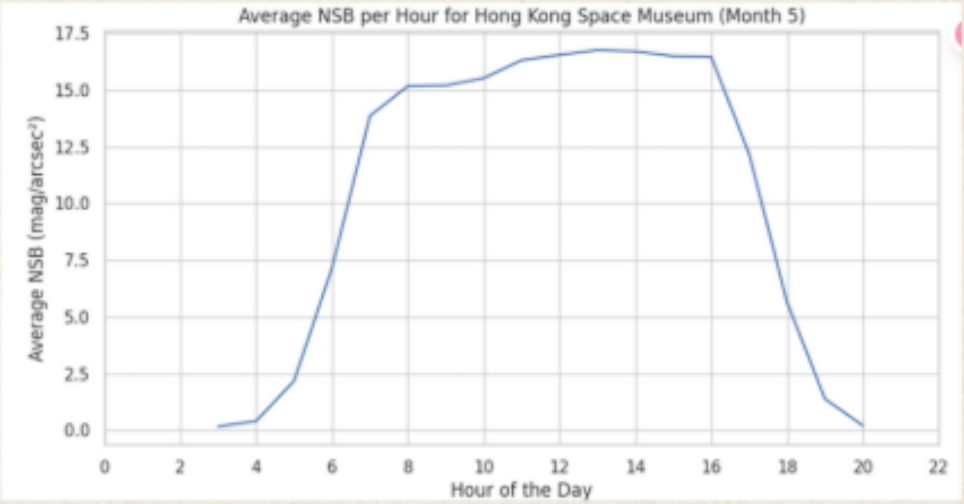


• Bar graph by quarter

Divide the 12 months into four quarters (Q1-Q4) and calculate the average light intensity values for each quarter. Compare the differences between different quarters through a bar graph, in order to analyze whether there are seasonal fluctuations in the city's night light intensity. For example, the fourth quarter (Q4) is generally higher than other quarters, showing a clear trend of "high in winter and low in summer".

• Line graph by hours

Refine the time granularity to 24 hours per day, calculate the average light intensity for each hour of each month for each observation point, and draw a line graph. This was used to analyze the temporal pattern of urban night lighting. For example, light intensity peaks in most areas between 7 and 10 p.m. and decreases after 2 a.m., but remains high in some commercial districts, reflecting continuous night activity.



2.3.4 Spatial feature extraction and statistics

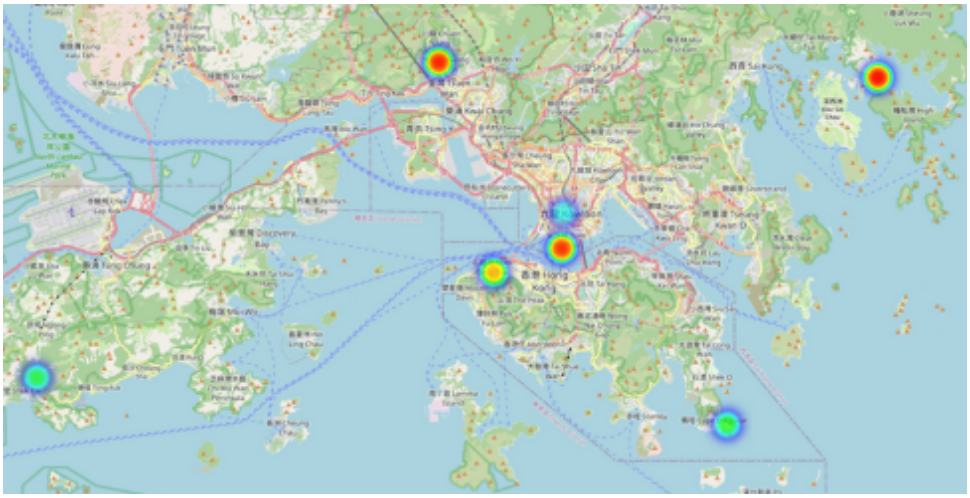
After completing the statistical analysis in the time dimension, in order to further reveal the spatial distribution characteristics of the night light intensity (NSB), we performed spatial mapping and modeling of the data, which mainly included: heat map visualization, K-Means clustering, DBSCAN clustering, and kriging interpolation

(1) Data spatial mapping and heat map visualization

In order to correlate the light intensity data with the geographic location, we merge the raw light intensity data with the map_rel and location files through the device number device_code, and finally get the latitude and longitude information of each observation point. Subsequently, the Folium library is used to create a map object and draw a HeatMap based on the average light intensity values with the following formula.

$$H(x,y) = \frac{1}{n}\sum_{i=1}^{n} w_i \cdot f(x_i, y_i)$$

H(x,y) denotes the thermal intensity at location (x,y), f(xi,yi) denotes the light intensity value at the ith observation point, and wi is the weighting factor. Through the heat map, we visualize the distribution of night light pollution in various regions of Hong Kong.



(2) K-Means based clustering analysis

We used K-Means clustering algorithm to group the observation points. Firstly, the data containing longitude, latitude and average light intensity were standardized:

$$x' = \frac{x - \mu}{\sigma}$$

x is the original value, μ is the mean value, σ is the standard deviation, and the number of clusters k=5.

The results show that there are significant differences in night light intensity in different areas. For example, the commercial area of Tsim Sha Tsui generally belongs to high luminance clusters, while mountainous areas such as Shatin are mostly distributed in low luminance clusters.

(3) DBSCAN Clustering to Identify Noise and Localized Dense Regions

Since K-Means is sensitive to the initial clustering center and fails to identify anomalies, we further employ the DBSCAN algorithm for additional analysis. DBSCAN can effectively identify isolated points and find non-spherical distribution of cluster structures, which is suitable for detecting localized high-intensity illumination areas or anomalous light pollution points in urban areas of Hong Kong.

(4) Kriging interpolation to construct a continuous light intensity surface

In order to obtain a finer spatial distribution of light intensity, we use the Kriging interpolation method to predict the region between observation points.

$$\hat{Z}(s_0) = \sum_{i=1}^{n} \lambda_i Z(s_i)$$

$\hat{Z}(s_0)$ is the estimated value at location s_0,

$Z(s_i)$ is the observed value at known location s_i,

λ_i are the weights assigned to each known observation point.

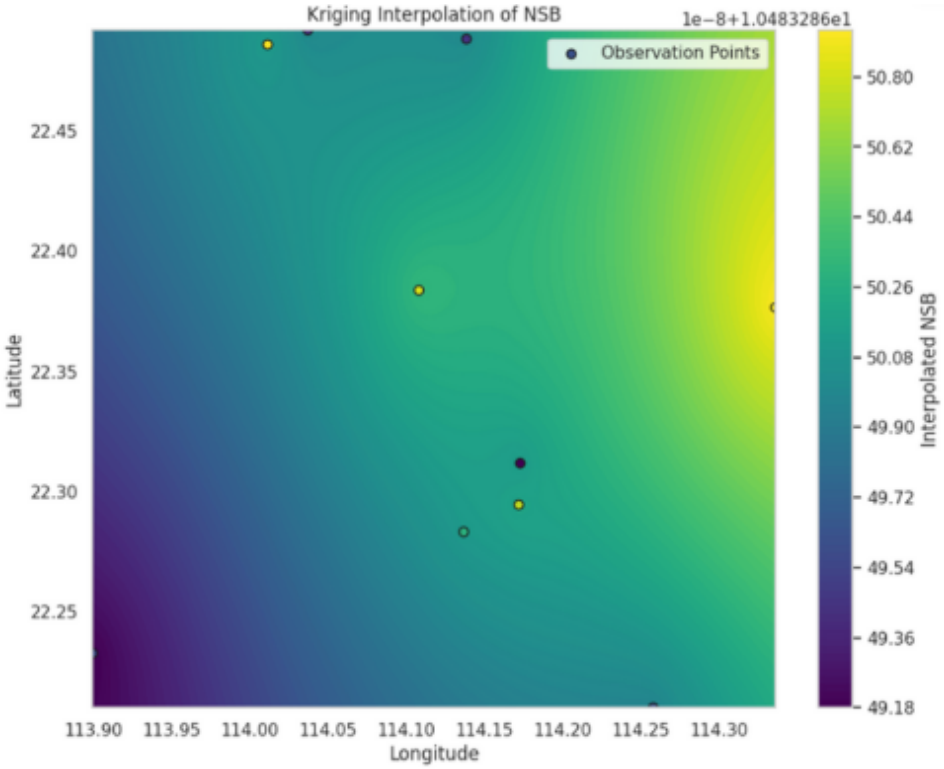
$$\sum_{i=1}^{n} \lambda_i = 1$$

$$\text{Var}\left(\hat{Z}(s_0) - Z(s_0)\right) \to \min$$

$\hat{Z}(s_0)$ is the estimated value at location s_0,

$Z(s_0)$ is the true (but unknown) value at location s_0.

The interpolation calculation is performed by the linear variational function model to generate a gridded light intensity map covering the whole study area. The interpolation results not only fill in the blank regions between observation points, but also provide a continuous description of the trend of light intensity changes.



2.4 Preprocessing for Occurrence Data of Hong Kong Species

2.4.1 Data Preprocessing

In this section, we use Python's Pandas library to clean, group, and perform statistical analysis on Hong Kong species data. First, we read the raw data using the `pd.read_csv()` function, and use `df['geometry'].astype(str)` to convert the geometry column to string format for grouping.

Then, we try to use `pd.to_datetime(df['date'])` to convert the date column to a standard date format. If that fails, we use `str.split()` to extract the month information in the date, and handle various date format anomalies such as `MM/DD/YYYY` or `YYYY-MM`.

Next, we use `groupby(['scientific', 'geometry_str', 'month'])` to group the data by species name, geographic information, and month, and use `agg({'gno': 'mean', 'OBJECTID': lambda x: list(x)})` to calculate the mean `gno` and `OBJECTID` list for each group, and use `apply(len)` to count the number of samples in each group.

Finally, we use `sort_values(['scientific', 'month'])` to sort by species name and month, save the result as a file through `to_csv`, and display the table results of the first 20 rows.

2.4.2 Visualization for Occurrence Data of Hong Kong Species

We use Python's Pandas, Numpy, and Plotly libraries to visualize Hong Kong species data in April 2022 in 3D. The data is read using `pd.read_csv(file_path)`, and invalid rows are filtered with `df['OBJECTID'].isna()`. The `extract_coordinates` function extracts longitude and latitude from `geometry_str`.

GNO values are logarithmically scaled using `np.log1p(april_data['gno_numeric']) * scale_factor`. A 3D scatter plot is created using `go.Scatter3d` to show species distribution. A thermal surface map (`go.Surface`) is generated via grid interpolation to illustrate spatial variation of GNO values.

The 3D map is saved as an HTML file using `fig.write_html(output_file)`. Species statistics, such as `april_data['scientific'].value_counts()`, and the top 10 common species are outputted, providing a tool for exploring species distribution.

3 Model Selection and Result

3.1 Model Selection

Since the distribution of light sky brightness observation stations is discrete, and the occurrence of HK species is continuous, therefore, we need to unify the types of these two datasets, converting them into either two continuous datasets or two discrete datasets. The following sections show three methods. The first method is a continuous method, namely changing the night sky brightness from a discrete dataset to a continuous dataset and using the regression model. The second and the third methods are discrete methods, namely changing the occurrence of HK species from a continuous dataset to a continuous dataset. To be specific, the second method is a regression model, and the third is a deep learning model, the Spatiotemporal prediction model.

3.2 Regression Model

3.2.1 Startup for regression model

We mainly use two columns of data from our two datasets. One is from the occurrence of HK species- the number of species (gno), another is from the global night sky brightness monitoring network- night sky brightness (nsb). Firstly, we group data with the same longitude, latitude, and month into one set, with each group representing an observation point and a month.

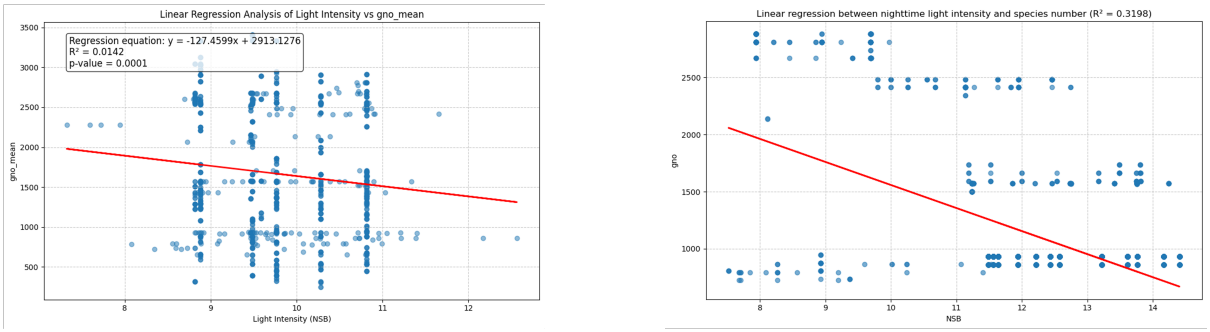
3.2.2 Simple Regression model- Continuous method

In this section, we need to change the night sky brightness from a discrete dataset to a continuous dataset. In processing night sky brightness data, we applied the Kriging interpolation method and selected the spherical model as the variogram function. Furthermore, we use a library, the PyKrige library, to interpolate the discrete light intensity observation data onto a regular grid, and generate a continuous light intensity distribution to cover a whole study area.

3.2.3 Simple Regression model- Discrete method

In this section, we need to change the occurrence data of HK species from a continuous dataset to a discrete dataset. Our objective is to match the coordinates of each species to the coordinates of sky brightness observation points. Since each sky brightness observation point has an effective range of approximately a circular area with a two-kilometer radius, we calculated the distance between the coordinates of each species and the coordinates of sky brightness observation points. If this distance is less than two kilometers, we consider that the species coordinates correspond to that sky brightness observation point.

3.2.4 Comparison between two simple regression model & Preliminary result



(a) Continuous Method (b) Discrete Method

Figure 1: Comparison of Discrete and Continuous Regression Methods

We can see from the above two pictures, the R square (0.3198) of the discrete method is greater than that of the continuous one (0.0142). Therefore, we eliminate the continuous

regression method and use the discrete method for our further study. Besides, we can find from both two results: light intensity has a weak negative correlation with number of species.

3.2.5 Multivariate Regression Model (category-simple)

Although we get a negative correlation between the number of species and the night sky brightness, the R square is quite small (0.3198). We considered that different types of species have different living habits, so we hypothesized that the number of species is related not only to sky brightness intensity but also to the biological category of the species itself. We attempted to classify more than 600 species from the species data according to biological taxonomy, dividing them into the following nine categories and corresponding each of them with one binary (dummy) variables from x2 to x10:

1: Birds (x2)
2: Mammals (x3)
3: Reptiles (x4)
4: Amphibians (x5)
5: Fish (x6)
6: Insects (Butterflies and Moths) (x7)
7: Insects (Dragonflies and Damselflies) (x8)
8: Insects (Fireflies and Beetles) (x9)
9: Other Invertebrates (x10)

Take x2 as an example, if an animal is a kind of bird, then x2=1, otherwise, x2=0.

After adding categories as the first dummy variables, we get the picture shown below.

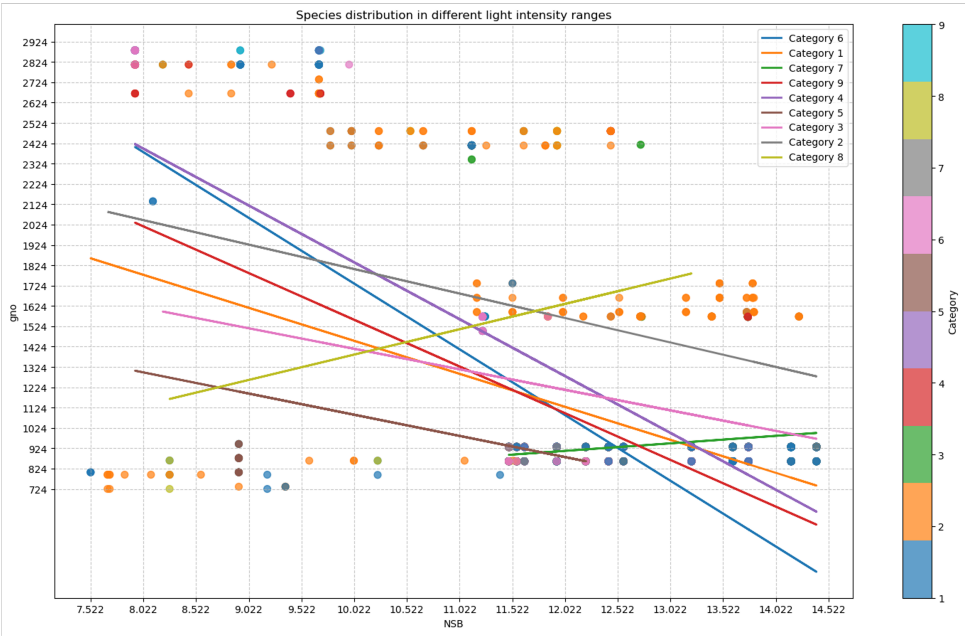


Figure 2: After adding categories as the first dummy variables

In this picture, we represented each species with different colors and connected the same colors, resulting in individual regression lines for each species. And the R square in this case is 0.41, which is greater than the previous 0.3198, showing that our consideration is reasonable.

3.2.6 Multivariate Regression Model (category-interaction-degree 1&2)

We consider that there may exist some interaction between "night sky brightness" and "category", hence we add some interaction term nsb*category(i), where 1<i<9 (inclusive). The result is shown below:

```
线性回归模型统计数据:
截距 (Intercept): -3553369824598.9478
系数 (Coefficients):
  nsb: -108895799809393.75
  category_1: 3553369827678.133
  category_2: 3553369827609.1987
  category_3: 3553369827019.9033
  category_4: 3553369829223.547
  category_5: 3553369826736.342
  category_6: 3553369829570.371
  category_7: 3553369825020.365
  category_8: 3553369824709.471
  category_9: 3553369828434.0034
  nsb_category_1: 108895799809231.47
  nsb_category_2: 108895799809273.58
  nsb_category_3: 108895799809293.25
  nsb_category_4: 108895799809115.48
  nsb_category_5: 108895799809288.78
  nsb_category_6: 108895799809070.75
  nsb_category_7: 108895799809434.31
  nsb_category_8: 108895799809520.31
  nsb_category_9: 108895799809165.78
R² 值: 0.46
```

Figure 3: Interaction between night sky brightness and category

Compared with the simple multivariate regression model, which has a R square equals to 0.41, after adding the interaction terms, the R square has a slight increase.

We further add some second degree order interactions, namely interaction term: nsb^2*category(i), where 1<i<9 (inclusive), and use different kinds of regression models to realize regularization. The following shows our result:

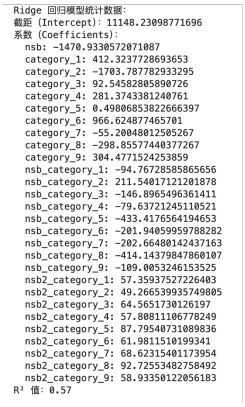


Figure 4: Ridge

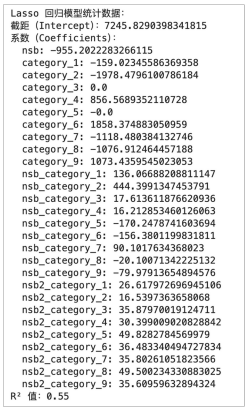


Figure 5: Lasso

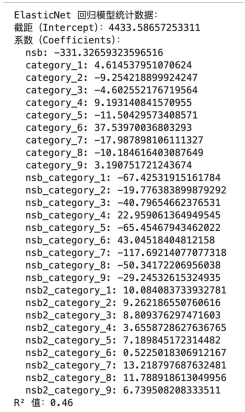


Figure 6: Elastic-Net

Among the three regression models (Ridge, Lasso and ElasticNet), Ridge performs the best with 0.57 R square.

3.2.7 Multivariate Regression Model (category-interaction-high degree)

In this section, we further add some higher degree interactions, namely interaction term: nsb^j*category(i), where 1<i<9 (inclusive), 1<j<5 (inclusive). The following are the results:

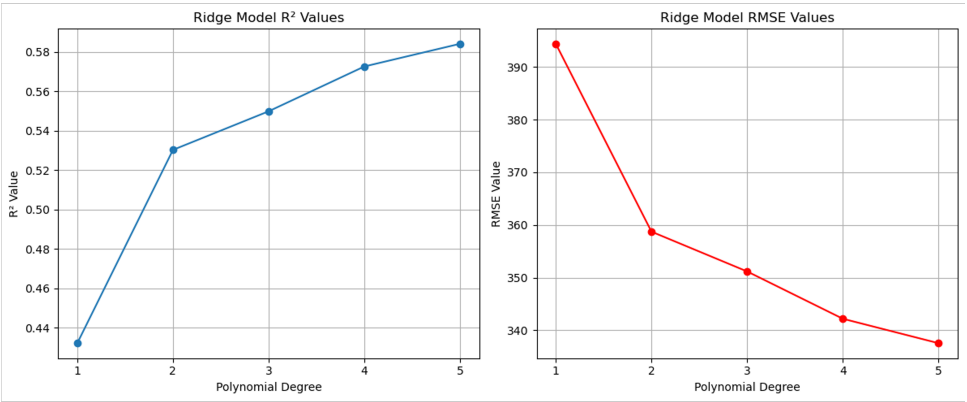


Figure 7: R square and RMSE of the Model

3.2.8 Multivariate Regression Model (category & station-simple)

Until now, we get the best result from the Ridge regression, which explains more than half of the variation. We want our model to be more accurate with a higher R square. After consideration, we discover that the location of species also influences the number of species. Previously, when determining the light intensity for each species, we calculated the distance between the species and observation points. If the distance was less than 2 kilometers, we established one-to-one correspondences. Therefore, we used the locations of observation points to represent the position of each species, and introduce observation points as the second dummy variable. In our project, we named this dummy variable

```
Ridge 回归模型包含到 nsb^1 的交互项:
截距 (Intercept): 1079.9542206475523
R² 值: 0.4321
RMSE 值: 394.4274

Ridge 回归模型包含到 nsb^2 的交互项:
截距 (Intercept): 1079.2541160811577
R² 值: 0.5303
RMSE 值: 358.7214

Ridge 回归模型包含到 nsb^3 的交互项:
截距 (Intercept): 1079.8378678481806
R² 值: 0.5499
RMSE 值: 351.1601

Ridge 回归模型包含到 nsb^4 的交互项:
截距 (Intercept): 1080.5329018017112
R² 值: 0.5726
RMSE 值: 342.1713

Ridge 回归模型包含到 nsb^5 的交互项:
截距 (Intercept): 1080.7358046566392
R² 值: 0.5841
RMSE 值: 337.5277
```

Figure 8: Value of RMSE

"station". Since there are no species matching with station1, and there are 10 observation points in total, we only introduced 9 new variables ranging from the following x11 to x19:

1: station2 (x11)
2: station3 (x12)
3: station4 (x13)
4: station5 (x14)
5: station6 (x15)
6: station7 (x16)
7: station8 (x17)
8: station9 (x18)
9: station10 (x19)

Take station2 as an example, if an observation point is station2, x11=1, otherwise, x11=0. We got an extremely surprising result after adding the second dummy variable, which shown below, the R square equals to ONE!

We think this result may be caused by "overfitting", so we introduce regularization and split the data into training and test sets with the percentage 80% to 20%. The results are shown below:

We can see that the R square has a slight decrease to 0.99, which is good enough. And also, the RMSE for both training set and testing set is small enough. Our model has been improved a lot!

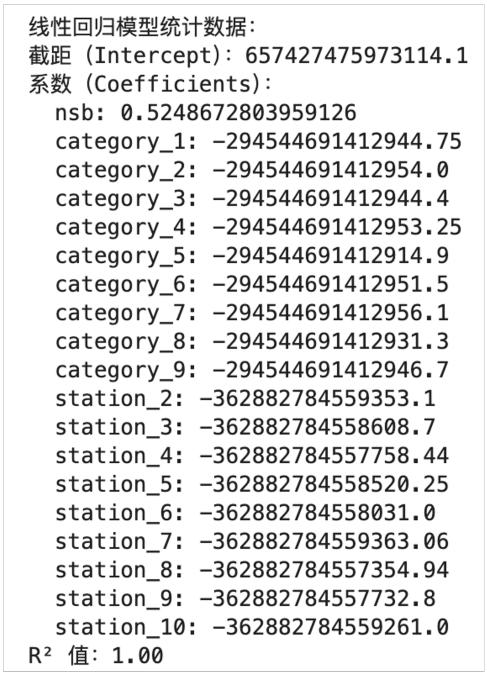


Figure 9: Statistical Results of the Linear Regression Model

Figure 10: Result of the Ridge Medel

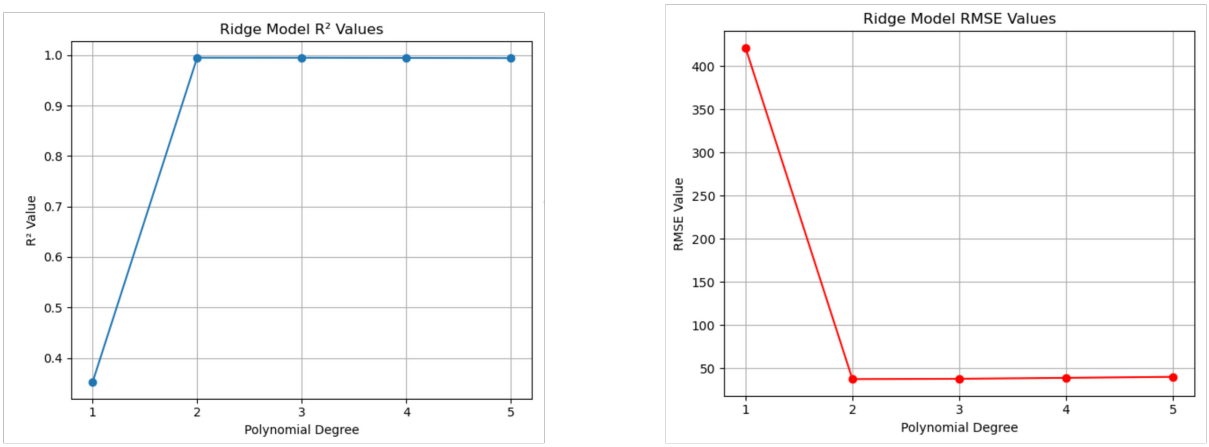
3.2.9 Multivariate Regression Model (Category & Station - Interaction)

Similar to the previous attempt, we further add some interaction terms among these variables. After considering the type of the variables (nsb → Quantitative, category → Qualitative, station → Qualitative), we add the following interaction terms, in which we use n to represent nsb, c to represent category, and s to represent station:

- **Degree 1:** n, c, s, nc, ns, cs, ncs
- **Degree 2:** n^2, n^2s, n^2c, n^2cs
- **Degree 3:** n^3, n^3c, n^3s, n^3cs

3.2.10 Final Result of Regression Model

Here are our final results obtained by splitting the 2022 data into training and testing sets, with 80% used for training and 20% for testing.



(a) R Square (b) RMSE

Figure 11: Final Results for Regression Model

We can clearly see that our multivariate regression model achieved good results in fitting historical data. The final R^2 reached a peak of 0.9949 at polynomial degree = 2, and then decreased slightly. The final RMSE reached a minimum of 37.3675 at polynomial degree = 2, and then increased slightly.

Furthermore, we increase our training data and test data. We try to use all the data from 2022 as training data and the 2024 data set as testing data. The final results are:

We can get our finalize result: Best polynomial degree for training set in 2022: degree 5, R^2: 0.9960, RMSE: 31.3082 Best polynomial degree for test set in 2024: degree 2, R^2: 0.9718, RMSE: 64.5335

3.3 Deep learning model (Spatiotemporal prediction model)

Deep learning model: learning the relationship between light intensity and species distribution

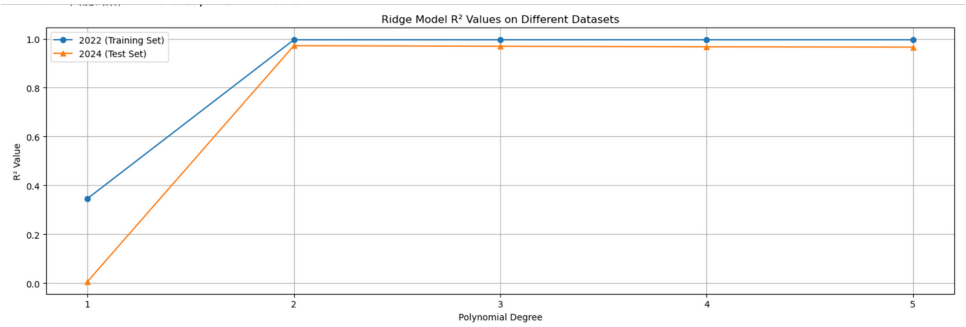


Figure 12: R Square of 2022 and 2024

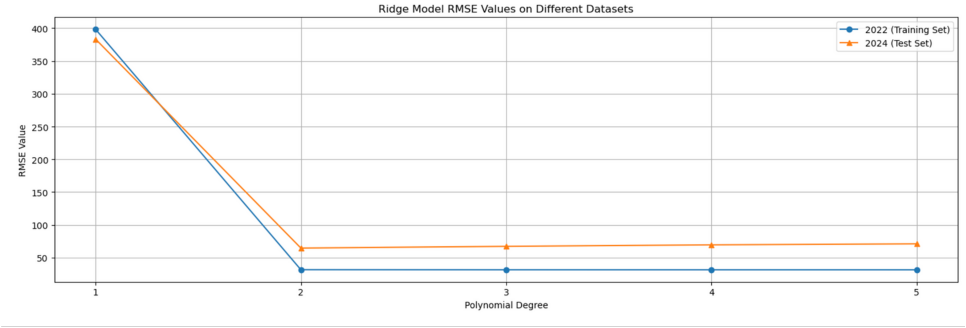


Figure 13: RMSE of the 2022 and 2024

To explore the potential relationship between night light intensity (NSB) and biological species distribution in Hong Kong, we constructed a spatio-temporal prediction model. The model incorporates spatial adjacency and time-series dependence, and is capable of modeling and predicting changes in the number of species in different regions at different times.

3.3.1 Data integration and missing value filling

The species data contain the latitude, longitude, month, species number (gno), and number of observations (count) for each observation, while the light intensity data record the night sky brightness values (NSB) for the same time and location. In order to achieve spatial matching between the two data sets, we rounded the latitude and longitude:

$$\text{lat}_{\text{round}} = \text{round}(\text{latitude}, 2), \quad \text{lon}_{\text{round}} = \text{round}(\text{longitude}, 2)$$

Left-link the species data with the light intensity data using month, lat_round, lon_round as keys to ensure that each species observation corresponds to the nearest light intensity value.

For the problem that some observations failed to match the NSB values, we used the K-nearest neighbor (KNN) algorithm to fill in the missing values.

3.3.2 Graph structure construction and node mapping

For subsequent graph neural network modeling, we consider each species observation point as a node in the graph and establish the following mapping relationships:

(1) Node ID Mapping

Each species number (gno) is mapped to a unique node ID to facilitate graph structure representation:

$$\text{node\_id}(gno_i) = i$$

(2) Unique Node Coordinate Extraction

Extract the latitude and longitude information of all unique nodes for constructing the vertex set of the graph:

$$\mathcal{V} = \{(lat_1, lon_1), (lat_2, lon_2), \ldots, (lat_N, lon_N)\}$$

3.3.3 Time series construction and normalization

For each node, time series were constructed for light intensity, latitude and longitude, and number of species, sorted by month.

$$S_i = \{(nsb_t, lat_t, lon_t, count_t) | t = 1, 2, ..., T_i\}$$

All feature quantities (including nsb and count) were subjected to histogram analysis and standardized using StandardScaler:

$$x' = \frac{x - \mu}{\sigma}$$

Where μ is the mean and σ is the standard deviation. The standardized data are used as input to the model to improve the stability of training and convergence speed.

3.3.4 Sliding Window and Temporal Sample Construction

To capture the dynamics in the time dimension, we use the sliding window method to extract samples from each time series. Setting the window size as W, the input-output pairs are constructed for any time step t:

$$X_t = [s_{t-W+1}, s_{t-W+2}, \ldots, s_t], \quad y_t = s_{t+1}$$

In this way, the model is able to learn the future trend of change from the history window.

3.3.5 Dynamic graph construction and graph attention mechanism

To model spatial correlation, we construct a dynamic graph structure based on the K Nearest Neighbor (KNN) method. For each set of node coordinates V, a neighbor matrix is constructed using kneighbors_graph and converted to a graph edge index E. This neighbor relationship is recalculated at each time step, which ensures that the graph structure changes dynamically over time. Subsequently, the graph structure is modeled using Graph Attention Network (GAT), which automatically learns the importance weights between nodes through the attention mechanism with the following core update rules:

$$h_i^{(l+1)} = \sum_{j \in \mathcal{N}_i} \alpha_{ij} W^{(l)} h_j^{(l)}$$

where α_{ij} is the attention coefficient between node i and neighbor j, $W^{(t)}$ is the learnable parameter matrix, and $\mathcal{N}_i$ is the set of neighbors of node i.

3.3.6 Model Architecture Design

In order to effectively capture the spatio-temporal correlation between light intensity and species distribution, we designed a hybrid spatial-temporal prediction model.

SpatialTemporalModel that incorporates the graph attention network (GAT) and the long-short-term memory network (LSTM).The model consists of the following four core components:

(1) Graph Attention Network (GAT)

Used to model the spatial dependencies between observation points. At each time step, the model extracts local spatial features based on the current node features and neighbor relationships by aggregating the neighbor information through the graph attention mechanism.

$$h_i^{(l+1)} = \sum_{j \in \mathcal{N}_i} \alpha_{ij} W^{(l)} h_j^{(l)}$$

where h is the embedding representation of node i in layer l, Ni denotes the set of its neighbors, and α_{ij} is the attention coefficient, which measures the importance of node j to node i.

The GAT layer receives the node feature matrix at each time step and outputs the updated node representation.

(2) Dropout layer

In order to prevent the model from overfitting, a Dropout layer is added after the GAT output, and some neuron connections are randomly discarded with a probability of 0.3.

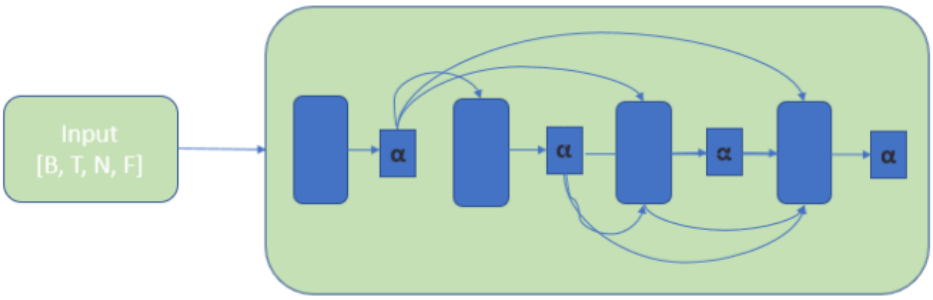
(3) Long Short-Term Memory (LSTM) network

Used to model long-term dependencies on time series. The GAT outputs from multiple time steps are spliced into a time series tensor X, which is input to the LSTM layer for sequence modeling.

The state update process of LSTM is as follows:

$$
\begin{aligned}
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
\tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\
C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
h_t &= o_t \odot \tanh(C_t)
\end{aligned}
$$

Finally, the hidden state h_t of the last time step is taken as the integrated representation of the whole sequence.



3.3.7 Innovation

(1) Integrating GAT and LSTM

The GAT module is responsible for extracting spatial dependencies between observation points and capturing geographic proximity and mutual influence through dynamically constructed graph structures. The LSTM module models time series and learns the evolutionary patterns of species numbers in the time dimension.

This combined method not only fully considers the spatio-temporal characteristics of data, but also enhances the modeling ability of the model for complex ecosystems, and has stronger expressiveness and generalization performance compared to single spatial or temporal models.

(2) Mechanism for constructing dynamic graph structures

Introduce a dynamic graph construction mechanism based on KNN. Unlike traditional fixed graph structures, this method independently constructs a graph structure at each time step, uses K-nearest neighbor algorithm to find the geographically closest neighbor for each node, and generates an edge_index based on it.

This model better adapts to real-world scenarios where spatial distribution changes over time, enhancing its sensitivity and adaptability to spatial topological changes.

3.3.8 Experimental results and analysis

(1) Model training and fitting effect

To verify the model's ability to learn the relationship between light intensity and species distribution, we used 80% of the historical data for training and the remaining 20% for testing. The mean square error (MSE) was used as the loss function during training:

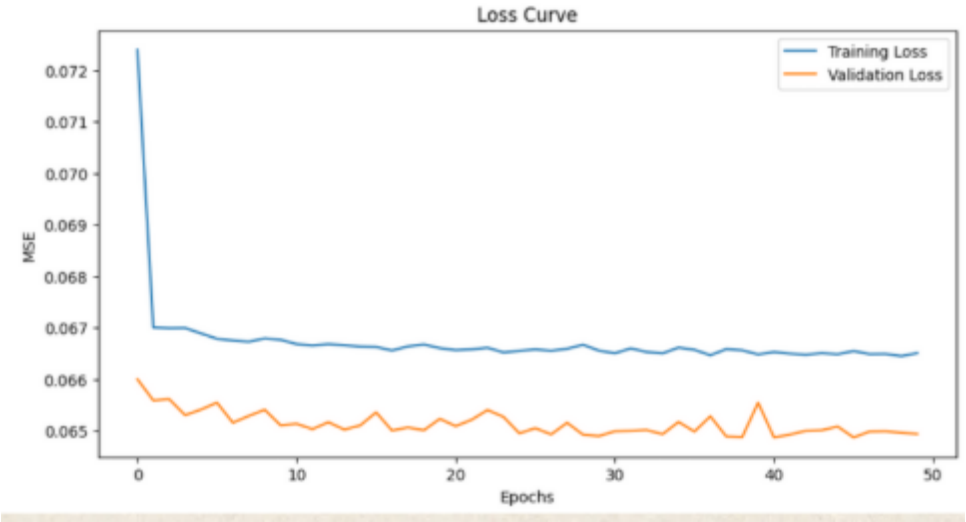
$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

The trends of Training Loss and Validation Loss of the model over 50 epochs are shown :

• Training loss : It decreases rapidly in the first 10 epochs, then stabilizes and finally converges to about 0.066.

• Validation loss : Again, it decreases rapidly in the early stage and remains relatively stable after the 10th epoch, eventually stabilizing at around 0.065.

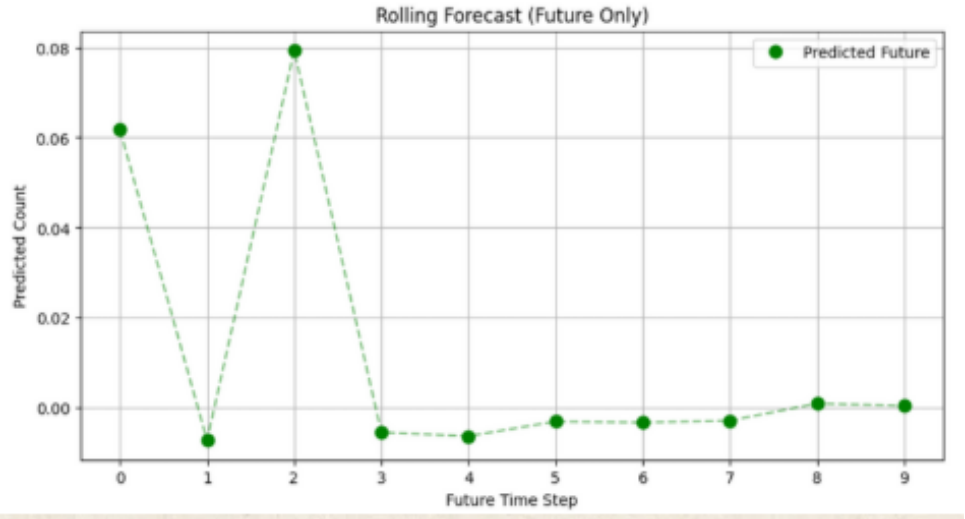
From the figure, we can see that the model basically reaches convergence at the 10th epoch, and the gap between the training loss and the validation loss is small, indicating that the model has good generalization ability and no overfitting phenomenon. Therefore, we believe that after 50 epochs of training, the model has an average loss of 0.066 on the training set and 0.065 on the validation set, both of which are at a low level. This indicates that the model is able to capture the spatio-temporal patterns in the historical data better and has some predictive ability.



(2) Testing and Rolling Forecast

To further evaluate the prediction performance of the model, we performed a Rolling Forecast on the test set using the trained model. by updating the input window step by step, predicting one future step at a time and adding the predicted values to the input sequence to generate a new prediction.

By showing the results of the model's Rolling Forecast for the next 10 steps, it can be seen that the model captures the trend of the number of species over time better. As the number of prediction steps increases, the predicted values gradually level off, indicating that the model shows good stability in long-term prediction.



3.3.9 Summarize

The deep learning model proposed in this project has achieved significant results in the learning task of the relationship between light intensity and species distribution. By fusing Graph Attention Network (GAT) and Long Short-Term Memory Network (LSTM), the model successfully captures complex patterns in the spatio-temporal dimension and shows good predictive ability in rolling prediction. The experimental results show that the model can not only accurately fit historical data, but also has strong generalization ability and prediction stability.

4 Failed Attempts

4.1 Failed topic 1: Predicting the unemployment rate

In this topic, we originally wanted to predict the unemployment rate of a specific group of people (for example, the unemployment rate of men aged 20-30 in the IT industry). However, since the government has already made predictions on unemployment rates for different age groups, genders, and industries, we wanted to find a more innovative project, so we gave up this topic.

These are the corresponding datasets:

- **Unemployed persons by previous industry, duration of unemployment and sex**
  https://data.gov.hk/sc-data/dataset/hk-censtatd-tablechart-210-06409
- **Unemployed persons by duration of unemployment, age and sex**
  https://data.gov.hk/sc-data/dataset/hk-censtatd-tablechart-210-06403
- **Median duration of unemployment by previous industry and sex**
  https://data.gov.hk/sc-data/dataset/hk-censtatd-tablechart-210-06410

- **Unemployed persons and unemployment rate by educational attainment and sex**
  https://data.gov.hk/sc-data/dataset/hk-censtatd-tablechart-210-06402
- **Unemployed persons and unemployment rate by age and sex**
  https://data.gov.hk/sc-data/dataset/hk-censtatd-tablechart-210-06401
- **Unemployed persons by previous industry, reason for leaving last job and sex**
  https://data.gov.hk/sc-data/dataset/hk-censtatd-tablechart-210-06408

4.2 Failed topic 2: Evaluating the effectiveness of different vaccines

This topic aims to evaluate the effects of different types of vaccines (inactivated vaccines and mRNA vaccines) on the incidence rate. However, we do not have data on whether individuals who have received different types of vaccines have been infected. This requires us to make predictions based on the ratio of the number of new cases to the number of people who have received different vaccines in the same period of time, and use some test methods (such as chi-square test) to calculate the reliability of such predictions. Therefore, we gave up this topic.

These are the corresponding datasets:

- **Data on COVID-19 (2019 Novel Coronavirus)** https://data.gov.hk/sc-data/dataset/hk-dh-chpsebcddr-novel-infectious-agent
- **Number of COVID-19 vaccine doses administered by age group** Shortened Link

5 Conclusion

Our project explored the impact of light pollution on species population changes through regression models and deep learning models. After introducing species categories and observation point locations, the prediction accuracy of the multivariate regression model was significantly improved ($R^2 = 0.99$). At the same time, the deep learning model combining the graph attention network (GAT) and the long short-term memory network (LSTM) successfully captured the spatiotemporal characteristics of the data, and the prediction results showed good stability and accuracy. Studies have shown that light pollution has a significant negative impact on biodiversity, and there are obvious differences in the impact on different species and regions. Our research not only reveals the ecological hazards of light pollution, but also provides a scientific analysis method and practical framework for ecological protection and research, emphasizing the importance of reducing light pollution.

6 References

Globe at Night. (n.d.). Global at Night Monitoring Network. http://globeatnight-network.org/global-at-night-monitoring-network.html

Unihedron. (n.d.). Sky Quality Meter - L (SQM-L). https://www.unihedron.com/projects/sqm-le/

Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. Neural Computation , 9 (8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P. (2018). Graph attention networks. International Conference on Learning Representations (ICLR) . https://openreview.net/forum?id=rJXMpikCZ