Journal of Computers Vol. 35 No. 4, August 2024, pp. 241-257
doi: 10.53106/199115992024083504017

Research on Rice Pest Forecasting Model Based on GCN-AGRU

Hang Ma[1], Jian-Chuang Wu[1], Xin-Tong Meng[1], Jing Feng[1],
Yan Wang[1], Yi-Qun Wang[1], Wen-Bai Chen[1*], and Xian-Shan Li[2]

[1] School of Automation, Beijing Information Science and Technology University,
100192, Beijing, China
{mh, wujc, 2023020368}@bistu.edu.cn, Fengj30@chinatelecom.cn
{2022010579, wangyiqun, wbchen}@bistu.edu.cn

[2] School of Information Science and Engineering, Yanshan University,
06004, Qinhuangdao, China
xjlxs@ysu.edu.cn

Received 19 July 2024; Revised 23 July 2024; Accepted 15 August 2024

Abstract. Rice, as one of the world's major staple crops, is highly susceptible to various factors such as extreme weather conditions, differences in cultivation types, and the diversity of varieties. Rice crops are increasingly threatened by various pests and diseases, particularly by "double-migration pest" (Brown planthopper and Rice leaf roller), which have shown a tendency for severe infestations. The current rice industry faces widespread issues of excessive pest control, leading to pesticide residue exceeding safe limits, causing environmental pollution in farmlands, and posing a threat to food security to some extent. This paper focuses on Brown planthopper and Rice leaf roller in Hunan Province, proposing a pest prediction method based on GCN-AGRU using multidimensional data collected from multiple pest monitoring stations in Hunan. This method considers the mutual influence of meteorological conditions and pest occurrences in various counties and cities, constructing a graph structure that reflects the spatial relationships between monitoring stations. By calculating the distance weights between stations, the model effectively identifies the spatial dependencies of pest occurrences. Additionally, GRU is introduced to enhance the ability to extract temporal sequence features, and an attention mechanism is employed to identify important features. Experiments demonstrate that the proposed GCN-AGRU prediction method achieves high accuracy and reliability in predicting pest trends over multiple days.

Keywords: rice pest, pest prediction, spatiotemporal features, graph convolution, gated recurrent unit

1 Introduction

The rice industry faces multifaceted challenges due to the combined effects of various factors such as planting systems, cultivation management, crop distribution, and climatic conditions, one of which is the high incidence of pests and diseases. Rice, as one of the world's major staple crops, is cultivated globally on approximately 180 million hectares, with China accounting for 18.5% of this area and contributing 31% of the global output [1]. Currently, the rice field area in China is around 29.92 million hectares, which represents 25% of the country's total arable land [2]. The production of rice is directly linked to food security and the livelihoods of farmers. However, rice crops are increasingly threatened by various pests and diseases due to extreme weather conditions, different rice cultivation methods, and complex varietal differences. Annually, pest-related diseases cause rice yield reductions ranging from 4 to 5 million tons, leading to significant attention on measures to control rice pests and diseases. Among these, the migratory pests of rice, such as Brown planthopper and Rice leaf roller, are particularly destructive during various growth stages of the crop. Moreover, the current practice of excessive pest control in the rice industry often leads to pesticide residues exceeding standard limits [3], thereby polluting the farmland environment and posing threats to food safety to some extent. This situation necessitates a more cautious consideration of the sustainability and environmental friendliness of pest control measures while addressing pest and disease issues.

Addressing the control of rice pest infestations requires attention to the forecasting of pest occurrence trends.

* Corresponding Author

Pest infestation is a highly complex system characterized by diversity, unpredictability, cyclical patterns, and predictable traits. Early research focused on analyzing the relationship between meteorological data and pest activity through statistical methods, utilizing weather parameters such as temperature, humidity, and rainfall to predict pest population dynamics. For instance, R. Mouly et al. [4] determined the impact of abiotic variables on leafhopper populations in organic mango orchards and developed an integrated weather forecast model, demonstrating the significant influence of maximum temperature and relative humidity on pest populations. Bao et al. [5] introduced new methodologies into the short-term forecasting of rice leaf folders using stepwise regression methods and the Kalman filter algorithm, further optimizing the timeliness and accuracy of the prediction models. Yan et al. [6] compared the performance of multivariate regression and artificial neural networks (ANN) in predicting pests such as melon thrips and diamondback moth, finding that ANN showed clear advantages in prediction accuracy. Furthermore, Jayanthi et al. [7] used ANNs and the Quasi-Newton (QN) algorithm to predict the catch of oriental fruit flies, showcasing the high accuracy of ANNs in estimating catch amounts. Narava et al. [8] combined autoregressive integrated moving average (ARIMA) and ANN methods, demonstrating that feedforward neural networks of ANN are best suited for effective pest prediction. Sharma [9] proposed a fuzzy inference system combined with multi-objective evolutionary algorithms and ANNs, offering a new perspective for predicting the timing of crop planting and pest occurrence rates. Skawsang et al. [10] used ANNs, random forests, and multivariate linear regression analysis methods, integrating ground meteorological variables and satellite-derived host plant variables to provide accurate short-term forecasts for brown planthopper populations in central Thailand. In deepening the application of deep learning technologies, Xiao et al. [11] used long short-term memory networks (LSTM) to address cotton pest occurrence issues, proving the advantages of LSTM in predicting pests such as cotton bollworm compared to other machine learning methods. Chen et al. [12] proposed a model based on bidirectional long short-term memory networks (Bi-LSTM), further revealing the strong potential of deep learning technologies in capturing the temporal relationships between pest occurrences and climate features.

This study focuses on significant pests in rice—Brown planthopper and Rice leaf roller—as the subjects of research, constructing a spatiotemporal prediction model for rice pest infestations based on GCN-AGRU. The objective of this model is to provide more targeted information to assist agricultural decision-making, thereby enhancing rice yields, reducing economic losses, and supporting the intelligent and sustainable development of agricultural production.

2 Related Works

2.1 Graph Convolution Network (GCN)

Graph neural network (GNN) is a class of deep learning models specifically designed to process graph-structured data. Their goal is to learn effective representations of nodes within the graph or of the entire graph itself, while preserving the structural information of the graph. These representations capture the characteristics of the nodes and their interactions, thereby revealing the intrinsic structural properties of the graph. GNN achieve this by iteratively propagating and aggregating information among the nodes of the graph, where each node updates its representation by integrating its own features with those of its neighbors. This dynamic aggregation process enables GNN to capture both local and global structural features within the graph, thus providing deep insights for various graph analysis tasks such as node classification, graph classification, and link prediction. Graph Neural Networks can be defined as follows:

$$h_v^{t+1} = f(x_v, x_c o[v], h_n^t e[v], x_n e[v]). \tag{1}$$

The function $f(\bullet)$ represents the updating of a node's hidden state, x_v denotes the features of node v, $x_c o[v]$ represents the features of the edge connecting nodes v, $h_n^t e[v]$ denotes the features obtained at the $t{-}th$ layer, and $x_n e[v]$ represents the features of the nodes connected to node v. The core of GNN lies in optimizing the function $f(\bullet)$ through deep learning methods to establish connections between network layers and capture node features.

GCN [13] represents an important implementation of graph neural networks, extending the convolution operation from Convolutional Neural Networks (CNN) to graph data, adapting to the complexities of non-Euclidean

spaces. Unlike CNN, which perform convolutions on regular grid structures to capture local spatial features, GCN implement convolutions on the nodes of graphs, taking into account the relationships between nodes and their diverse neighbors. This shift enables GCN to directly identify and extract significant patterns within graph structures, making them suitable for various node-level and graph-level prediction tasks. The graph convolutional framework is illustrated in Fig. 1.

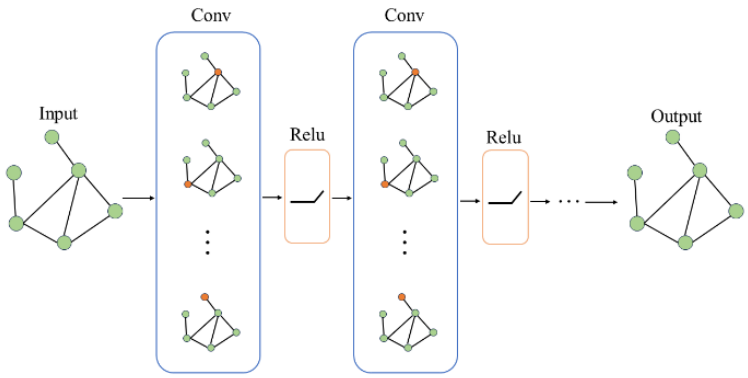

Fig. 1. Diagram of the GCN architecture

The core of GCN based on spectral methods lies in transforming the operations of GCN into linear computations within the spectral domain of the graph Laplacian matrix L, utilizing the spectral structure of the graph to capture complex relationships between nodes [14]. Initially, define an undirected graph $G = (V, E, A)$ with its adjacency matrix A and $D \in \mathbb{R}^{N\times N}$ as the corresponding degree matrix. The Laplacian matrix L is defined as $L = D - A$. Due to its symmetry, the Laplacian matrix can be eigen-decomposed as $L = U \wedge U^T$, where U is an orthogonal matrix composed of the eigenvectors of L, serving as the graph's Fourier basis, and $\Lambda = diag\left(\left[\lambda_0,...,\lambda_{N-1}\right]\right) \in \mathbb{R}^{N\times N}$ is a diagonal matrix with the eigenvalues of L on its diagonal. A signal $x \in R^N$ on the graph U can undergo a Fourier transform $\hat{x} \in U^T x$, and the inverse Fourier transform is realized through $x = U\hat{x}$, where the matrix involved in the inverse Fourier transform process can be represented as:

$$\begin{aligned} x = U\hat{x} &= \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \left[\vec{V}_1 ... \vec{V}_n\right] \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \\ &= \hat{x}_1\vec{V}_1 + x_2\vec{V}_2 + \ldots + x_n\vec{V}_n \\ &= \sum_{k=1}^{N} \hat{x}_k\vec{V}_k \end{aligned} \tag{2}$$

In this context, $\vec{V}$ represents the column vectors that constitute the Fourier basis matrix, and x_k is the $k{-}th$ row vector of the signal matrix on the graph. This notation is designed to simplify the inverse Fourier transform process, thus making convolution operations in the frequency domain more direct and straightforward. The graph filter g_θ is defined as a function related to the eigenvalues of the Laplacian matrix, enabling the creation of diverse filters to regulate the influence of different spectral components on the ultimate signal. The graph convolution operation can be defined as:

$$y = g_\theta(L)x = Ug_\theta(\Lambda)U^T x. \tag{3}$$

The graph signal x is processed in the spectral domain through the filter g_θ and then transformed back to the spatial domain. This processing method allows the GCN model to filter the signal, enhancing or attenuating specific frequency components in the spectrum. However, directly calculating the eigen-decomposition of the graph's Laplacian matrix L incurs a very high computational cost. In order to address this problem, Defferrard et al. [15] proposed a spectral-domain GCN named ChebyNet, which uses Chebyshev polynomials to approximate the eigen-decomposition of the graph Laplacian matrix L, reducing complexity. ChebyNet defines its convolution kernels through a truncation at the K-step of Chebyshev polynomials, and its expression is as follows:

$$g_\theta(L)x \approx \sum_{k=0}^{K-1} \theta_k T_k(\tilde{L})x. \tag{4}$$

$$\tilde{L} = \frac{2L}{\lambda_{\max}} - I. \tag{5}$$

$$T_k(\tilde{L}) = 2\tilde{L}T_{k-1}(\tilde{L}) - T_{k-2}(\tilde{L}). \tag{6}$$

In this context, $\tilde{L}$ represents the normalized Laplacian matrix, I is the identity matrix, and $\lambda_{\max}$ is the maximum eigenvalue of L. $T_k(\tilde{L})$ denotes the k–th order Chebyshev polynomial, starting from $T_0 = I$ and $T_1 = \tilde{L}$, which can recursively compute polynomials of any order. Utilizing this polynomial approximation, the graph convolution operation of ChebyNet can be rewritten as:

$$g_\theta(L)x = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{L})x. \tag{7}$$

The spatial approach begins with the spatial structure of the graph and defines graph convolution by aggregating the neighborhood features of nodes. This method operates directly on the nodes of the graph. For node i, its feature vector h_i is updated after the convolution operation to h_i'. The calculation process can be expressed by the following formula:

$$h_i' = \sigma(W \bullet AGGREGATE(\{h_j : j \in \mathrm{N}(i)\}) + b). \tag{8}$$

In this formula, W is the learned weight matrix, b is the bias term, $\sigma(\bullet)$ is a nonlinear activation function, and $AGGREGATE(\bullet)$ is an aggregation function that collects the feature vectors h_j of the neighboring nodes j of node i. The function $\mathrm{N}(i)$ represents the set of neighboring nodes of node i. This process can be used to capture the local topological structure of the node. By stacking multiple such convolutional layers, deeper structural information of the graph can be learned.

2.2 Gated Recurrent Unit (GRU)

The Gated Recurrent Unit (GRU) is an effective improvement on the Recurrent Neural Network (RNN), which is specifically designed for processing sequential data. A distinctive characteristic of RNN is it recurrent connections within the network, which enable them to preserve and leverage information from previous time steps as they process the current input. However, RNN often face issues with gradient vanishing or exploding when dealing with long sequences, making it challenging for the model to learn long-term dependencies within the sequence. The instability of gradients occurs because, during backpropagation, the gradients are continuously multiplied by the weight matrices between time steps, leading to drastic increases or decreases in the gradient values.

Moreover, the recursive nature of RNN requires each time step's computation to depend on the output of the previous time step, limiting the model's parallel processing capabilities and making the training of large-scale RNN models time-consuming and resource-intensive. To overcome these drawbacks, the Long Short-Term Memory (LSTM) network [16] was proposed. LSTM introduces three gating mechanisms—forget gate, input gate, and output gate—that effectively manage the storage, updating, and retrieval of information. These mech-

anisms enable LSTM to capture long-term dependencies, significantly enhancing the model's performance and stability in handling sequential data.

GRU simplifies the model architecture based on the LSTM network by optimizing the three gating mechanisms in LSTM into two—namely, the update gate and the reset gate. The update gate assists the model in determining how much of the previous information to retain in the current state, while the reset gate decides how much of the previous state information to forget. This design retains LSTM's ability to handle long-term dependencies while simplifying the network structure and reducing computational complexity. The simplified gating mechanism in GRU ensures that the model can efficiently process sequential data without the overhead associated with LSTM's more intricate structure.

By having fewer gates, GRU reduces the number of parameters and operations needed per time step, leading to faster training and inference times. This reduction in complexity makes GRU particularly advantageous when working with large datasets or in resource-constrained environments where computational efficiency is crucial. Despite the simplification, GRU has been shown to perform comparably to LSTM on a variety of tasks, often achieving similar levels of accuracy and robustness in capturing temporal dependencies in sequential data.

The architecture of a GRU, as depicted in Fig. 2, illustrates its streamlined structure. The update gate (z) and reset gate (r) work together to control the flow of information within the unit. The update gate controls the amount of the previous hidden state that is transferred to the current time step. Conversely, the reset gate adjusts how much of the past information should be discarded, thus enabling the unit to reset its memory as necessary.

This efficient mechanism allows GRU to capture complex temporal patterns without the extensive computational overhead typically associated with LSTM. Consequently, GRU is extensively employed in diverse applications such as natural language processing, speech recognition, and time-series forecasting, where they provide robust performance and enhanced efficiency.

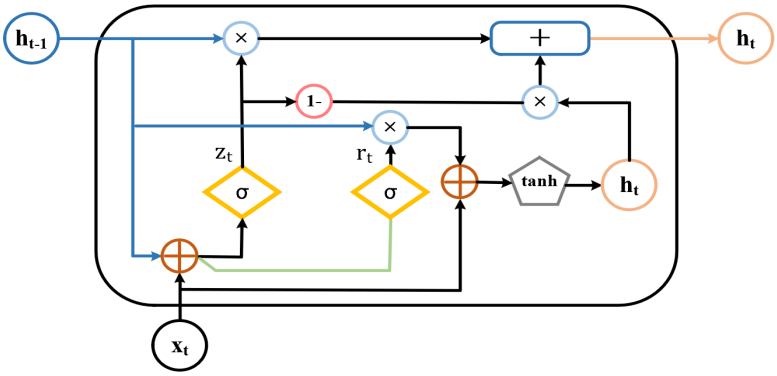

Fig. 2. GRU network architecture diagram

The internal formula for GRU is as follows:

$$z_t = \sigma(W_z \bullet [h_{t-1}, x_t] + b_z). \tag{9}$$

$$r_t = \sigma(W_r \bullet [h_{t-1}, x_t] + b_r). \tag{10}$$

$$\widetilde{h}_t = \tanh(W_h \bullet [r_t * h_{t-1}, x_t] + b_h). \tag{11}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h}_t). \tag{12}$$

Where z_t, r_t, $\widetilde{h}_t$ and h_t represent the update gate, reset gate, candidate hidden state, and final hidden state, respectively. W_z, W_r and W_h are the weight matrices, while b_z, b_r and b_h are the corresponding bias vectors.

2.3 Attention

The attention mechanism, originating from the field of natural language processing, primarily functions by computing attention scores to selectively focus on key parts of the information [17]. These scores represent the importance of the information at each time point in the sequence, enabling the model to prioritize the most crucial information for the prediction task. The calculation of the attention mechanism involves the following steps: Initially, the output feature vectors from the GRU are transformed through a fully connected network. This transformation employs the tanh activation function to map the vectors into a specific range, preparing the data for the computation of attention scores. This step essentially adjusts the feature representations to make them suitable for the subsequent attention calculation. Next, another fully connected network, followed by a Softmax function, is used to compute the attention scores for each time point. These scores reflect the relative importance of each time point with respect to the prediction task. The Softmax function ensures that the attention scores sum to one, effectively normalizing them into a probability distribution. Finally, the original GRU output features are weighted by the computed attention scores. This results in an attention-weighted feature vector that highlights the critical temporal features in the sequence. By emphasizing these key points, the model can focus on the most relevant information, improving its performance on the prediction task. The calculation of the attention mechanism can be represented as:

$$M = \tanh(W_1 H_{out}). \tag{13}$$

$$\alpha = \mathrm{softmax}(W_2^{\mathrm{T}} M + b_2). \tag{14}$$

$$\beta = H \otimes \alpha^{\mathrm{T}}. \tag{15}$$

where $H_{out} = (h_1, h_2, ..., h_l)$ is the output feature vector, W_1 and W_2 are the weight matrices of the fully connected layers, and b_2 is the bias term.

2.4 GCN-AGRU Prediction Model for Rice Insect Pests

A graph-based approach is employed to predict pest occurrences between monitoring points while also considering their meteorological factors, resulting in the creation of a predictive model that combines GCN and AGRU. This model aims to delve deeply into the spatial correlations and temporal patterns of pest occurrences in rice fields. The input to the model includes data collected from pest monitoring points, which possess both spatial and temporal dimensions. In the spatial dimension, the input data include the geographical location information of the monitoring stations, represented through a constructed weighted graph, where each monitoring station serves as a node. The edges between nodes represent the spatial correlations between stations, and this associative strength is calculated based on geographical distances. In the temporal dimension, the input data comprise historical meteorological data and pest occurrence data at various time points, organized into feature matrices that are updated over time, providing dynamic inputs for the model.

Spectral methods are used for the extraction of spatial features through GCN analysis. This approach leverages the graph Laplacian matrix and the framework of Fourier transforms to extract spatial features from the interconnections between monitoring stations. Spectral methods are stable and, based on the global properties of the graph, can effectively identify spatial patterns within the monitoring network. These patterns include not only local connectivity but also the global layout of the monitoring stations, making spectral methods an effective means to handle irregular graph structures. Moreover, spectral methods provide a more natural and mathematical approach to expressing and processing graph convolutions, enabling the use of existing deep learning architectures and optimization techniques. Compared to spatial methods, spectral methods do not require the explicit design of filters to handle different spatial configurations; instead, filters are directly defined by the inherent spectral properties of the graph, simplifying the model's design and training process.

Following spatial feature analysis, the model progresses to the extraction of temporal features, a process managed by the GRU network. The time series data from each monitoring point are input into the GRU layer, where GRU, through its unique gating mechanism, selectively remembers or forgets past information, providing a feature representation at each time point that integrates the impact of historical data. In the attention mechanism part of the model, a self-attention mechanism is employed to further enhance the model's performance. It enhances

the model's ability to capture important information by learning different subspace representations of the input data in parallel. Fig. 3 illustrates the overall structure of the model.

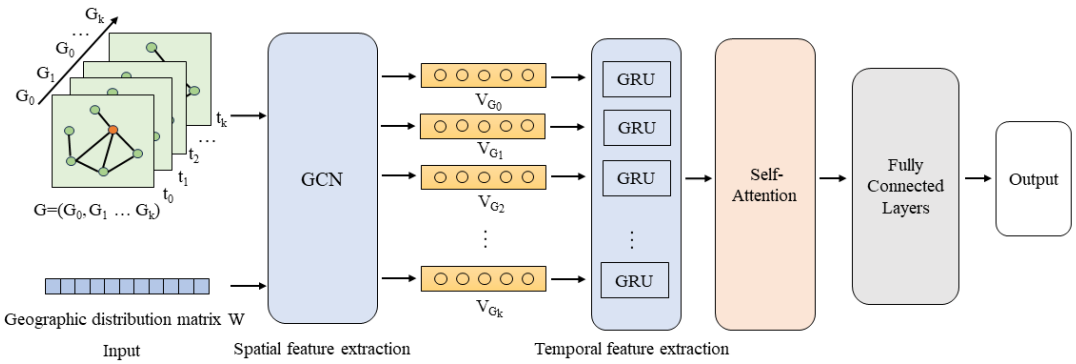

Fig. 3. GCN-AGRU model architecture diagram

2.5 Selection of Evaluation Indicators

To comprehensively evaluate the performance of the GCN-AGRU model in time-series prediction of rice pest infestations, multiple evaluation metrics were considered. These metrics provide a multidimensional view of the model's performance, assessing not only the accuracy of predictions but also the magnitude of errors and the correlation of the predicted results. The main metrics used include Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and the Coefficient of Determination (R^2) [18]. RMSE and MAE primarily reflect the average magnitude of errors between the predicted values and the actual values, providing a benchmark for intuitively assessing the model's performance in quantifying the precision of pest occurrence time-series predictions. The R^2 coefficient measures the consistency between model predictions and actual values, evaluating the model's accuracy in capturing the relationships between variables. MAPE offers a percentage evaluation of prediction accuracy, suitable for assessing performance in time-series forecasting problems. The specific formulas for these metrics are as follows:

$$\mathrm{RMSE}=\sqrt{\frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i-y_i)^2}. \tag{16}$$

$$\mathrm{MAE}=\frac{1}{m}\sum_{i=1}^{m}\left|\hat{y}_i-y_i\right|. \tag{17}$$

$$\mathrm{MAPE}=\frac{1}{m}\sum_{i=1}^{m}\left(\frac{\left|\hat{y}_i-y_i\right|}{y_i}\right)\times 100\%. \tag{18}$$

$$\mathrm{R}^2=1-\frac{\sum_{i=1}^{m}\left(\hat{y}_i-y_i\right)^2}{\sum_{i=1}^{m}\left(y_i-\overline{y}\right)^2}. \tag{19}$$

Where m represents the number of samples, y_i denotes the actual value of the $i{-}th$ observation, and $\hat{y}_i$ represents the model's predicted value for the $i{-}th$ observation. $\overline{y}$ is the average of all observed values.

In classification tasks, the confusion matrix is a key tool for evaluating the performance of a model, providing a clear visual representation of the model's classification ability across different categories. It primarily consists of four parts: True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). True Positives and True Negatives refer to the counts of positive and negative instances that the model correctly classifies, respectively, while False Positives and False Negatives represent the counts of instances where the model incorrectly classifies positives as negatives and negatives as positives, respectively. Table 1 shows a structured representation of this concept.

Table 1. Confusion matrix

Confusion matrix	Positive forecast	Negative forecast
Actual positive	TP	FN
Actual negative	FP	TN

The confusion matrix can be analyzed using multiple evaluation metrics to describe the performance of the model from different perspectives. Weighted accuracy takes into account the proportion of samples in each category, assigning different weights to correct predictions for different categories. This approach can address the bias issue of traditional accuracy on unbalanced datasets. The formula for calculation is as follows:

$$Weighted\ \ Accuracy = \sum_{i=1}^{n} w_i \times \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}. \tag{20}$$

In the formula, n represents the total number of categories, and w_i is the weight for the i category.

The $F1$ *Score*, as the harmonic mean of precision and recall, considers both the accuracy and sensitivity of the model. It effectively balances the impact of false positives and false negatives on unbalanced datasets. For multi-class classification problems, calculating the $F1$ *Score* for each category and taking its weighted average allows for a comprehensive assessment of the model's overall performance. The weighted $F1$ *Score* allocates weights according to the sample size of each category, ensuring that minority classes are adequately represented in model evaluations. The formula for calculation is as follows:

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{21}$$

$$Weighted\ \ F1\ Score = \sum_{i=1}^{n} w_i \times F1_i. \tag{22}$$

In the formula, $F1_i$ is the $F1$ *Score* for the *i−th* category. Precision is the proportion of true positive predictions in all positive predictions made by the model, and recall is the proportion of true positives detected by the model out of all actual positives. The formulas for calculating these metrics are as follows:

$$Precision = \frac{TP}{TP + FP}. \tag{23}$$

$$Recall = \frac{TP}{TP + FN}. \tag{24}$$

3 Study Area and Data Collection

Hunan Province has been selected as the study area, located between 24°38'N to 30°08'N latitude and 108°47'E to 114°15'E longitude, with a total area of about 211,800 square kilometers. It hosts major water systems, in-

cluding Dongting Lake, providing abundant water resources that create favorable conditions for agricultural production. Hunan features a subtropical humid monsoon climate with distinct seasons, plentiful rainfall, and ample sunlight. The average annual temperature ranges from 16°C to 18°C, with annual precipitation between 1,200 mm and 1,700 mm, offering excellent natural conditions for the growth of crops such as rice. Hunan is one of China's important bases for the production of grain and economic crops, consistently leading the nation in rice cultivation area, with double-cropping rice areas comprising a quarter of the national total. The province's agricultural production also faces threats from various pests, particularly migratory pests, which significantly impact the yield and quality of rice. In light of this, the study has selected 19 counties and cities in Hunan where pest infestations are particularly severe as key research subjects, as shown in Fig. 4. These areas provide an ideal environment for constructing a deep learning-based spatiotemporal predictive model for rice pests.

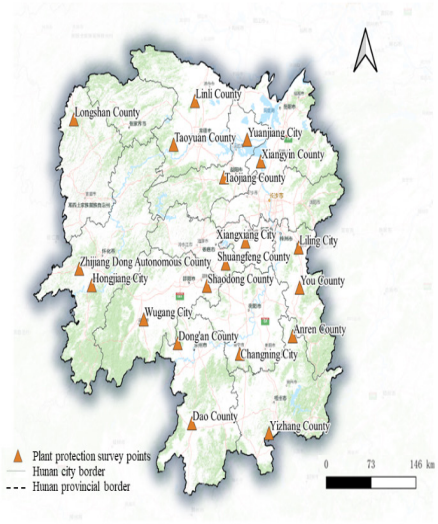

Fig. 4. Geographic distribution of the study area

Pest monitoring data were sourced from the Hunan Plant Protection and Plant Inspection Information Network, selecting daily monitored pest counts of rice leaf folders and planthoppers during the rice growing season from multiple counties and cities in Hunan Province from 2010 to 2023 as research subjects. Meteorological data were obtained from the "ERA5-Land Daily Aggregated-ECMWF Climate Reanalysis" dataset, which aggregates the original hourly data of ERA5-Land into daily summaries. This dataset is provided jointly by Google and the Copernicus Climate Change Service Centre and accessed via the Google Earth Engine (GEE) platform. Before proceeding with data analysis, daily Relative Humidity (RH) calculations were performed on the ERA5-Land data. Relative humidity is calculated based on daily recorded 2-meter air temperature and 2-meter dew point temperature, as described in formulas (18) and (19) [19]:

$$RH = \frac{e(T_d)}{e(T)}. \tag{25}$$

$$e(t) = 6.112 \exp\left(\frac{17.67t}{t + 243.5}\right). \tag{26}$$

Where T_d represents the dew point temperature at 2 meters height, and T represents the air temperature at 2 meters height.

4 Experiments and Analysis

4.1 Data Preprocessing and Model Parameterization

In the data preprocessing section, this chapter selects meteorological and pest data from the rice growing period of May to September from 2010 to 2023 in Shaodong County, Hunan Province, as the dataset. The data on two major rice pests—Rice leaf roller and Brown planthopper—are used as the targets for prediction. The collected data undergo standardization and quality filtering to ensure consistency in format before being input into the model. Initially, pest protection data are screened for missing and anomalous values. Detected anomalies are removed to clear any potential errors or inconsistencies in the data. Subsequently, missing values in the dataset are filled using linear interpolation. The processed data are then combined with meteorological data to construct a comprehensive dataset for predicting rice pest infestations. Afterwards, a graph structure is constructed to represent spatial connections between monitoring stations, using the longitude and latitude information of the stations to calculate the distance weights. These calculated distance weights are integrated into the graph structure, connecting monitoring stations with edges to form the foundational framework of the graph convolutional network. After establishing the graph structure and calculating weights, the input features of the model are standardized.

This article employs the haversine formula to calculate the shortest great-circle distance between two points. This distance not only represents the spatial relationship in geographical terms but also reflects the ecological and meteorological connections that may influence the spread of pestilence. The haversine formula is defined as follows:

$$d_{ij} = 2r \arcsin\left(\sqrt{\sin^2(\frac{\Delta\varphi_{ij}}{2}) + \cos(\varphi_i)\cos(\varphi_j)\sin^2(\frac{\Delta\lambda_{ij}}{2})}\right). \tag{27}$$

In the formula, d_{ij} represents the haversine distance between nodes i and j, where φ_i and φ_j denote the latitudes, and λ_i and λ_i denote the longitudes of the two points, respectively. r represents the radius of the Earth, while $\Delta\varphi_{ij}$ and $\Delta\lambda_{ij}$ represent the differences in latitude and longitude between the two points, respectively. The calculated distance weights are integrated into the graph structure, connecting monitoring stations as edges to form the foundational framework of the graph convolutional network. These edge weights directly influence the intensity of information propagation in the graph convolution operations and are crucial for the model to capture spatial dependencies.

To ensure that the model can effectively learn and predict the temporal variations in rice pest infestations, data preprocessing steps are taken. To address the issue of inconsistent dimensions and significant differences in numerical ranges among different data features, this study employs two common methods of data normalization: Min-max normalization and Z-score normalization. Min-max normalization linearly transforms the original data to fit within the [0,1] interval. This method effectively adjusts the scale of the data while preserving the relative relationships between the data points. The formula for Min-max normalization is as follows:

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}}. \tag{28}$$

In the formula, $x_{\min}$ and $x_{\max}$ represent the minimum and maximum values in the dataset, respectively.

Z-score normalization involves standardizing the data by calculating the mean and standard deviation. This process transforms the data such that the resulting distribution has a mean of 0 and a variance of 1, conforming to a standard normal distribution. The formula for Z-score normalization is as follows:

$$x^* = \frac{x - \mu}{\sigma}. \tag{29}$$

In the formula, μ represents the mean of the data, and σ represents the standard deviation of the data.

To determine which data normalization method is more suitable for the rice pest prediction model in this study, two methods—Min-max normalization and Z-score normalization—are used to preprocess the data. The processed data are then used to train deep learning models of the same structure to compare the differences in model training effects between these two normalization methods. Special attention is given to analyzing their loss function value decline curves during the model training process. Specific experimental results are shown in Fig. 5.

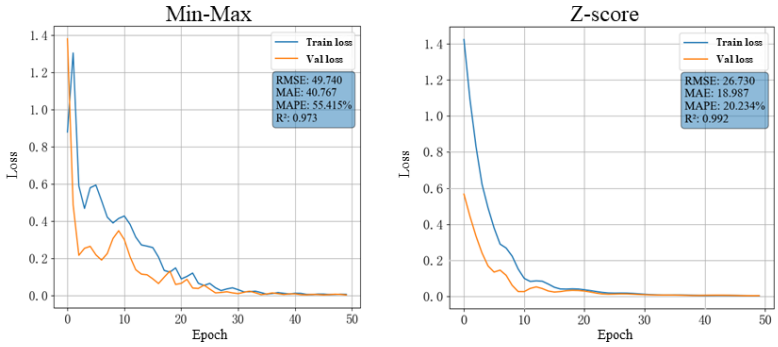

Fig. 5. Loss curve

The results indicate that data processed with Z-score normalization exhibit a smoother loss decline curve and faster convergence. Therefore, Z-score normalization is selected as the data preprocessing method. The dataset is divided into three parts: training set (60%), test set (20%), and validation set (20%).

This experiment was conducted on a system equipped with 16GB of RAM, an NVIDIA GeForce RTX 3060 Laptop GPU, and a 12th Gen Intel(R) Core(TM) i7-12700H CPU @ 2.30GHz processor. The GCN-AGRU network model was built using Python 3.10 and the PyTorch deep learning framework. The settings for the model parameters are presented in Table 2.

Table 2. Model Parameter Setting

Name	Parameter	Name	Parameter
GCN layers	2	GCN activation	ReLU
Number of GCN hidden units	64	GRU activation	Tanh
GRU layers	2	Optimizer	Adam
Number of GRU units	128	Learning rate	0.005
Chebyshev polynomials of order K	5	Training epoch	200
Dropout ratio	0.05	Batch size	32

4.2 FCN-AGRU Model Validation and Result Analysis

To evaluate the performance of the proposed method, meteorological and pest data from Shaodong County, Hunan Province, during the rice growing stages from May to September of 2010 to 2023 were initially selected as the dataset, focusing on two major rice pests—rice leaf folders and brown planthoppers—as prediction targets. The effectiveness and accuracy of various models in predicting pest time series were assessed by comparing the performance of statistical models, machine learning models, and commonly used deep learning models. Data spanning fourteen days were used as input for the models to predict pest occurrences for the following day. All models underwent fifty rounds of training to ensure thorough learning, and Bayesian optimization was employed to fine-tune all parameters to their optimal combinations, allowing for a comprehensive evaluation and comparison of each model's performance.

As can be seen from Table 3, compared to deep learning models, traditional statistical models such as ARIMA and SARIMA exhibit weaker performance in forecasting, especially in predicting planthopper infestations. The

higher MAPE values indicate that these models have limitations in capturing complex nonlinear relationships within the data. Overall, deep learning models demonstrate superior performance over traditional statistical models in this study, with the FCN-AGRU model achieving the best results in predicting rice pest infestations. A comparison of the prediction results of each model with actual pest occurrences is shown in Fig. 6. From May to September, the relationship between the actual records of rice leaf folder and brown planthopper occurrences and the predicted values from each model reveals that the GCN-AGRU model's prediction curve aligns most closely with the actual data points, particularly at the peaks and troughs.

Table 3. Comparison results of rice pest prediction models

Model	Rice leaf roller				Brown planthopper			
	MAPE	RMSE	MAE	R^2	MAPE	RMSE	MAE	R^2
SVM	42.181%	107.309	36.924	0.871	112.728%	132.135	50.292	0.226
ARIMA	39.026%	113.956	37.691	0.855	104.574%	123.207	39.863	0.325
SARIMA	22.156%	75.401	16.519	0.872	45.138%	81.566	17.678	0.355
LSTM	14.057%	138.274	37.277	0.958	13.912%	33.602	19.160	0.987
GRU	11.335%	112.308	29.668	0.973	14.644%	25.457	18.010	0.993
FCN-GRU	17.430%	118.512	38.176	0.970	16.257%	25.571	18.467	0.992
FCN-AGRU	22.932%	28.151	19.308	0.991	12.710%	18.097	12.408	0.996

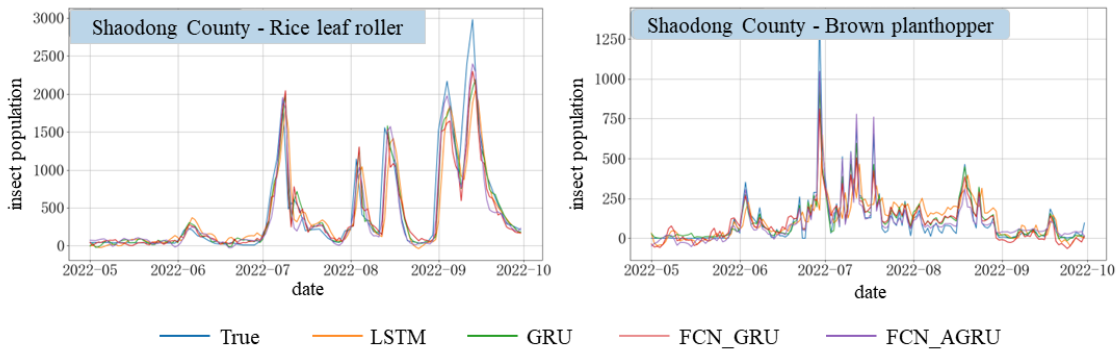

Fig. 6. Comparison of model predictions with actual

The most effective model is applied to forecasts over a longer time span, specifically using fourteen days of historical data as input to predict pest occurrences for the next three and six days. When predicting for three and six days, the model is similarly trained for only 50 epochs, with parameters adjusted to their optimal combination.

Table 4. Model comparison results of FCN-AGRU in different days

Days	Rice leaf roller				Brown planthopper			
	MAPE	RMSE	MAE	R^2	MAPE	RMSE	MAE	R^2
One day	22.932%	28.151	19.308	0.991	12.710%	18.097	12.408	0.996
Three days	353.458%	249.103	101.555	0.866	249.971%	245.448	144.589	0.307
Six days	297.591%	387.668	163.035	0.659	273.468%	204.998	126.903	0.294

The Table 4 shows that the FCN-AGRU model can make relatively accurate predictions for short-term forecasts, specifically for one-day predictions of pest occurrences. However, the performance of the model exhibits a significant decline when the forecasting period is extended to three and six days. Fig. 7 compares the quantity predictions of the FCN-AGRU model at different days with the actual quantities, further validating the results of the numerical analysis.

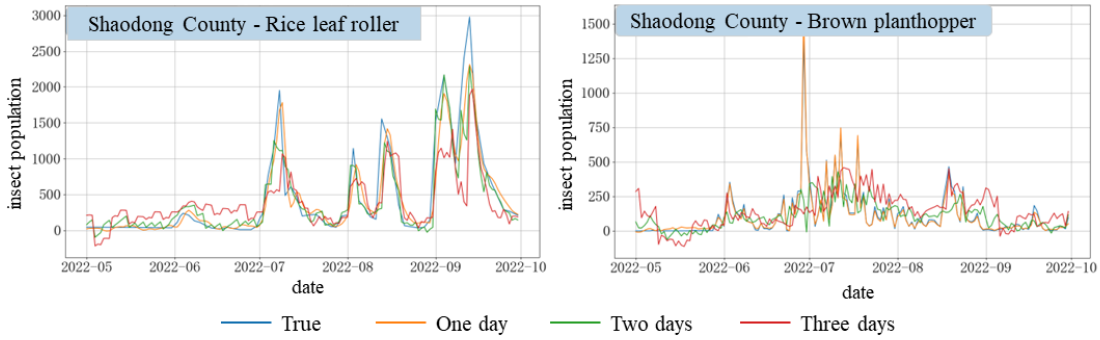

Fig. 7. Comparison between the predicted and actual number of FCN-AGRUs in different days

4.3 GCN-AGRU Model Validation and Result Analysis

To meet the demand for multi-day forecasts of rice pest infestations, the interaction between local meteorological conditions and the occurrence of pests is considered, and spatial information is introduced to compare the GCN-AGRU model with several other models. Firstly, for datasets with spatio-temporal characteristics, Yu et al. [20] proposed a Spatio-Temporal Graph Convolutional Network model (STGCN), which has achieved certain performance results in handling data with combined temporal and spatial features. Additionally, Wang et al. [21] proposed a further advanced network model, STGNN, which also shows performance improvements on datasets with spatio-temporal dependencies. Ablation experiments on agricultural pest datasets with spatio-temporal characteristics are conducted to verify the prediction effects and metric values under the same experimental conditions. This experiment uses fourteen days of data as input to predict pest occurrences over the next three days, with Shaodong County as the prediction output region for the model. All models are trained for 200 rounds, and all parameters are tuned to their optimal combinations through Bayesian optimization methods. The specific comparison results are shown in Table 5.

Table 5. Comparison results of forecasting models

Model	Rice leaf roller				Brown planthopper			
	MAPE	RMSE	MAE	R^2	MAPE	RMSE	MAE	R^2
GRU	297.591%	387.668	163.035	0.659	273.468%	204.998	126.903	0.294
STGCN	296.404%	102.533	60.301	0.857	264.768%	156.140	75.613	0.818
STGNN	247.355%	88.627	51.742	0.881	240.113%	133.628	64.159	0.845
GCN-GRU	171.432%	68.502	38.177	0.921	196.297%	95.075	62.417	0.889
GCN-AGRU	122.952%	38.051	23.338	0.984	142.710%	72.037	47.158	0.906

The Table 4 indicates that integrating spatial information from different regions into the model can effectively enhance its performance. On the four key performance indicators—MAPE, RMSE, MAE, and R^2—the GCN-AGRU model, which incorporates spatial information, performs best in predicting the occurrences of rice leaf folder and brown planthopper infestations over the next three days. Models based on GRU are more effective in processing time series data, but their performance is limited when spatial features are also important. Both STGCN and STGNN models show significant improvements over the GRU model on these four key performance indicators. Although both STGCN and STGNN models effectively handle spatio-temporal data, the GCN-AGRU provides the best performance on this dataset. A comparison of the prediction results of each model with actual pest occurrences is shown in Figure 8. From May to September, the relationship between the actual records of rice leaf folder and brown planthopper occurrences and the predicted values from each model reveals that the GCN-AGRU model's prediction curve aligns most closely with the actual data points, particularly at the peaks and troughs.

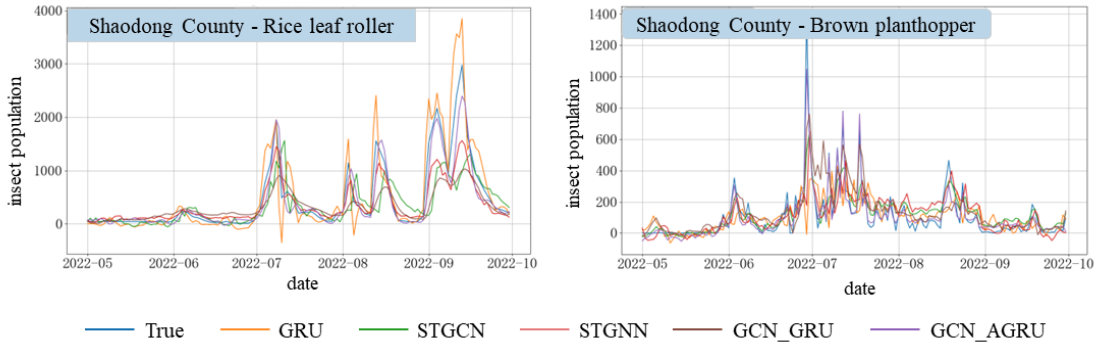

Fig. 8. Comparison of Model Predictions with Actual

To assess the model's generalizability across geographic distributions and its predictive effectiveness, a further analysis of the GCN-AGRU model's performance in predicting rice pest infestations in different counties and cities is conducted. Table 6 shows the comparison results of prediction accuracy across different counties and cities, indicating that the GCN-AGRU model has smaller prediction errors in most counties and cities. The model's predictions for rice leaf folder and brown planthopper are relatively accurate, with R^2 values generally exceeding 0.8, demonstrating that the model can explain most of the variability and has high predictive accuracy. Fig. 9 further displays the model's performance, where the predicted curves of the model are consistent with the actual pest occurrences over most of the time periods.

Table 6. Comparison results of forecast accuracy in different counties and cities

Area	Rice leaf roller				Brown planthopper			
	MAPE	RMSE	MAE	R^2	MAPE	RMSE	M AE	R^2
Hong Jiang City	55.776%	14.134	12.316	0.887	168.182%	76.277	58.127	0.817
Long Shan County	133.781%	59.906	36.847	0.931	61.396%	17.560	13.271	0.899
Shao Dong City	122.952%	38.051	23.338	0.984	142.710%	72.037	47.158	0.906
Ruan Jiang City	143.599%	43.268	24.579	0.902	60.104%	10.767	10.123	0.924

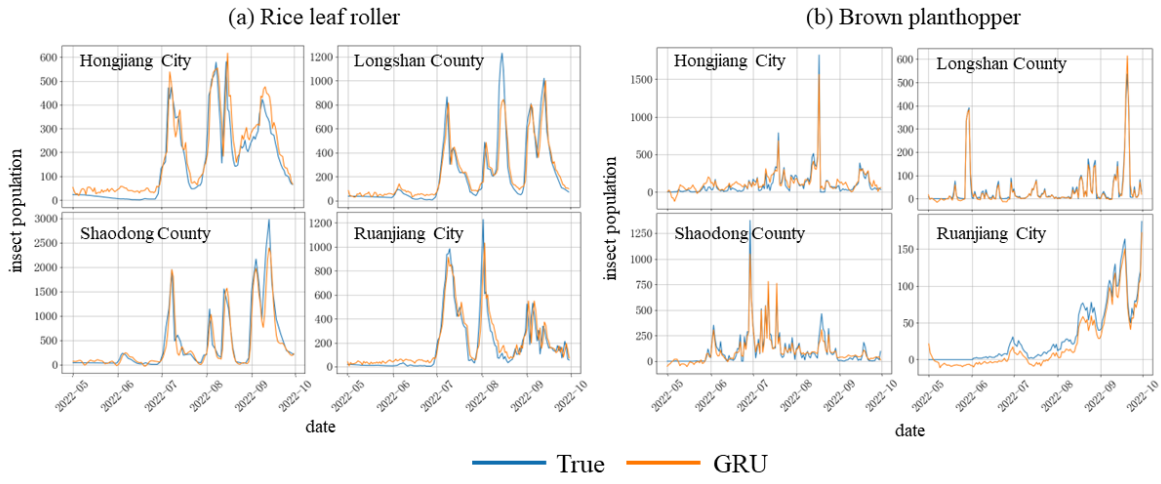

Fig. 9. Comparison of model GCN-AGRU's prediction results of the next three days in different counties and cities with the actual ones

To comprehensively assess the performance of the model, the GCN-AGRU model's predictive capabilities at different phenological stages in Shaodong County were evaluated. Table 7 provides the prediction results for the first, third, and sixth days, and Fig. 10 illustrates the comparison between the model's predictions at these stages and the actual occurrences. The table indicates that as the forecast duration increases, the prediction error (MAPE) gradually increases, and the ability to explain data variability (R^2) also decreases. This is expected, as the uncertainty of predictions generally increases over time. Despite this, the R^2 value remains above 0.764 even at the sixth day, indicating that the GCN-AGRU model maintains a relatively high accuracy over longer forecast durations. The curves in the figure show that although the fluctuation of the prediction curves increases at the third and sixth days, the model still captures the main trends of the actual curves. The introduction of spatial factors has significantly enhanced the model's performance. Although accuracy in multi-day predictions is affected, the GCN-AGRU model still maintains a low error range.

Table 7. Model comparison results of GCN-AGRU in different days

Days	Rice leaf roller				Brown planthopper			
	MAPE	RMSE	MAE	R^2	MAPE	RMSE	M AE	R^2
One day	32.632%	29.021	18.344	0.996	62.716%	58.387	32.288	0.991
Three days	122.952%	38.051	23.338	0.984	142.710%	72.037	47.158	0.906
Six days	254.723%	274.713	133.259	0.836	266.21%	143.103	81.098	0.764

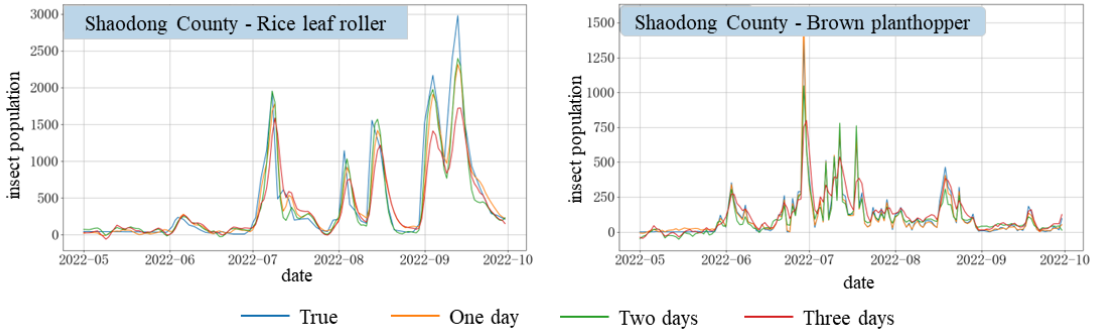

Fig. 10. Comparison of GCN-AGRU predicted results with actual numbers at one, three and six days

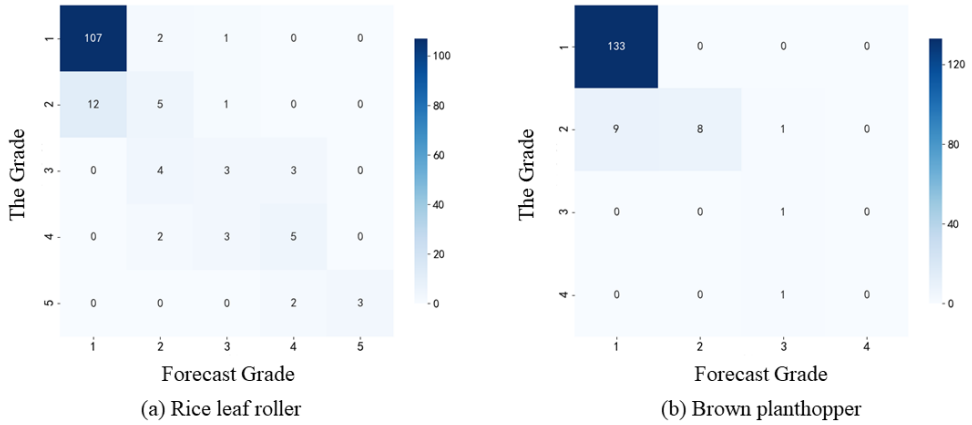

Fig. 11. Confusion matrix

To enhance the practicality of the predictive model, a comprehensive review of the related forecasting standards for rice "double-migration pests" was conducted, and these pests were categorized into five levels of occurrence: light, slightly light, moderate, slightly severe, and severe. The three-day forecast results of the GCN-AGRU model were classified and compared with the actual classification results. Fig. 11 displays the confusion matrices for the rice leaf folder and rice planthopper infestations. The results from the figure show that Level 1 (light occurrence) predominates among the two types of pest infestations, with fewer samples at higher infestation levels. From the confusion matrix for rice leaf folder (Fig. 11(a)), it can be seen that the model is capable of accurately classifying Level 5, which represents severe infestations. The model achieved a weighted accuracy of 80.4% and a weighted F1 score of 78.9%, demonstrating good performance in classification. For the classification of rice planthoppers (Fig. 11(b)), the model's weighted accuracy in predicting rice planthoppers was 0.928, with a weighted F1 score of 0.916, indicating favorable results in this classification task as well.

5 Conclusions

This study focuses on the occurrence of pest infestations during the rice growing stages in counties and cities of Hunan Province that experienced severe pest problems from 2010 to 2023. It integrates spatial information about rice pest infestations and employs the GCN-AGRU model for spatio-temporal forecasting. Comparative analyses confirm that the GCN-AGRU model outperforms other forecasting models on key performance indicators such as MAPE, RMSE, MAE, and R^2 in predicting rice pest infestations across multiple stages. In one-day forecasts, this model demonstrates high prediction accuracy, and although accuracy slightly decreases in three-day and six-day forecasts, the model still effectively captures the main trends of actual pest occurrences and maintains a high R^2 value over longer forecast intervals. Further analysis using confusion matrices shows that the GCN-AGRU model exhibits high precision in classifying low-level infestations of rice leaf folder and rice planthopper and performs well in classifying less common, higher-level infestations. This study indicates that the GCN-AGRU model exhibits superior comprehensive performance both in time series analysis and in generalizing across geographical locations, with its robust spatio-temporal data handling capabilities, accurate prediction performance across various time scales, and precise classification of pest occurrence levels, providing a valuable reference for forecasting rice pest infestations.

6 Acknowledgement

This work is granted by The Major Project of Scientific and Technological Innovation 2030 (2021ZD0113603), and Hebei Province Central Leading Local Science and Technology Development Project (246Z1817G).

References

[1] R. Prasad, Y.S. Shivay, D. Kumar, Current status, challenges, and opportunities in rice production, Rice production worldwide (2017) 1-32. DOI: 10.1007/978-3-319-47516-5_1.

[2] L. X. Chang, X. R. Liang, L. Wang, Z. R. Li, F.D. Zhan, Y. M. He, Characteristics and Influencing Factors of Soil Organic Carbon Sink in Paddy Fields in China: A Review, Soils 55(3)(2023) 487-493.

[3] Y.T. Shao, Y.P. Wang, Y.W. Yuan, Food safety and government regulation in rural Chiana, Journal of agriculture and food research 5(2021) 100170.

[4] R. Mouly, T.N. Shivananda, A. Verghese, Weather based forecasting models for prediction of leafhopper population Idioscopus nitidulus Walker; (Hemiptera: Cicadellidae) in mango orchard, Journal of Entomology and Zoology Studies 5(1)(2017) 163-168.

[5] Y.X. Bao, X.Y. Chen, X.J. Xie, L. Wang, M.H. Lu, Short-term Forecasting Models on Occurrence of Rice Leaf Roller Based on Kalman Filter Algorithm, Chinese Journal of Agrometeorology 37(5)(2016) 578.

[6] Y.W. Yan, C.C. Feng, M.P.H. Wan, K.T.T. Chang, Multiple regression and artificial neural network for the prediction of crop pest risks, in: Proc. 2015 Information Systems for Crisis Response and Management in Mediterranean Countries, 2015.

[7] P.D.K. Jayanthi, A. Verghese, P.D. Sreekanth, Predicting the oriental fruit fly Bactrocera dorsalis (Diptera: Tephritidae) trap catch using artificial neural networks: A case study, International Journal of Tropical Insect Science 31(4)(2011) 205-211.

[8] R. Narava, S.R.K. DV, J. Jaba, A.K. P, R. R. GV, S.R. V, S.P. Mishra, V. Kukanur, Development of temporal model for forecasting of Helicoverpa armigera (Noctuidae: Lepidopetra) using Arima and Artificial Neural Networks, Journal of Insect Science 22(3)(2022) 2.
[9] R.P. Sharma, R. Dharavath, D.R. Edla, IoFT-FIS: Internet of farm things based prediction for crop pest infestation using optimized fuzzy inference system, Internet of Things 21(2023) 100658.
[10] S. Skawsang, M. Nagai, N.K. Tripathi, P. Soni, Predicting rice pest population occurrence with satellite-derived crop phenology, ground meteorological observation, and machine learning: a case study for the Central Plain of Thailand, Applied Sciences 9(22)(2019) 4846.
[11] Q.X. Xiao, W. Li, P. Chen, B. Wang, Prediction of crop pests and diseases in cotton by long short term memory network, in: Proc. 2018 Intelligent Computing Theories and Application, 2018.
[12] P. Chen, Q.X. Xiao, J. Zhang, C.J. Xie, B. Wang, Occurrence prediction of cotton pests and diseases by bidirectional long short-term memory networks with climate and atmosphere circulation, Computers and Electronics in Agriculture 176(2020) 105612.
[13] C.Y. Zhong, L. Hu, Z.H. Zhang, Y.J. Ye, S.H. Xia, Spatio-temporal gating-adjacency gcn for human motion prediction, in: Proc. 2022 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022.
[14] J. Bruna, W. Zaremba, A. Szlam, Y. LeCun, Spectral networks and locally connected networks on graphs. <https://arxiv.org/abs/1312.6203>, 2013 (accessed 13.03.2024).
[15] M. Defferrard, X.Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering. <https://arxiv.org/abs/1606.09375>, 2016 (accessed 18.04.2024).
[16] Y. Liu, Z. Song, X. Xu, W. Rafique, X. Zhang, J. Shen, M.R. Khosravi, L. Qi, Bidirectional GRU networks-based next POI category prediction for healthcare, International Journal of Intelligent Systems 37(7)(2022) 4020-4040.
[17] A. Galassi, M. Lippi, P. Torroni, Attention in natural language processing, IEEE transactions on neural networks and learning systems 32(10)2020 4291-4308.
[18] Z. Yuan, L. Meng, X. Gu, Y. Bai, H. Cui, C. Jiang, Prediction of NOx emissions for coal-fired power plants with stacked-generalization ensemble method, Fuel 289(2021) 119748.
[19] M.G. Lawrence, The relationship between relative humidity and the dewpoint temperature in moist air: A simple conversion and applications, Bulletin of the American Meteorological Society 86(2)(2005) 225-234.
[20] B. Yu, H. Yin, Z. Zhu, Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. < https://arxiv.org/abs/1709.04875>, 2017 (accessed 16.05.2024).
[21] X.Y. Wang, Y. Ma, Y.Q. Wang, W. Jin, X. Wang, J.L. Tang, Traffic flow prediction via spatial temporal graph neural network, in: Proc. 2020 Proceedings of the web conference, 2020.