Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 15, No. 1, 2024

Machine Learning-Based Morel Mushroom Environmental Monitoring System

Lei Ye, Yun Xu

Sichuan University of Science & Engineering, YiBin 644002, China

Abstract: As living standards improve, the demand for edible fungi increases, making their efficient cultivation crucial for food supply and agricultural productivity. Traditional manual monitoring has limited accuracy, but the introduction of IoT and embedded technology enhances the intelligence and precision of agricultural monitoring, saving both labor and resources. This study designs a machine learning-based environmental monitoring and prediction system for edible fungi, utilizing low-power ZigBee networks and Kalman filtering to improve data accuracy. The system ensures real-time data transmission via the MQTT protocol and enhances security and scalability through cloud storage. By comparing Transformer, LSTM, and LSTM-Attention models, it was found that the Transformer model performed best in predicting environmental parameters, enabling proactive regulation, boosting yield and quality, and promoting the development of intelligent agriculture.

Keywords: Environmental parameters; Machine Learning; Embedded; Internet of Things.

1. System Hardware Design

The system adopts a modular design structure. Based on functional analysis, the system hardware architecture is mainly divided into an STM32 main controller module, a communication module, a Zigbee environmental information collection module, and a relay module.

1.1. Main Control Module

The main control module of this system is an STM32 development board, a cost-effective ARM series product available on the market. This development board uses a Cortex-M3 processor, which offers more internal resources compared to the 51 microcontrollers in the same price range. In terms of hardware and software compatibility, the STM32 series performs exceptionally well. Especially when switching to chips with a different number of pins, its portability allows developers to upgrade the hardware without needing to modify the existing code.

1.2. Relay Module

A relay is an electrically operated switch that uses a small current to control a larger current, enabling safe and efficient circuit management. This characteristic makes relays highly useful in various automation control systems, especially in scenarios requiring remote or automatic control of high-power loads. In this system, relays are extensively used to control the switching operations of heating equipment, ventilation systems, water supply devices, and supplementary lighting equipment.

1.3. Communication module

Since its launch, the ESP8266 module has gained widespread recognition and application in Internet of Things (IoT) projects due to its low cost, high integration, ease of use, flexibility, and low power consumption. Developed by Espressif Systems, this Wi-Fi microcontroller chip is not only economical but also integrates a 32-bit processor, TCP/IP network stack, 10-bit ADC, and Wi-Fi functionality, significantly simplifying the development process of wireless network applications. It supports various development environments. In standby mode, the ESP8266 consumes very little power, making it ideal for applications requiring long-term battery operation. Additionally, its robust developer community provides users with abundant resources and support, making the ESP8266 the preferred module for implementing IoT connectivity. Whether serving as a standalone application processor or as a Wi-Fi access device for other main control chips, the ESP8266 demonstrates exceptional performance and application value. Therefore, this system's communication module utilizes the ESP8266. Its primary function is to transmit sensor-collected data to the cloud server in real-time and receive control commands issued from the cloud

1.4. Zigbee Environmental Information Collection Module

The ZigBee environmental information collection module primarily comprises ZigBee end devices, a coordinator, and various sensors, with the CC2530 chip serving as the core controller. In this setup, the ZigBee end devices are responsible for receiving and processing data collected from the sensors. The coordinator, a crucial component of the ZigBee network, gathers signals from these sensors and transmits them to the main controller. Each ZigBee end device consists of several independent yet interconnected subunits, while the coordinator communicates with the main controller via a serial interface.

2. System Software Design

2.1. Cloud Platform Application Design

Using cloud service platforms, the system can upload and store data for monitoring, while also supporting remote presentation of the data. The cloud service further allows for remote control of greenhouse equipment. By accessing the environmental monitoring management system, users can remotely review the environmental indicators of the greenhouse and perform manual or automatic adjustments to the greenhouse facilities. The human-computer interaction interface adopted by this system is the Things Cloud platform. First, log in to the console to create a project. Then, enter the project, create connected devices, and finally open the visualization dashboard to design the monitoring system's visual interface.

2.2. Zigbee Module Software Design

This system is designed based on the C language using a Zigbee development board, with IAR Embedded Workbench selected as the development environment. The workflow begins with data collection from terminal devices, followed by the transmission of data to the coordinator via the Zigbee network's router. After receiving the data, the coordinator forwards the information through the serial port.

2.3. Main Control Module Software Design

This system uses an STM32 development board, with programming based on C and C++ languages, and employs the efficient MDK development environment for software development and debugging. During system operation, the STM32 core module is responsible for receiving environmental monitoring data and effectively forwarding the information to a dedicated communication module via serial communication. Additionally, the system features an intelligent data analysis mechanism that can logically assess the received data to identify any anomalies. Once an anomaly is detected, the system automatically triggers an output signal to activate a relay, thereby controlling external devices. The main controller can also receive control commands issued from the cloud to turn the corresponding devices on or off, thus regulating the greenhouse environment.

2.4. Communication Module Software Design

The communication program of this system is written in Lua scripts and then flashed into the ESP8266 communication module using ESP8266Flasher. The program receives serial data, connects to the MQTT server, and acts as an MQTT client to publish topic information, transmitting data to the cloud.

3. Introduction to the Transformer Model

The Transformer model is a neural network based on the self-attention mechanism, introduced by Vaswani et al. in the paper "Attention is All You Need". The Transformer model has become a foundational model in NLP (Natural Language Processing) due to its excellent performance in tasks like machine translation and text generation, along with its parallel processing advantages. Below is an introduction to the Transformer model, including its primary formulas and calculation processes. The Transformer model's structure mainly consists of two modules: an encoder and a decoder. The encoder processes the input sequence, while the decoder generates the output sequence. Each encoder and decoder module include multiple layers, and each layer has a multi-head self-attention mechanism and a feed-forward neural network [1].

3.1. Self-Attention Mechanism

The self-attention mechanism is the core idea of the Transformer. It enables the model to

focus on relevant information within a sequence when processing sequence data

For an input sequence $\{x_1, x_2, \ldots, x_n\}$,each word generates three vectors: Query, Key, and

Value. These vectors are generated by the following linear transformations:

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V$$

where Q, K and V are the matrices for queries, keys, and values, and W_Q, W_K, W_V are the learnable weight matrices. Then, these matrices are used to compute self-attention scores (also called scaled dot-product attention):

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Here, $\sqrt{d_k}$ is a scaling factor, where d_k is the dimensionality of the key vector. This scaling prevents the dot-product values from becoming too large when the input length is long, which could lead to gradient vanishing issues. The resulting attention scores represent how much each word in the sequence focuses on other words [2].

3.2. Multi-Head Attention Mechanism

To allow the model to capture information from different subspaces, the Transformer uses multi-head attention. Multiple heads calculate attention matrices in parallel, which are then concatenated and linearly transformed. The formula is as follows:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W_O$$

where each attention head head_i is computed using different W_{Q_i}, W_{K_i}, and W_{V_i} matrices:

$$\text{head}_i = \text{Attention}\left(QW_{Q_i}, KW_{K_i}, VW_{V_i}\right)$$

3.3. Feed-Forward Network (FFN)

Each encoder and decoder layer also includes a feed-forward neural network, which applies a nonlinear transformation to the representation of each word position. The formula for the feed-forward network is:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

where W_1 and W_2 are weight matrices, and b_1 and b_2 are bias vectors. This structure usually uses the ReLU activation function.

3.4. Positional Encoding

Since the Transformer model lacks convolutional or recurrent structures, it requires positional information to represent the order of words. Positional encoding can be calculated using sine and cosine functions as follows:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

where pos is the position, i is the dimension index, and d_{model} is the dimensionality of the word embeddings [4].

3.5. System Prediction Evaluation Metrics

In time series forecasting, selecting the appropriate model, performing accurate model evaluation, and tuning the parameters are crucial. Commonly used model evaluation metrics include Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE).The smaller the RMSE, the better the model's predictive ability. MAE provides the average level of prediction error, with smaller values indicating better performance. The lower the

MAPE, the higher the model's prediction accuracy [5].

4. Experiment

Since temperature has the most significant impact on the growth of Morchella (morel), temperature is chosen as the subject of this experiment. Data from the large greenhouse is collected every ten minutes. The data is then processed using multiple imputation methods and the Kalman filtering algorithm. The daily average temperature is calculated, and the average temperature for a consecutive 200 days is selected as the dataset for this experiment. In the process of using the Transformer model for temperature prediction, the proper division of the data is crucial to ensuring the effectiveness of model training and the accuracy of test evaluations. In this study, the entire dataset is divided into two parts: 90% of the data is selected as the training set for model training and parameter optimization, while the remaining 10% is reserved as the test set to evaluate the model's ability to predict unseen data, as shown in Figure 1. This division strategy aims to leverage the majority of the data for in-depth learning, while reserving a small portion for later model performance validation to ensure the objectivity and reliability of the evaluation results.

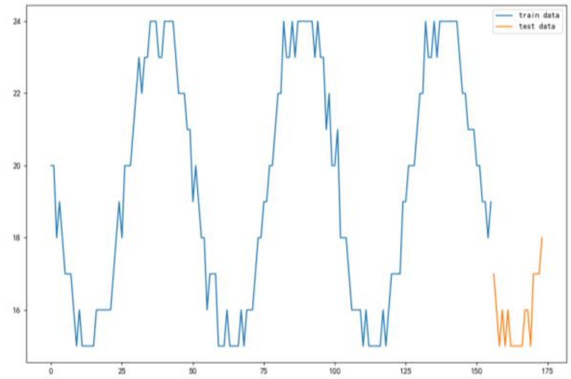

Figure 1. Training Set and Test Set

In the temperature prediction results of the Transformer model shown in Figure 2, the predicted values are highly similar to the actual temperature values in the test set, indicating that the model can accurately capture temperature trends. By comparing the predicted curve with the actual curve, the minimal difference verifies the effectiveness and potential of the Transformer model in real-world applications. The model evaluation metrics are: MAE = 0.5049, RMSE = 0.6745, MAPE = 3.2442, demonstrating the model's excellent performance in prediction accuracy.

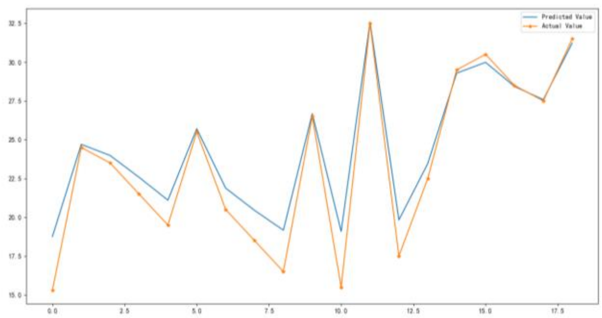

Figure 2. Transformer Model Prediction Results

The comparison models selected in this paper are the LSTM model and the LSTM-Attention model. Similar to the Transformer model, the evaluation metrics of the system are obtained through model training and prediction. Then, the predictive performance of the three models is analyzed and compared.

The prediction results of the LSTM model are shown in Figure 3. The evaluation metrics of the model's predictions are calculated as follows: MAE = 0.6708, RMSE = 0.8061, MAPE=4.1193.

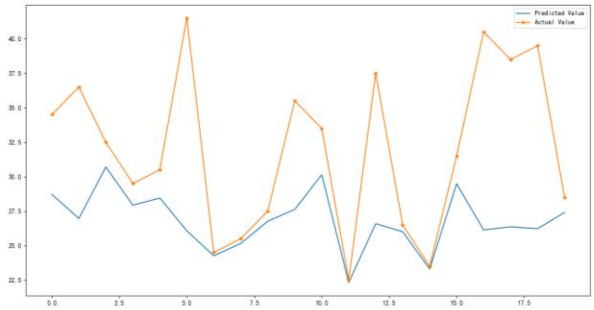

Figure 3. LSTM Model Prediction Results

The prediction results of the LSTM-Attention model are shown in Figure 4. The evaluation metrics of the model's predictions are calculated as follows: MAE = 0.6708, RMSE = 0.8061, MAPE = 4.1193.

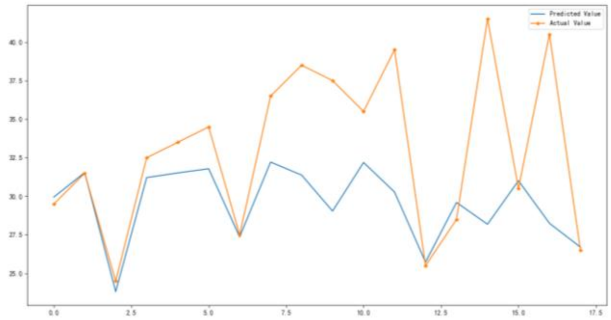

Figure 4. LSTM-Attention Model Prediction Results

The evaluation metrics are plotted as a bar chart in Figure 5. After a comparative analysis of the Transformer, LSTM-Attention, and LSTM models based on the MAE, RMSE, and MAPE performance metrics, the results show that the Transformer model performs the best in terms of prediction accuracy.

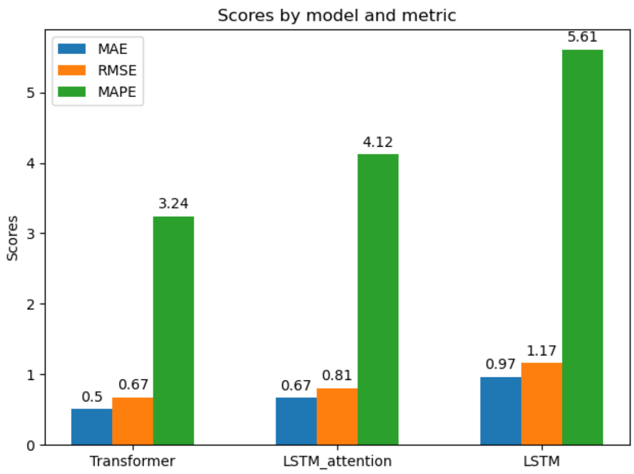

Figure 5. Bar Chart of Evaluation Metrics

References

[1] Park S O .Transformer for Inverter: 11/630997 [P]. US20080291353 [2024-11-07]. DOI:US20080291353.

[2] Shen T, Zhou T , Long G ,et al.DiSAN: Directional Self-Attention Network for RNN/CNN-free Language Understanding[J]. 2017.DOI:10.48550/arXiv.1709.04696.

[3] Sulong A, Martono W , Siddiqi M U .Intelligent keystroke pressure-based typing biometricsauthentication system using multilayer feed-forward network[J]. 2011.

[4] [1] Baniata L H, Kang S , He M E .Switching Self-Attention Text Classification Model with Innovative Reverse Positional Encoding for Right-to-Left Languages: A Focus on Arabic Dialects[J]. 2024.

[5] Demorest P B, Pennucci T , Ransom S M ,et al.A two-solar-mass neutron star measured using Shapiro delay[J].Nature, 2010, 467.DOI:10.1038/nature09466.