sustainability

MDPI

Article

Using Time-Series Generative Adversarial Networks to Synthesize Sensing Data for Pest Incidence Forecasting on Sustainable Agriculture

Chen-Yu Tai, Wun-Jhe Wang and Yueh-Min Huang *

Department of Engineering Science, National Cheng Kung University, Tainan 70403, Taiwan
* Correspondence: huang@mail.ncku.edu.tw

Abstract: A sufficient amount of data is crucial for high-performance and accurate trend prediction. However, it is difficult and time-consuming to collect agricultural data over long periods of time; the consequence of such difficulty is datasets that are characterized by missing data. In this study we use a time-series generative adversarial network (TimeGAN) to synthesize multivariate agricultural sensing data and train RNN (Recurrent Neural Network), LSTM (Long Short-Term Memory), and GRU (Gated Recurrent Unit) neural network prediction models on the original and generated data to predict future pest populations. After our experiment, the data generated using TimeGAN and the original data have the smallest *EC* value in the GRU model, which is 9.86. The results show that the generative model effectively synthesizes multivariate agricultural sensing data and can be used to make up for the lack of actual data. The pest prediction model trained on synthetic data using time-series data generation yields results that are similar to that of the model trained on actual data. Accurate prediction of pest populations would represent a breakthrough in allowing for accurate and timely pest control.

Keywords: time series; data augmentation; deep learning; pest forecasting; generative adversarial network (GAN)

check for updates

Citation: Tai, C.-Y.; Wang, W.-J.; Huang, Y.-M. Using Time-Series Generative Adversarial Networks to Synthesize Sensing Data for Pest Incidence Forecasting on Sustainable Agriculture. *Sustainability* **2023**, *15*, 7834. https://doi.org/10.3390/su15107834

Academic Editors: Barlin Orlando Olivares Campos, Edgloris E. Marys and Miguel Araya-Alman

Received: 17 April 2023
Revised: 6 May 2023
Accepted: 8 May 2023
Published: 10 May 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The global demand for sustainable development is increasing, and agriculture is no exception. Sustainable agriculture is a production method that emphasizes sustainable development in economic, social, and environmental aspects [1,2]. It primarily focuses on protecting the environment, improving the productivity of crops and animals, enhancing farmers' income, and improving the quality of life in local communities. In order to improve agricultural productivity and reduce production costs, agricultural production is increasing thanks to the Internet of Things (IoT) and big data analytics [3]. For example, IoT devices receive sensor data [4], RGB images, or multispectrum images and then analyze these data for pest control [5], crop yield prediction [6], precise agricultural irrigation [7], and so on.

Pest prediction is vital in agriculture. A lack of effective pest control directly affects crop yields [8]. In general, it is difficult for farmers to gain a complete picture of crop growth due to the large area of farmland. In recent years, weather conditions have become more severe due to climate change, making pests more adaptable to environmental changes. Multiple sensors placed in remote fields would enable farmers to constantly monitor the environmental data in each area. By using environmental data and an understanding of pest species, experts could build a pest prediction system [9,10] for integrated pest management (IPM). Farmers could plan their control operations in advance given early warnings generated by the prediction system.

Effective analysis of meteorological and pest data is crucial for accurate pest prediction systems. Since both types of data are constantly monitored in the field, early warning is key to pest forecasting [11]. For sudden increases in pest populations, immediate control

may not be possible. Technological advances have yielded more sophisticated methods for time-variant analysis for data prediction [12]. However, such prediction relies on a large amount of historical data. Historical data is rich in time dynamics. Greater amounts of sensing data facilitates more accurate pest prediction. Conversely, insufficient sensing data degrades the effectiveness of the prediction system.

Environmental sensing data is now obtained generally from IoT edge devices. The effectiveness of such edge devices deployed in farms can be degraded by climate, flora, fauna, and other natural factors, resulting in incorrect sensor values or even damage to the devices [13], causing problems in subsequent data analysis, such as data errors during preprocessing or data losses due to edge device damage; these conditions must be corrected to ensure continued data availability.

Data augmentation algorithms are one way to resolve data deficiencies. Since prediction models are affected by the amount of data, it is more important to address the fundamental problem of data deficiency than to use complex prediction models [14]. Data augmentation increases the variability of data by generating data samples. For instance, augmentation methods for image recognition use rotation, translation, and scaling to produce multiple images from the same image [15]. In contrast, a prediction system for linear problems uses time series data, for which data augmentation is dissimilar to that for images [16]. Here, we investigate time series data augmentation to address the problem of insufficient data for pest prediction.

Samples with missing data can be divided into those that are missing for a long period of time and those that are missing for only a short instant. The amount of missing agricultural sensing data can significantly affect the accuracy of the amplification [17]. When a small sample of data is missing, common scaling methods generate simulations with similar results. However, when more data samples are missing and the data are multivariate, such scaling does not preserve both the temporal features and the characteristics of multivariate data. Therefore, we take into account both temporal and multivariate data features, and generate data that retains both features.

Time-series data is a regression problem in which sequence values change according to time dynamics. Clearly, data generation requires that we preserve the temporal characteristics of the original data and increase the number of samples. However, depending on the number of data samples to be augmented and the number of data variables, basic augmentation cannot be used to generate long-term or multivariate sensing data. In this case, a time-series generative adversarial network (TimeGAN) is an effective model for generating long-term data [18]. The temporal features of the data are learned by the autoregressive (AR) model, and a large amount of data is synthesized by the generative adversarial network (GAN). In this study, we use TimeGAN for data augmentation to synthesize multivariate sensing data.

The target pest in this study is *Scirtothrips dorsalis* Hood (*S. dorsalis*), which in Taiwan primarily affects mangoes. We seek to use agricultural sensing data as the main analysis. Firstly, we use multivariate agricultural sensing data to better understand the environmental conditions, and further use a deep generative model to synthesize more time-series data. We then verify the effect of the generated data using time prediction models such as recurrent neural networks (RNN), long short-term memory (LSTM) models, and gated recurrent unit (GRU) models [19]. Our purpose is to predict future trends and alert farmers of them in order to control pest populations in a timely manner. We analyze the effectiveness of time series data augmentation using data visualization and evaluation metrics for regression.

In summary, we have put forth the following two hypotheses and designed corresponding experimental procedures to verify them:

(1) To examine whether TimeGAN can generate synthetic data with the same temporal characteristics as the original data.
(2) To investigate whether the pest prediction model trained using the synthetic data yields comparable results to the model trained using the original data.

The remaining sections of this paper are organized as follows: Section 2 surveys IoT smart agriculture applications for pest prediction and time series data augmentation, Section 3 describes the system architecture and our experimental methods, and Section 4 presents a comparison and discussion of the experimental results, including the processing of environmental data, a correlation analysis of meteorological factors, the construction of the generation model, and an evaluation of the prediction models. In Section 5, we conclude and suggest future research directions for pest prediction.

2. Related Literature

Below, we describe the current agricultural sensing data to implement time series tasks, after which we explore practical techniques for pest prediction based on image recognition and time series data. We then discuss time series data augmentation, focusing particularly on data augmentation methods based on deep generative models.

2.1. Agricultural Sensing Data

Smart agriculture uses sensors to collect agriculture-related data and transfer them to a cloud database. The stored data are analyzed to identify potential problems and to advance agricultural technology. Big data time analysis plays a crucial role in this because agricultural sensors often produce huge amounts of time-series data [20–22]. As the method used for data processing can affect downstream results, the challenge is how to analyze past data, adjust the dataset appropriately, and come up with the best processing strategy. Ren et al. [23] use data preprocessing methods to clean up specific features, transform structures and formats, and reduce complexity. By using multiple data preprocessing steps, we retain important features and improve efficiency when constructing deep learning models.

Analysis methods in agriculture vary in terms of the sensing data format and the needs of the task. Time-series tasks can be divided into classification, anomaly detection, and prediction. In time-series classification, sensor data identify crop species [24] and monitor growth status [25]; values from the spectral sensors are converted into a vegetation index which reveals information about crop growth and environmental health. In time-series abnormality detection, sensor data can reveal abnormalities in the environment or in crop yields [26]. In time-series prediction, past dynamic series provide potential trends by which to estimate future crop yields [27] and pest populations [28]. Selecting important features in multivariate agricultural data and improving the rationality of predictions facilitates subsequent crop and pest management.

2.2. Pest Prediction

The yield of high-value crops in the agricultural sector is of concern. As pest infestation during the production season can significantly reduce crop yields [29], pest infestation is an important and immediate problem. Delayed control or low doses of pesticides can cause economic crop losses [30]. According to Heeb et al. [31], up to 40% of crop production in the world is currently lost due to unmanaged pest damage. If experts could provide farmers with advance warnings of increases in pest populations, farmers would have enough time to apply the recommended amount of pesticides, and could thus effectively control the pest situations in their fields and prevent the endless spread of pest populations. Below, we survey research on pest prediction.

2.2.1. Image Recognition-Based Approaches

In recent years, the dilemma of traditional pest prediction has changed due to the advanced development of artificial intelligence (AI) and two technological innovations: intelligent image recognition technology [32] and time-series data prediction [33]. In image recognition technology, features of visible or multispectral images are analyzed to select and create pest-specific features by resolving color features and texture features that are different from those of the crop [34]. Deep learning models learn the characteristics of

the pest, mark the location of the pest in the image using object detection, classify the pest species, and report the probability that it is in fact this pest. Given the ease of image collection, deep research is now available both for close-up images from cameras or cell phones [35,36], as well as for overhead drone imagery [37,38].

2.2.2. Time-Series Data-Based Approaches

In time-series data prediction, it is common to combine multiple types of sensing devices to monitor and record real-time environmental and pest conditions in the field [39,40]. Prediction given time-series data is similar to weather prediction in that the prediction model learns past environmental and pest development trends from a large amount of historical data [41,42]. Since potential feature changes are sequential, temporal dynamics can be used to estimate the number of pests that may occur in the future. Figure 1 shows the number of studies on pest prediction from time-series data from 2012 to 2021. We retrieved the keywords *pest forecasting*, *pest prediction*, *time-series* from the Web of Science (WOS) database and recorded the number of papers published in each year.

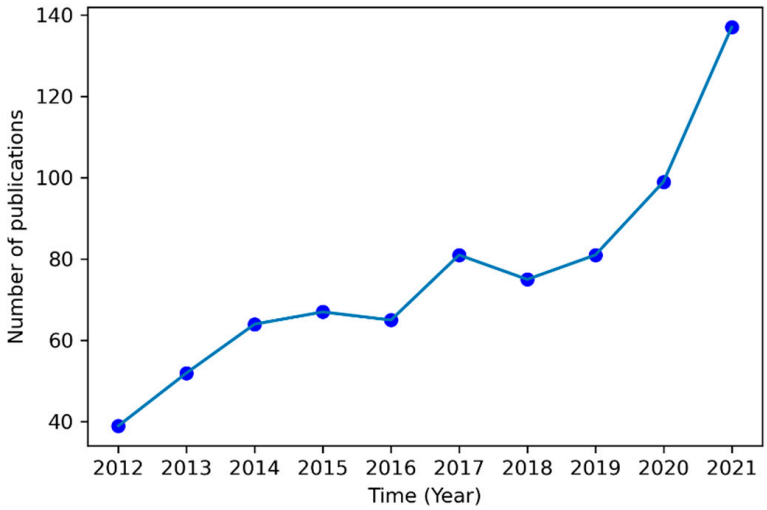

Figure 1. Publications on time-series data for pest prediction.

As shown in Figure 1, the publication of pest prediction-related papers increased gradually and steadily between 2012 and 2018. Pest prediction research then exploded beginning in 2019, both in terms of the stability of the values collected by IoT devices [43] and the more in-depth research on prediction models [44], which have contributed to the development of pest prediction. However, bottlenecks in pest prediction remain, including the use of a small number of variable features, the failure to analyze the validity of weather factors on prediction, the failure to investigate the effects of different model architectures on prediction, and the scarcity of dataset samples [45]. Therefore, we review the literature to further analyze and discuss pest prediction studies from recent years.

Techniques applied for data prediction include statistical methods, machine learning, and deep learning. Pest prediction using statistical methods is often achieved using the autoregressive integrated moving average (ARIMA) model, which does not take into account variations in relevant random variables. Narava et al. [46] use an ARIMA model to predict the population dynamics of *Helicoverpa armigera* and show that ARIMA is easier to implement for pest prediction.

Machine learning-based pest prediction is often implemented using the *k* nearest neighbors algorithm, which searches for the *k* closest samples in the feature space. Gómez et al. [47] build and compare the performance of six machine learning algorithms to predict desert locusts based on soil moisture and demonstrate that the model prediction performance is limited by the space and time of the data.

Deep model structures are often used to predict pests in deep learning. Tan et al. [48] use deep autoregressive (DeepAR) models in deep learning to predict *Chilo suppressalis*, a major threat to rice production, and show that deep learning predicts pest dynamics more accurately by integrating meteorological data, pest data, and temporal features. Therefore, we used a deep learning model for pest prediction due to our need to focus on the temporal dynamics of the sensing data and due to the ease of implementing pest prediction.

2.3. Time-Series Data Augmentation

Recent time-series prediction models include a large number of additional neural structures and use deeper architectures. Since the model is trained using a deeper structure, the dataset size affects the accuracy of the model prediction, resulting in overfitting for small datasets [49]. Note that the use of real-time datasets is limited by factors such as the completeness, ownership, and access of the dataset. Therefore, various methods are used to augment the original dataset and expand the amount of training data to increase its comprehensiveness. Wen et al. [50] has classified augmentation methods for time series data into two categories: basic approaches and advanced approaches. The basic approaches encompass techniques such as time domain, frequency domain, and time-frequency domain. Meanwhile, the advanced approaches consist of decomposition methods, statistical generative models, and learning methods. The learning methods can further be categorized into embedding space, deep generative models, and automated data augmentation. In the next section, we introduce basic time data augmentation methods and describe data augmentation algorithms based on the depth generation model.

2.3.1. Basic Data Augmentation

The most important feature of timeseries data are their time and frequency characteristics. Data values are fused with time and frequency to increase the amplitude of the signal to produce a time-series dataset. Therefore, data augmentation algorithms include time- and frequency-domain variants, for which data are transformed to the appropriate domain for subsequent task requirements. In time-series prediction, the effectiveness of time- and frequency-domain transformations has been demonstrated in the literature [51]. Common time-domain methods include window slicing, window warping, flipping, and noise injection; frequency-domain conversion algorithms include amplitude and phase perturbations (APP) and Fourier transforms (FT).

Um et al. [52] state that a single augmentation method has limited effectiveness, and that it is crucial to select and combine the right combination of transformations for the task to improve the performance via subsequent model analysis. Thus, it is important to select and combine methods for different tasks to develop a combined strategy for the task. Standard practices for data augmentation in the time domain include flipping, down-sampling, and adding slope. In the frequency domain, methods such as amplitude-adjusted Fourier transform (AAFT), iterated amplitude-adjusted Fourier transform (IAAFT), amplitude-phase permutation (APP), and short-time Fourier transform (STFT) are commonly used.

2.3.2. Deep Generative Model Approaches

Time-series data augmentation should not be limited to the generation of diverse data; for the dataset to be used with confidence, the features of the generated data should resemble the distribution of the actual data features. With the rise of deep learning, deep generative models (DGMs) have been created that combine the advantages of generative models and deep neural networks. The most advanced techniques in DGM are variational auto-encoders (VAEs), normalizing flow (NF), and generative adversarial networks (GANs) [53]. Many different architectures are GAN derivatives, including recurrent GAN (RGAN) and recurrent conditional GAN (RCGAN) [54]. Given the recent trend of using GAN approaches to synthesize simulated training sets, we select the time-series generative adversarial network (TimeGAN) [18] as the basis for the data augmentation model based on the characteristics of the sensing dataset.

3. Research Method

Here we describe the system architecture and the experimental implementation process. We present the specific design of the experimental framework, after which we describe the implementation method from the selection of the depth generation model and the data augmentation and model training to the final prediction and analysis by time-prediction models. Then, we list common evaluation metrics and explain the usage of validation metrics to adjust the training parameters.

3.1. System Design

The system records environmental data at fixed time intervals using sensor equipment at the experimental site and monitors pest conditions at various locations in the field as agricultural sensor data. After preprocessing this time-series data, the time units of the data are converted to weeks to merge the two datasets. The multidimensional time series is then augmented and analyzed for predictive purposes. Here we estimate the timing of pest emergence and recommend subsequent plans for pest control. This requires four steps: (1) agricultural sensing data collection and production, (2) time series data preprocessing and fusion, (3) multivariate time series data augmentation, and (4) pest prediction from synthetic data. The system architecture of this study is shown in Figure 2.

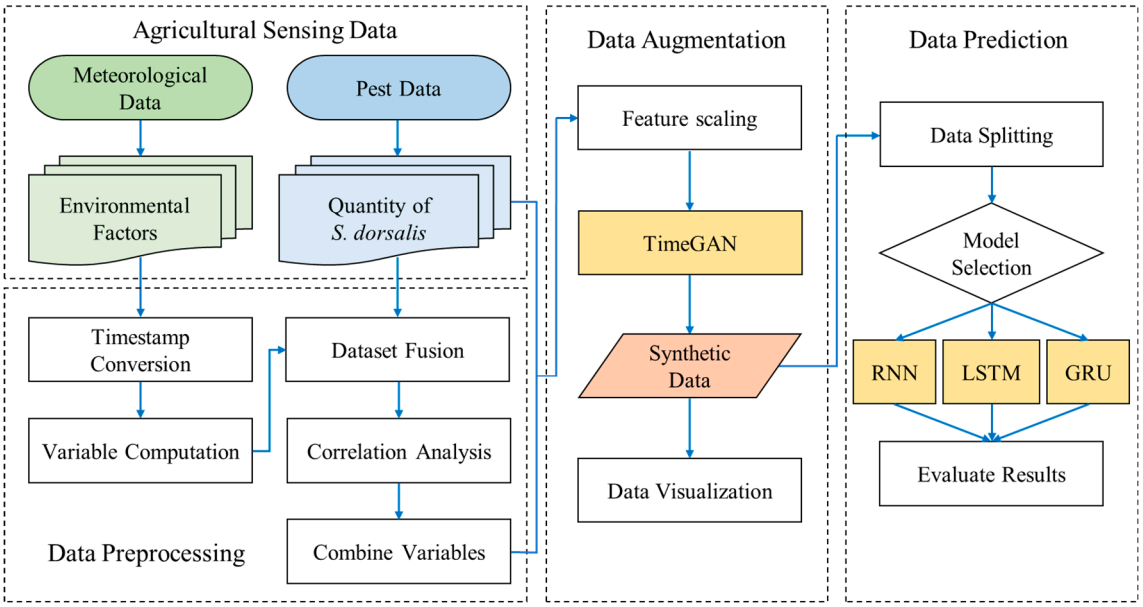

Figure 2. Proposed pest prediction system.

3.2. Time-Series Generative Adversarial Network

This study uses TimeGAN, a deep generative model, to build a deep learning model for agricultural sensing data [18]. TimeGAN was first used to improve the temporal correlation of raw time-series data that are not taken into account when GAN generates data. TimeGAN employs a generator-discriminator architecture for sequential generative adversarial training while fusing the embedding and recovery functions to learn encoding features of the data and generating representations in a large number of training epochs. The embedding and recovery functions construct a hidden space in which the adversarial networks are trained so that the feature space and the hidden space echo each other, generating features of the currently generated data with representations of the original sequence, and finally learning the temporal features of the time series.

3.2.1. TimeGAN Training Data

The training data comprised 59 instances of raw data with 22 characteristics, 21 including ground temperature, dew point temperature, relative humidity, and other weather

data, along with one pest count. The weather data came from the Agricultural Weather Observation Network Monitoring System of the Central Weather Bureau [55], and the pest data came from the Bureau of Agriculture, Kaohsiung City Government [56]. More details about the dataset are provided in Section 4.1.

Since the system uses the current week's sensing data to predict the next week's pest population (Section 3.3, TimeGAN's training data are produced as two-week samples and contain all agricultural sensing features, of which there are 22 sensing features after preprocessing. Therefore, TimeGAN generates a realistic random vector by setting the window size of the real data to 2, which means the generated data simulates two weeks of real data, and sets the feature length to 22, which means that the generated data simulates changes in the real 22 sensing data features.

Sliding the entire dataset completely with 59 samples of the original data using a moving window yields 57 two-week matrices for subsequent model training, where each matrix contains two weeks of sensing data, and the sensing data is preprocessed to obtain 22 features. As described above, real-time series input data trained by TimeGAN is produced to obtain a dataset of dimensions (57, (2, 22)) with 57 samples, each with 2 columns (weeks) and 22 variables (agricultural sensing features). Figure 3 shows a schematic of the TimeGAN training data, the features of which are normalized to [0, 1] to facilitate subsequent model training.

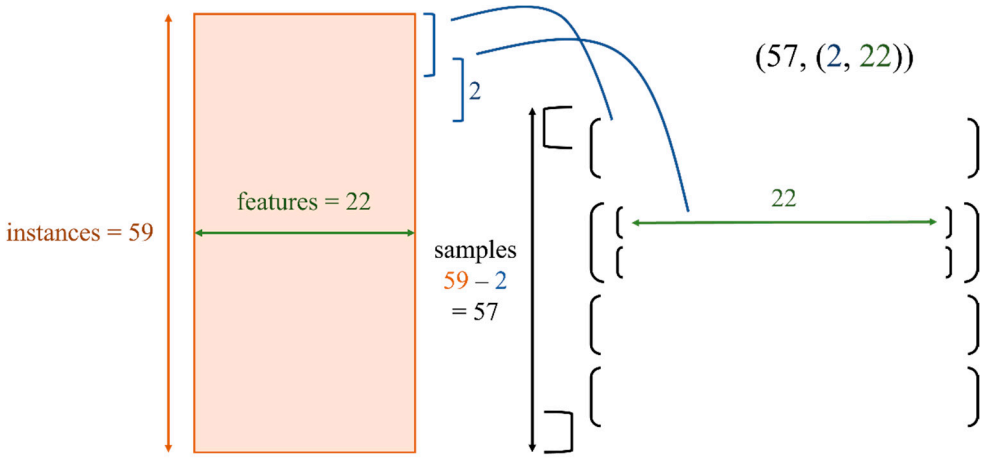

Figure 3. Schematic diagram of TimeGAN training data.

3.2.2. TimeGAN Training

The TimeGAN training process is shown in Figure 4. Augmented model training is composed of three stages. In stage 1, the embedding and recovery stage, actual data are used to achieve the best reconstruction of original data and obtain historical data features. In stage 2, the supervised stage, samples are generated to learn temporal features from real samples in matrix space. In stage 3, the joint stage, all of the networks (embedding, generator, recovery, and discriminator) are trained simultaneously to optimize all loss functions based on backpropagation.

In Figure 4, the TimeGAN model in this study uses a different loss function for optimization in each stage. The embedding stage trains using reconstruction loss ($\mathcal{L}_R$), the supervised stage optimizes using supervised loss ($\mathcal{L}_S$) and determines the time correlation between the training samples and the generated samples, and the joint stage optimizes all loss functions ($\mathcal{L}_R$, $\mathcal{L}_S$) and adjusts the generation network ($\mathcal{L}_U$) using unsupervised loss feedback; thus, this last stage is the most time-consuming.

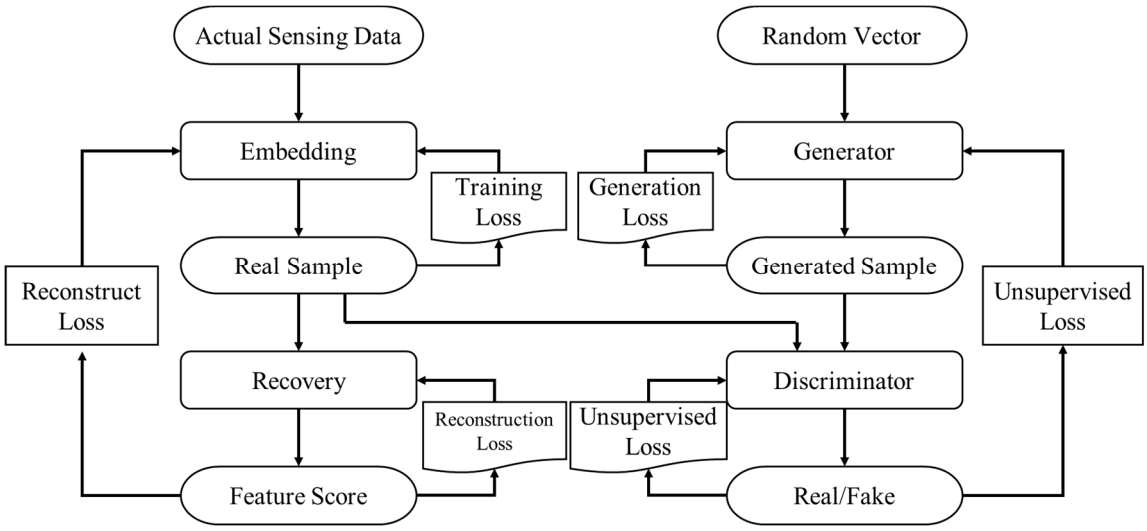

Figure 4. Training the TimeGAN model to generate data.

Here, the reconstruction loss ($\mathcal{L}_{\mathrm{R}}$) is calculated as the difference between the actual data (X_t) and the restored training sequence ($\widetilde{\mathrm{X}}_t$), as shown in (1). The original data are downscaled and restored to the original data. Loss reconstruction means that the real data are similar to the restored data after downscaling.

$$\mathcal{L}_{\mathrm{R}} = \mathbb{E}_{S,X(x,y)\sim\mathcal{P}data(x,y)}\left[\sum_t \left|\left|\mathrm{X}_t - \widetilde{\mathrm{X}}_t\right|\right|_2\right] \tag{1}$$

In this study, the supervision loss ($\mathcal{L}_{\mathrm{S}}$) is calculated as the difference between the temporal features of the training sequence (h_t) and the temporal features of the generated sequence ($\widetilde{\mathrm{h}_t}$), as shown in (2). The original data is downscaled to retain the temporal features of the data, and the generator adjusts to and learns the temporal features. Low supervised loss means that the training sequence has temporal features that are similar to the generated sequence.

$$\mathcal{L}_{\mathrm{S}} = \mathbb{E}_{S,\ X(x,y)\sim\mathcal{P}data(x,y)}\left[\sum_t \left|\left|\mathrm{h}_t - \widetilde{\mathrm{h}_t}\right|\right|_2\right] \tag{2}$$

Here, unsupervised loss ($\mathcal{L}_{\mathrm{U}}$) indicates the correct classification of the generated sequences as shown in (3). Temporal features retained after the original data is downscaled are used as a basis by which to identify whether the generated data created by the generator fit the true temporal features.

$$\mathcal{L}_{\mathrm{U}} = \mathbb{E}_{X(x,y)\sim\mathcal{P}data(x,y)}\left[\sum_t \log \mathcal{Y}_t\right] + \mathbb{E}_{S,X(x,y)\sim\mathcal{P}data(x,y)}\left[\sum_t \log\left(1-\hat{\mathcal{Y}}_t\right)\right] \tag{3}$$

As such, the objective function of this study is to minimize the very large generator and the identifier (G^*, D^*) by maximizing $\mathcal{L}_{\mathrm{U}}$ (for the identifier) and minimizing $\mathcal{L}_{\mathrm{R}}$ and $\mathcal{L}_{\mathrm{S}}$ (for the generator), as shown in (4). Since it is more important to preserve the temporal characteristics of the original data, in the final stage we seek the best generator and identifier solutions simultaneously.

$$G^*, D^* = \min_{\theta_e,\theta_g,\theta_r}\left(\mathcal{L}_{\mathrm{R}} + (\lambda+\eta)\mathcal{L}_{\mathrm{S}} + \max_{\theta_d}\mathcal{L}_{\mathrm{U}}\right) \tag{4}$$

3.2.3. TimeGAN Architecture

The multivariate agricultural sensing data augmentation model constructed in this study is implemented by the ydata-synthetic Python package [57]. The TimeGAN model is composed of an embedding network, a recovery network, a generator, and a discrimi-

nator, which are connected through a serial connection. Below, we describe the network architecture and settings of the four components.

Here, the generator input is a random sequence of size (2, 22) as described in Section 3.2.1, and the generator and discriminator are both networks with three layers of GRU neurons stacked on top of each other. The number of GRU neurons cascaded in each layer is hidden_dim, the hidden dimension of the network. Since we seek to expand the data in two-week increments, we set hidden_dim to 2, which causes the TimeGAN model to generate and identify two-week sample data. In addition, the generator input is a random vector of size (2, 22), and the generator output and the discriminator input are also sequences of size (2, 22), but the discriminator output is a discriminative score of size 1, used to determine whether the current generated sequence consists of true or fake data.

In this study, the embedded and recovered networks are trained with real data of size (2, 22), as discussed in Section 3.2.1. As the size of the generated target data must be the same as the real data, the input size of the network is the same. The embedding and recovery network is a three-layer LSTM stacked autoencoder because the dimension of the original data is too large to be used directly as the discriminator's judgment criterion; hence, the embedding and recovery network is used to downscale the data, and the training sequence is fitted through multiple generations to provide the discriminator for the generated sequence. The hidden_dim is set to 2 to match the data size of the generator and the discriminator. In addition, the input of the embedded network consists of real data of size (2, 22) and the output of the embedded network and the input of the recovered network are training sequences of size (2, 22), but the output of the recovered network has a size of 22 features scored for the coding and decoding effect of the autoencoder.

3.2.4. TimeGAN Training Validation

The purpose of generating new agricultural sensing data based on a train-on-synthetic, test-on-real (TSTR) [58] methodology and augmented data is to provide input to the subsequent time-series prediction model. The pest predictions are used to evaluate whether the new data generated are superior to the original data in the area of agricultural prediction tasks where data are not available. In this phase, therefore, the real model is trained independently from the synthetic model, where the real model represents training with real data and validation with real data, and the synthetic model represents validation with real data but training with augmented data. The model is evaluated to measure the effectiveness of generated data against real data.

3.2.5. Visualization of Synthetic Data

Data visualization makes it possible to present the generated multivariate agricultural sensing data as a picture of their model training effect and then analyze them for consistency between the generated and real data. Generally, basic visualization is used to compare the curves of the generated data with the real data. However, as the experimental agricultural sensing data is multivariate, the data dimensionality is reduced using algorithms such as principal components analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE) [59] to observe the distribution of the dataset generated by TimeGAN and that of the original agricultural sensing data.

3.3. Predictive Models

Here, we generate agricultural sensing data and then use a simple model to verify the difference between generated and real data. The initial validated evaluation metrics are real and simulated data, which include all weather variables but not pest prediction; thus, models for pest prediction must be designed. The two-stage model is used to evaluate the indicators and the combination of different independent and dependent variables. Results demonstrate the effect of multivariate time-series data augmentation, and experiments reveal which factors are significant in predicting *S. dorsalis* population trends.

Prediction of future pest populations requires time-series data as inputs to the model. The input data are a combination of independent variables based on weather factors investigated in the correlation analysis, and the output data are weekly pest counts. After confirming the input and output data, the data are normalized, and the experimental dataset is partitioned into a training dataset for input during model training and a test dataset to evaluate the performance of a trained model. We use three time-series prediction models (RNN, LSTM, and GRU), and set their parameters, architectures, and optimizers accordingly. Finally, we use the evaluation metrics to compare the prediction errors of the three models and calculate the prediction errors of the original and generated data. The time-series model prediction flow is shown in Figure 5.

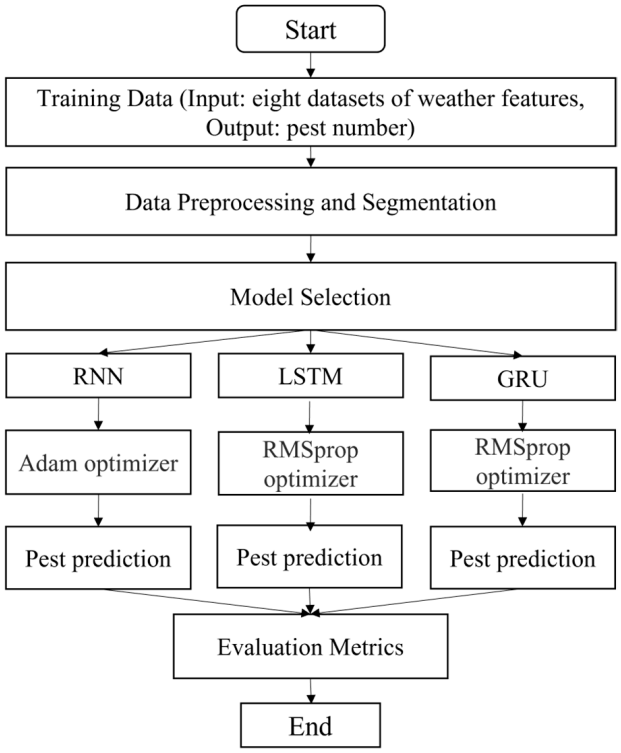

Figure 5. Pest prediction based on weather factors.

3.4. Evaluation Metrics

The experimental dataset combines observed environmental meteorological data and field surveys of pest infestation by investigators. Assuming that there is too few data, we use the TimeGAN model to expand the time-series data with more variables, and compare the three prediction models (RNN, LSTM, GRU) before and after this expansion. We measure the model effectiveness using time-series prediction error metrics. Since future pest populations are predicted as a continuous value, the predicted value should be close to the real data. Therefore, we use four common regression metrics: mean absolute error (*MAE*), mean absolute percentage error (*MAPE*), root mean squared error (*RMSE*), and the coefficient of determination (R^2) [60]. In addition, since the original data and the generated data are used as training data, the error change (*EC*) is used to compare the predicted error change before and after data augmentation. The above index equations are as follows:

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y_i - \hat{y}_i| \tag{5}$$

$$MAPE = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \tag{6}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_i - \hat{y}_i)^2} \quad (7)$$

$$R^2 = 1 - \sum_i (y_i - f_i)^2 / \sum_i (y_i - \hat{y})^2 \quad (8)$$

$$EC = MAE_{w/\ aug} - MAE_{w/o\ aug} \quad (9)$$

As *MAE* is the sum of the absolute values of the difference between the actual and predicted values, it reflects the actual difference between the predicted and sample values, as shown in (5); *MAPE* explains the difference between the predicted and actual values intuitively in terms of a percentage, as shown in (6); *RMSE* is the sum of the squares of the difference between the actual and predicted values, and is used to evaluate the error between observed and actual values. Since *RMSE* is calculated by squaring, it amplifies the error in addition to maintaining the absolute value estimation, as shown in (7); R^2 represents the strength of the correlation between the dependent variable and the independent variable, and is used to explain the degree of fit between the predicted and actual values in the model, as shown in (8); *EC* is the difference between the *MAE* of the augmented data and the *MAE* of the original data. *MAE* results are compared to analyze the performance of the training model between the augmented data and the original data, as shown in (9). The smaller the value of the first three error indicators (*MAE*, *MAPE*, and *RMSE*), the lower the error, and the larger the fourth validation indicator (R^2), the better. The smaller the *EC* value, the better the fit of the expansion to the original data. Therefore, for regression, the same evaluation metric can be used for different network architectures. The evaluation metrics correspond directly to the model performance and accuracy.

4. Experimental Procedure and Discussion

4.1. Research Site

The site for this experiment was an orchard in Liugui District, Kaohsiung City, Taiwan, as shown in Figure 6. According to the annual Kaohsiung City Agricultural Statistics report [61], the total production of mangoes in Kaohsiung City is about 13,300 tons, of which the production in Liugou District is about 3100 tons. Since this area is the highest yielding area in Kaohsiung, we chose Liugui as the most representative site for pest threats. In addition, since mango is the most abundant fruit in the experimental area, the main research issue is how to maintain the quality of the mangoes and reduce the threat of pests and diseases. In summary, we seek to investigate the historical data of this area and estimate future pest trends via smart agriculture to provide farmers with real-time spraying guidance to control pest populations.

4.1.1. Pest Data

The pest surveyed in this study was *Scirtothrips dorsalis* Hood (*S. dorsalis*), also known as the yellow tea thrip. The experimental pest dataset was obtained from public data monitored from 2019 to 2021 by the Kaohsiung City Bureau of Agriculture [56]. Since the current *S. dorsalis* control focuses mainly on issuing early warnings when the number of insects rises to remind farmers to prepare for pest control in advance, the annual monitoring period ends when the mangoes are in season, so no monitoring was conducted from September to December. Therefore, the purpose of this study is to estimate the timing of pest emergence so farmers know when to spray insecticides.

The pest dataset is a collection of data collected from farmland locations in response to experimental needs. Eight mango orchards larger than one hectare were recorded, and for each mango orchard we selected, the most representative locations were affected by only the small yellow thistle and no other pests. The number of crops in the vicinity of each site was approximately the same. Ten sticky boards were deployed at each site for trapping and counting, and the boards were replaced at regular intervals every week. The data were

recorded in terms of the number of pests, and the average of each sticky board was taken, so the pest data are expressed as the average number of pests at the site for that week. The sum of the eight sites represents the pest level corresponding to the weather information for that week.

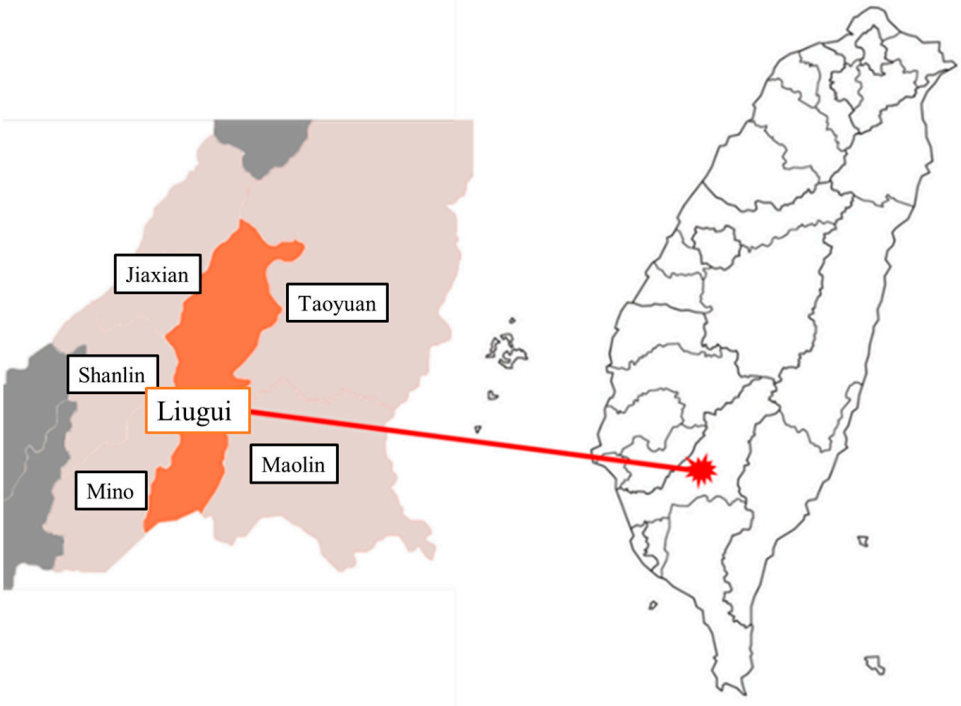

Figure 6. Experimental site: Liugui District, Kaohsiung City.

4.1.2. Meteorological Data

As pest populations are affected by environmental variables such as *temperature*, *humidity*, and *light*, we used meteorological data to construct a multivariate agricultural dataset. Since the number of pests and the weather dataset must be monitored at the same time and location, for the meteorological data we also used the environmental data for Liugui District, Kaohsiung City. The experimental meteorological dataset was obtained from the Agricultural Weather Watch monitoring system of the Central Weather Bureau [55] of Taiwan. On the CWB's weather monitoring website, there is one station for Liugui District (code E2P980), so the data recorded for this station were used as the meteorological data, as they are most representative of the weather in the orchards in Liugui District.

We filtered the meteorological data from 2020 to 2021. The weather data were sampled according to time attributes, and the raw data included 16 weather factors, among which are *temperature*, *precipitation*, *relative humidity*, *atmospheric pressure*, *all-sky solar radiation*, and *wind speed*, as shown in Table 1. The weather data were monitored at hourly intervals, and there were 17,000 data instances. We rescaled the time units of the environmental features to match the temporal attributes of the meteorological data and the pest data, and fused the data to generate the agricultural sensing data for this experiment.

4.2. Data Preprocessing

For data preprocessing, we used data cleaning, missing value processing, resampling, and data fusion. In addition, we conducted a correlation analysis to better understand the relationship between variables of agricultural sensing data.

Table 1. Meteorological data of weather variables and units.

Weather Variable	Unit	Weather Variable	Unit
Temperature (*T*)	°C	Rainfall (*RF*)	mm
Ground temperature 0 cm (*GT0*)	°C	Relative humidity (*RH*)	%
Ground temperature 5 cm (*GT5*)	°C	Vapor pressure (*VP*)	hPa
Ground temperature 10 cm (*GT10*)	°C	Sea level pressure (*SLP*)	hPa
Ground temperature 20 cm (*GT20*)	°C	Solar irradiance (*LUX*)	MJ/m^2
Ground temperature 50 cm (*GT50*)	°C	Insolation duration (*LUXTIME*)	h
Ground temperature 100 cm (*GT100*)	°C	Mean wind speed (*WS*)	m/s
Dew point temperature (*DPT*)	°C	Maximum gust wind speed (*MWS*)	m/s

4.2.1. Data Preprocessing

As is typical, the environmental monitoring data were characterized by missing values and outliers, which were preprocessed using data purification and missing value processing. Since no outliers were found in the weather data samples, data purification did not change the original dataset. We observed missing values in each feature, so we removed the fields in which the missing values were located. After preprocessing and data correction, we converted all feature data from hourly to weekly dimensions. We resampled using four functions: average, maximum, minimum, and cumulative. After resampling, the amount of data for each feature was reduced from 17,000 to 105 samples. The temporal attributes of the meteorological data matched that of the pest data, and the complexity of the meteorological dataset was reduced and potential trends were identified.

4.2.2. Correlation Analysis

As shown in Table 2, we used Pearson's *r* to calculate the correlation between meteorological variables and pest populations based on agricultural sensing data from the site and pest data surveyed by agricultural experts. *Temperature*, *light*, and *wind speed* were positively correlated with pest populations, whereas *humidity*, *pressure*, and *rainfall* were negatively correlated. These correlations constitute weather indicators of pest incidence for farmers. In addition, the positive correlations of *temperature*, *light*, and *wind speed* were combined with eight input data sets to compare the performance of the predictive models.

Table 2. Pearson's *r* coefficients between meteorological variables and pest population.

Variable	*r*	Variable	*r*	Variable	*r*
T_AVG	0.426 **	GT20_AVG	0.589 ***	VP_MIN	−0.179
T_MAX	0.421 **	GT50_AVG	0.551 ***	SLP_AVG	−0.269
T_MIN	0.319 *	GT100_AVG	0.469 ***	RF_SUM	−0.107
DPT_AVG	0.281 *	RH_AVG	−0.274	LUX_SUM	0.322 *
GT0_AVG	0.571 ***	RH_MIN	−0.014	LUXTIME_SUM	0.400 **
GT5_AVG	0.575 ***	VP_AVG	−0.266	WS_AVG	0.474 ***
GT10_AVG	0.583 ***	VP_MAX	−0.280 *	WS_MAX	0.122

Note. *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$.

4.3. Data Augmentation Analysis

We investigated the data augmentation from three aspects: the parameter settings of the TimeGAN model, a visual presentation of the generated data, and an evaluation of the training effectiveness of the generated and real data.

4.3.1. TimeGAN Parameters

First, we set the parameters of the TimeGAN model. As described in Section 3.2.1, we set the size of the training data to (57, (2, 22)). We used trial and error to set *batch_size*, *num_layer*, *hidden_dim*, *noise_dim*, and *learning_rate*.

After testing and adjusting the TimeGAN parameters, the hyperparameter settings were as follows: we used a GRU generator and authenticator architecture and an LSTM

embedding and recovery network architecture; *num_layer* = 3, *hidden_dim* = 2, *noise_dim* = 32, *batch_size* = 32, *learning_rate* = 0.0005, and the number of generations (*train_epoch*) was set to 50,000.

4.3.2. Visualization of Synthetic Data

After training the TimeGAN model, we used it to generate a new serialized object (pickle file, .pkl) for each iteration. For example, a data sample of 64 samples was generated by reading the pickle file, where the size of the new dataset was (64, (2, 22)): 64 samples with 2 weeks and 22 data features (21 weather variables and 1 pest count).

The generated data were evaluated by visualizing the data distribution. First, we directly compared the generated data with the original data, where all the features were selected at the same time and a step size of 2 was used to perform the graphing. The generated TimeGAN data is represented by the basic visualization line graph shown in Figure 7, in which the solid blue line represents the real data, and the dashed orange line represents the generated data. In this case, given the multivariate nature of the trained agricultural sensing data, it was not possible to visualize all of the features and present a dataset with a step size of 1.

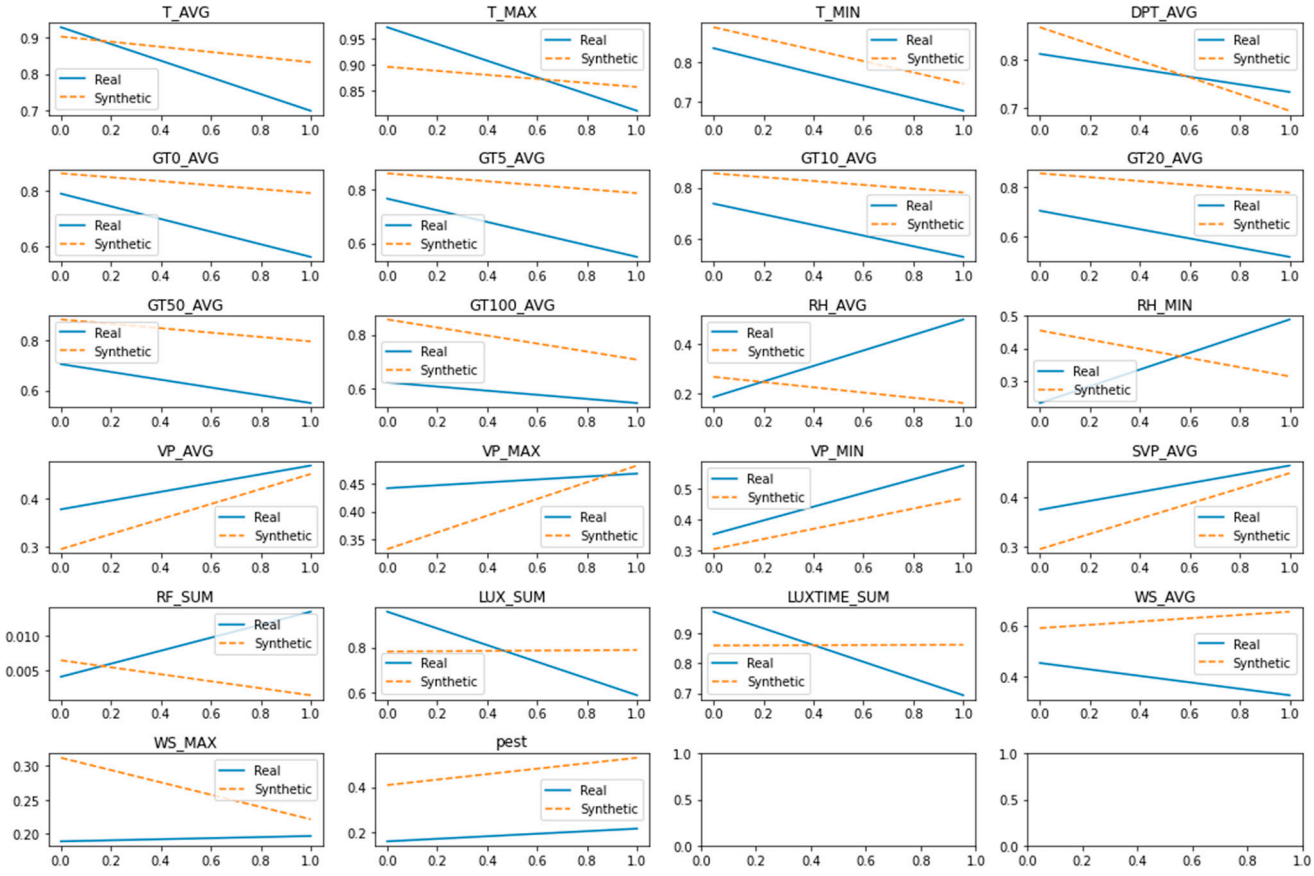

Figure 7. Basic visualization of generated data (*train_epoch* = 50,000).

The second visualization approach used the PCA and t-SNE algorithms to reduce the dimensionality of the dataset from 22 to 2. Figure 8 shows the distribution of the TimeGAN data: the left panel shows PCA results, and the right panel shows the t-SNE results. Since PCA uses linear downscaling, many features are discarded, resulting in underfitting (the distribution is not well-fitted); conversely, as t-SNE is nonlinear, the similarity between sample points in high and low dimensions is defined by chance. The black and red points in t-SNE almost overlap.

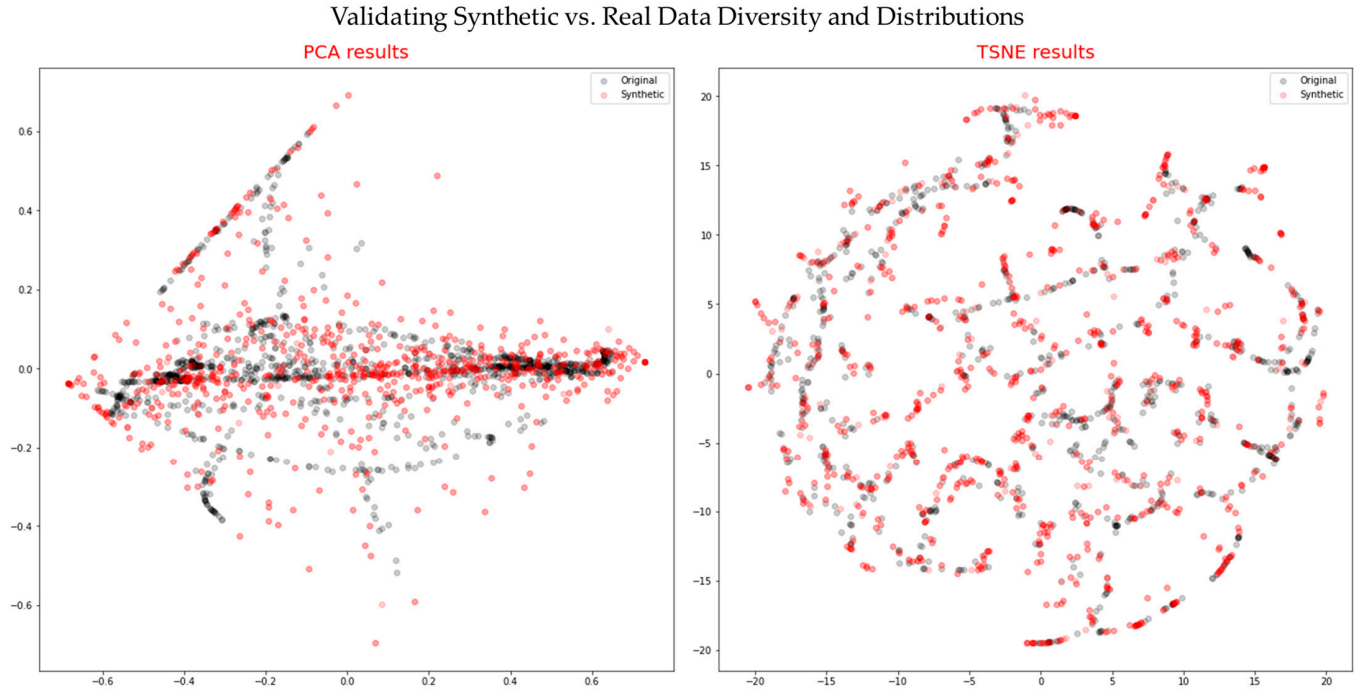

Figure 8. PCA and t-SNE visualizations of generated data (*train_epoch* = 50,000).

4.3.3. Evaluation of Synthetic Data

After training the TimeGAN model, we evaluated the generated data. We thus trained two models, one on real data and one on generated data. For this phase, the test set used real data; the evaluation metrics confirm the difference between the generated data and the model trained with real data. Table 3 shows the results of experiments using trial and error to find a good dataset cut ratio and model parameter settings based on the TSTR premise. After comparing the effectiveness of various dataset cutting ratios, we set the cut ratio between the training set and the test set to 8:2. We built a single-layer GRU to evaluate the model, where the number of neurons (*num_cell*) was 16, *batch_size* was 64, we used the Adam optimizer, and we used early stopping to prevent overfitting. The R^2, *MAE*, and *RMSE* results in Table 3 reveal a small difference between *MAE* and *RMSE*, and show that the R^2 of the model with generated data is close to that of the model with real data.

Table 3. Performance of TSTR-based models.

Metric / Type of Data	R^2	*MAE*	*RMSE*
Actual data	0.335843	0.142426	0.204147
Synthetic data	0.379916	0.141800	0.196959

4.4. Data Forecast Analysis

We divided the pest prediction experiment into four parts. First, we optimized the pest prediction model parameters. Second, based on the results of variable correlation analysis, we calculated the evaluation indicators by the prediction model under eight types of independent variables, and selected suitable independent variable data for subsequent multivariate data augmentation experiments. Third, based on the resultant prediction model and independent variable data, we calculated evaluation indicators using three prediction models, and observed the degree of fit between the augmented and original data in the model prediction. Fourth, we presented the overall pest prediction system.

4.4.1. Optimization of Predictive Models

The optimal model parameters use GRU as the base model and a one-to-one framework to design a predictive model that uses all weather variables of the current week to predict the next week's pest population. Recall that the training data accounts for 80% of the total data in this study. GRU model parameters including *batch_size*, *num_cell*, *num_hidden* layer, and dropout were adjusted sequentially by scaling the *RMSE* of the unreturned data to the same evaluation metric to optimize pest prediction, thus optimizing the parameters and structure of the model. Figure 9 shows the *RMSE* curves for optimizing the parameters of the GRU pest prediction model.

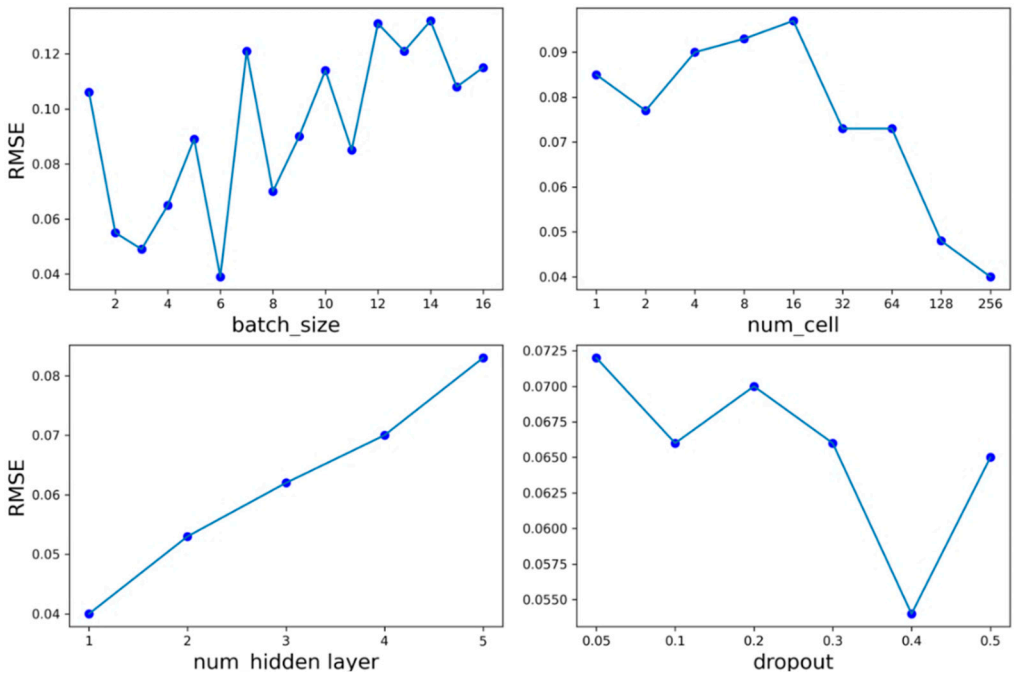

Figure 9. Optimization of model parameter curves.

In Figure 9, the optimal *batch_size* for prediction model training is 6, *num_cell* of GRU is 256, GRU cells are not stacked, the dropout of cells is 0.4, the training model optimizer was RMSprop, and the number of epochs is controlled by early stopping to avoid overfitting. The parameters of the pest prediction model were optimally combined with those of the GRU model, and the RNN and LSTM models used the same parameters and architecture. In addition, we used the Adam optimizer for RNN.

4.4.2. Selection of Input Variables

We selected three sets of positively correlated variables (*temperature*, *light*, and *wind speed*) and combined them to classify the pest prediction independent variables into eight types of data. We used three models (RNN, LSTM, and GRU) to evaluate and predict pest populations. We selected the best-performing independent variables for subsequent experiments to augment the independent variable data. Figure 10 shows the results of pest prediction using the three models. The black dashed line represents the real data, and the green solid line, red solid line, and blue solid line are the predicted values of the RNN model, the LSTM model, and the GRU model, respectively. The y-axis unit is the unnormalized pest counts. Table 4 shows the evaluation results of the eight types of independent variables using the three models.

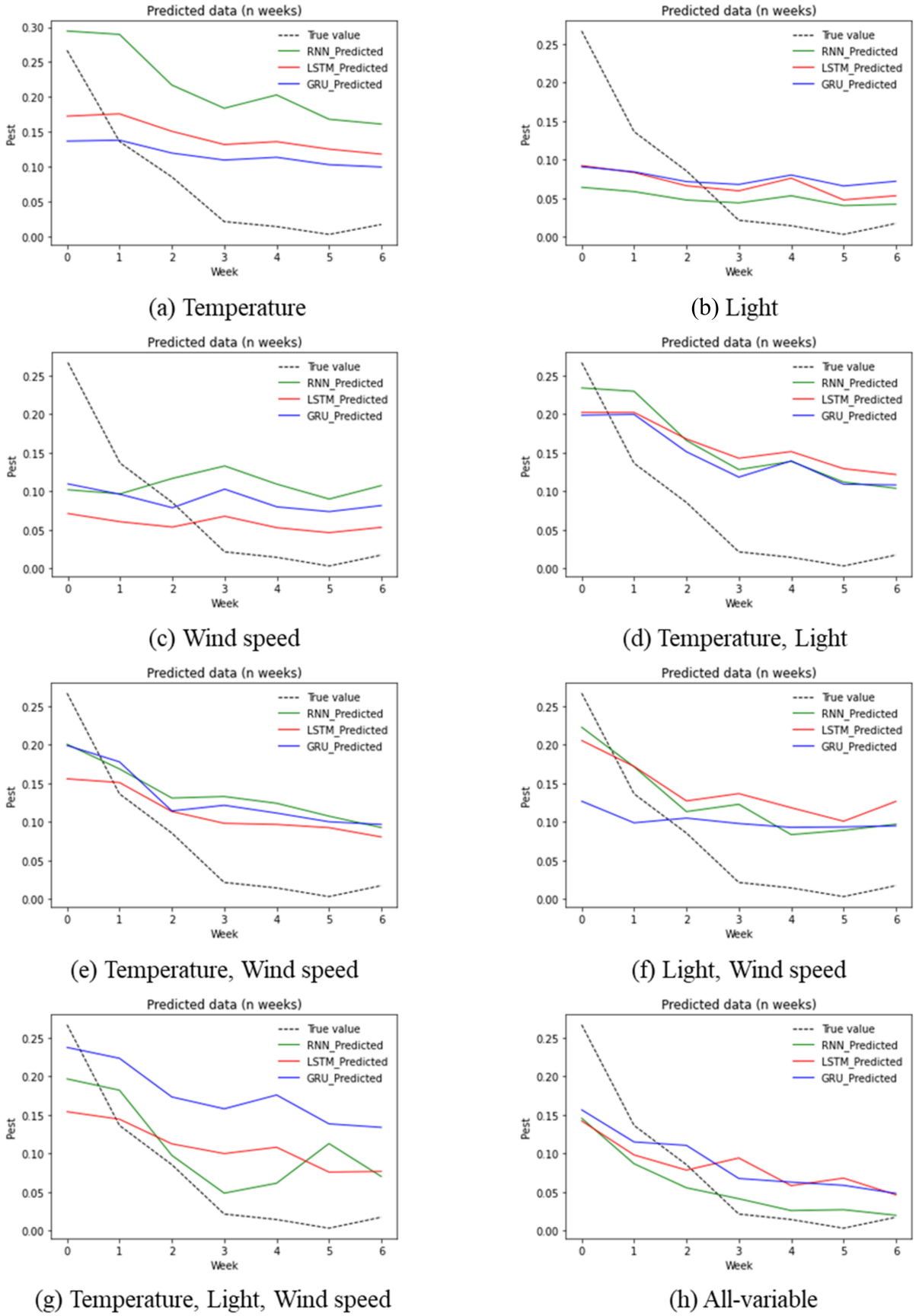

Figure 10. Weather feature prediction results.

Table 4. Results for weather feature datasets.

Input Factor	Model	*MAE*	*MAPE*	*RMSE*
Temperature	RNN	153.12	11.89%	162.07
	LSTM	103.34	8.42%	107.98
	GRU	84.80	6.84%	95.82
Light	RNN	70.12	2.66%	95.52
	LSTM	67.85	3.34%	86.03
	GRU	74.78	4.39%	91.00
Wind Speed	RNN	97.78	6.48%	107.96
	LSTM	74.08	3.15%	95.23
	GRU	77.00	5.00%	90.13
Temperature, *Light*	RNN	100.03	7.76%	104.43
	LSTM	110.81	8.90%	114.84
	GRU	97.38	7.58%	100.25
Temperature, *Wind Speed*	RNN	86.23	7.25%	92.17
	LSTM	73.77	5.96%	81.63
	GRU	81.02	6.74%	86.11
Light, *Wind Speed*	RNN	70.32	5.96%	75.87
	LSTM	89.35	7.20%	95.69
	GRU	82.32	6.11%	91.29
Temperature, *Light*, *Wind Speed*	RNN	58.04	6.09%	66.36
	LSTM	71.71	5.30%	80.87
	GRU	118.94	9.76%	127.04
All Variables	RNN	41.47	1.51%	58.27
	LSTM	60.52	4.17%	71.87
	GRU	53.85	3.65%	62.07

In Figure 10a–c, we observe a single combination of independent variables (*temperature*, *light*, and *wind speed*) where the predicted values curve smoothly and differ more from the actual values; in Figure 10d–g, the combined meteorological variables are predicted with a larger magnitude and gradually approximate the real data curve, but still produce large errors at some moments; in Figure 10h, when training with the agricultural sensing data that covers all variables, the predicted trend curve is clearly closer to the real line than the other seven results. We thus infer that the complete data is more comprehensive except for the variables, and the combination of positive correlation variables does not reduce the prediction error. Therefore, for the data augmentation experiment we use the complete multivariate agricultural sensor data as input values and compare the training set with the unaugmented and augmented improvement.

Note in Table 4 that for models without augmentation (RNN, LSTM, and GRU), the *MAPE* values are significantly smaller than those for the *temperature* features when *light* and *wind speed* are the independent variables, but the predicted curves in Figure 10b,c do not approximate the real data, indicating that a single variable does not contain enough information to fit real-world situations. For [*temperature*, *light*], [*temperature*, *wind speed*], [*light*, *wind speed*], and [*temperature*, *light*, *wind speed*], the *MAPE* values are slightly higher. Likewise, due to the additional data features, the results in Figure 10d–g more closely approximate the true values. When the independent variable is [*all variables*], the *MAPE* results for RNN, LSTM, and GRU are 1.51%, 4.17%, and 3.65%, respectively, which are the lowest values in all cases, and the *MAE* and *RMSE* values are the lowest among all combinations. Thus, using all variables yields the most complete characteristics for model learning; note that both predictions are close to the real situation in Figure 10h. Consequently, we generate data using all variables and compare the prediction errors before and after augmentation in RNN, LSTM, and GRU.

4.4.3. Comparison of Predictive Metrics

We independently tested the prediction effects of different independent variables and selected all variables as the original augmented input data; thus, we used the all of the multivariate agricultural sensing data, and used RNN, LSTM, and GRU to compare the six results in terms of the prediction error of the two data sets (*MAE*) and the error difference (*EC*). Table 5 shows the performance of the time-series prediction model using different data.

Table 5. Prediction error for different training data.

Model	Training Data			Test Data		
	$MAE_{w/o\ aug}$	$MAE_{w/\ aug}$	*EC*	$MAE_{w/o\ aug}$	$MAE_{w/\ aug}$	*EC*
RNN	13.05	91.91	78.86	79.86	120.39	40.53
LSTM	19.62	76.67	57.05	73.29	107.34	34.05
GRU	17.43	85.34	67.91	77.67	87.53	9.86

In Table 5, when the original and augmented data are used for training, the *MAE* values of the RNN model are 13.05 and 91.91 and the *EC* value is 78.86. The LSTM model trained with augmented data has the lowest prediction error, and its *MAE* value of 76.67 is the lowest of the three models. The difference between the prediction errors calculated from the real and simulated data using the LSTM model is also the smallest, and the *EC* value of 57.05 is the best of the three models.

In addition, as shown in Table 5, the *MAE* values of the RNN model are 79.86 and 120.39 and the *EC* value is 40.53 when the original and augmented data is used for the test data. The *MAE* values of the LSTM model are 73.29 and 107.34 and the *EC* value is 34.05. The *MAE* values of the GRU model are 77.67 and 87.53 and the *EC* value is 9.86. The prediction error of the GRU model trained with the augmented data is the lowest, and the *MAE* value of 87.53 is the lowest of the three models. The difference between the prediction error of real and simulated data using the GRU model is also the smallest, and the *EC* value of 9.86 is the lowest of the three models.

4.4.4. Overall Pest Prediction System

The final objective of this study was to present a multivariate pest prediction system for agricultural sensing data. After optimizing the prediction model parameters, we selected the independent variables for pest prediction, used these independent variables to generate data, and then used the generated data together with the original data as the training set for the prediction models. After the initial optimization of the pest prediction model parameters, we constructed six pest prediction models after the model overlay training: the original RNN model, the original LSTM model, the original GRU model, an RNN model trained with generated data, an LSTM model trained with generated data, and a GRU model trained with generated data. Figure 11 shows the prediction results on the training set using the six pretrained models. Figure 12 shows the prediction results of the pretrained models on the test set. The x-axis unit is weeks, and the y-axis unit is the number of pests that were restored without data normalization.

In Figures 11 and 12, the black dashed line represents the real data, the green solid line represents the original RNN model, the green dashed line represents the RNN model trained with the generated data, the red solid line represents the original LSTM model, the red dashed line represents the LSTM model trained with the generated data, the blue solid line represents the original GRU model, and the blue dashed line represents the GRU model trained with the generated data.

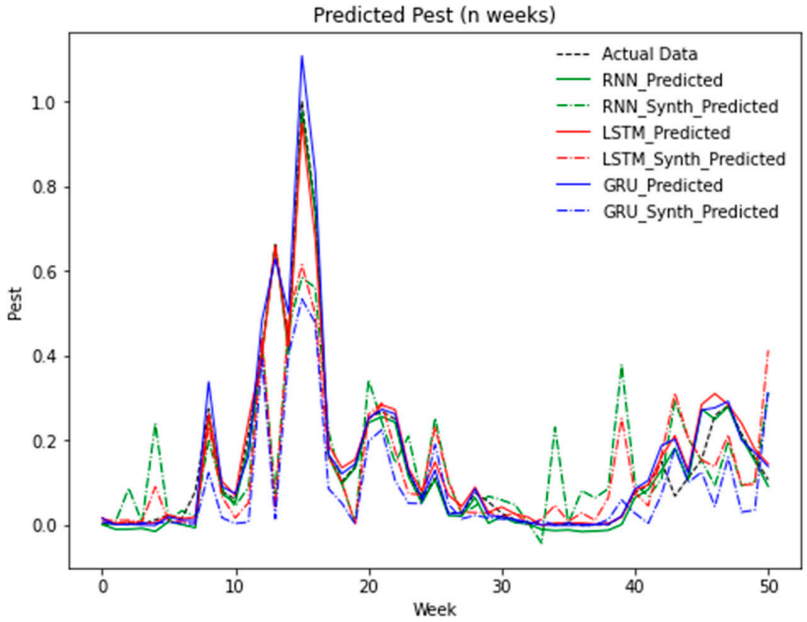

Figure 11. Pest prediction on training set.

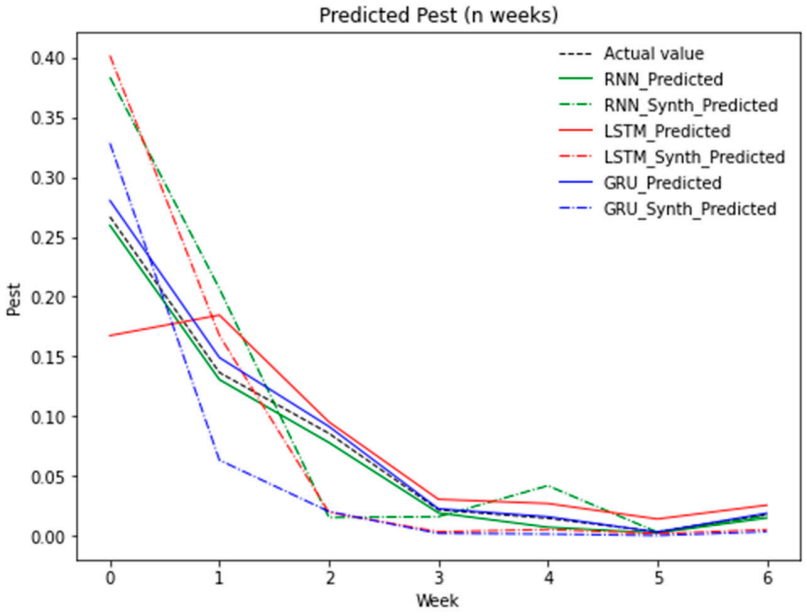

Figure 12. Pest prediction on test set.

5. Conclusions and Prospects

In Section 4.2.1, we use downsampling to convert the raw data into more weather characteristics and then use the prediction model to learn more complete weather trends. In addition, to highlight which combination of weather factors is most beneficial for the task, we analyze eight combinations of independent variables on pest trends by evaluating the results and visualization predictions in Section 4.4.2: when we use all of the variables as the independent variables for the prediction model, the *MAE*, *MAPE*, and *RMSE* are best compared to the other combinations, and the prediction curves most closely approximate the real data. In Section 4.4.1, we further optimize the model parameter configuration and architecture of the predictive model in terms of the same evaluation metrics. Finally, in Sections 4.4.2–4.4.4, we compare and collate the differences in model effectiveness between the optimized models by considering the effects of multiple models simultaneously in different pest prediction experiments.

Data augmentation has not yet been applied to agricultural sensing data. Nevertheless, as the agricultural sensing data is in the form of a time series, they are suited for data augmentation in the form of data generation of sensing data. We generate the original time-series data and use this data for pest trend prediction. The experimental results in Section 4.3.3 show that the generated data can be used to simulate real situations in the absence of sufficient real data. Moreover, in Sections 4.4.3 and 4.4.4, we conduct experiments to evaluate the effect of the generated data on pest prediction. Finally, in Section 4.3.2, we visualize the augmented sensing data and graphically represent the differences between the data via dimension reduction. The t-SNE algorithm shows that the markers of the generated data are generally the same as those for the real data.

Currently, most of the applications of Generative Adversarial Networks (GANs) in agriculture are focused on image generation. To improve the usability of generated data, many variations of GANs have emerged, such as conditional GANs and CycleGANs. While these applications have the potential to enhance various aspects of agriculture, there is a growing emphasis on sustainable agriculture practices that prioritize the long-term health of the environment and ecosystems. TimeGAN, which is mainly used for generating time-series data, is highly suitable for sustainable agriculture applications. It can help generate data that inform sustainable agricultural practices, such as optimizing crop yields, improving resource efficiency, and reducing the environmental impact of agricultural activities. To the best of our knowledge, there has not yet been any research on generating time-series data in this field, but the potential for TimeGAN to contribute to sustainable agriculture practices is promising.

In summary, we address the fundamental problem of insufficient data for pest prediction through time-series data augmentation. We use three common temporal prediction models to evaluate the differences between a model trained on large amounts of generated data and a model trained on raw data. The results show that complex LSTM and GRU models produce prediction results that are similar to those produced by models trained on the raw data. For future work, we suggest using a more complete dataset to predict pest occurrences. In addition, the scaling method proposed in this study compares only the prediction difference between the original data and the generated data, and does not consider the scaling principle of other time-series data. Future studies could investigate the generation effects of different augmentation methods and the improvement in pest prediction results due to augmentation methods such as interpolation, which is commonly used to augment linear problems.

Author Contributions: Conceptualization, Y.-M.H. and C.-Y.T.; methodology, C.-Y.T.; software, C.-Y.T.; validation, C.-Y.T.; investigation, W.-J.W.; data curation, W.-J.W.; writing—original draft preparation, C.-Y.T.; writing—review and editing, Y.-M.H.; visualization, C.-Y.T.; supervision, Y.-M.H.; project administration, Y.-M.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available in a publicly accessible repository that does not issue DOIs. Publicly available datasets were analyzed in this study. This data can be found here: [https://agr.cwb.gov.tw], [https://data.kcg.gov.tw/organization/3ff96cb3-b16d-4612-98cf-88e63aa6a012?tags=%E5%B0%8F%E9%BB%83%E8%96%8A%E9%A6%AC]. All accessed on 15 June 2022.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Reganold, J.P.; Papendick, R.I.; Parr, J.F. Sustainable agriculture. *Sci. Am.* **1990**, *262*, 112–121. [CrossRef]
2. Harwood, R.R. A history of sustainable agriculture. In *Sustainable Agricultural Systems*; CRC Press: Boca Raton, FL, USA, 2020; pp. 3–19.

3. Qazi, S.; Khawaja, B.A.; Farooq, Q.U. IoT-equipped and AI-enabled next generation smart agriculture: A critical review, current challenges and future trends. *IEEE Access* **2022**, *10*, 21219–21235. [CrossRef]
4. Jamil, F.; Ibrahim, M.; Ullah, I.; Kim, S.; Kahng, H.K.; Kim, D.-H. Optimal smart contract for autonomous greenhouse environment based on IoT blockchain network in agriculture. *Comput. Electron. Agric.* **2021**, *192*, 106573. [CrossRef]
5. Tian, E.; Li, Z.; Huang, W.; Ma, H. Distributed and Parallel simulation methods for pest control and crop monitoring with IoT assistance. *Acta Agric. Scand. Sect. B Soil Plant Sci.* **2021**, *71*, 884–898. [CrossRef]
6. van Klompenburg, T.; Kassahun, A.; Catal, C. Crop yield prediction using machine learning: A systematic literature review. *Comput. Electron. Agric.* **2020**, *177*, 105709. [CrossRef]
7. García, L.; Parra, L.; Jimenez, J.M.; Lloret, J.; Lorenz, P. IoT-Based Smart Irrigation Systems: An Overview on the Recent Trends on Sensors and IoT Systems for Irrigation in Precision Agriculture. *Sensors* **2020**, *20*, 1042. [CrossRef] [PubMed]
8. Boudwin, R.; Magarey, R.; Jess, L. Integrated Pest Management Data for Regulation, Research, and Education: Crop Profiles and Pest Management Strategic Plans. *J. Integr. Pest Manag.* **2022**, *13*, 13. [CrossRef]
9. Ibrahim, E.A.; Salifu, D.; Mwalili, S.; Dubois, T.; Collins, R.; Tonnang, H.E. An expert system for insect pest population dynamics prediction. *Comput. Electron. Agric.* **2022**, *198*, 107124. [CrossRef]
10. Bradhurst, R.; Spring, D.; Stanaway, M.; Milner, J.; Kompas, T. A generalised and scalable framework for modelling incursions, surveillance and control of plant and environmental pests. *Environ. Model. Softw.* **2021**, *139*, 105004. [CrossRef]
11. Aharoni, R.; Klymiuk, V.; Sarusi, B.; Young, S.; Fahima, T.; Fishbain, B.; Kendler, S. Spectral light-reflection data dimensionality reduction for timely detection of yellow rust. *Precis. Agric.* **2020**, *22*, 267–286. [CrossRef]
12. Sasikala, V.; Venkatesan, G. Time Variant Multi Feature Census Analysis for Efficient Prediction of Migration Risks in Agriculture. *J. Comput. Theor. Nanosci.* **2020**, *17*, 5323–5333. [CrossRef]
13. Morais, R.; Mendes, J.; Silva, R.; Silva, N.; Sousa, J.J.; Peres, E. A Versatile, Low-Power and Low-Cost IoT Device for Field Data Gathering in Precision Agriculture Practices. *Agriculture* **2021**, *11*, 619. [CrossRef]
14. Ayyoubzadeh, S.M.; Zahedi, H.; Ahmadi, M.; Kalhori, S.R.N. Predicting COVID-19 Incidence Through Analysis of Google Trends Data in Iran: Data Mining and Deep Learning Pilot Study. *JMIR Public Health Surveill.* **2020**, *6*, e18828. [CrossRef] [PubMed]
15. Su, D.; Kong, H.; Qiao, Y.; Sukkarieh, S. Data augmentation for deep learning based semantic segmentation and crop-weed classification in agricultural robotics. *Comput. Electron. Agric.* **2021**, *190*, 106418. [CrossRef]
16. Singhal, A.; Cheriyamparambil, A.; Jha, S.K. Spatial extrapolation of statistically downscaled weather data over the Northwest Himalayas at major glacier sites. *Environ. Model. Softw.* **2022**, *149*, 105317. [CrossRef]
17. Zhang, Y.; Thorburn, P.J. Handling missing data in near real-time environmental monitoring: A system and a review of selected methods. *Futur. Gener. Comput. Syst.* **2021**, *128*, 63–72. [CrossRef]
18. Yoon, J.; Jarrett, D.; Van der Schaar, M. Time-series generative adversarial networks. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 5508–5518.
19. Dupuis, A.; Dadouchi, C.; Agard, B. Predicting crop rotations using process mining techniques and Markov principals. *Comput. Electron. Agric.* **2022**, *194*, 106686. [CrossRef]
20. Bhat, S.A.; Huang, N.-F. Big Data and AI Revolution in Precision Agriculture: Survey and Challenges. *IEEE Access* **2021**, *9*, 110209–110222. [CrossRef]
21. Husin, N.A.; Khairunniza-Bejo, S.; Abdullah, A.F.; Kassim, M.S.M.; Ahmad, D. Multi-temporal analysis of terrestrial laser scanning data to detect basal stem rot in oil palm trees. *Precis. Agric.* **2021**, *23*, 101–126. [CrossRef]
22. Du, X.; Elbakidze, L.; Lu, L.; Taylor, R.G. Climate Smart Pest Management. *Sustainability* **2022**, *14*, 9832. [CrossRef]
23. Ren, C.; Kim, D.-K.; Jeong, D. A survey of deep learning in agriculture: Techniques and their applications. *J. Inf. Process. Syst.* **2020**, *16*, 1015–1033.
24. Xu, L.; Zhang, H.; Wang, C.; Zhang, B.; Liu, M. Crop Classification Based on Temporal Information Using Sentinel-1 SAR Time-Series Data. *Remote Sens.* **2018**, *11*, 53. [CrossRef]
25. Htitiou, A.; Boudhar, A.; Lebrini, Y.; Lionboui, H.; Chehbouni, A.; Benabdelouahab, T. Classification and status monitoring of agricultural crops in central Morocco: A synergistic combination of OBIA approach and fused Landsat-Sentinel-2 data. *J. Appl. Remote Sens.* **2021**, *15*, 014504. [CrossRef]
26. Rembold, F.; Meroni, M.; Urbano, F.; Csak, G.; Kerdiles, H.; Perez-Hoyos, A.; Lemoine, G.; Leo, O.; Negre, T. ASAP: A new global early warning system to detect anomaly hot spots of agricultural production for food security analysis. *Agric. Syst.* **2018**, *168*, 247–257. [CrossRef] [PubMed]
27. Filippi, P.; Jones, E.J.; Wimalathunge, N.S.; Somarathna, P.D.S.N.; Pozza, L.E.; Ugbaje, S.U.; Jephcott, T.G.; Paterson, S.E.; Whelan, B.M. An approach to forecast grain crop yield using multi-layered, multi-farm data sets and machine learning. *Precis. Agric.* **2019**, *20*, 1015–1029. [CrossRef]
28. Dong, Y.; Xu, F.; Liu, L.; Du, X.; Ren, B.; Guo, A.; Geng, Y.; Ruan, C.; Ye, H.; Huang, W.; et al. Automatic System for Crop Pest and Disease Dynamic Monitoring and Early Forecasting. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 4410–4418. [CrossRef]
29. van Mourik, S.; van der Tol, R.; Linker, R.; Reyes-Lastiri, D.; Kootstra, G.; Koerkamp, P.G.; van Henten, E.J. Introductory overview: Systems and control methods for operational management support in agricultural production systems. *Environ. Model. Softw.* **2021**, *139*, 105031. [CrossRef]
30. Sharma, R. Artificial Intelligence in Agriculture: A Review. In Proceedings of the 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 6–8 May 2021; pp. 937–942.

31. Heeb, L.; Jenner, E.; Cock, M.J.W. Climate-smart pest management: Building resilience of farms and landscapes to changing pest threats. *J. Pest Sci.* **2019**, *92*, 951–969. [CrossRef]
32. Meshram, V.; Patil, K.; Meshram, V.; Hanchate, D.; Ramkteke, S. Machine learning in agriculture domain: A state-of-art survey. *Artif. Intell. Life Sci.* **2021**, *1*, 100010. [CrossRef]
33. Issad, H.A.; Aoudjit, R.; Rodrigues, J. A comprehensive review of Data Mining techniques in smart agriculture. *Eng. Agric. Environ. Food* **2019**, *12*, 511–525. [CrossRef]
34. Yang, J.; Ma, S.; Li, Y.; Zhang, Z. Efficient Data-Driven Crop Pest Identification Based on Edge Distance-Entropy for Sustainable Agriculture. *Sustainability* **2022**, *14*, 7825. [CrossRef]
35. Chen, J.-W.; Lin, W.-J.; Cheng, H.-J.; Hung, C.-L.; Lin, C.-Y.; Chen, S.-P. A Smartphone-Based Application for Scale Pest Detection Using Multiple-Object Detection Methods. *Electronics* **2021**, *10*, 372. [CrossRef]
36. Liu, L.; Wang, R.; Xie, C.; Yang, P.; Wang, F.; Sudirman, S.; Liu, W. PestNet: An End-to-End Deep Learning Approach for Large-Scale Multi-Class Pest Detection and Classification. *IEEE Access* **2019**, *7*, 45301–45312. [CrossRef]
37. Roosjen, P.P.; Kellenberger, B.; Kooistra, L.; Green, D.R.; Fahrentrapp, J. Deep learning for automated detection of *Drosophila suzukii*: Potential for UAV -based monitoring. *Pest Manag. Sci.* **2020**, *76*, 2994–3002. [CrossRef] [PubMed]
38. Yu, R.; Luo, Y.; Zhou, Q.; Zhang, X.; Wu, D.; Ren, L. A machine learning algorithm to detect pine wilt disease using UAV-based hyperspectral imagery and LiDAR data at the tree level. *Int. J. Appl. Earth Obs. Geoinf.* **2021**, *101*, 102363. [CrossRef]
39. Mekonnen, Y.; Namuduri, S.; Burton, L.; Sarwat, A.; Bhansali, S. Review—Machine Learning Techniques in Wireless Sensor Network Based Precision Agriculture. *J. Electrochem. Soc.* **2019**, *167*, 037522. [CrossRef]
40. Materne, N.; Inoue, M. IoT Monitoring System for Early Detection of Agricultural Pests and Diseases. In Proceedings of the 2018 12th South East Asian Technical University Consortium (SEATUC), Yogyakarta, Indonesia, 12–13 March 2018; pp. 1–5. [CrossRef]
41. Bourhis, Y.; Bell, J.R.; Bosch, F.V.D.; Milne, A.E. Artificial neural networks for monitoring network optimisation—A practical example using a national insect survey. *Environ. Model. Softw.* **2020**, *135*, 104925. [CrossRef]
42. Chen, P.; Xiao, Q.; Zhang, J.; Xie, C.; Wang, B. Occurrence prediction of cotton pests and diseases by bidirectional long short-term memory networks with climate and atmosphere circulation. *Comput. Electron. Agric.* **2020**, *176*, 105612. [CrossRef]
43. Saleem, M.H.; Potgieter, J.; Arif, K.M. Automation in agriculture by machine and deep learning techniques: A review of recent developments. *Precis. Agric.* **2021**, *22*, 2053–2091. [CrossRef]
44. Jiang, J.-A.; Syue, C.-H.; Wang, C.-H.; Liao, M.-S.; Shieh, J.-S.; Wang, J.-C. Precisely forecasting population dynamics of agricultural pests based on an interval type-2 fuzzy logic system: Case study for oriental fruit flies and the tobacco cutworms. *Precis. Agric.* **2022**, *23*, 1302–1332. [CrossRef]
45. Zhang, J.; Huang, Y.; Pu, R.; Gonzalez-Moreno, P.; Yuan, L.; Wu, K.; Huang, W. Monitoring plant diseases and pests through remote sensing technology: A review. *Comput. Electron. Agric.* **2019**, *165*, 104943. [CrossRef]
46. Narava, R.; Sai Ram Kumar, D.V.; Jaba, J.; Anil Kumar, P.; Ranga Rao, G.V.; Srinivasa Rao, V.; Mishra, S.P.; Kukanur, V. Development of Temporal Model for Forecasting of *Helicoverpa armigera* (Noctuidae: Lepidopetra) Using Arima and Artificial Neural Networks. *J. Insect Sci.* **2022**, *22*, 2. [CrossRef]
47. Gómez, D.; Salvador, P.; Sanz, J.; Rodrigo, J.F.; Gil, J.; Casanova, J.L. Prediction of desert locust breeding areas using machine learning methods and SMOS (MIR_SMNRT2) Near Real Time product. *J. Arid. Environ.* **2021**, *194*, 104599. [CrossRef]
48. Tan, S.; Liang, Y.; Zheng, R.; Yuan, H.; Zhang, Z.; Long, C. Dynamic Prediction of *Chilo suppressalis* Occurrence in Rice Based on Deep Learning. *Processes* **2021**, *9*, 2166. [CrossRef]
49. Delgado, J.M.D.; Oyedele, L. Deep learning with small datasets: Using autoencoders to address limited datasets in construction management. *Appl. Soft Comput.* **2021**, *112*, 107836. [CrossRef]
50. Wen, Q.; Sun, L.; Yang, F.; Song, X.; Gao, J.; Wang, X.; Xu, H. Time Series Data Augmentation for Deep Learning: A Survey. *arXiv* **2020**, arXiv:2002.12478.
51. Barunik, J.; Krehlik, T.; Vacha, L. Modeling and forecasting exchange rate volatility in time-frequency domain. *Eur. J. Oper. Res.* **2016**, *251*, 329–340. [CrossRef]
52. Um, T.T.; Pfister, F.M.J.; Pichler, D.; Endo, S.; Lang, M.; Hirche, S.; Fietzek, U.; Kulić, D. Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In Proceedings of the 19th ACM International Conference on Multimodal Interaction, Glasgow, UK, 13–17 November 2017; pp. 216–220.
53. Dumas, J.; Wehenkel, A.; Lanaspeze, D.; Cornélusse, B.; Sutera, A. A deep generative model for probabilistic energy forecasting in power systems: Normalizing flows. *Appl. Energy* **2022**, *305*, 117871. [CrossRef]
54. Asimopoulos, D.C.; Nitsiou, M.; Lazaridis, L.; Fragulis, G.F. Generative Adversarial Networks: A systematic review and applications. *SHS Web Conf.* **2022**, *139*, 03012. [CrossRef]
55. Agricultural Meteorological Observation Network Monitoring System. Available online: https://agr.cwb.gov.tw/ (accessed on 15 June 2022).
56. Government. Active Investigation and Monitoring of Epidemic Diseases and Insect Pests in Kaohsiung City—Information Table of the Number of Scirtothrips dorsalis. Available online: https://data.kcg.gov.tw/organization/3ff96cb3-b16d-4612-98cf-88e63aa6a012?tags=%E5%B0%8F%E9%BB%83%E8%96%8A%E9%A6%AC (accessed on 15 June 2022).
57. YData. YData-Synthetic. Available online: https://github.com/ydataai/ydata-synthetic (accessed on 15 June 2022).

58. Jordon, J.; Yoon, J.; Van Der Schaar, M. PATE-GAN: Generating Synthetic Data with Differential Privacy Guarantees. In International Conference on Learning Representations. 2018. Available online: https://openreview.net/forum?id=S1zk9iRqF7 (accessed on 15 June 2022).
59. Anowar, F.; Sadaoui, S.; Selim, B. Conceptual and empirical comparison of dimensionality reduction algorithms (pca, kpca, lda, mds, svd, lle, isomap, le, ica, t-sne). *Comput. Sci. Rev.* **2021**, *40*, 100378. [CrossRef]
60. Chicco, D.; Warrens, M.J.; Jurman, G. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Comput. Sci.* **2021**, *7*, e623. [CrossRef] [PubMed]
61. Government. Kaohsiung City Agricultural Statistics Annual Report in 110 Years of the Republic of China. Available online: https://agri.kcg.gov.tw/FileDownLoad/FileUpload/20220502105956424644.pdf (accessed on 15 June 2022).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.