A Novel Combined Model for Predicting Humidity in Sheep Housing Facilities

Dachun Feng 1,2,3,4,5,†, Bing Zhou 1,2,3,4,5,†, Qianyu Han 1,2,3,4,5, Longqin Xu 1,2,3,4,5, Jianjun Guo 1,2,3,4,5, Liang Cao 1,2,3,4,5, Lvhan Zhuang 1,2,3,4,5, Shuangyin Liu 1,2,3,4,5,* and Tonglai Liu 1,2,3,4,5,*

1 College of Information Science and Technology, Zhongkai University of Agriculture and Engineering, Guangzhou 510225, China
2 Smart Agriculture Engineering Technology Research Center of Guangdong Higher Education Institutes, Zhongkai University of Agriculture and Engineering, Guangzhou 510225, China
3 Guangzhou Key Laboratory of Agricultural Products Quality & Safety Traceability Information Technology, Zhongkai University of Agriculture and Engineering, Guangzhou 510225, China
4 Academy of Smart Agricultural Engineering Innovations, Zhongkai University of Agriculture and Engineering, Guangzhou 510225, China
5 Guangdong Province Key Laboratory of Waterfowl Healthy Breeding, Guangzhou 510225, China

* Correspondence: shuangyinliu@zhku.edu.cn (S.L.); tonglailiu@zhku.edu.cn (T.L.)
† These authors contributed equally to this work.

Simple Summary: The hybrid model is proposed to predict humidity in sheep barns, based on a machine learning model combining a light gradient boosting machine with gray wolf optimization and support-vector regression (LightGBM–CGWO–SVR). Influencing factors with a high contribution to humidity were extracted using LightGBM to reduce the complexity of the model, and required hyperparameters in SVR were optimized, adopting the CGWO algorithm to avoid the local extremum problem. The experimental results indicated that the proposed LightGBM–CGWO–SVR model outperformed eight existing models used for comparison on all evaluation metrics; it achieved minimum values of 0.0662, 0.2284, 0.0521, and 0.0083, in terms of MAE, RMSE, MSE, and NRMSE, respectively, and a maximum value of 0.9973, in terms of the R² index.

Abstract: Accurately predicting humidity changes in sheep barns is important to ensure the healthy growth of the animals and to improve the economic returns of sheep farming. In this study, to address the limitations of conventional methods in establishing accurate mathematical models of dynamic changes in humidity in sheep barns, we propose a method to predict humidity in sheep barns based on a machine learning model combining a light gradient boosting machine with gray wolf optimization and support-vector regression (LightGBM–CGWO–SVR). Influencing factors with a high contribution to humidity were extracted using LightGBM to reduce the complexity of the model. To avoid the local extremum problem, the CGWO algorithm was used to optimize the required hyperparameters in SVR and determine the optimal hyperparameter combination. The combined algorithm was applied to predict the humidity of an intensive sheep-breeding facility in Manas, Xinjiang, China, in real time for the next 10 min. The experimental results indicated that the proposed LightGBM–CGWO–SVR model outperformed eight existing models used for comparison on all evaluation metrics. It achieved minimum values of 0.0662, 0.2284, 0.0521, and 0.0083 in terms of mean absolute error, root mean square error, mean squared error, and normalized root mean square error, respectively, and a maximum value of 0.9973 in terms of the R² index.

Keywords: sheep barn; humidity; SVR algorithm; hybrid prediction model

1. Introduction
Sheep farming for meat is a leading industry in the livestock economy of rural inland Northwest China (i.e., Xinjiang), and large-scale intensive farming is the primary mode
of production [1]. However, while large-scale farming reduces farming costs, it does involve problems with the physical welfare of the animals, primarily because sheep barns often exhibit high humidity, which tends to breed a great deal of molds and parasites that affect their physiological functions (e.g., reproduction and metabolism) and may lead to various diseases and health problems [2]. Therefore, predicting humidity in sheep barns can provide a theoretical foundation for regulating the barn environment. Commonly used mechanistic prediction models suffer from defects caused by large errors and low accuracy in predicting complex data [3], especially for such humidity data, which typically exhibit complex nonlinear and multi-coupling characteristics, as well as a substantial dynamic time lag [4,5]. Data-driven modeling techniques, based on machine learning and enhanced artificial intelligence, can effectively address these problems, and have been widely discussed in the field of humidity prediction based on time-series data.

As a typical machine learning method, the linear method was discussed earlier in the prediction of pig house environment and humidity. Banhazi et al. [6,7] used a general linear model (GLM) to construct predictive models of environmental factors in pig barns and, subsequently, improved the statistical model using the “leave-one-out” cross-validation technique, which proved helpful for managing pig barn environments. Daskalov [8] used the dynamic discrete autoregressive moving average (ARMA) algorithm to construct an environmental humidity model for pig barns. Even though the linear prediction model was relatively simple, the prediction accuracy of the model was not satisfactory. Other single machine learning methods are often used for univariate rolling prediction of air humidity, for example, Yang et al. [9] developed an air humidity model based on the extended Fourier series model in the least squares method, which overcame the low prediction accuracy of traditional Fourier models. Qadeer et al. [10] proposed the use of a random forest model to predict air humidity, and their prediction results proved that random forest models could accurately predict highly nonlinear data. Although the single variable time series prediction models have high prediction accuracy, the model input parameters do not consider the influence factors of air humidity, which are difficult to apply to the prediction of humidity in livestock houses. In recent years, in addition to traditional machine learning models, the research community has focused on prediction models based on neural networks. Hongkang et al. [11] constructed a dynamic backpropagation neural network (BPNN) model to predict the humidity of a solar greenhouse in northern China, based on recurrent neural networks (RNNs), and took five parameters, such as enumeration and CO₂ concentration, as input variables. Zou et al. [12] used a convex bidirectional extreme machine-learning-based model to predict the humidity of a solar greenhouse. Wind speed, solar radiation, indoor and outdoor temperature and humidity were used as input variables in the model. Jung et al. [13] used three neural-network-based algorithms, i.e., an artificial neural network (ANN), a nonlinear autoregressive exogenous model (NARX), and a recurrent neural network long short-term memory (RNN-LSTM) to construct greenhouse prediction models for temperature, humidity, and CO₂; however, the prediction accuracy for humidity was very low for all models. Although these studies on the prediction of greenhouse humidity, based on neural network, provide some convenience for production management, they also have the disadvantage of low prediction accuracy. Besides unsatisfactory prediction accuracy, the above single prediction methods, based on data drive, face other issues, such as poor robustness, the complexity of constructing the models, and difficult selection of neural network parameters in the actual prediction process. In particular, environmental data collected by sensors from large-scale sheep breeding bases are characterized by strong coupling, high dimensionality, nonlinearity, and a large time lag. As a single prediction model considers all the parameters, this approach leads to a complex model structure and poor generalization performance, and, thus, limits the final prediction accuracy. Therefore, the use of a hybrid model to predict such strongly coupled, high-dimensional, and nonlinear parameters has attracted considerable attention and has neem a topic of active research in recent years.
A typical hybrid design helps remove noise, reduce dimensionality, or extract features from high-dimensional input data of models, e.g., wavelet transform and principal component analysis, and, subsequently, uses the obtained data as an input to construct a predictive model using machine learning or deep learning methods. Some hybrid models have been applied to predict humidity and other environmental parameters in livestock houses or greenhouses. For example, Besteiro et al. [14] developed a commercial piglet farm CO\textsubscript{2} prediction model based on a wavelet neural network designed to consider indoor and outdoor temperatures, indoor humidity, and ventilation rates as input variables. He and Ma [15] extracted four main factors from eight parameters, namely, outside air temperature and humidity, wind speed, and others, based on principal component analysis (PCA) and then constructed a prediction model for indoor greenhouse humidity in north China in winter based on a BPNN model. The test set results showed that the model was not accurate enough. In addition to PCA for feature selection, the light gradient boosting machine (LightGBM) algorithm, developed in recent years [16] as a variant of the improved gradient-boosted decision tree (GDBT), has been widely used as a prediction model in several fields [17–19]. Since the heuristic information of the iterative tree can be used as an important measure of features [20], an iterative tree can be constructed using LightGBM to calculate the feature importance measure of a sample to perform feature selection. The built-in exclusive feature bundling algorithm of the LightGBM model compensates for the drawback of PCA that useful information tends to be lost in calculating the dimensionality reduction for less relevant high-dimensional features, and mutually exclusive feature bundling allows for certain cases of non-mutual exclusion among a small number of features, which can better separate redundant features from bundled features to maintaining the accuracy of information filtering. A few works have used LightGBM to reduce the dimensionality of the features of an input model [16,21,22].

In data-driven predictive modeling algorithms, the support-vector regression (SVR) approach rests on a stronger mathematical theoretical foundation than neural-network-based “black box” predictive models [23]. It can effectively solve the problem of constructing high-dimensional data models under limited sample conditions and possesses the advantages of strong generalization ability, convergence to the global optimum, and insensitivity to dimensionality. This method has shown considerable potential to learn nonlinear time-series data, and has been widely used in several fields for prediction, such as late potato blight [24], apricot yields [25], solar radiation [26], and so on.

As the SVR parameter selection is often stochastic and empirical, inappropriate parameter combinations affect the performance and prediction accuracy of such models. Thus, for SVR prediction models, the best parameter combination should be selected to obtain good prediction results. The gray wolf optimization (GWO) algorithm proposed by Mirjalili et al. [27] possesses the advantages of high stability and search capability and has been widely applied to optimize SVR [28,29], BPNN [30] and other models. It has been proven to effectively improve the accuracy and generalization ability of prediction models. As a new metaheuristic optimization technique, GWO possesses stronger global search capability and multimodal function exploration capability than genetic algorithms (GA) and particle swarm optimization (PSO) algorithms [31]. However, it is relatively inefficient and has a tendency to obtain a local optimum [32]. A few researchers have improved the GWO algorithm by introducing new operators, changing control parameters, or updating the GW position. Song et al. [33] proposed a nonlinear model predictive controller, based on the shaded GWO algorithm, to predict the maximum wind energy of wind turbines, and the algorithm was balanced between local and global search by changing the formulae of the gray wolves’ social ranks and position updates to prevent the algorithm from reaching a local optimum. Zhang and Hong [34] improved the search capability and search efficiency of the GWO algorithm by introducing a chaotic tent mapping function to solve the problems of incomplete diversity and premature convergence of wolf packs. Zhao et al. [35] optimized the parameters of a gated recurrent unit (GRU) model using an improved version called IGWO to balance the overfitting and underfitting of the traditional
GWO algorithm by improving the objective function. Liu et al. [36] addressed the tendency of the GWO algorithm to fall into local extremes in early stages by introducing a new operator to construct a dynamic variational operator to adjust the difference vector of the wolf pack. Based on these works, and by combining the advantages of other algorithms, in this study, we propose an algorithm called Chaos-GWO (CGWO), based on chaos theory, which is designed to avoid falling into local optima. The proposed approach is further discussed in Sections 2.4 and 3.3.

Therefore, to improve the lower performance of the existing humidity prediction methods for livestock houses or greenhouses, and overcome the shortcomings of a single SVR model, which may be difficult to adapt to sheep barn humidity prediction, we propose a hybrid prediction model for ambient air humidity in sheep barns based on the proposed combined LightGBM–CGWO–SVR method, to predict the air humidity after 10 min. First, redundant or overlapping information on the environmental features obtained from the sensors is filtered out using the LightGBM model’s built-in dimensionality reduction technique, to avoid excessive computational complexity and overfitting caused by the direct input of multidimensional features. Subsequently, the SVR model hyperparameters are searched and optimized using the GWO algorithm, improved by the chaos operator for better accuracy and reliability.

2. Materials and Methods

2.1. Data Sources

The source of the data was a sheep breeding facility for meat production located at 44°27′18″ N latitude and 86°10′47″ E longitude in Manas County, Changji Hui Autonomous Prefecture, Xinjiang Uygur Autonomous Region, China. The main space of the facility, with an area of ~422 m², was selected to collect the experimental data. The facility mainly breeds and raises Suffolk sheep, and the sheep barn structure is semi-enclosed and generally divided into three parts, as shown in Figure 1, which included the main area, used as the daily resting area for sheep located in the middle of the barn, a shaded area located in the north, and an eating area located in the south. The main area of the farm was ~33.75 m long and 12.5 m wide, with brick and concrete walls on all sides, windows and doors, a roof made of steel plating, and a floor of compacted soil. In summer, the sheep barn is protected from heat by natural ventilation and a shaded tent on the north side. In winter, the sheep are kept in the main area for warmth and enclosed breeding, and ventilation is provided by exhaust fans. To monitor the environmental parameters of the sheep barn online, temperature, humidity, noise, light, particulate matter 2.5 (PM2.5), particulate matter 10 (PM10), total suspended particulate (TSP), CO2, ammonia, and hydrogen sulfide sensors were deployed in the main area. The ammonia, hydrogen sulfide, noise, and other sensors were installed at 3.2, 3.7, 3.1, and 3.3 m, respectively, from the ground.

![Figure 1. Schematic of the Xinjiang sheep house monitoring.](image-url)
2.2. Experimental Data Acquisition

The sensors installed in the sheep barn performed online data collection at 10-min intervals, and the data were uploaded to a remote IoT cloud service platform through a gateway. The 2000 units of data collected online from 8 to 22 February 2021, were used as the data source for this study; 70% of these data were selected for the training set, and 30% were used for the testing set to train and validate the performance of the proposed humidity prediction model. The raw environmental data are provided in Table 1, and the sequence of the observed air humidity data is illustrated in Figure 2.

Table 1. Experimental original data collected from 8 to 22 February 2021.

| Time            | Air Temperature °C | Air Humidity %  | CO₂ /mL/m³ | PM₁₀ /μg/m³ | PM₂.₅ /μg/m³ | Light Intensity /lx | Noise /dB | TSP /μg/m³ | NH₃ /mL/m³ | H₂S /mL/m³ |
|-----------------|--------------------|-----------------|------------|-------------|--------------|--------------------|-----------|------------|------------|------------|
| 2021/2/8 17:11:56 | 4.2                | 78              | 1355       | 16.3        | 81.3         | 146                | 31.1      | 120.7      | 0          | 5.2        |
| 2021/2/8 17:21:35 | 3.9                | 78.4            | 1330       | 26.5        | 107.1        | 97                 | 72.7      | 165.3      | 0          | 5.2        |
| 2021/2/22 17:12:15 | 6.6                | 81.2            | 855        | 12.5        | 24           | 97                 | 40.2      | 45.3       | 0          | 5.2        |
| 2021/2/22 17:22:15 | 6.4                | 81.4            | 780        | 9.3         | 24.6         | 183                | 34.1      | 42         | 0          | 5          |
| 2021/2/22 17:32:16 | 6.3                | 81.4            | 770        | 12.2        | 34.8         | 159                | 31.8      | 58.2       | 0          | 1.4        |

Figure 2. Original nonlinear curve of the values of air humidity.

2.3. Experimental Data Preprocessing

As the parameters in the sheep barn environment had different magnitudes and units, which differed significantly and affected the data analysis and learning results, the influence of magnitudes between parameter sets needed to be eliminated through normalization to enable the algorithms to learn and analyze the hidden structural relations in the data more easily [37]. In addition, standardization was performed to accelerate the training of learning models, and was more stable than an unstandardized training procedure under the same conditions, providing a more uniform fit and better prediction results. The standardization equation is provided as Equation (1):

\[ Z_m^* = \frac{Z_m - \bar{Z}_m}{Z_s}, \]

where \(Z_s\), \(\bar{Z}_m\), and \(Z_m^*\) denote the standard deviation, mean, and values obtained after normalization, respectively.
2.4. Construction of the Combined Forecasting Model for Sheep Barn Humidity

2.4.1. LightGBM for Feature Selection

LightGBM is a lightweight framework implementing a gradient-boosting decision tree, which is an algorithm that converts multiple decision trees into a strong learning machine using weighted linear combinations. It retains samples with large gradients and randomly selects samples with small gradients during training. To counteract the sampling effect in the data distribution, LightGBM introduces a constant multiplier for samples with small gradients in calculating the information gain. For the training set \( D = \{Y_1, Y_2, \cdots, Y_n\} \), the negative gradient of the loss function at each iteration corresponds to \( \{h_1, h_2, \cdots, h_n\} \), the sampling rate of the large gradient samples is \( o \), and the sampling rate of the small gradient samples is \( p \). The steps used to learn the feature weighting are performed as follows:

Step 1: Sort the sample points in descending order according to the absolute values of their gradients.

Step 2: Select the top \( o \times n \) samples of the sorted results to generate a subset of the large gradient samples \( O_D \).

Step 3: Randomly select \( p \times n \) samples from the remaining samples and generate them as a small gradient sample subset \( P_D \).

Step 4: Combine the large and small gradient samples \( (O_D \cup P_D) \) to learn a new decision tree; to calculate the information gain (IG), the small gradient samples are multiplied by a weight factor \((1 - o) / p\), and the information gain of the split feature \( m \) for the split point \( d \) is calculated as:

\[
V_m(d) = \frac{1}{n} \left( \frac{\left( \sum_{Y_i \in O_D, Y_i \leq d} h_i + \frac{1-o}{p} \sum_{Y_i \in P_D, Y_i \leq d} h_i \right)^2}{\sum F(Y_i)} + \frac{\left( \sum_{Y_i \in O_D, Y_i > d} h_i + \frac{1-o}{p} \sum_{Y_i \in P_D, Y_i > d} h_i \right)^2}{\sum [1 - F(Y_i)]} \right),
\]

of which:

\[
F(Y_i) = \begin{cases} 
1 & Y_i \in (O_D \cup P_D), Y_i \leq d \\
0 & Y_i \in (O_D \cup P_D), Y_i > d 
\end{cases}
\]

Step 5: Repeat Steps 1–4 for a specified number of iterations or until convergence is reached. After traversing the entire model, the sum of IG of all split nodes for a feature is considered an important weight.

The collected environmental parameters of sheep barns were ranked based on feature importance using LightGBM, and the weights obtained are listed in Table 2. To select strongly correlated features, the key influencing factors of CO\(_2\) concentration, light intensity, air temperature and PM\(_{2.5}\) were screened and applied to the LightGBM model, and the training and testing sets of the Xinjiang sheep barn humidity prediction model were constructed using the screened parameter data.

**Table 2.** Parameter feature importance ranking based on the LightGBM model.

| Parameter | Feature Weights |
|-----------|-----------------|
| CO\(_2\)   | 29.741152       |
| Light intensity | 13.195553     |
| Air temperature | 13.092886     |
| PM\(_{2.5}\) | 6.540839        |
| H\(_2\)S   | 5.084741        |
| PM\(_{10}\) | 3.862321        |
| Noise     | 3.290505        |
| TSP       | 1.477563        |
| NH\(_3\)  | 0.000000        |
2.4.2. GWO Algorithm

The GWO algorithm was inspired by the putative hierarchy of Gray Wolves and an associated hunting mechanism. The model is based on the assumption of a strict social hierarchy in a GW pack, which is divided into four levels. The first level is an \( \alpha \) wolf, the leader of the pack and responsible for the decision-making of the whole population; the second level is the \( \beta \) wolves, which assist the \( \alpha \) wolf in decision-making or other hunting activities; the third level is the \( \delta \) wolves, which obey the commands of the first two levels and perform tasks, such as scouting and sentry duty; the fourth level comprises \( \omega \) wolves, which obey the leadership and command of the first three levels. The main components of the GWO algorithm are described below.

(1) Population structure

In the GWO algorithm, each GW represents a potential solution, where the optimal solution is denoted as \( \alpha \), the second best as \( \beta \), the third best as \( \delta \), and \( \omega \) denotes the remaining candidate solutions and is guided by \( \alpha \), \( \beta \), and \( \delta \) in its search.

(2) Surrounding the prey

The GW encircle the prey in the GWO algorithm as described by a mathematical model defined in Equations (4) and (5):

\[
\begin{align*}
D &= |C \cdot U_i(n) - U(n)|, \quad \text{and} \\
U(n + 1) &= U_i(n) - A \cdot D,
\end{align*}
\]

where \( n \) denotes the current number of iterations, \( A \cdot D \) denotes the bracketing step, \( U_i \) denotes the position vector of the prey, and \( U \) denotes the current position vector of an individual GW. \( A \) and \( C \) denote the vectors of random coefficients, which are calculated as given in Equations (6) and (7):

\[
\begin{align*}
A &= 2a \cdot r_1 - a, \quad \text{and} \\
C &= 2 \cdot r_2,
\end{align*}
\]

where \( C \) varies in the range of \((0, 2)\), which increases the randomness of the distances of other individual GWs from the \( \alpha \), \( \beta \), and \( \delta \) wolves, which influences the detection ability of the group and facilitates the GWO algorithm to jump out of local optima. \( r_1 \) and \( r_2 \) denote randomly generated vectors on interval \((0, 1)\), respectively, and \( a \) denotes a nonlinear factor that decreases linearly from 2 to 0 with the number of iterations, which is calculated as:

\[
a = 2 - 2 \left( \frac{n}{n_{\text{max}}} \right),
\]

where \( n \) and \( n_{\text{max}} \) denote the current number of iterations and the maximum number of iterations. As the value of \( a \) decreases from 2 to 0, \( A \) varies in the range of \((-a, a)\). When \(|A| < 1\), the GW launches an attack on its prey; that is, an optimal solution is achieved. When \(|A| > 1\), the GW has performed a search but abandoned the current prey; thus, this search avoids getting stuck in a local optimal solution and causes the GWO algorithm to continue to search for a solution that is optimal solution.

(3) Hunting process

The \( \alpha \), \( \beta \), and \( \delta \) wolves are more aware of the potential location of their prey and occupy an optimal position to attack. Other individual GWs are guided by the \( \alpha \), \( \beta \), and \( \delta \) wolves to move toward the prey during the hunting phase, to surround the prey, and to update their positions by moving and repositioning multiple times to eventually reach the prey’s position, respectively. The specific equations for the change in the positions of the GW pack siege are given as:

\[
\begin{align*}
L_\alpha &= |C_1 \cdot U_\alpha(n) - U(n)|, \quad L_\beta = |C_2 \cdot U_\beta(n) - U(n)|, \quad L_\delta = |C_3 \cdot U_\delta(n) - U(n)|,
\end{align*}
\]
Step 1: Initialize the algorithm parameters of the GW population size $M$, with a maximum number of iterations $n_{\text{max}}$.

Step 2: Population initialization. Randomly generate $M$ individual GWs $\{U_i, i=1,2,\cdots,M\}$, where $U_i$ denote the current positions of $\alpha$, $\beta$, and $\delta$ wolves, respectively; $U_1$, $U_2$, and $U_3$ denote the updated positions of the individual GWs toward $\alpha$, $\beta$, and $\delta$ wolves, respectively; and $U_i(n+1)$ denotes the mean of the updated positions of the individual GWs along the directions of $\alpha$, $\beta$, and $\delta$ wolves.

Step 3: Set $n=1$ (current number of iterations).

Step 4: If $n < n_{\text{max}}$ (or when the iteration condition is satisfied).

Step 5: Update the distances and directions of the movement of individual GWs in the current pack from the $\alpha$, $\beta$, and $\delta$ wolves using Equations (9)–(11), respectively, and update the location of each individual GW using Equations (4) and (5).

Step 6: Update the random exploration vectors $A$ and $C$ using Equations (6)–(8).

Step 7: Calculate the fitness of each individual GW for the current iteration $\{f(U_i(n)), i=1,2,\cdots,M\}$, and compare it with the fitness of the last iteration. If the new fitness is higher than the current fitness, the new fitness replaces the current fitness as the global optimal fitness and updates the wolf’s position. If the new fitness value is not higher than the current fitness, it remains unchanged.

Equations (6)–(8).

Step 8: Iterate through all the individual wolves, calculate the fitness values of these wolves, and identify the best fitness value as the global optimal fitness and updates the wolf’s position. If the new fitness is higher than the current fitness, the new fitness replaces the current fitness as the global optimal fitness and updates the wolf’s position. If the new fitness value is not higher than the current fitness, it remains unchanged.

where $L_{\alpha}$, $L_{\beta}$, and $L_{\delta}$ denote the distances of the individual GWs from the $\alpha$, $\beta$, and $\delta$ wolves, respectively; $U_{\alpha}$, $U_{\beta}$, and $U_{\delta}$ denote the current positions of $\alpha$, $\beta$, and $\delta$ wolves, respectively; $U_1$, $U_2$, and $U_3$ denote the updated positions of the individual GWs toward $\alpha$, $\beta$, and $\delta$ wolves, respectively; and $U_i(n+1)$ denotes the mean of the updated positions of the individual GWs along the directions of $\alpha$, $\beta$, and $\delta$ wolves.

2.4.3. Improvement of GWO Based on Chaotic Operators

For GWO, exploration and exploitation capabilities are crucial to optimal performance and mainly depend on the value of the stochastic exploration coefficient $C$. Thus, choosing a suitable value of $C$ to coordinate the exploration and exploitation capabilities of GWO is critical.

Inspired by chaos theory, we introduced time-varying acceleration constants in $C_1$, differential mean perturbed time-varying parameters in $C_2$, and sigmoid-based acceleration coefficients in $C_3$ to enhance the randomness of $C$. Compared with random search, a chaotic algorithm could help in a thorough search with a higher probability and speed. It overcomes the tendency of the GWO algorithm to fall into a local optimum, owing to existing performance, and distributes the search more uniformly to obtain better global convergence performance and faster convergence. The improved equations, $C_1$, $C_2$, and $C_3$, are defined as:

$$C_1 = 0.5 + 0.5 \cdot \exp\left(-\frac{n}{500}\right) + 1.4 \cdot \frac{\sin(n)}{30},$$

$$C_2 = 1 + 1.4 \cdot \left(1 - \exp\left(-\frac{n}{500}\right)\right) + 1.4 \cdot \frac{\sin(n)}{30},$$

$$C_3 = 1 \cdot \left(1 + \exp\left(-\frac{0.0001 \cdot n}{n_{\text{max}}}\right)\right) (0.5 - 2.5) \left(\frac{n}{n_{\text{max}}}\right)^2,$$

where $n$ and $n_{\text{max}}$ denote the current iteration number and maximum iteration number, respectively. Equations (12)–(14) are used instead of Equation (7).

(2) Algorithm implementation of CGWO

In summary, the implementation process of the proposed CGWO algorithm is given as follows:

Step 1: Initialize the algorithm parameters of the GW population size $M$, with a maximum number of iterations $n_{\text{max}}$.

Step 2: Population initialization. Randomly generate $M$ individual GWs $\{U_i, i=1,2,\cdots,M\}$, where $U_i$ denote the current positions of $\alpha$, $\beta$, and $\delta$ wolves, respectively; $U_1$, $U_2$, and $U_3$ denote the updated positions of the individual GWs toward $\alpha$, $\beta$, and $\delta$ wolves, respectively; and $U_i(n+1)$ denotes the mean of the updated positions of the individual GWs along the directions of $\alpha$, $\beta$, and $\delta$ wolves.

Step 3: Set $n=1$ (current number of iterations).

Step 4: If $n < n_{\text{max}}$ (or when the iteration condition is satisfied).

Step 5: Update the distances and directions of the movement of individual GWs in the current pack from the $\alpha$, $\beta$, and $\delta$ wolves using Equations (9)–(11), respectively, and update the location of each individual GW using Equations (4) and (5).

Step 6: Update the random exploration vectors $A$ and $C$, and the convergence factor $a$ using Equations (6)–(8).

Step 7: Calculate the fitness of each individual GW for the current iteration $\{f(U_i(n)), i=1,2,\cdots,M\}$, and compare it with the fitness of the last iteration. If the new fitness is higher than the current fitness, the new fitness replaces the current fitness as the global optimal fitness and updates the wolf’s position. If the new fitness value is not higher than the current fitness, it remains unchanged.
Step 8: Update $\alpha$, $\beta$, and $\delta$ wolves: $U_\alpha$, $U_\beta$, and $U_\delta$.

Step 9: $n = n + 1$, return to Step 4 when the iteration condition is satisfied, exit the loop when the maximum number of iterations ($n_{max}$) is reached, obtain the position of the $\alpha$ wolf, which is the best value for the SVR parameters $C$, $\gamma$, and $\epsilon$, and the algorithm ends.

2.4.4. SVR

For a given sample training set $(m_i, n_i), i = 1, 2, \ldots, N$, $m$ denotes the independent variable in the sample training set, $n$ is the dependent variable in the sample training set and $N$ is the number of samples. The SVR is linearly fitted by the nonlinear feature mapping function $\phi(m)$, which maps the data $m$ onto a high-dimensional linear space. The linear function of SVR in the high-dimensional feature space can be expressed as:

$$ f(m) = \mu \phi(m) + b, \quad (15) $$

where $\mu$ denotes the normal vector and $b$ denotes the bias term.

To control the accuracy of the fit, an insensitive loss function $\varepsilon$ is introduced. If the difference between the predicted value $f(m)$ of $m$ and the actual value $n$ is less than $\varepsilon$, the predicted value does not suffer any loss. Otherwise, the predicted value is considered to have suffered a loss, and the optimization model is expressed in Equation (16):

$$ F_\varepsilon(f(m), n) = \begin{cases} 0 & |f(m), n| < \varepsilon \\ |f(m) - n| - \varepsilon & |f(m), n| \geq \varepsilon \end{cases} \quad (16) $$

Penalty coefficients $C$ were introduced for error-free fitting at positive and negative relaxation levels, i.e., $\tau_i$ and $\tau_i^*$, respectively, with the minimum of the sum of the complexity of the regression function and the fitting error as the final optimization objective, as expressed in Equation (17):

$$ \min \frac{1}{2} ||\mu||^2 + C \sum_{i=1}^{N} (\tau_i + \tau_i^*), \quad (17) $$

s.t. \( \begin{cases} n_i - \mu \phi(m_i) - b \leq \varepsilon + \tau_i^* \\
\mu \phi(m_i) + b - n_i \leq \varepsilon + \tau_i \\
\tau_i, \tau_i^* \geq 0 (i = 1, 2, \ldots, N). \end{cases} \quad (18) $$

Using the Lagrange multipliers $\lambda_i$ and $\lambda_i^*$ such that the partial derivatives of the variables $\mu$, $b$, $\tau_i$, and $\tau_i^*$ are zero, the abovementioned functions were transformed into the pairwise problem of SVR, as expressed in Equation (19):

$$ \begin{cases} \max \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} (\lambda_i - \lambda_i^*)(\lambda_j - \lambda_j^*) \cdot \kappa(m_i, m_j) + \varepsilon \sum_{i=1}^{N} (\lambda_i + \lambda_i^*) + \sum_{i=1}^{N} m_i(\lambda_i - \lambda_i^*) \\
\text{s.t.} \quad \sum_{i=1}^{N} (\lambda_i - \lambda_i^*) = 0 \\
\lambda_i, \lambda_i^* \in [0, C] \end{cases} \quad (19) $$

where $\kappa(m_i, m_j)$ denotes the kernel function, which implies that the inner product is calculated after mapping $m_i$ and $m_j$ onto a high-dimensional feature space.

By solving the pairwise problem, the SVR regression function could be expressed as:

$$ f(m) = \sum_{i=1}^{N} (\lambda_i - \lambda_i^*) \kappa(m_i, m) + b \quad (20) $$

In this experiment, the radial basis kernel function RBF, which is currently widely used, was considered, and expressed as:

$$ \kappa(m_i, m_j) = \exp \left( -\frac{|m_i - m_j|^2}{g^2} \right), \quad (21) $$

where $g$ denotes the width parameter of the radial basis kernel function.
2.4.5. Humidity Prediction Model of Sheep Barns Based on LightGBM-CGWO-SVR

A hybrid prediction model, based on the proposed LightGBM-CGWO-SVR method, was implemented for sheep barn humidity (Figure 3). The specific algorithmic steps are as follows:

Step 1: Perform data repair on the collected data.
Step 2: Use LightGBM to calculate the IG for the sheep barn environmental data after the data have been restored, and each parameter factor has been ranked in terms of feature importance, selecting strongly correlated features.
Step 3: Normalize the data after dimensionality reduction, based on LightGBM, and divide the training and test sets in a ratio of 7:3.
Step 4: Based on the training set, optimize $C$, $g$, and $\varepsilon$ for the SVR model using the CGWO algorithm described in Section 2.4.3.
Step 5: Apply the best hyperparameters obtained using CGWO, that is, $C$, $g$, and $\varepsilon$, to the SVR model to obtain the combined LightGBM–CGWO–SVR prediction model.

![Figure 3. Flowchart of the LightGBM–CGWO–SVR algorithm.](image)

3. Results and Discussion

3.1. Experimental Setup and Parameter Determination

The experimental calculations were performed using a computer with an 11th Gen Intel(R) Core (TM) i7-1165G7 processor at 2.80 GHz and 16 GB of memory, running the Windows 10 64-bit operating system with the Anaconda3 platform and the Python 3.6 programming language (64-bit). The simulation environment was completed by installing the NumPy, Sklearn, and LightGBM packages.

The CGWO parameters were set as follows. The size of the wolf pack was set to 20, the random exploration vectors $A$ and $C$ were set to 20, and the SVR parameters were searched in the range of $[0.01, 10]$. SVR uses RBF, and the optimal combination of hyperparameters was obtained by searching the SVR model globally using the CGWO algorithm. The
maximum number of iterations was 20, although the fourth iteration started to converge in the experiment.

3.2. Performance Criteria

In this study, the coefficient of determination ($R^2$), mean absolute error (MAE), root mean square error (RMSE), normalized root mean square error (NRMSE), and mean squared error (MSE) were used as evaluation metrics to computationally assess the prediction results.

1. $R^2$ shown in Equation (1) was used to assess the fit between the predicted and actual humidity, and considered a value in the range of $(0, 1)$, a value closer to 1 represented a better fit between the predicted and actual humidity values.

$$R^2 = \frac{\sum_{k=1}^{r} (f'_k - \bar{f})^2}{\sum_{k=1}^{r} (f_k - \bar{f})^2},$$  \hspace{1cm} (22)

2. MAE was the average distance between the individual and predicted humidity data; a smaller MAE implied that the predicted value was closer to the actual humidity data (Equation (23)).

$$MAE = \frac{1}{r} \sum_{k=1}^{r} |f_k - f'_k| \hspace{1cm} (23)$$

3. MSE represented the mean of the square of the difference between individual humidity data and the predicted humidity data. RMSE was the root mean square result of the MSE, which could be intuitive, in terms of the order of magnitude performance; normalized RMSE converted the RMSE value within $(0, 1)$. The smaller the MSE, RMSE, and NRMSE, the smaller the error between the predicted and true values, and the more accurate the model. They were expressed as:

$$RMSE = \sqrt{\frac{1}{r} \sum_{k=1}^{r} (f_k - f'_k)^2}, \hspace{1cm} (24)$$

$$NRMSE = \sqrt{\frac{1}{\bar{f}} \sum_{k=1}^{r} (f_k - f'_k)^2}, \hspace{1cm} (25)$$

$$MSE = \frac{1}{r} \sum_{k=1}^{r} (f_k - f'_k)^2 \hspace{1cm} (26)$$

In the equations above, $f_k$, $f'_k$, $r$, and $\bar{f}$ denoted the original data value, predicted value of $f_k$, amount of data, and average value of $f_k$, respectively.

3.3. Results and Analysis

To demonstrate the superior performance and effectiveness of the proposed LightGBM–CGWO–SVR model in predicting the humidity of sheep barn facilities, our approach was compared with the following models: GRU, SVR, extreme gradient boosting (XGBoost), GA-SVR, GWO-SVR, CGWO-XGBoost, CGWO-SVR, and LightGBM-SVR. These were computed using the same training and validation datasets, and parameters of different SVR models are shown in Table 3. The prediction performance for 10 min details of the different models on the final validation dataset are listed in Table 4. The real humidity measurements on the validation set with the predicted values of each model are shown in Figure 4. According to the results of the comparison presented in Table 4, the evaluation indices of the proposed LightGBM-CGWO-SVR-based sheep barn humidity prediction model were better than those of the other prediction models. The minimum values of MAE, RMSE, MSE, and NRMSE of the model were 0.0662, 0.2284, 0.0521, and 0.0083, respectively, among all the models, and $R^2$ achieved the highest value of 0.9973. As shown in Figure 4, the prediction curve of the LightGBM–CGWO–SVR model was able to significantly fit the
true value curve, indicating that the LightGBM–CGWO–SVR-based sheep barn humidity prediction model exhibited high accuracy and fitting performance.

Table 3. Parameters of different SVR models in experiment.

| Prediction Model | C  | g   | ε   |
|------------------|----|-----|-----|
| SVR              | 1.000 | 0.100 | 0.100 |
| GA–SVR           | 6.564 | 0.831 | 0.011 |
| GWO–SVR          | 8.958 | 0.023 | 0.016 |
| CGWO–SVR         | 9.346 | 0.011 | 0.040 |
| LightGBM–SVR     | 1.000 | 0.100 | 0.100 |
| LightGBM–CGWO–SVR | 9.849 | 0.010 | 0.059 |

Table 4. Experimental results of different prediction models.

| Prediction Model   | Error Type |
|--------------------|-------------|
|                    | MAE | RMSE | MSE | NRMSE | \(R^2\) |
| GRU \(^1\)         | 1.4307 | 1.7019 | 2.8964 | 0.0625 | 0.8522 |
| SVR                | 0.3451 | 1.1878 | 1.4110 | 0.0436 | 0.9280 |
| XGBoost \(^2\)     | 1.4372 | 1.4645 | 2.1450 | 0.0538 | 0.8906 |
| GA–SVR            | 0.3732 | 0.5897 | 0.3477 | 0.0216 | 0.9822 |
| GWO–SVR           | 0.0763 | 0.4230 | 0.1789 | 0.0155 | 0.9908 |
| CGWO–XGBoost \(^3\) | 0.0772 | 0.4025 | 0.1620 | 0.0147 | 0.9917 |
| CGWO–SVR         | 0.0731 | 0.3048 | 0.0929 | 0.0112 | 0.9952 |
| LightGBM–SVR     | 0.2596 | 1.1545 | 1.3328 | 0.0424 | 0.9320 |
| LightGBM–CGWO–SVR | 0.0662 | 0.2284 | 0.0521 | 0.0083 | 0.9973 |

\(^1\) learning_rate = 0.079, subsample = 0.5, n_estimators = 50. \(^2\) learning_rate = 0.659, subsample = 0.5, n_estimators = 50. \(^3\) learning_rate = 0.08, batch_size = 256, nb_epoch = 500.

Figure 4. Cont.
Figure 4. Cont.
Figure 4. Cont.
For a single model, the prediction accuracy of the SVR model was higher than that of the GRU and XGBoost models. The MAE, RMSE, MSE, and NRMSE values of the SVR model were reduced by 76%, 30%, 51%, and 30%, respectively, compared with those of GRU, and 76%, 19%, 34%, and 19%, respectively, compared with those of XGBoost. Compared with GRU and XGBoost, the $R^2$ of SVR improved by 9% and 4%, respectively. This indicated that the SVR model exhibited good adaptability to sheep barn humidity data. The real observed values of the local humidity and the predicted values of the GRU, XGBoost, and SVR models are shown in Figure 5. The prediction curves of the SVR model were closer to the real values than those of the GRU and XGBoost models, particularly for the prediction curves of the GRU model, which fluctuated more than those of the XGBoost and SVR models. The values of the GRU and XGBoost models deviated more from the predicted value when the humidity value was approximately 85%, whereas SVR achieved better coverage. This might have occurred because GRU models exhibit issues of gradient disappearance or explosion for increasing amounts of data and model volume, and XGBoost suffers from the difficulty of tuning parameters to cope with high-dimensional nonlinear features. In contrast, SVR models, the computational complexity of which is not highly correlated with the dimensionality of the input parameters, demonstrate better generalization ability and prediction accuracy when dealing with high-dimensional data.
Animals 2022, 12, x FOR PEER REVIEW

Figure 5. Measured versus computed humidity of GRU, XGBoost, and SVR (part of validation data).

When the SVR algorithm hyperparameters were globally searched and optimized using three different search algorithms (i.e., GA, GWO, and CGWO), the SVR algorithm optimized using CGWO exhibited lower MAE, RMSE, MSE, and NRMSE values of 80%, 43%, 73%, and 48%, respectively, compared to the SVR algorithm optimized using GA, whereas $R^2$ was improved by 1%. Compared with the GWO-optimized SVR algorithm, the MAE, RMSE, MSE, and NRMSE values were reduced by 4%, 30%, 48%, and 28%, respectively, and $R^2$ was improved by 0.4%. Thus, CGWO outperformed GA and GWO in terms of optimization and could maintain a good balance between local and global searches. Moreover, the real measurement values of humidity localized in the validation set were compared with the predicted values obtained from the GA-SVR, GWO-SVR, and CGWO-SVR models (Figure 6). Using the optimization algorithm to perform a global search of the SVR hyperparameters significantly improved the prediction performance of the SVR model, and the CGWO-optimized SVR model was able to fit the original curve more ideally. The result was closer to the true values in the prediction of peaks than the models optimized by GA and GWO. Compared with GA, the GWO algorithm introduces nonlinear decreasing factors and random coefficient vectors to increase the randomness of the distance between other GW individuals and the head, which improved the detection ability of the group. The results obtained in this study, by introducing chaos operators into the GWO optimization algorithm to form the CGWO optimization algorithm, further validated that chaos operators can overcome the shortcomings of the GWO algorithm’s tendency to fall into local optima and its low efficiency in path planning.

Figure 6. Measured versus computed humidity of GA-SVR, GWO-SVR, and CGW-SVR (part of validation data).
The data in Table 4 indicate that the MAE, RMSE, MSE, and NRMSE of the SVR model optimized by CGWO were reduced by 5%, 24%, 43%, and 24%, respectively, compared with those of the model optimized by CGWO with XGBoost, which further demonstrated the superior fitting ability of SVR for nonlinear problems. The MAE, RMSE, MSE, and NRMSE of the LightGBM–CGWO–SVR model obtained by further reducing the dimensionality of the data using LightGBM on top of the CGWO–SVR model were reduced by 9%, 25%, 44%, and 26%, respectively, compared with those of CGWO–SVR. This demonstrated that feature extraction of sheep barn data using LightGBM could reduce the input dimensionality of the model, filter redundant information and irrelevant features, reduce the size of the algorithm, and enhance the generalization performance of the model. In addition, the MAE, RMSE, MSE, and NRMSE of the LightGBM–SVR model obtained using LightGBM to reduce the dimensionality were 25%, 3%, 6%, and 3%, respectively, lower than those of the single SVR model, which implied that using LightGBM to reduce the dimensionality of the data could improve the accuracy of the model. This was supported by the comparison of the local datasets predicted by the single SVR, CGWO–SVR, and LightGBM–CGWO–SVR models with respect to humidity (Figure 7).

![Figure 7. Actual versus computed humidity of SVR, CGW–SVR, and LightGBM–CGWO–SVR (part of validation data).](image)

Figure 8 shows the box plot scheme of the predicted moisture value spatial diffusion at the validation level, based on different models (i.e., GRU, SVR, XGBoost, GA–SVR, GWO–SVR, CGWO–XGBoost, CGWO–SVR, and LightGBM–SVR), where GRU and XGBoost predictions on the validation set faced difficulties in matching the actual true observations. The predictions of the LightGBM–CGWO–SVR model were more stable, with the upper quartile, lower quartile, and median close to the actual values and agreeing with the actual observations over the entire range (e.g., maximum and minimum). The predictions of the LightGBM–CGWO–SVR model were more accurate than those of the other models for the offset values that appeared in the original environmental data. The Taylor plots of the accuracy of the predicted humidity values of each model on the validation set compared with the actual values are shown in Figure 9, where each point represents the error statistic of the predicted humidity value of a model compared with the actual value. Among them, the LightGBM–CGWO–SVR model was closest to the arc where the actual values were located, and had the shortest distance from the actual values, indicating that the model possessed the smallest root mean square error between the predicted and actual values. Moreover, the Taylor plot shows that the proposed model exhibited the highest correlation coefficient compared to the other models.
In addition, autocorrelations of the observed time series data are often an important factor affecting prediction performance. Table 5 shows the accuracy results of the model
predicted by the proposed LightGBM–CGWO–SVR, which used 10-, 30-, 60-, and 90-min lag time datasets, respectively. The results showed the accuracy of the model developed slowly decreased over time. Despite all this, the observed humidity data with a lag of less than 100 min were sound information, and the prediction accuracy of the dataset with a lag of 90 min also showed good performance. Its $R^2$ reached 0.9940, which was only 0.39% less than that of the dataset with a lag of 10 min.

Table 5. Comparison of LightGBM–CGWO–SVR prediction accuracy at time steps of 10, 30, 60, and 90 min for humidity.

|        | MAE   | RMSE  | MSE   | NRMSE | $R^2$ |
|--------|-------|-------|-------|-------|-------|
| 10 min | 0.0597| 0.1950| 0.0380| 0.0073| 0.9979|
| 30 min | 0.0674| 0.2494| 0.0622| 0.0092| 0.9972|
| 60 min | 0.2181| 0.2786| 0.0776| 0.0149| 0.9956|
| 90 min | 0.2254| 0.3298| 0.1087| 0.0179| 0.9940|

3.4. Discussion

In a previous related study, He and Ma [15] obtained good performance on the training set using the BPNN and PCA combination forecasting model, with $R^2$ of 0.9768 and RMSE of 1.6745. However, the test set results with $R^2$ of 0.8842 showed that the model generalization ability was still insufficient. The RNN–LSTM prediction model developed by Hongkang et al. [11] achieved good performance to temperature in the greenhouse, but it was not satisfactory to humidity prediction, and the $R^2$ was only 0.82 for the dataset with a 10 min lag.

The proposed LightGBM–CGWO–SVR prediction model exhibited high prediction accuracy for the collected time-series data. It could help accurately predict changes in humidity pattern in sheep barn facilities at intervals of 10 min or even 90 min. Furthermore, it could effectively adapt to nonlinear and high-dimensional data on humidity in sheep barns and exhibited superior generalization ability and prediction accuracy, compared to advanced algorithms, such as GRU and XGBoost models. It is possible to apply constructed combination models to predict and regulate air humidity in intensive sheep farming barns.

It should be noted that the data was acquired in February, when Xinjiang is in winter and very cold. For this reason, the sheep house was not always strongly ventilated, but ventilated intermittently and regularly. That is to say, a large part of the humidity data in this data set was not significantly affected by ventilation. Of course, the ventilation factor should not be ignored in model construction for the warm season, when the sheep house would be frequently ventilated.

4. Conclusions

The final simulation results of the proposed LightGBM–CGWO–SVR model and a comparison with other advanced models implied the following points:

1. The characteristics of the environmental factors of the sheep barn were screened using LightGBM without considering environmental variables with small effects on humidity. The key influencing factors of $CO_2$ concentration, light intensity, air temperature, and $PM_{2.5}$ were finally screened from the nine variables, including air humidity, $CO_2$ concentration, $PM_{2.5}$, $PM_{10}$, light intensity, noise, TSP, $NH_3$ concentration, and $H_2S$ concentration, and used for modeling, which effectively improved the computational efficiency of the model and its accuracy.

2. The CGWO algorithm obtained by introducing chaotic operators in this study retained the advantages of a simple structure, few control parameters, and straightforward implementation of the GWO algorithm, while improving the global search capability of the traditional GWO algorithm and solving the problem of randomness and empiricity in the selection of SVR parameters. The optimized SVR algorithm exhibited higher prediction accuracy and better generalization ability than an SVR algorithm optimized using GA and GWO.
(3) The performance of the proposed LightGBM–CGWO–SVR model in predicting the humidity for 10 min in a sheep barn facility exceeded that of the GRU, SVR, XGBoost, GA–SVR, GWO–SVR, CGWO–XGBoost, CGWO-SVR, and LightGBM–SVR models. The minimum values of the MAE, RMSE, MSE, and NRMSE indices were 0.0662, 0.2284, 0.0521, and 0.0083, respectively, and $R^2$ achieved the highest value of 0.9973. The results indicated that the proposed model can provide an effective guide for the accurate prediction of humidity in sheep barns in for Suffolk sheep, in Xinjiang, and could provide a reference for future research on the prediction of other environmental factors, such as NH$_3$ and H$_2$S concentration.

Author Contributions: Conceptualization, Writing-original draft, Writing-review & editing, D.F.; Methodology and Computing, Writing-original draft, B.Z.; Investigation, Formal analysis, software, Q.H.; Investigation, data curation, L.X., J.G., L.C. and L.Z.; Project administration, S.L.; Data curation, T.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported partly by Meizhou City S&T Planed Projects under Grant 2021A0305010, the National Natural Science Foundation of China under Grant 61871475, Guangzhou Key Research and Development Project under Grant 202103000033, Rural Science and Technology Commissioner Project in Guangdong Province under Grant KA1901002, KA22Y0113, Innovation Team Project of Universities in Guangdong Province under Grant 2021KCXTD019, Guangdong Science and Technology Project under Grant 2020A1414050060, 2020B0202080002, Quality Engineering Project of Universities in Guangdong Province under Grant KA220160149, the Technical Service Project for Xinnging Pigeon Industrial Park of Zhongkai University of Agriculture and Engineering College (Construction and Promotion of Visual Information Platform), Guangdong Natural Science Foundation under Grant 2021A1515011605.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References
1. Shi, J.; Li, X.; Lu, F.; Zhuge, H.; Dong, S. Variation in group sizes of sympatric Wild yak, Tibetan wild ass and Tibetan antelope in Arjin Shan National Nature Reserve of Xinjiang Province, China. Glob. Ecol. Conserv. 2019, 20, e00749. [CrossRef]
2. Qiao, Y.; Huang, Z.; Li, Q.; Liu, Z.; Hao, C.; Shi, G.; Dai, R.; Xie, Z. Developmental Changes of the FAS and HSL mRNA Expression and Their Effects on the Content of Intramuscular Fat in Kazak and Xinjiang Sheep. J. Genet. Genom. 2007, 34, 909–917. [CrossRef]
3. Huang, J.; Liu, S.; Hassan, S.G.; Xu, L.; Huang, C. A hybrid model for short-term dissolved oxygen content prediction. Comput. Electron. Agric. 2021, 186, 106216. [CrossRef]
4. Sanikhani, H.; Deo, R.C.; Samui, P.; Kisi, O.; Mirabassi, R.; Gavili, S.; Yaseen, Z.M. Survey of different data-intelligent modeling strategies for forecasting air temperature using geographic information as model predictors. Comput. Electron. Agric. 2018, 152, 242–260. [CrossRef]
5. Tong, X.; Zhao, L.; Heber, A.J.; Ni, J.-Q. Development of a farm-scale, quasi-mechanistic model to estimate ammonia emissions from commercial manure-belt layer houses. Biosyst. Eng. 2020, 196, 67–87. [CrossRef]
6. Banhazi, T.; Rutley, D.; Pitchford, W. Validation and fine-tuning of a predictive model for air quality in livestock buildings. Biosyst. Eng. 2010, 105, 395–401. [CrossRef]
7. Banhazi, T.M.; Seedorf, J.; Rutley, D.L.; Pitchford, W.S. Identification of risk factors for sub-optimal housing conditions in Australian piggeries: Part 1. Study justification and design. J. Agric. Saf. Health 2008, 14, 5–20. [CrossRef] [PubMed]
8. Daskalov, P. Prediction of Temperature and Humidity in a Naturally Ventilated Pig Building. J. Agric. Eng. Res. 1997, 68, 329–339. [CrossRef]
9. Yang, Z.-C. Hourly ambient air humidity fluctuation evaluation and forecasting based on the least-squares Fourier-model. Measurement 2019, 133, 112–123. [CrossRef]
10. Qadeer, K.; Ahmad, A.; Qyyum, M.A.; Nizami, A.-S.; Lee, M. Developing machine learning models for relative humidity prediction in air-based energy systems and environmental management applications. J. Environ. Manag. 2021, 292, 112736. [CrossRef]
11. Hongkang, W.; Li, L.; Yong, W.; Fanjia, M.; Hailhua, W.; Sigrimis, N. Recurrent Neural Network Model for Prediction of Microclimate in Solar Greenhouse. IFAC-PapersOnLine 2018, 51, 790–795. [CrossRef]
12. Zou, W.; Yao, F.; Zhang, B.; He, C.; Guan, Z. Verification and predicting temperature and humidity in a solar greenhouse based on convex bidirectional extreme learning machine algorithm. *Neurocomputing* 2017, 249, 72–85. [CrossRef]

13. Jung, D.-H.; Kim, H.S.; Jhin, C.; Kim, H.-J.; Park, S.H. Time-serial analysis of deep neural network models for prediction of climatic conditions inside a greenhouse. *Comput. Electron. Agric.* 2020, 173, 105402. [CrossRef]

14. Besteiro, R.; Arango, T.; Ortega, J.A.; Rodriguez, M.R.; Fernández, M.D.; Velo, R. Prediction of carbon dioxide concentration in weaned piglet buildings by wavelet neural network models. *Comput. Electron. Agric.* 2017, 143, 201–207. [CrossRef]

15. He, F.; Ma, C. Modeling greenhouse air humidity by means of artificial neural network and principal component analysis. *Comput. Electron. Agric.* 2010, 71, S19–S23. [CrossRef]

16. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.-Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 3149–3157.

17. Hao, X.; Zhang, Z.; Xu, Q.; Huang, G.; Wang, K. Prediction of f-CaO content in cement clinker: A novel prediction method based on LightGBM and Bayesian optimization. *Chemos. Intell. Lab. Syst.* 2022, 220, 104461. [CrossRef]

18. Li, Y.; Yang, C.; Zhang, H.; Jia, Y.; Park, S.; Kim, Y.; Park, S.; Kim, J.; Lim, J. BLITE-SVR: New forecasting model for late blight on potato using support-vector regression. *Comput. Electron. Agric.* 2016, 130, 169–176. [CrossRef]

19. Candido, C.; Blanco, A.; Medina, J.; Gubatang, E.; Santos, A.; Ana, R.S.; Reyes, R. Improving the consistency of multi-temporal land cover mapping of Laguna lake watershed using light gradient boosting machine (LightGBM) approach, change detection analysis, and Markov chain. *Remote Sens. Appl. Soc. Environ.* 2021, 23, 100565. [CrossRef]

20. Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* 2003, 3, 1157–1182. [CrossRef]

21. Zhang, Y.; Jiang, Z.; Chen, C.; Wei, Q.; Gu, H.; Yu, B. DeepStack-DTIs: Predicting Drug–Target Interactions Using LightGBM Feature Selection and Deep-Stacked Ensemble Classifier. *Interdiscip. Sci. Comput. Life Sci.* 2022, 14, 311–330. [CrossRef]

22. Zhang, Y.; Zhang, R.; Ma, Q.; Wang, Y.; Wang, Q.; Huang, Z.; Huang, L. A feature selection and multi-model fusion-based approach of predicting air quality. *ISA Trans.* 2020, 100, 210–220. [CrossRef]

23. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* 2004, 14, 199–222. [CrossRef]

24. Gu, Y.; Yoo, S.; Park, C.; Kim, Y.; Park, S.; Kim, J.; Lim, J. BLITE-SVR: New forecasting model for late blight on potato using support-vector regression. *Comput. Electron. Agric.* 2016, 130, 169–176. [CrossRef]

25. Esfandiarpour-Boroujeni, I.; Karimi, E.; Shirani, H.; Esmaeilizadeh, M.; Mosleh, Z. Yield prediction of apricot using a hybrid particle swarm optimization-imperialist competitive algorithm- support vector regression (PSO-ICA-SVR) method. *Sci. Hortic.* 2019, 257, 108756. [CrossRef]

26. Ghimire, S.; Deo, R.C.; Raj, N.; Mi, J. Wavelet-based 3-phase hybrid SVR model trained with satellite-derived predictors, particle swarm optimization and maximum overlap discrete wavelet transform for solar radiation prediction. *Renew. Sustain. Energy Rev.* 2019, 113, 109247. [CrossRef]

27. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* 2014, 69, 46–61. [CrossRef]

28. Niu, M.; Wang, Y.; Sun, S.; Li, Y. A novel hybrid decomposition-and-ensemble model based on CEEMD and GWO for short-term PM2.5 concentration forecasting. *Atmos. Environ.* 2016, 134, 168–180. [CrossRef]

29. Tikhomarine, Y.; Souag-Gamane, D.; Ahmed, A.N.; Kisi, O.; El-Shafie, A. Improving artificial intelligence models accuracy for monthly streamflow forecasting model using gated recurrent unit based on improved grey wolf optimizer. *J. Hydrocl.* 2020, 582, 124435. [CrossRef]

30. Tian, Y.; Yu, J.; Zhao, A. Predictive model of energy consumption for office building by using improved GWO-BP. *Energy Rep.* 2020, 6, 620–627. [CrossRef]

31. Emary, E.; Zawbaa, H.M.; Grosan, C.; Hassanien, A.E. Feature Subset Selection Approach by Gray-Wolf Optimization. In Proceedings of the Afro-European Conference for Industrial Advancement, Addis Ababa, Ethiopia, 17–19 November 2014; Springer: Cham, Switzerland, 2015; pp. 1–13.

32. Lu, C.; Gao, L.; Yi, J. Grey wolf optimizer with cellular topological structure. *Expert Syst. Appl.* 2018, 107, 89–114. [CrossRef]

33. Song, D.; Liu, J.; Yang, Y.; Yang, J.; Su, M.; Wang, Y.; Gui, N.; Yang, X.; Huang, L.; Joo, Y.H. Maximum wind energy extraction of large-scale wind turbines using nonlinear model predictive control via Yin-Yang grey wolf optimization algorithm. *Energy 2021, 220, 119866. [CrossRef]*

34. Zhang, Z.; Hong, W.-C. Application of variational mode decomposition and chaotic grey wolf optimizer with support vector regression for forecasting electric loads. *Knowl.-Based Syst.* 2021, 228, 107297. [CrossRef]

35. Zhao, X.; Lv, H.; Lv, S.; Sang, Y.; Wei, Y.; Zhu, X. Enhancing robustness of monthly streamflow forecasting model using gated recurrent unit based on improved grey wolf optimizer. *J. Hydrocl.* 2021, 601, 126607. [CrossRef]

36. Liu, D.; Li, M.; Ji, Y.; Fu, Q.; Li, M.; Faiz, M.A.; Ali, S.; Li, T.; Cui, S.; Khan, M.I. Spatial-temporal characteristics analysis of water resource system resilience in irrigation areas based on a support vector machine model optimized by the modified gray wolf algorithm. *J. Hydrocl.* 2021, 597, 125758. [CrossRef]

37. Yu, C.; Li, Z.; Yang, Z.; Chen, X.; Su, M. A feedforward neural network based on normalization and error correction for predicting water resources carrying capacity of a city. *Ecol. Indic.* 2020, 118, 106724. [CrossRef]