processes

MDPI

Article

An Enhanced Stacking Ensemble Method for Granule Moisture Prediction in Fluidized Bed Granulation

Binbin Chen [1,2], Panling Huang [1,2,*], Jun Zhou [1,2] and Mindong Li [1,2]

[1] School of Mechanical Engineering, Shandong University, Jinan 250061, China;
201913892@mail.sdu.edu.cn (B.C.); zhoujun@sdu.edu.cn (J.Z.); 202034339@mail.sdu.edu.cn (M.L.)

[2] Key Laboratory of High Efficiency and Clean Mechanical Manufacture, Ministry of Education,
Jinan 250061, China

* Correspondence: hfpl@sdu.edu.cn; Tel.: +86-150-0531-1566

Abstract: Moisture is a crucial quality property for granules in fluidized bed granulation (FBG) and accurate prediction of the granule moisture is significant for decision making. This study proposed a novel stacking ensemble method to predict the granule moisture based on granulation process parameters. The proposed method employed k-nearest neighbor (KNN), random forest (RF), light gradient boosting machine (LightGBM) and deep neural networks (DNNs) as the base learners, and ridge regression (RR) as the meta learner. To improve the diversity of the base learners, perturbations of the input variables and network structures were adopted in the proposed method, implemented by feature construction and combination of multiple DNNs with a different number of hidden layers, respectively. In the feature construction, a SHapley Additive exPlanations (SHAP) approach was innovatively utilized to construct effective synthetic features, which enhanced the prediction performance of the base learners. The cross-validation results demonstrated that the proposed stacking ensemble method outperformed other machine learning (ML) algorithms in terms of performance evaluation criteria, for which the parameters *MAE*, *MAPE*, *RMSE*, and *Adj.* R^2 were 0.0596, 1.5819, 0.0844, and 0.99485, respectively.

Keywords: stacking ensemble method; granule moisture prediction; fluidized bed granulation; process parameters; feature construction; SHapley Additive exPlanations (SHAP)

check for updates

Citation: Chen, B.; Huang, P.; Zhou, J.; Li, M. An Enhanced Stacking Ensemble Method for Granule Moisture Prediction in Fluidized Bed Granulation. *Processes* **2022**, *10*, 725. https://doi.org/10.3390/pr10040725

Received: 2 March 2022
Accepted: 6 April 2022
Published: 9 April 2022

1. Introduction

Granulation, defined as the process of particle enlargement by agglomeration technique, has been widely applied in the production of pharmaceutical solid dosage forms, mostly tablets and capsules [1]. Granulation can be divided into wet granulation and dry granulation according to whether liquid is utilized in the process. In dry granulation, mechanical compression or roll compaction is employed to agglomerate the dry powder particles. While in wet granulation, a granulation liquid (binder/solvent) is added into the pharmaceutical powders to bind the particles together by cohesive forces [2,3]. With the advantages of strong cohesiveness, high compressibility, good distribution, and uniform content, wet granulation is the most widespread granulation technique used in the pharmaceutical industry [4,5]. Fluidized bed granulation (FBG) is one main approach of wet granulation.

Moisture is one of crucial quality properties for granules, which has a significant influence on the fluidity, homogeneity, hardness of the granules, and the stability of the active pharmaceutical ingredient (API) [6,7]. In addition, the moisture content will indirectly affect the subsequent processes. While too high a moisture content may lead to the sticking of tablets during compression, too low a moisture content can result in tablet friability and disintegration issues [8]. A previous study has proved that the characteristics of the tablet compressed from granules are dependent not only on the residual moisture content of the granules but also on the moisture profiles during the entire FBG process [9]. Thus, real-time monitoring and timely control of the moisture content of the granules are of great importance to improve the quality of pharmaceuticals. Currently, inline methods using near-infrared (NIR) spectroscopy are effectively utilized to detect the moisture content of the granules during the granulation process [10,11]. Nevertheless, the NIR analyzers are costly and prone to fouling. Process modeling is another common practice to predict granule moisture based on process parameters, such as product temperature, spraying rate, inlet airflow rate, and inlet air temperature [12]. An empirical global process model (GPM) was reported to simulate the temporal evolution of the bed moisture–temperature, correlating the evaporation parameters to process parameters via two multi-linear functions. The GPM demonstrated a good capability in predicting the impact of process parameters on the moisture–temperature evolution [13]. Building mechanistic models requires a comprehensive understanding of the process and, which is still a scientific challenge. Data-driven models fit granule properties based on process data, and they can be developed using various modeling techniques.

Previous modeling studies of the FBG process were mostly based on various single algorithms. These models mine knowledge from different perspectives, which causes some limitations for a holistic understanding of the data. To overcome this deficiency of single-model-based approaches, an ensemble learning method has been developed by researchers. It combines multiple weak models to improve the prediction performance and generalization capability [14]. Numerous studies have indicated that ensemble models increased the accuracy and robustness with respect to single models [15–18]. Based on diverse algorithms and frameworks, different ensemble models were applied in various research fields and attained comparably high predictive power, which allowed to reduce prediction errors and decision-making risks [15–20]. Specifically, in the analysis of granulation, Wafa'H et al. developed an ensemble framework to model a high shear granulation (HSG) process, incorporated with a two-level integrated network, based on different RBF sub models, and a Gaussian mixture model [21]. A more predominant modelling performance and generalization capability of the proposed ensemble model was proved compared to other models in predicting the properties of the granules. Nevertheless, a similar application of ensemble approaches for modelling FBG process has still not been found in the literature.

In this paper, a novel ensemble method based on the stacking technique was proposed to predict the moisture content of granules during the FBG process. K-nearest neighbor (KNN), random forest (RF), light gradient boosting machine (LightGBM), and deep neural networks (DNNs) were determined as the base learners, and correspondingly the ridge regression (RR) as a meta learner, which formed a two-level stacking framework. Comprehensive performance evaluations were conducted to ascertain the prediction performance of the proposed stacking method base on cross-validation tests. Further, a comparative study with other well-known machine learning (ML) models was carried out to properly verify the superiority of the enhanced ensemble framework.

The remainder of the article is organized as follows. In Section 2, the experimental dataset of the granule moisture and relevant process parameters measured during the FBG process is briefly described. Section 3 provides an overview of the ML models and stacking ensemble approach utilized in this study. The preprocessing and feature construction of the dataset are discussed in Section 4; the proposed stacking ensemble method is also detailed in this section. The prediction results and performance comparison of diverse models are provided in Section 5. Finally, Section 6 concludes the study and gives the direction for future research.

2. Materials and Experiments

2.1. Materials

Each batch consisting of 375 g of starch, 495 g of lactose (Granulac 200, Wasserburg, Germany), 75 g of acetaminophen (active ingredient, DeYao Pharmaceutical Co., Ltd., Dezhou, China), and 555 g of microcrystalline cellulose (SH-CG1, Anhui Sunhere Pharmaceutical Excipients Co., Ltd., Huainan, China) was granulated, utilizing 3% hydroxypropyl methyl cellulose (Anhui Sunhere Pharmaceutical Excipients Co., Ltd., Huainan, China) as binder.

2.2. Granulation Batches

The granulation experiments were performed in a top-spray fluidized bed granulator (LGL002; Shandong Xinma Pharmaceutical Equipment Co., Ltd., Zibo, China). Before the formal experiments, a viable design space was determined under some operating requirements. The inlet airflow must keep the powders in a good fluidized state and the height of the flowing powders was not allowed to exceed the one of the spray nozzle. The spray pressure and the binder spray rate needed to be controlled in appropriate operating ranges for the successful nebulization of the binder. In addition, the binder spray rate and the inlet air temperature played main impacts on the temperature of the materials, which had an upper limit required of 40 °C. Table 1 shows the operating range of each process parameter obtained based on a series of trial experiments. In this research, ten batches of granulation experiments in different operating space were conducted totally.

Table 1. Operating ranges of the process parameters.

Process Parameter	Operating Range	
	Minimum	Maximum
Inlet air temperature (°C)	50	70
Inlet airflow rate ($m^3 \cdot h^{-1}$)	20	60
Spray pressure (bar)	0.8	1.8
Binder spray rate ($mL \cdot min^{-1}$)	3.75	14.25

2.3. Data Acquisition

During the FBG, a micro near-infrared spectrometer (MicroNIR PAT-U) was used to measure the moisture content of the granules every 2 s, which was installed in a position with the same height of the sampling port. The spectral data were collected via MicroNIR™ Pro v2.5.1 software. The process parameters studied were as follows: material temperature (F_1), inlet air temperature (F_2), inlet airflow rate (F_3), outlet air temperature (F_4), spray pressure (F_5), and binder spray rate (F_6). The scheme of the data acquisition for these process parameters during granulation is depicted in Figure 1. The six process parameters for data analysis were measured by the corresponding sensors and saved in the form of time stamps. The size of the data collected was 17,715 sets, including all the ten batches of experiments.

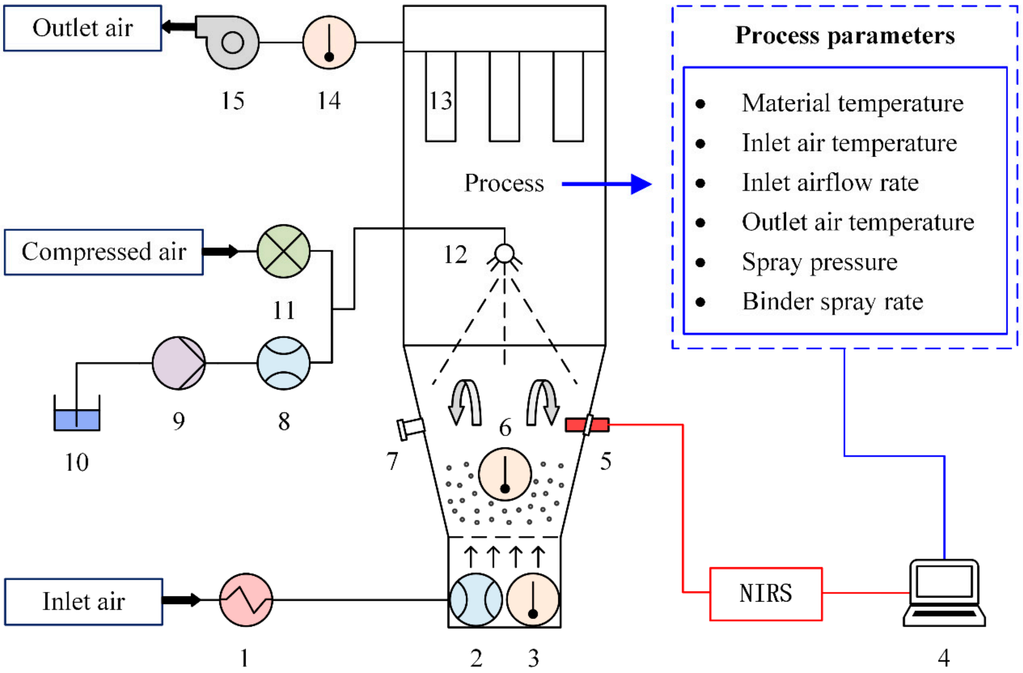

Figure 1. Scheme for data acquisition of the process parameters in top-spray fluidized bed granulation. (1) Electric heater; (2) airflow rate sensor; (3) temperature sensor 1; (4) data acquisition system; (5) in-line probe; (6) temperature sensor 2; (7) sampling port; (8) flowmeter; (9) peristaltic pump; (10) binder liquid; (11) pressure sensor; (12) nozzle; (13) bag filter; (14) temperature sensor 3; (15) exhaust fan.

3. Theoretical Aspects

The theories of four machine learning models, employed as the base learners, and the stacking ensemble approach are briefly described in this section.

3.1. Machine Learning Models

3.1.1. K-Nearest Neighbors Regression

K-nearest neighbors algorithm implements regression function by assigning the property value for the objective point to be the average of its *k*-nearest neighbors. Methods used to measure the distance between the query point and each training point include Euclidean distance (*ED*),

$$ED = \sqrt{\sum_{d=1}^{n} (x_{1d} - x_{2d})^2} \tag{1}$$

and Manhattan distance (*MD*),

$$MD = \sum_{d=1}^{n} |x_{1d} - x_{2d}| \tag{2}$$

where x_{1d} and x_{2d} are the values at the d-th dimension of two random n-dimensional data points, x_1 and x_2, respectively. The scaling of all data should be considered to prevent the domination of the variables with higher values in the distance calculation.

As the most significant parameter in the KNN model, the value of k determines the k-nearest training points in the feature space. A weight function can be useful for prediction, weighting the contribution of neighbors by the inverse of their distance. The formula of the weight function is formed as

$$y' = \frac{\sum_{i=1}^{k} \omega_i \cdot y_i}{\sum_{i=1}^{n} \omega_i} \tag{3}$$

$$\omega_i = \frac{1}{d(x', x_i)} \quad (4)$$

where y' is the predicted value of the objective point, y_i is the property value of the i-th k-nearest point, ω_i is the weight corresponding to y_i, and $d(x', x_i)$ is the distance between the point x' and the point x_i. In this study, the KNN models were implemented using the sklearn library in Python. The weighted function was used in the predictions of the models. The optimal number of the nearest neighbors was optimized by grid search.

3.1.2. Random Forests

Random forests are a bagging ensemble algorithm that combines copious randomized decision trees and averages their predictions as output [22]. The bootstrap sampling method is used to generate a random subset for each decision tree, and a random feature selection is applied into node splitting. The diversities of base learners are formed by the randomness, which enables the RF to have a powerful performance of generalization. The accuracy of the RF depends on the strengths of individual trees and the correlations between any two of them. Therefore, the most critical parameters are the number of trees n and the size of the feature subsets in the RF. Increasing these parameters can improve the accuracy of the prediction, but leads to the computation overhead. In this study, the RF models were implemented using the sklearn library in Python. The hyperparameters of the RF models were optimized by grid search based on the training dataset.

3.1.3. Light Gradient Boosting Machine

Light gradient boosting machine is a powerful tree-based gradient boosting framework, due to its efficiency and accuracy, which has been widely used in various machine learning tasks. To tackle the computational complexity problem of gradient boosting decision tree (GBDT), LightGBM employs two novel techniques: Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) [23]. GOSS aims to reduce the data size by keeping those data instances with large gradients and randomly dropping those instances with small gradients. The EFB is a method to reduce the number of effective features by bundling the exclusive features. LightGBM uses the histogram algorithm to find an appropriate split point, which can reduce the computational expense and prevent overfitting. In addition, LightGBM employs a leaf-wise growth strategy for growing trees, compared to the conventional level-wise one, which can reduce more loss under the same splitting times. Meanwhile, LightGBM adds a depth limit on the leaf-wise to avoid overfitting. In this study, LightGBM models were implemented using the lightgbm package in Python, and their hyperparameters were optimized by grid search.

3.1.4. Deep Neural Network

Artificial neural networks (ANNs) are computational models that process information by simulating the structure and function of biological neural networks. An ANN consists of a number of artificial neurons that are organized into layers; i.e., the input layer, hidden layer, and output layer. The model performance of an ANN is affected by three aspects: the activation function of the neurons, the learning rule, and the neural architecture itself [24]. A DNN is defined as an ANN with two or more hidden layers. Benefitting from the deep architecture, DNN can model the nonlinear relationships with high complexity between the inputs and the outputs. This makes DNN a promising modeling technique of pharmaceutical processes, in which nonlinear relationships are frequently encountered [25]. Better network performance comes from deeper layers, but which may bring an overfitting problem and also increase the training difficulty. In this study, three DNN models, respectively with 4 layers (DNN_4), 5 layers (DNN_5), and 6 layers (DNN_6), were built to construct to the stacking ensemble. Batch Normalization (BN) is a powerful technique that can not only accelerate deep network convergence but also alleviate overfitting risk [26]. By inserting a normalization layer after each hidden layer, BN was performed in the three DNNs. Moreover, to reduce overfitting, Gaussian dropout was utilized in the training of the DNNs with a drop rate of 0.02. These DNNs were implemented by the Keras package in Python and then applied to conduct all experiments in this research.

3.2. Stacking Ensemble Approach

Stacking is a prominent ensemble approach that integrates the predictions of multiple base learners via a meta learner to achieve higher prediction performance [27]. Considering a stacking ensemble with two levels (level-0 and level-1), the base learners in level-0 are trained and whose predictions are subsequently used as the input set of the meta learner in level-1. The prediction of the meta learner is the end output result. In general, each learner in level-0 provides its best estimation and appropriate meta learner needs to be selected for avoiding overfitting in level-1. The most critical principle in stacking ensemble is that the base learners should be "mutually orthogonal and span the space" [27]. The improvement on prediction performance, when the stacking approach is applied, is apparent in the presence of diversity among the base learners. The diversity can be improved mainly by the following ways: using models based on different learning strategies, training models with different data characteristics and/or samples, as well as setting different parameter values for the models. In this study, a statistical ML model (KNN), two types of ensemble models (RF and LightGBM), and three deep learning model (DNNs), described above, were employed to construct the proposed stacking ensemble model.

4. Methods

The proposed stacking ensemble method and the main modelling steps are described in detail in this section. First, data preprocessing work is introduced, which ensures the quality of the dataset. Second, an innovative feature construction method is elaborated, which was employed to improve the prediction performances of the models in this study. Third, the framework of the proposed stacking ensemble method is illustrated in detail.

4.1. Data Preprocessing

The quality of the dataset always exerts a significant effect on the predictive performance of supervised learning models. This makes data preprocessing an essential step to ensure the success of the data mining. Preprocessing work, mainly including outlier detection, was described in this subsection.

The dataset acquired consists of feature and label values, respectively, corresponding to the six simultaneous process parameters (F_1–F_6) and the real-time moisture content values. Before training, preprocessing of both datasets is necessary to yield accurate prediction. The data of the process parameters were collected by standard sensors with satisfactory quality. Therefore, the preprocessing for these feature data was only to delete several empty data therein. The moisture content data of the granules were predicted by an established NIR model with spectrum data. However, abnormal spectrums were inevitably measured under the influence of internal and external factors (the form of the material, the equipment error of the spectrometer, etc.). Further, abnormal moisture content values were predicted due to the anomalies of the spectrums, and which would finally influence the prediction accuracy of the models to be built. It was taken into consideration that the moisture content of the granules changed continuously when the process parameters were tuned. In this research, an exponentially weighted moving average (*EWMA*) approach [28] was employed to reflect the trend of change; the data values deviated, and those over a certain threshold were identified as outliers (see Figure 2). In *EWMA*, exponentially decreasing weights are assigned to further data points when calculating the average, which is in line with the actual process of granulation. Thus, the *EWMA* is formulated as

$$EWMA_t = \begin{cases} y_1, & t = 1 \\ \alpha y_t + (1-\alpha)EWMA_{t-1}, & t > 1 \end{cases} \tag{5}$$

where $EWMA_t$ is the average value at time t, y_t is the original value, and α is a constant ($0 < \alpha < 1$) that determines the decay degree of the weight. The *EWMA* was conducted on the data of each batch with an α of 0.2. After calculating the *EWMA*, the deviations of the original values were further obtained, as shown in Figure 2b; the normality of the deviations is observed in Figure 2c. The point whose deviation value exceeded the threshold can be treated as an outlier and, accordingly, deleted. The thresholds for outliers were determined based on Tukey's Method: lower threshold = lower quartile (Q_1)-step and upper threshold = upper quartile (Q_3) + step [29]. In this study, to obtain more data for the training of the prediction models, the step was set as three times the interquartile range (IQR = $Q_3 - Q_1$), as shown in Figure 2d. Finally, 175 outliers were identified—less than one percent of all data—and deleted. The remaining dataset was randomly split into a training set and testing set with a ratio of 80:20.

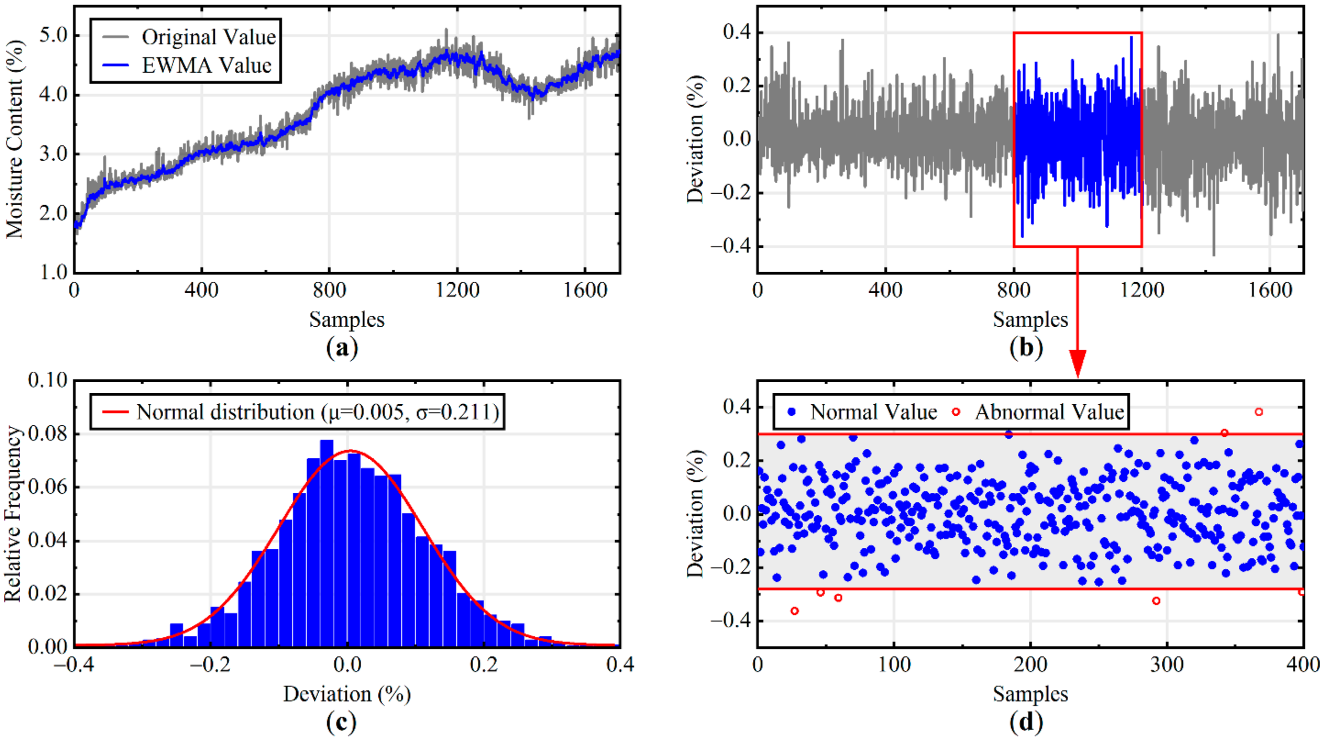

Figure 2. Outlier detection of the moisture content data: (**a**) EWMA value of the moisture content; (**b**) line graph of the deviations; (**c**) distribution of the deviations; (**d**) result of outlier detection.

4.2. Feature Construction

Feature engineering plays a crucial role in machine learning, which extracts representative features from the given data to improve the prediction performance of models. As a significant segment of feature engineering, feature construction combines original features with the objective of seeking highly predictive features. However, feature construction is a complex, time-consuming exercise in great request of domain knowledge. In this study, a SHapley Additive exPlanations (SHAP) approach was innovatively employed to provide information for the construction of new features. SHAP is a game-theoretic method used to interpret model predictions. It was recently developed by Lundberg and Lee [30], the detail of which can be found in the literature. SHAP explains the output value as a linear addition of input variables using an additive feature attribution method, and the attribution value of each variable is the SHAP value. SHAP interaction values are defined as a generalization of the SHAP values to a higher order interaction. These values reveal the hidden relationships between the variables. The summary plot of the SHAP interaction value matrix for the LightGBM model trained is shown in Figure 3. In the plot, the main effects of those features on the prediction are presented on the diagonal with the interaction effects off the diagonal. Each point represents a SHAP interaction value of that sample, colored by the variable value from low (blue) to high (red). The size of the

SHAP interaction value, i.e., the magnitude of the pairwise interaction, is indicated by the horizontal axis. In order of importance, the input variables are presented on the vertical axis from top to bottom.

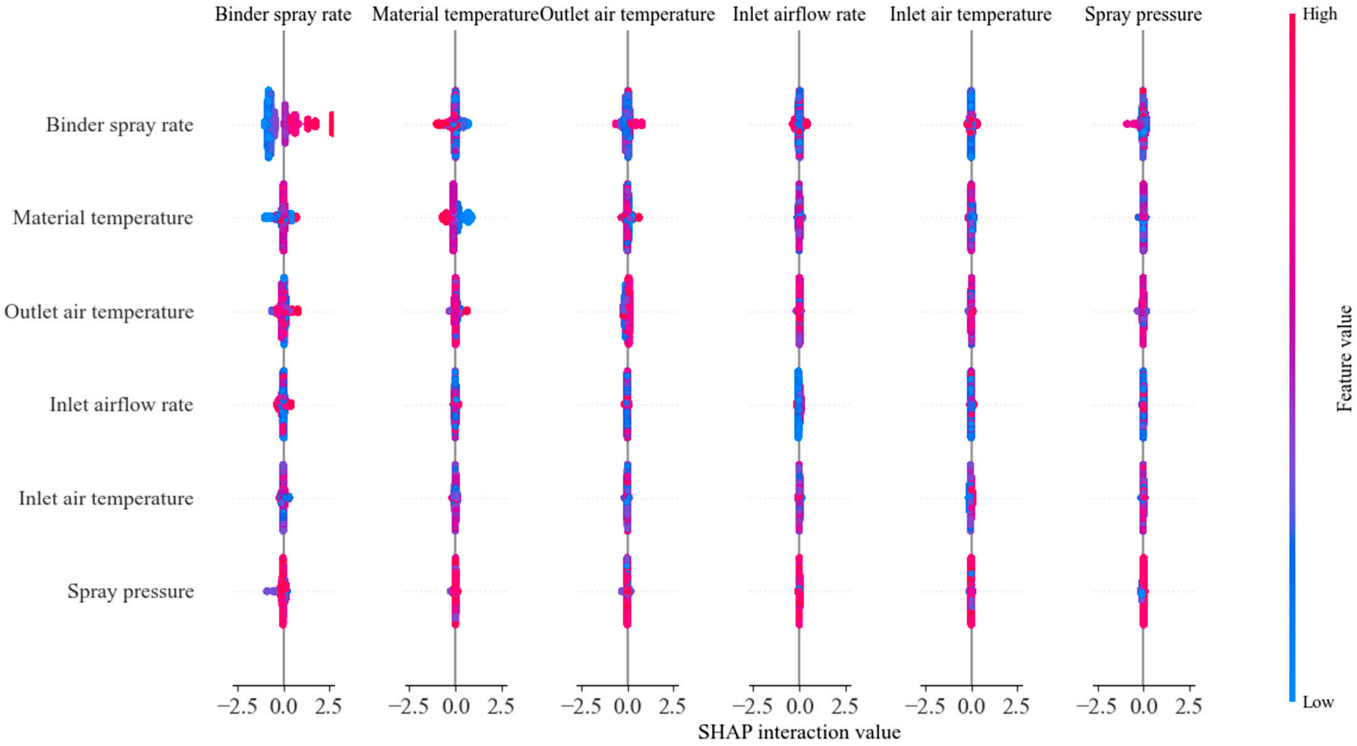

Figure 3. Summary plot for the SHapley Additive exPlanations (SHAP) interaction values of the light gradient boosting machine (LightGBM) model.

As shown in Figure 3, the parameters binder spray rate (F_6) and material temperature (F_1) are the top two most important parameters, with conspicuous main effects on the prediction. Moreover, a relatively large interaction effects exist between the pairs of the features: F_1 and F_6, F_4 and F_6, and F_5 and F_6. The SHAP value of an input variable is the sum of its main effect and interaction values. Since the SHAP value is the direct attribution value of the model prediction, the interaction between the parameters with a high SHAP interaction value can be considered to have a large impact on the model prediction. Based on this, synthetic features were constructed by combining these parameters, and their effects on different base learners were analyzed to select the optimal combination in this study.

4.3. Proposed Stacking Ensemble Method

This subsection elaborates the proposed stacking ensemble method for predicting granule moisture. Figure 4 shows the stacking ensemble framework, which consists of two learning components: the base-level learning component and meta-level combining component. In the base-level learning component, diverse base learners are trained with the given input datasets, comprising different features selected for each learner. The meta-level combining component builds an optimal ensemble of the base learners to improve prediction performance. The descriptions of the two components are provided in the following subsections.

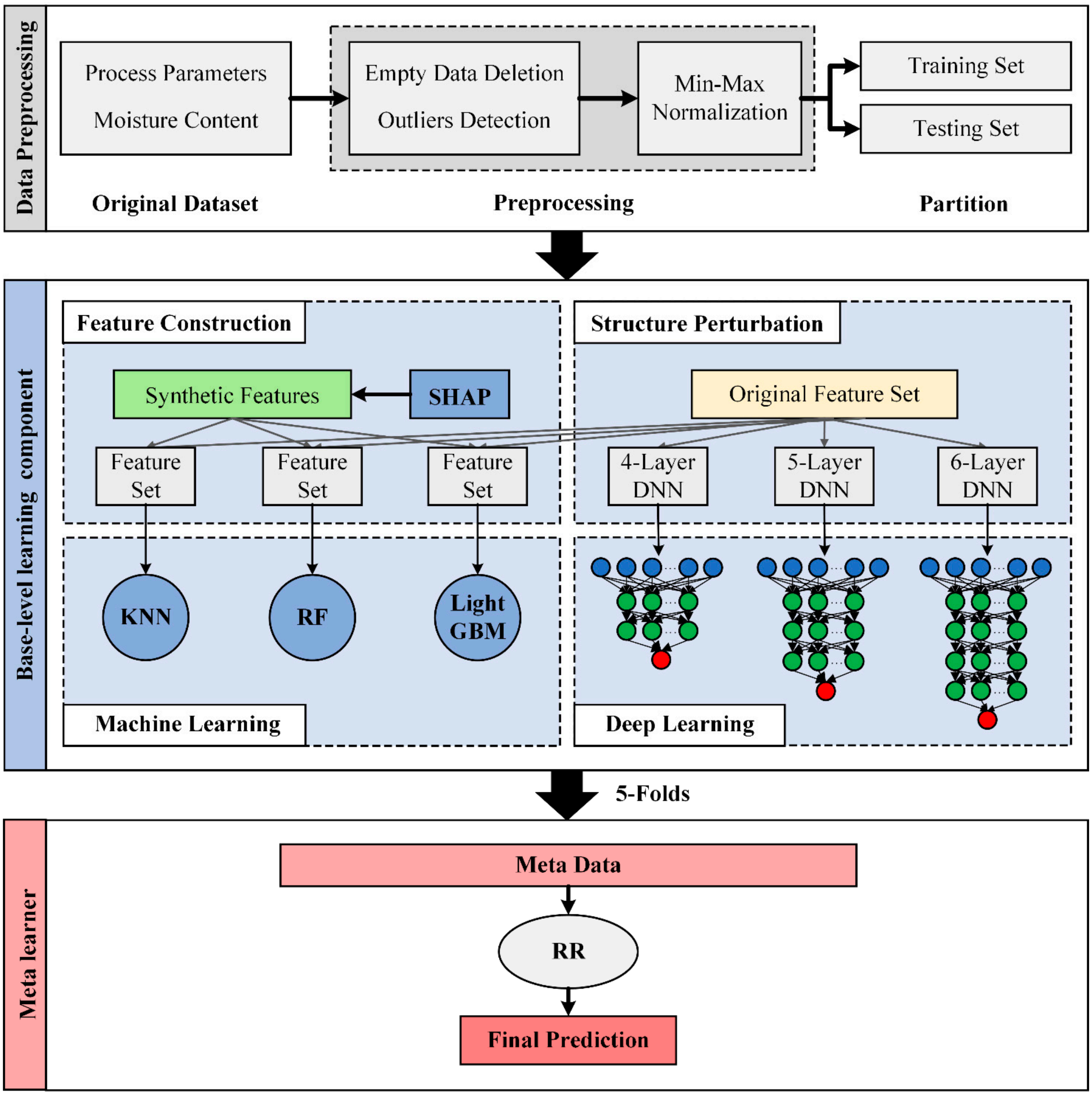

Figure 4. Framework of the proposed stacking ensemble method.

4.3.1. Base-Level Learning Component

The primary objective of the base-level learning component is to construct adequately diverse base learners with high prediction performance. To achieve the requirement of model diversity, the KNN, RF, LightGBM, and DNNs models were selected as the base learners. The KNN is a nearest-neighbor statistical algorithm base on the distance metric. The RF is a bagging ensemble algorithm with the characteristic of reducing prediction variance. Different from the RF, LightGBM is another efficient algorithm based on boosting the ensemble method, which focuses on reducing prediction bias. In addition, three DNNs with a different number of layers are constructed considering the diversity of the network structures. The divergent learning strategies of these algorithms ensure the diversity and complementarity of the base learners. The good prediction performance of the base learners also plays a crucial role in an efficient stacking ensemble. In this component, synthetic features are constructed by utilizing the SHAP approach to improve the prediction performance of the base learners. A detailed description of the feature construction is provided in Section 4.2. Different feature sets were determined as the inputs for the trainings of the base learners. In this way, not only the performance of the base learners was enhanced, but the perturbation of the input variables was implemented, which further improved the diversity. The DNNs have powerful learning capabilities but the deep learning building process is expensive. Therefore, the original feature set has been used as the input variables for the DNNs constructed in this learning component. In addition, to form the network diversity, the DNNs were constructed with a different number of layers, including 4 layers, 5 layers, and 6 layers. Combining multiple DNNs can also overcome the overfitting problem that a separate DNN model likely encounters.

The hyperparameters of the KNN, RF, and LightGBM models were optimized by a grid search with 10-fold cross-validation based on their training datasets. For the hyperparameter setting of the DNNs, the node number was 200 in each hidden layer. Moreover, rectified linear units (ReLU) was employed as the activation function to accelerate training. The DNNs were trained with the adaptive moment estimation (Adam) in 2000 epochs (batch_size: 256, learing_rate: 0.02, decay: 0.001, loss: mean square error). Before training the base learners, a Min–Max normalization was performed to eliminate the influence of dimension. In this learning level, 5-fold cross-validation was applied to train the base learners and prepare training data for the meta learner. The training dataset (80%) was divided into five mutually exclusive subsets in equal size. Each time, four subsets were used together for training the base learners and the remaining one was used for testing. This process was repeated five times for each base learner, the prediction values of which were then combined as the input training data for the meta learner in the next level. In addition, the base learners trained each time were simultaneously used to predict on the testing dataset (20%). Every learner generated five parallel prediction results, which were further averaged by learner separately. The average prediction values of each base learner were then combined as the testing data for the meta learner.

4.3.2. Meta-Level Combining Component

The combination of base learners plays a significant role in the construction of an ensemble model. Unlike the simple ensemble strategies widely used (average scoring and majority voting), the stacking strategy implements a combination by feeding the outputs of the base learners into an appropriate meta learner. The meta learner can automatically integrate the respective strengths of the base learners, by which a better and more stable performance is potentially provided. For identifying the appropriate meta learner, different types of learners, namely, least absolute shrinkage and selection operator (LASSO), KNN, support vector regression (SVR), ridge regression (RR), extra trees (ETs), gradient boosting decision tree (GBDT), and extreme gradient boosting (XGBoost), were utilized. These learners were implemented with the sklearn library in Python. Experiments based on the diverse meta learners were conducted to identify the best meta learner. From the experiment results, the RR learner achieved the best use of the base learners and their combination. Thus, the RR was determined as the final meta learner of the proposed stacking ensemble method.

4.4. Performance Evaluation

In this paper, four evaluation criteria, including mean absolute error (*MAE*), mean absolute percentage error (*MAPE*), root mean squared error (*RMSE*), and adjusted R-squared (*Adj. R^2*), were used to assess the performance of the models referred. These statistical indicators are calculated as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{6}$$

$$MAPE = 100 \times \sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{7}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{8}$$

$$Adj.R^2 = 1 - \frac{(n-1)\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{(n-k-1)\sum_{i=1}^{n}(y_i - \overline{y})^2} \tag{9}$$

where y_i and $\hat{y}_i$ denote the observed and predicted value, $\overline{y}$ is the mean for the observed values, n is the number of the predictions, and k is the number of the features.

5. Results and Discussion

This paper mainly focuses on the feature construction and the proposed stacking ensemble method. In this section, the effects of the different synthetic features constructed are firstly analyzed and, based on that, the optimal input feature set is selected for each base learner. Secondly, to validate the superior performance of the proposed method, it was compared with that of the individual base learners (KNN, RF, LightGBM, and DNNs) and five other well-known ML algorithms, namely, bagging regressor (BR), ETs, GBDT, XGBoost, and Generalized Regression Neural Network (GRNN). All experiments were conducted on a workstation with an Intel® Core$^{\text{TM}}$ i9-10940X 3.30 GHz CPU with 128 GB RAM running a Windows 10 64-bit operating system.

5.1. Results of Hyperparameters Setting

The performance of ML algorithms is highly dependent on their hyperparameters setting. This makes the determination of the optimal hyperparameters the biggest problem when using these algorithms. Accordingly, the grid search method was applied, which iterates through every parameter combination and identifies the optimal one by cross-validation [31]. In this study, the hyperparameters of each ML algorithm used were optimized by a 10-fold GridSearchCV (provided by the sklearn library) on the training dataset. Table 2 presents the optimum values of hyperparameters for these ML algorithms. Based on the selected optimal hyperparameters, the prediction performance of the algorithms was compared in the subsequent experiments.

Table 2. Optimum values of the hyperparameters.

Model	Optimum Value
KNN	n_neighbors = 12, weights = "distance", p = 1
RF	max_depth = 15, n_estimators = 500, min_samples_split = 10
LightGBM	max_depth = 12, n_estimators = 1000, learning_rate = 0.02, num_leaves = 100
DNNs	activation = "relu", optimizer = "adam", learning_rate = 0.02, epochs = 2000
BR	max_samples = 0.5, n_estimators = 500
ETs	n_estimators = 500, min_samples_split = 14, min_samples_leaf = 1
GBDT	max_depth = 10, n_estimators = 500, learning_rate = 0.01
XGBoost	max_depth = 14, n_estimators = 500, learning_rate = 0.025, min_child_weight = 10
GRNN	std = 0.008

5.2. Performance Comparison of Different Feature Sets

For better learning of non-line relation, feature discretization was performed before the feature combination. In this research, continuous features (F_1, F_4) were discretized into integers from 1 to 5 by the equal-width discretization method (EWD) [32]. The spray pressure (F_5) and the binder spray rate (F_6) were separately operated in a phased change mode during the granulation experiments. These two parameters had some discontinuity, and as a result, the raw data of which were used for the feature combination. According to the existence of an interaction, feature crosses of F_1–F_6, F_4–F_6, and F_5–F_6 were conducted to construct synthetic features in polynomial forms, as listed in Table 3.

To check the validity of the synthetic features for improving the prediction performance, 10-fold cross-validation tests repeated five times were performed based on the training set. Figure 5 shows the performance results of the four base models, separately trained on the original feature set (OFS) and the new feature sets with different synthetic features added. The performance was quantified using *Adj. R^2*. The closer the *Adj. R^2* value to 1, the better the model performance. As shown in Figure 5, the performance of the KNN model was not improved with the addition of the synthetic features. However, valid performance enhancements for the RF and LightGBM models were all obtained by the incorporation of synthetic features F_7, F_8, and F_9. In addition, the synthetic features F_{10}, F_{11}, and F_{12}, had boosting effects on the RF model only. After adding the synthetic features F_{13} and F_{14}, the performance of the LightGBM model was enhanced. It can be seen that the effects of different synthetic features vary on different models. The synthetic features with best enhancement effects were selected as the final features constructed for each group of feature cross. As a result, the synthetic features F_9 and F_{11} were selected for the RF model, F_9 and F_{13}, for the LightGBM model. Further, the prediction performance of these models was measured combining all the selected synthetic features and the OFS as input variables. Figure 5d,e shows the prediction performance of the RF and LightGBM models was further improved on the feature sets with two synthetic features compared to the ones with single synthetic features. Thus, the feature sets OFS + F_9 + F_{11} and OFS + F_9 + F_{13} were separately determined as the final input feature sets for the RF and LightGBM models. Furthermore, the OFS was determined as the final input feature set for the KNN model due to its insensitivity to the synthetic features.

Table 3. Synthetic features of the feature crosses.

Synthetic Feature	Feature Cross	Synthetic Feature	Feature Cross
F_7	F_1 * × F_6	F_{12}	F_4 * × $(F_6)^2$
F_8	$(F_1$ *$)^2$ × F_6	F_{13}	F_5 × F_6
F_9	F_1 * × $(F_6)^2$	F_{14}	$(F_5$ *$)^2$ × F_6
F_{10}	F_4 * × F_6	F_{15}	F_5 × $(F_6)^2$
F_{11}	$(F_4$ *$)^2$ × F_6		

*: Feature discretized.

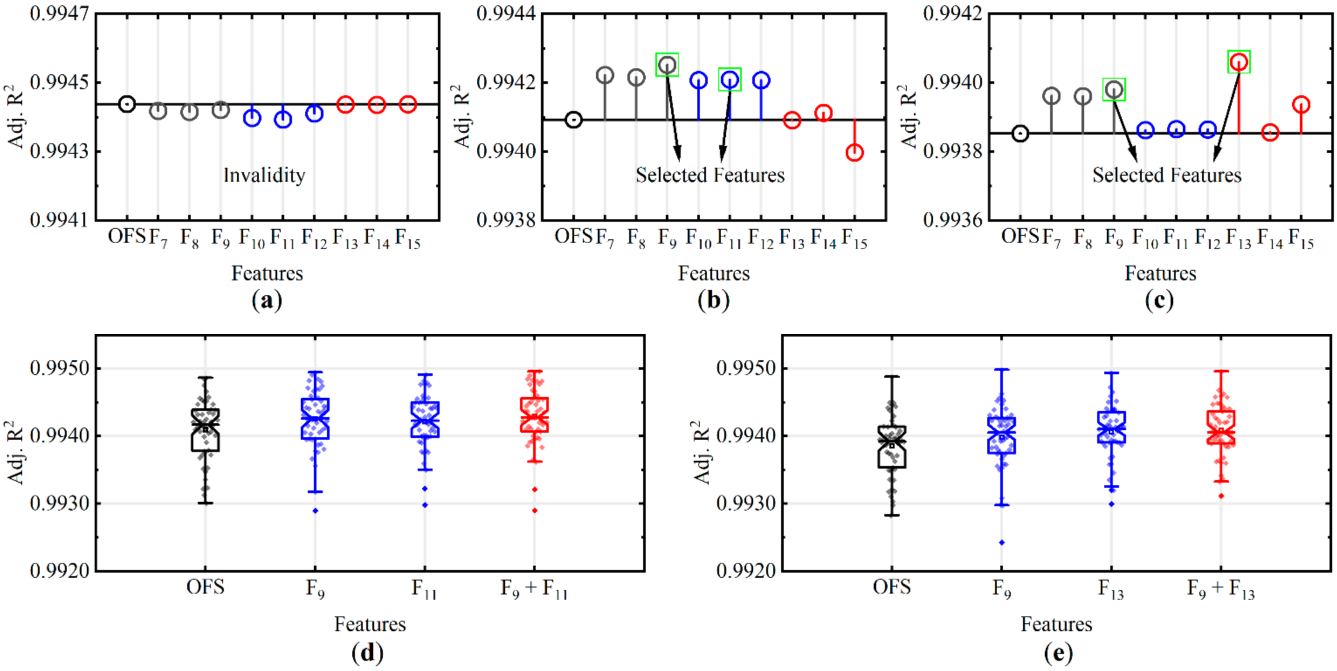

Figure 5. Performance comparison of the base learners based on feature sets composed of the original feature set (OFS) and different synthetic features: (**a**) result of k-nearest neighbors regression (KNN); (**b**) result of random forests (RF); (**c**) result of LightGBM; (**d**) boxplot for the result of RF; (**e**) boxplot for the result of LightGBM.

5.3. Performance Comparison of Different Fusion Methods

A stacking ensemble approach can be divided into two steps: base learner generation and meta-data (predictions of base learners) fusion. The performance of the stacking ensemble approach is highly dependent on its fusion method [33]. In this paper, the RR model was employed as the meta learner of the proposed stacking ensemble method. To evaluate the performance of the RR fusion, it was compared to the widely used average method and other ML algorithms. Table 4 presents the 10-fold cross-validation results for the performance evaluation of the different fusion methods. The results demonstrate the employed RR fusion strategy exhibits the best performance in terms of the four evaluation criteria. The RR is a linear regression applying an L2 regularization to avoid the over-fitting problem [34]. This makes the RR an appropriate fusion method to effectively combine the prediction results of the base learners. From Table 4, the superiority of the linear fusion methods (Lasso, RR, and average) is observed in relation to the nonlinear fusion methods (KNN, SVR, Ets, GBDT, and XGBoost). This may be associated to the fact that only six base learners were used to form the level-0 in the proposed stacking ensemble method; this results in a deficiency in the nonlinear supplementary information among the base learners.

Table 4. Performance comparison of the different fusion methods.

Method	***MAE*** **(%)**	***MAPE*** **(%)**	***RMSE*** **(%)**	***Adj. R*** ^{2}
Lasso	0.0597	1.5840	0.0845	0.99484
KNN	0.0604	1.6022	0.0849	0.99478
SVR	0.0607	1.6046	0.0849	0.99480
RR	0.0596	1.5819	0.0844	0.99485
ETs	0.0599	1.5892	0.0846	0.99483
GBDT	0.0605	1.6032	0.0853	0.99474
XGBoost	0.0605	1.6040	0.0852	0.99475
Average	0.0597	1.5841	0.0845	0.99482

In addition, to verify the effectiveness of the feature construction in the proposed method, the model performance before and after adding the synthetic features (SFs) was further compared. As shown in Figure 6, the prediction RMSEs of the model reduced after adding the SFs when employing the Lasso, KNN, SVR, RR, Ets, and average methods. Only when the GBDT and XGBoost were used as the meta learners, the prediction RMSEs increased, which suggested an over-fitting problem. The GBDT and XGBoost are boosting-based approaches and tend to over-fitting. Despite the fact that the SFs enhanced the performance of the base learners (the RF and LightGBM), they simultaneously increased the risk of over-fitting and finally resulted in the degradation of the model performance. Benefitting from L2 regularization, the RR method could solve the over-fitting problem and effectively combined the prediction results from the base learners. Therefore, the superiority of the RR fusion method was proved, and that was reasonably determined as the meta learner in the proposed stacking ensemble method.

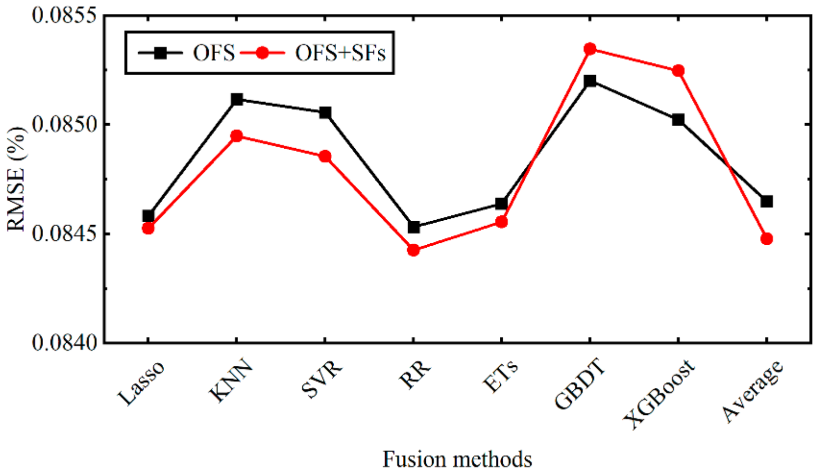

Figure 6. Prediction RMSEs of the proposed method with and without the synthetic features (SFs).

5.4. Performance Comparison of Different Models

The performance of the proposed stacking ensemble method was compared with other models in terms of *MAE*, *MAPE*, *RMSE*, and *Adj. R*^{2}. Figure 7 presents the boxplots for the 10-fold cross-validation results of each model. From Figure 7a–c, it can be seen that the proposed method gains the minimum average values of *MAE*, *MAPE* and *RMSE*, separately. This means the prediction error of the proposed method was smallest during the test. Compared to other evaluated models, the prediction results of the proposed method also have smaller distribution intervals, showing to be robust. In Figure 7d, for the proposed model, relatively higher *Adj. R*^{2} values are observed, which confirms the proposed model outperforms the other models on prediction performance.

Detailed experimental results of all the evaluated models are provided in Table 5. From the results, the *MAE*, *MAPE*, and *RMSE* for the base learners (the KNN, RF, LightGBM, DNN$_4$, DNN$_5$, and DNN$_6$) range between 0.0863–0.0912, 1.616–1.710, and 0.0612–0.0644, and they are reduced to 0.0844, 1.582, and 0.0596 with the proposed stacking method, respectively. Furthermore, the result for the *Adj. R*^{2} (0.9948) of the proposed model obtains an improvement compared to the ones (0.9940–0.9946) of the base learners. The above comparison results verify that the proposed method implemented an effective ensemble of the base learners and attained a better prediction performance.

Table 5. Performance comparison of the proposed stacking ensemble method and other models.

Model	*MAE* (%)	*MAPE* (%)	*RMSE* (%)	*Adj. R*^{2}
KNN	0.0608	1.6157	0.0867	0.99457
BR	0.0624	1.6586	0.0897	0.99418
ETs	0.0604	1.6051	0.0857	0.99469
RF	0.0625	1.6610	0.0899	0.99416
GBDT	0.0654	1.7325	0.0935	0.99368
LightGBM	0.0644	1.7104	0.0912	0.99399
XGBoost	0.0636	1.6904	0.0904	0.99409
GRNN	0.0635	1.6852	0.0908	0.99405
DNN$_4$	0.0617	1.6347	0.0871	0.99452
DNN$_5$	0.0613	1.6241	0.0865	0.99459
DNN$_6$	0.0612	1.6212	0.0863	0.99461
Proposed stacking ensemble method	0.0596	1.5819	0.0844	0.99485

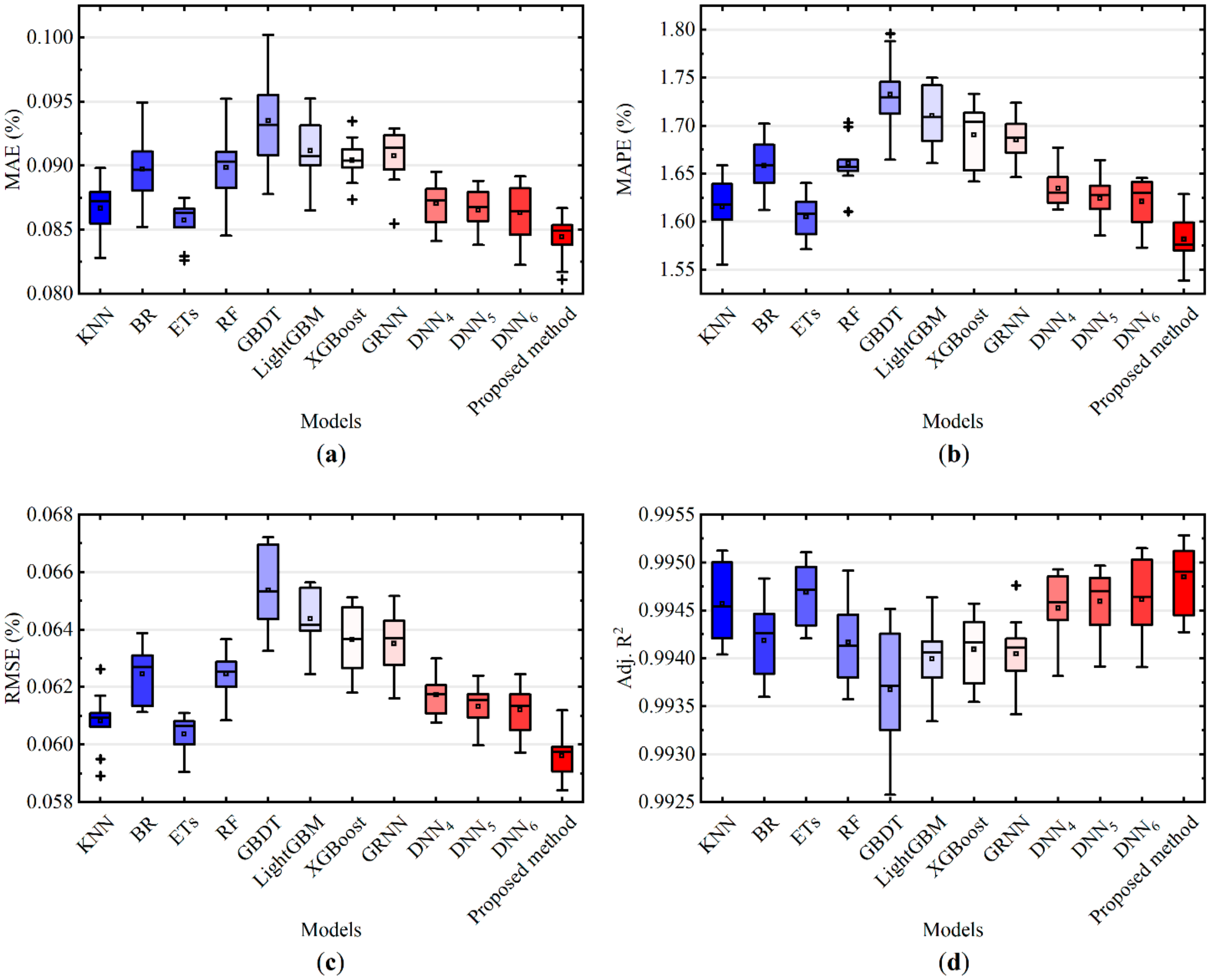

Figure 7. Results for performance evaluation of the proposed stacking method and other models: (**a**) *MAE*; (**b**) *MAPE*; (**c**) *RMSE*; (**d**) *Adj. R*^{2}.

In addition, it can be observed from Figure 7 and Table 5 that the performance of the boosting-based models (the GBDT, LightGBM, and XGBoost) is inferior to the performance of the bagging-based models (the BR and RF) in general. It is inferred that the boosting ensemble method, used to reduce prediction bias, causes the GBDT, LightGBM, and XGBoost models to overfit the dataset. Unlike the boosting ensemble method, the bagging ensemble method can reduce prediction variance, which allows the BR and RF models to mitigate the over-fitting problem. The proposed stacking method employed the RF and LightGBM as base learners, integrating the boosting and bagging ensemble methods simultaneously. This can be considered as a reason contributing to the high precision and strong robustness of the proposed model. Alongside this, Figure 7 illustrates that the KNN, DNN$_4$, DNN$_5$, and DNN$_6$ models are competitive, with a similar performance. This may be due to the fact that the information provided by the data was finite and the learning of these models for the information was close to the limit. The proposed method employed them as the base learners, fusing their respective superiority, and attained an even better performance.

For the GRNN model, it exhibits an inferior performance in relation to the three DNNs (DNN$_4$, DNN$_5$, and DNN$_6$). This suggests that the designed DNNs are more suitable for the prediction of granule moisture. In addition, the results of the three DNNs indicate that their prediction accuracy is improved while robustness becomes worse with the number of network layers increases, as shown in Figure 7. Therefore, considering both model precision and robustness, the proposed stacking ensemble method combined all the three DNNs to obtain a better performance.

From the above comparison of prediction performance, it can be concluded that the generalization capacity and robustness of the proposed stacking ensemble method outperforms the other models compared. The proposed method is proved to be a superior tool for the prediction of granule moisture content.

6. Conclusions

In this study, a novel stacking ensemble method was proposed for the prediction of granule moisture during the FBG process. For an ensemble modeled in levels, it attains good results depending on the diversity and performance of the base-level learners. In this context, the proposed method employed KNN, RF, LightGBM, and three DNNs models as the base learners. These models are orthogonal on the learning strategy, which complies with the requirement of diversity. To further improve the diversity, perturbations of the input variable and network structure were applied in the proposed method, implemented by feature construction and combination of multiple DNNs with a different number of hidden layers, respectively. In the feature construction, a SHAP approach was innovatively utilized to analyze the interactions between the process parameters and, according to which, effective synthetic features were constructed for the base learner RF and LightGBM. With the incorporation of synthetic features, the prediction performance of the RF and LightGBM was improved and it contributed to an enhanced performance of the proposed stacking ensemble method finally. This shows the SHAP approach to be an appropriate tool for feature construction, removing the great requirement of domain knowledge. In addition, experiments base on diverse fusion methods were conducted and, from the comparison results, RR method achieved the best combination of the base learners. Thus, the RR was determined as the meta learner of the proposed stacking approach.

The superior performance of the proposed stacking ensemble method for granule moisture prediction was verified through cross-validation experiments compared to other algorithms. The comparison results demonstrated that the proposed method reduced the prediction error in relation to the base learners. In addition, the proposed method outperformed other machine learning algorithms, exhibiting higher prediction performance and better robustness. Thus, the proposed stacking ensemble was proved to be an effective tool for the prediction of granule moisture during the FBG process, which could provide reliable decision support. The proposed method was validated on a dataset from only one piece of fluidized bed equipment. Therefore, further research is intended to enhance the generalization capacity of the proposed method using data from other equipment with various manufacturing conditions.

Author Contributions: Conceptualization, B.C.; methodology, B.C.; software, B.C.; validation, P.H. and J.Z.; formal analysis, B.C.; resources, M.L.; data curation, M.L.; writing—original draft preparation, B.C.; writing—review and editing, P.H. and J.Z.; visualization, B.C.; supervision, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key R&D Program of China, grant number 2019YFC1711200; the Key R&D project of Shandong Province of China, grant number 2020CXGC011001; the Shandong Key Laboratory of Computer Networks open project, grant number SKLCN-2020-08.

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. Parikh, D.M. *Handbook of Pharmaceutical Granulation Technology*; CRC Press: Boca Raton, FL, USA, 2016.
2. Arndt, O.R.; Baggio, R.; Adam, A.K.; Harting, J.; Franceschinis, E.; Kleinebudde, P. Impact of Different Dry and Wet Granulation Techniques on Granule and Tablet Properties: A Comparative Study. *J. Pharm. Sci.* **2018**, *107*, 3143–3152. [CrossRef]
3. Iveson, S.M.; Litster, J.D.; Hapgood, K.; Ennis, B.J. Nucleation, growth and breakage phenomena in agitated wet granulation processes: A review. *Powder Technol.* **2001**, *117*, 3–39. [CrossRef]
4. Esratun, J.; Abdullah Al, A.; Hasan, M.M.; Abdullah Bin, Z.; Harun Ar, R. Granulation techniques & its updated modules. *Pharma Innov.* **2016**, *5*, 134–141.
5. Shanmugam, S. Granulation techniques and technologies: Recent progresses. *Bioimpacts* **2015**, *5*, 55–63. [CrossRef]
6. Buschmuller, C.; Wiedey, W.; Doscher, C.; Dressler, J.; Breitkreutz, J. In-line monitoring of granule moisture in fluidized-bed dryers using microwave resonance technology. *Eur. J. Pharm. Biopharm.* **2008**, *69*, 380–387. [CrossRef]
7. Rossteuscher-Carl, K.; Fricke, S.; Hacker, M.C.; Schulz-Siegmund, M. Influence of in line monitored fluid bed granulation process parameters on the stability of Ethinylestradiol. *Int. J. Pharm.* **2015**, *496*, 751–758. [CrossRef]
8. Chablani, L.; Taylor, M.K.; Mehrotra, A.; Rameas, P.; Stagner, W.C. Inline real-time near-infrared granule moisture measurements of a continuous granulation-drying-milling process. *AAPS PharmSciTech* **2011**, *12*, 1050–1055. [CrossRef]
9. Hartung, A.; Knoell, M.; Schmidt, U.; Langguth, P. Role of continuous moisture profile monitoring by inline NIR spectroscopy during fluid bed granulation of an Enalapril formulation. *Drug Dev. Ind. Pharm.* **2011**, *37*, 274–280. [CrossRef]
10. Kona, R.; Qu, H.; Mattes, R.; Jancsik, B.; Fahmy, R.M.; Hoag, S.W. Application of in-line near infrared spectroscopy and multivariate batch modeling for process monitoring in fluid bed granulation. *Int. J. Pharm.* **2013**, *452*, 63–72. [CrossRef]
11. Rantanen, J.; Rasanen, E.; Tenhunen, J.; Kansakoski, M.; Mannermaa, J.; Yliruusi, J. In-line moisture measurement during granulation with a four-wavelength near infrared sensor: An evaluation of particle size and binder effects. *Eur. J. Pharm. Biopharm.* **2000**, *50*, 271–276. [CrossRef]
12. Hu, X.; Cunningham, J.; Winstead, D. Understanding and Predicting Bed humidity in Fluidized Bed Granulation. *J. Pharm. Sci.* **2008**, *97*, 1564–1577. [CrossRef]
13. Amini, H.; He, X.; Tseng, Y.C.; Kucuk, G.; Schwabe, R.; Schultz, L.; Maus, M.; Schroder, D.; Rajniak, P.; Bilgili, E. A semi-theoretical model for simulating the temporal evolution of moisture-temperature during industrial fluidized bed granulation. *Eur. J. Pharm. Biopharm.* **2020**, *151*, 137–152. [CrossRef]
14. Mendes-Moreira, J.; Soares, C.; Jorge, A.M.; Sousa, J.F.D. Ensemble approaches for regression: A survey. *ACM Comput. Surv.* **2012**, *45*, 1–40. [CrossRef]
15. Guo, H.; Nguyen, H.; Bui, X.-N.; Armaghani, D.J. A new technique to predict fly-rock in bench blasting based on an ensemble of support vector regression and GLMNET. *Eng. Comput.* **2021**, *37*, 421–435. [CrossRef]
16. Wen, L.; Hughes, M. Coastal Wetland Mapping Using Ensemble Learning Algorithms: A Comparative Study of Bagging, Boosting and Stacking Techniques. *Remote Sens.* **2020**, *12*, 1683. [CrossRef]
17. Ribeiro, M.H.D.M.; dos Santos Coelho, L. Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series. *Appl. Soft Comput.* **2020**, *86*, 105837. [CrossRef]
18. Soares, E.; Costa, P.; Costa, B.; Leite, D. Ensemble of evolving data clouds and fuzzy models for weather time series prediction. *Appl. Soft Comput.* **2018**, *64*, 445–453. [CrossRef]
19. Pinto, T.; Praça, I.; Vale, Z.; Silva, J. Ensemble learning for electricity consumption forecasting in office buildings. *Neurocomputing* **2021**, *423*, 747–755. [CrossRef]
20. Choubin, B.; Moradi, E.; Golshan, M.; Adamowski, J.; Sajedi-Hosseini, F.; Mosavi, A. An ensemble prediction of flood susceptibility using multivariate discriminant analysis, classification and regression trees, and support vector machines. *Sci. Total Environ.* **2019**, *651*, 2087–2096. [CrossRef]
21. AlAlaween, W.H.; Mahfouf, M.; Salman, A.D. Predictive modelling of the granulation process using a systems-engineering approach. *Powder Technol.* **2016**, *302*, 265–274. [CrossRef]
22. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
23. 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. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 3149–3157.
24. Nguyen, N.D.; Nguyen, V.T. Development of ANN structural optimization framework for data-driven prediction of local two-phase flow parameters. *Prog. Nucl. Energ.* **2022**, *146*, 104176. [CrossRef]
25. Zhang, K.; Lin, N.T.; Yang, J.Q.; Jin, Z.W.; Li, G.H.; Ding, R.W. Predicting gas-bearing distribution using DNN based on multi-component seismic data: Quality evaluation using structural and fracture factors. *Pet. Sci.* **2022**. [CrossRef]
26. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on International Conference on Machine Learning, Lille, France, 7–9 July 2015; Volume 37, pp. 448–456.
27. Li, Z.; Zhang, C.; Liu, H.; Zhang, C.; Zhao, M.; Gong, Q.; Fu, G. Developing stacking ensemble models for multivariate contamination detection in water distribution systems. *Sci. Total Environ.* **2022**, *828*, 154284. [CrossRef]
28. Kim, S.H.; Lee, D.H.; Kim, K.J. EWMA-PRIM: Process optimization based on time-series process operational data using the exponentially weighted moving average and patient rule induction method. *Expert Syst. Appl.* **2022**, *195*, 116606. [CrossRef]
29. Tukey, J.W. *Exploratory Data Analysis*; Addison-Wesley Pub. Co.: Boston, MA, USA, 1977; pp. 131–160.

30. Lundberg, S.M.; Lee, S.-I. A unified approach to interpreting model predictions. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 4768–4777.
31. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
32. Kotsiantis, S.; Kanellopoulos, D. In Discretization Techniques: A recent survey. *GESTS Int. Trans. Comput. Sci. Eng.* **2006**, *32*, 47–58.
33. Todorovski, L.; Džeroski, S. Combining Classifiers with Meta Decision Trees. *Mach. Learn.* **2003**, *50*, 223–249. [CrossRef]
34. Schreiber-Gregory, D.N. Ridge Regression and multicollinearity: An in-depth review. *Model Assist. Stat. Appl.* **2018**, *13*, 359–365. [CrossRef]