

## Article

# An HGA-LSTM-Based Intelligent Model for Ore Pulp Density in the Hydrometallurgical Process

Guobin Zou <sup>1,2</sup>, Junwu Zhou <sup>1</sup>, Kang Li <sup>3,\*</sup> and Hongliang Zhao <sup>4</sup> <sup>1</sup> College of Information Science and Engineering, Northeastern University, Shenyang 110004, China<sup>2</sup> State Key Laboratory of Process Automation in Mining and Metallurgy Research, Beijing 100160, China<sup>3</sup> BGRIMM Technology Group, Beijing 100160, China<sup>4</sup> School of Metallurgical and Ecological Engineering, University of Science and Technology, Beijing 100083, China

\* Correspondence: likang@bgrimm.com; Tel.: +86-10-5906-9777; Fax: +86-10-5906-6979

**Abstract:** This study focused on the intelligent model for ore pulp density in the hydrometallurgical process. However, owing to the limitations of existing instruments and devices, the feed ore pulp density of thickener, a key hydrometallurgical equipment, cannot be accurately measured online. Therefore, aiming at the problem of accurately measuring the feed ore pulp density, we proposed a new intelligent model based on the long short-term memory (LSTM) and hybrid genetic algorithm (HGA). Specifically, the HGA refers to a novel optimization search algorithm model that can optimize the hyperparameters and improve the modeling performance of the LSTM. Finally, the proposed intelligent model was successfully applied to an actual thickener case in China. The intelligent model prediction results demonstrated that the hybrid model outperformed other models and satisfied the measurement accuracy requirements in the factory well.

**Keywords:** intelligent model; thickening process; ore pulp density; long short-term memory; hybrid genetic algorithm



**Citation:** Zou, G.; Zhou, J.; Li, K.; Zhao, H. An HGA-LSTM-Based Intelligent Model for Ore Pulp Density in the Hydrometallurgical Process. *Materials* **2022**, *15*, 7586. <https://doi.org/10.3390/ma15217586>

Academic Editors: Jaroslaw Krzywanski, Yunfei Gao, Marcin Sosnowski, Karolina Grabowska, Dorian Skrobek, Ghulam Moeen Uddin, Anna Kulakowska, Anna Zylka and Bachil El Fil

Received: 9 October 2022

Accepted: 26 October 2022

Published: 28 October 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Hydrometallurgy is important in mineral resources. The hydrometallurgical process can deal with low-grade mines, complex ores, and generates fewer emissions to the environment [1]. The thickening process is a typical process of hydrometallurgy. The optimized control technology for hydrometallurgical processing is of great applicational value for the efficient utilization of metal mineral resources [2]. Optimal control of the thickening process usually depends on quality variables, such as feed density [3], which are difficult to measure online because the density of the feeding ore usually fluctuates substantially due to the existence of nonlinearity [4]. However, there is no research on the application of real-time online measurement methods of thickener feed pulp density in the actual production process [5].

To alleviate these problems, intelligent models have been used to predict wind flow around buildings by establishing inferential mathematical prediction models [6]. Owing to rapid response, accurate prediction results, and low maintenance costs, intelligent models have currently become one of the main methods for detecting quality variables in industrial processes, such as wind-induced pressure prediction [7], temperature prediction for roller kiln [8], and surface crack detection [9].

Considering the wide implementation of distributed control systems and the massive amount of available data, soft sensors based on data-driven systems are receiving increasing attention [10]. Typical data-driven modeling methods include many multivariate statistical and machine-learning methods [11–13]. Because of the limitations of the structures and parameters, some methods are limited to present strong nonlinearities and dynamics. In the past decade, deep learning has drawn increasing attention in many fields, such as intelligent

model applications [14], image classification [15], and process monitoring [16]. Compared with existing modeling methods, deep neural networks (DNNs) have a significant ability to express complex functions and learn the primary highlights of data [17]. The DNN model has shown excellent performance in processing complex and strongly nonlinear data for the development of intelligent industrial models. Most of these existing intelligent models are deep static models based on the assumptions of steady state, such as stacked auto-encoders (SAEs) [18] and deep belief networks (DBNs) [19]. Nevertheless, industrial processes are naturally dynamic, and the data series is sampled in real-time from a continuous process. Thus, to model such data sequences more accurately, the dynamic characteristics must also be considered; that is, the models must utilize past states and information to predict the present state.

Recurrent neural networks (RNNs) are dynamic neural networks. They can suffer from gradient explosion and gradient disappearance because of the memory function of past information [20]. Thus, a long short-term memory (LSTM) network which adds gate units to retain short and long-term memories is proposed to deal with this problem [21]. Recently, Zhang set up an LSTM-based network in the zinc flotation circuit to estimate the tailings grade of the first rougher [22]. Pan proposed an intelligent model based on an LSTM network to estimate the oxygen content of boiler flue gas [23]. Wensi developed a soft sensor method based on an LSTM network structure to handle the strong dynamics and nonlinearity of the process and verified its power using a sulfur recovery unit benchmark [24].

To satisfy the measurement requirements, the intelligent model focuses on constructing an accurate estimation. However, it is worth noting that LSTM network models have numerous hyperparameters that need to be continuously modified to obtain the most suitable results, such as the time window size and network structure [25]. Selecting the best hyperparameters is essential to optimize the validation errors, but it is extremely time-consuming. Therefore, the most commonly used method in hyperparameter estimation is the trial-and-error method based on heuristics. However, the limitations of computation level and time make it impossible to traverse the entire parameter space [26]. Thus, a new method is needed to optimize the verification error for both boosting accuracy and saving time, which can ensure the accuracy of the soft sensors in industrial processes. The genetic algorithm (GA) is a classic global optimal method developed by imitating the natural biological evolution mechanism and has attracted much attention in hyperparameter optimization. Most recently, Alshwaheen proposed an LSTM-RNN model to forecast the deterioration of ICU patients and used a modified GA to optimize the observation window to increase the accuracy [27]. Danial developed both ANN and GA-based ANN techniques for the prediction of AOP [28]. Zhang et al. combined a support vector machine (SVM) with GA to predict the moisture in oil-immersed insulations and obtained highly accurate results [29].

In this research, an intelligent model method based on LSTM and the hybrid genetic algorithm (HGA) was proposed to measure the feed ore pulp density in the thickener process. In particular, we applied the sequential quadratic programming (SQP) algorithm, which can perform fast and accurate local searches in GA and significantly increase the global searching ability of the algorithm. The GA-SQP, also called HGA, is used to optimize the hyperparameters to determine the time window and the structure parameters of the LSTM network based on the lowest verification error, which can enhance the performance of the LSTM. Finally, an intelligent modeling method was applied to a real thickener in China.

The contributions of this study:

1. We introduced an intelligent model to resolve the difficulty encountered in measuring the feed density in the thickener through online real-time detection in hydrometallurgy.
2. A novel intelligent modeling method combining SQP, GA, and LSTM was developed to address the nonlinear and dynamic background. The HGA algorithm was used to optimize the hyperparameters of the LSTM.

- From the perspective of actual cases, the results show that the method fulfills the measurement requirements in the factory.

## 2. Methodology

### 2.1. LSTM Network

An RNN is a dynamic neural network with an internal connection that can utilize past information and past states for the present state estimation [30]. Figure 1 is the structure of the RNN in the time step. RNNs also have a hidden state vector or memory and generate an output. The RNN has difficulty learning long input sequences and can easily produce gradient explosion or disappearance.

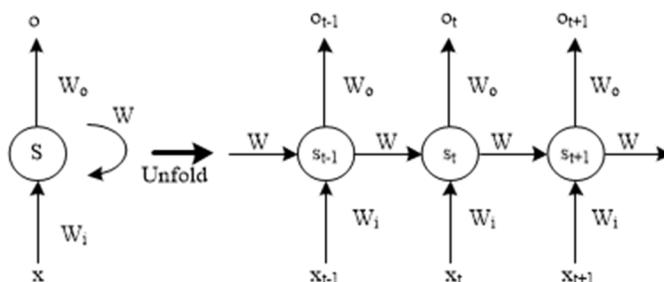


Figure 1. The simplified structure of RNN.

LSTM, a variant of the RNN architecture, designs a unique LSTM unit that can preserve past information and past states and learn sequential information with long-term dependencies. In Figure 2, the structure of the LSTM is presented, and the LSTM forward calculation formulas are as follows [21]:

$$f_t = \sigma(W_f \times [h_{t-1}, x_t, c_{t-1}] + b_f) \tag{1}$$

$$i_t = \sigma(W_i \times [h_{t-1}, x_t, c_{t-1}] + b_i) \tag{2}$$

$$\hat{c}_t = \tanh(W_c \times [h_{t-1}, x_t, c_{t-1}] + b_c) \tag{3}$$

$$c_t = f_t \times c_{t-1} + i_t \times \hat{c}_t \tag{4}$$

$$o_t = \sigma(W_o \times [h_{t-1}, x_t, c_{t-1}] + b_o) \tag{5}$$

$$h_t = o_t \times \tanh(c_t) \tag{6}$$

$$\hat{y}_t = (W_y \times h_t + b_y) \tag{7}$$

where  $\sigma$  and  $\tanh$  represent the sigmoid and tanh activation functions,  $W$  and  $b$  represent the matrices of the weight parameter and the bias, respectively, and the subscripts “ $i$ ”, “ $f$ ”, “ $c$ ”, “ $o$ ”, and “ $y$ ” represent the input-gate, forgetting-gate, update-gate, and output-gate, respectively.

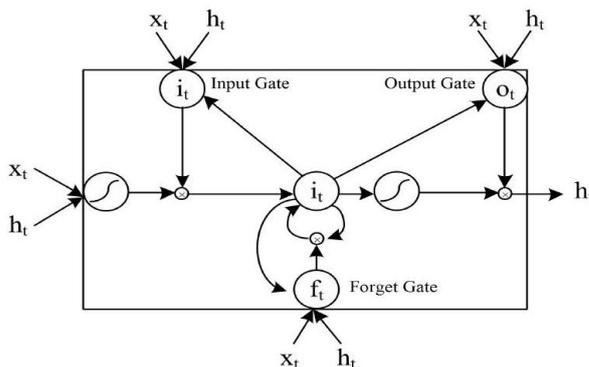


Figure 2. Internal structure of LSTM.

Through the calculation method, the long-term dependence on traditional RNN training can be overcome by the LSTM architecture effectively [31].

## 2.2. Hybrid Genetic Algorithm

GA is an adaptive heuristic optimization algorithm based on a computational model simulating the natural evolution process and has been used to determine near-optimal solutions. The algorithm uses mathematics and computer simulation to transform the process of problem-solving into a process of chromosome mutation and crossover in natural biological evolution mechanisms. Compared with some traditional optimization methods, the GA can usually obtain better optimization results more quickly when solving more complex system problems. However, the local search efficiency of a typical GA is low and time-consuming. With the stage of evolution, lower search efficiency and multiple calculations are required to achieve the final convergence [32].

The SQP algorithm is an effective method for solving nonlinear optimization problems [33]. Compared with other methods, the SQP has high computational efficiency, good convergence, and strong boundary-searching ability. The nonlinear optimization problem is expressed as follows:

$$\begin{cases} \min f(x) \\ \text{subject to } g_i(x) \leq 0 \quad (i = 1, 2, \dots, m_p) \\ g_i(x) = 0 \quad (i = m_p + 1, \dots, m) \end{cases} \quad (8)$$

where  $f(x)$  represents the objective optimization function and  $g_i(x)$  represents the boundary conditions. The subproblem is obtained by approximating the language function quadratically and linearizing the nonlinear constraints.

$$L(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i g_i \quad (9)$$

where  $\lambda_i$  is a language factor. The Hessian matrix is approximated by the quasi-Newtonian. At each  $x_k$ , the quadratic programming (QP) subproblem is obtained by linearizing the nonlinear constraints.

$$\begin{cases} \min \frac{1}{2} d^T H_k d + \nabla f(x_k)^T d \\ \text{subject to } \nabla g_i(x_k)^T + g_i(x_k) \leq 0 \quad (i = 1, 2, \dots, m_p) \\ \nabla g_i(x_k)^T + g_i(x_k) = 0 \quad (i = m_p + 1, \dots, m) \end{cases} \quad (10)$$

$x_k$ ,  $\lambda_k$ , and  $H_k$  are the approximations of the solution, multiplier, and Hessian of the language function, respectively.

The search direction  $d_k$  of the current iteration can be obtained using the above formula, and an iteration point is calculated using the formula:

$$x_{k+1} = x_k + a_k d_k \quad (11)$$

As a result, an HGA has been proposed by integrating the SQP algorithm and GA [34]. The HGA refers to a novel optimization search algorithm model, and the SQP can perform a fast and accurate local search in the GA to significantly increase the global search ability. First, using the excellent global search ability of the GA, all solutions in the solution space can be searched quickly, and some convergence values can be obtained without falling into the trap of local optimal solutions with rapid gradient descent. Moreover, the convergent result of each iteration can be the initial value of the SQP. Subsequently, the SQP search algorithm is used to implement a powerful local search designed to pursue a global optimal solution. Briefly, the HGA perfectly unites excellent global and fast local search capabilities.

In Figure 3, the specific steps of the hybrid algorithm are listed below. First, we determined a series of convergent populations that fulfilled the constraints through the

GA and selected and retained suitable individuals to solve the initial value of the SQP. Second, we constructed a multiplier function to determine whether the prediction criteria were satisfied. When the prediction criteria were not met, the vector of the local search was identified, and the minimum point in the direction was continuously determined.

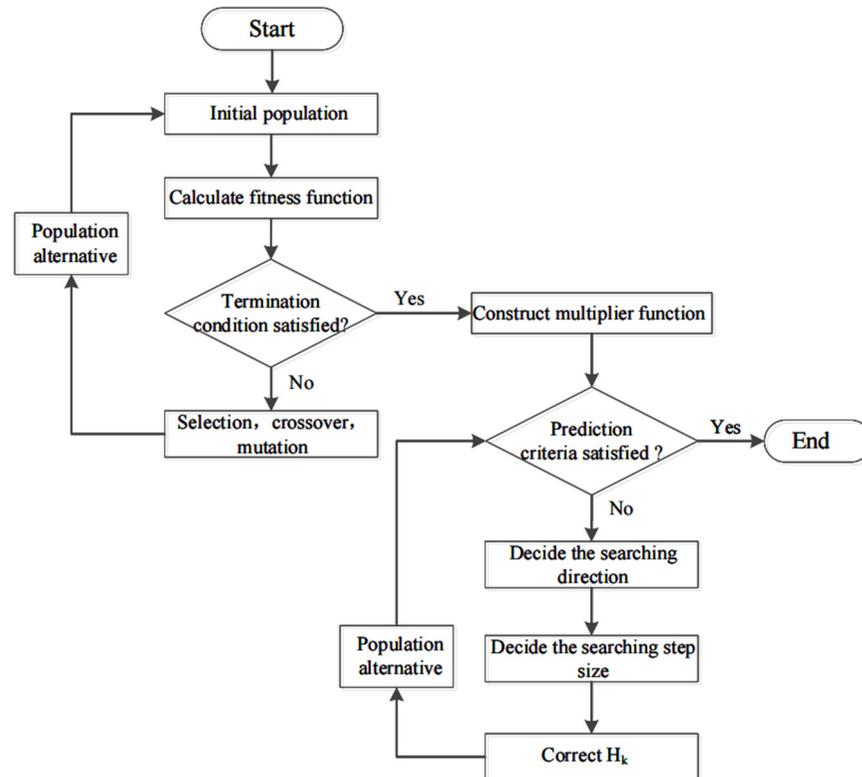


Figure 3. Flowchart of hybrid genetic algorithm.

### 3. Process Description

This work focused on mineral processing and attempted to solve the dilemma of the online measurement of the feed density in a thickener for gold smelting in China, as shown in Figure 4.



Figure 4. Layout of the thickener.

Under the flotation process, the ore slurry is concentrated by a thickener. In this work, to measure the feed concentration properly, we focused on the feed process. The slurry produced by the flotation process is merged into the slurry pump pool and then discharged from the slurry pump to the thickener for the thickening process. The slurry pipeline is equipped with a flow meter. The velocity of the feed slurry discharged into the

thickener can be detected online but not the density. The feed slurry enters the thickener for settlement, and the slurry flows out from the bottom discharge port into the dehydration process. It is impossible to calculate the real-time output and cumulative output of the slurry, which makes it difficult to achieve optimal control of the production process. Because the feed density in the thickener is currently the key index for the thickening process, this study focused on the online measurement of the feed density. The process flow chart from flotation to thickening is shown in Figure 5.

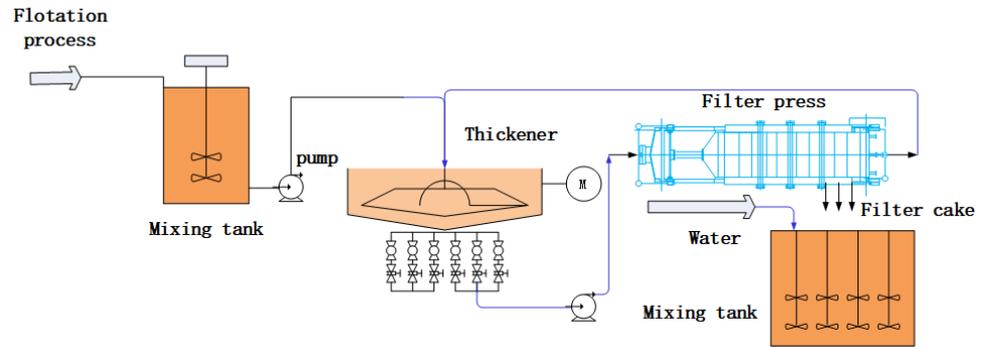


Figure 5. Process flow chart from flotation to thickening.

#### 4. Intelligent Model Based on HGA-LSTM

The intelligent model modeling process based on the LSTM network is shown in Figure 6.

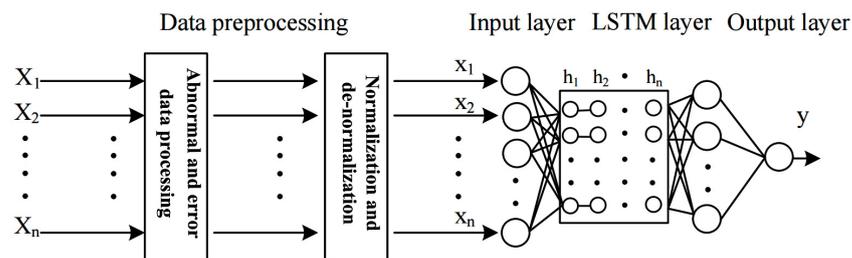


Figure 6. Modeling process of the intelligent model.

##### 4.1. Data Preprocessing

The data were measured using local detection devices, which have gross errors and random errors. Therefore, raw data cleaning was necessary.

The  $3\sigma$  criterion was used to eliminate abnormal data in this study, as follows:

The sample was set as  $x_1, x_2, \dots, x_n$ , and then the  $3\sigma$  of the sample was calculated according to the following formulas.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \tag{12}$$

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \tag{13}$$

When data  $x_d (1 \leq d \leq n)$  satisfies Equation (13), the data are considered abnormal data or error data and should be removed.

$$|x_d - \bar{x}| > 3\sigma \tag{14}$$

In addition, the normalized and de-normalization equations are expressed as:

$$X_i = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{15}$$

$$x_i = X_i \times (x_{max} - x_{min}) + x_{min} \quad (16)$$

$x_{max}$  and  $x_{min}$  represent the maximum and minimum values of the sample set,  $X_i$  represents the normalized value, and  $x_i$  represents the de-normalized value.

#### 4.2. LSTM Network Training and Hyperparameter Optimization

As mentioned above, hyperparameter optimization, including the time window and network structure parameters, can affect the results of the LSTM network. Therefore, this work adopted an HGA-LSTM network model. Usually, neural networks with better structures have advantages in updating weights, which also may lead to additional calculations and longer training and testing times. Thus, the structure parameters of the neural network must be suitable for the training set. In addition, because the LSTM network can make good use of the past time in the training process, selecting an appropriate sliding time window size results in a vast difference in the performance of the network. A window with a small size causes the model to ignore significant information, and that with a large size overuses the data during training. As a result, to achieve better performance of the intelligent model, it is necessary to identify the best parameters, especially the time window and the network structure.

The learning process of the HGA-LSTM algorithm has two main stages. In the first stage, the learning involved designing and selecting reasonable LSTM network parameters. Using the HGA method, the time window size, number of units per hidden layer, and number of hidden layers are calculated. Two activation functions are commonly used in the LSTM model: the tanh function, which is utilized as a state activation function of the input nodes and hidden nodes, and the sigmoid function is used for the gates. To improve the generalization ability of the LSTM model, dropout is necessary to effectively reduce data overfitting. Furthermore, a gradient-based “Adam” optimizer adjusts the initialized random weight of the network, which is appropriate for problems with large-scale parameters and data.

In the second stage, to evaluate the fitness of the HGA strategy, various optimization parameters were utilized. First, the population of chromosomes with a feasible solution was initialized with random values. Each chromosome[N] contains the hyperparameters in LSTM, i.e., [N] = [time windows, number of LSTM hidden-layers, number of fully connected hidden-layers, number of units per hidden layer].

In addition, the initialized chromosomes were encoded in binary bits in this study, which represented the time window size, the number of hidden layers, and the number of units per hidden layer. Based on the selection, crossover, and mutation, the solution space was constantly searched to identify the optimal solution. In the fitness function, the performance of this model is evaluated by the root mean square error (RMSE) and the average relative error (ARGE). The RMSE and ARGE were calculated using the following formulas:

$$ARGE = \frac{\sum_{i=1}^N \left| \frac{y_i - pred_i}{y_i} \right|}{N} \quad (17)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (y_i - pred_i)^2}{N}} \quad (18)$$

For population selection, this study used both the roulette wheel selection and the elitism policy such that chromosomes with higher fitness values had a higher probability of being selected, and the best chromosome in the current chromosome could always be selected. Meanwhile, the SQP algorithm was utilized to quadratically optimize the convergence value of the GA; that is, the set convergence value was used as the SQP initial value, and a new fitness function was fitted such that the SQP could be used for an accurate search to achieve a more accurate convergence. When selecting additional decision vectors

of the GA to fit a new fitness function, bad initial vectors may be introduced into the result, and when fewer decision vectors are selected, the fitness function cannot be fitted more realistically. Therefore, we selected five decision vectors to fit the SQP fitness function through trial experiments. Finally, we discretized the decision vector obtained through the optimization of the SQP algorithm.

$$\begin{cases} \min fitness(x) \\ \text{subject to } g_i(x) \leq 0 \quad (i = 1, 2, \dots, m_p) \\ g_i(x) = 0 \quad (i = m_p + 1, \dots, m) \end{cases} \quad (19)$$

$fitness(x)$  represents the objective optimization fitted by the decision vector, that is, the RMSE of the LSTM test set, and  $g_i(x)$  represents the boundary conditions or the limitation of the LSTM network structure and time window. The HGA-LSTM flow diagram is shown in Figure 7.

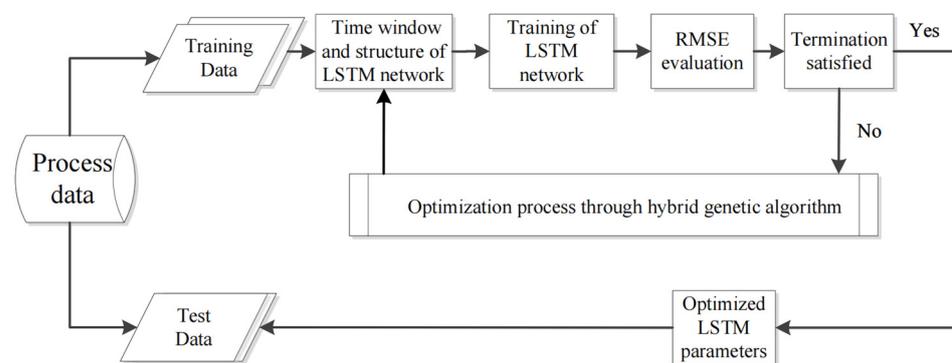


Figure 7. Flow diagram of the HGA-LSTM algorithm.

In Algorithm 1, the pseudo-code of the HGA-LSTM is shown.

---

#### Algorithm 1 HGA-LSTM steps

---

1. Divide the raw data into a test set and training set;
  2. Use the test data to evaluate the LSTM;
  3. Set GA parameters, and initialize population  $p$  randomly;
  4. Select the RMSE of LSTM in the testing set as the fitness function of GA;
  5. While the prediction criteria are not satisfied:
    - (a) Select befitting parents from the population;
    - (b) Generate a new population through crossover and mutation of chromosomes;
    - (c) Consider the individual chromosome that includes the time window, hidden layers, and number of hidden units per hidden layer into the LSTM to evaluate the fitness of the new population;
 End
  6. Set the five fast convergence values  $x_0$  of output GA as the initial values of SQP, fit a modified fitness function, and set  $k = 0$ ;
  7. Calculate the quasi-Newton approximation matrix  $H_k$  of the language function using the BFGS method at  $x_k$ ;
  8. Calculate the search direction  $d_k$ , and select the appropriate step length parameter  $a_k$ ;  
If satisfactory, stop; else set  $x_{k+1} = x_k + a_k d_k$ ,  $k = k + 1$ , and return to step 7;
  9. Discrete and output the optimal solution of the SQP, which is the hyperparameter of the LSTM network;
  10. Use the well-trained LSTM network for soft sensor modeling, and evaluate the predicted results.
- 

## 5. Experimental Results

### 5.1. Dataset Description

In industrial applications, when the nature of the conveying feed slurry is stable, the pump runs stably, the power frequency of the pump motor is fixed, and the pipeline

characteristics are stable, there is a corresponding relationship between the feed density and the current, frequency, and velocity of flow [35]. Thus, the three related indicators were selected as the auxiliary variable set and are listed in Table 1.

**Table 1.** Auxiliary variables used in the soft sensing models.

Variable	Description
$x_1$	current of the pump
$x_2$	frequency of the pump
$x_3$	flow rate of the pump

Raw data is collected at a sampling rate of one high-quality sample every 10 s in 24 h from an actual thickener working in China. There were 8640 samples in the dataset, including one target variable and three process variables. The time series data of the variables listed in Table 1 were selected as the input, and the feed density in the dataset served as the output. To develop the intelligent model, the initial 80% of the data is used to train the network; 20% of the data is used to test the performance of the method.

## 5.2. Results Analysis

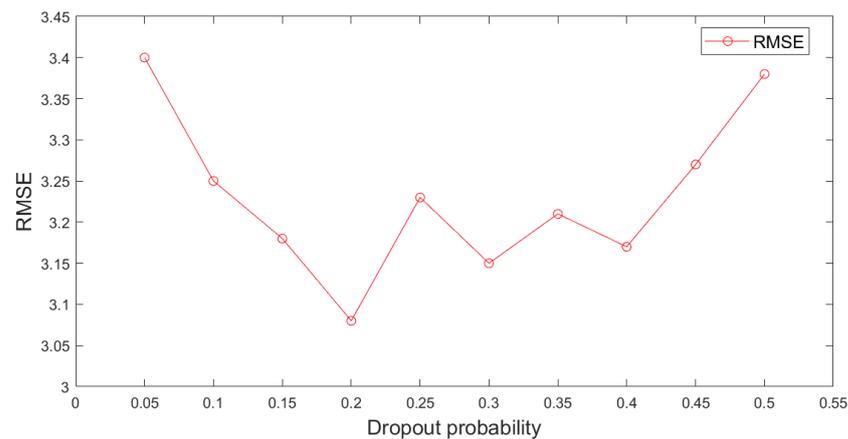
The development work of the intelligent model was conducted on a computing server with an Intel(R) Xeon(R) CPU E5-2620 v4 @2.10 GHz (two processors) and NVIDIA GeForce RTX 2080 Ti. The software environment was Windows Server 2019, Tensorflow-gpu 1.14.0, Python 3.7, and Keras 2.3.1. Taking full advantage of GPU computing, the LSTM training time for each epoch was less than 2mathrm{~s} with a total of 30 epochs. In view of the proposed HGA-LSTM model, a simulation on the abovementioned dataset was conducted. The experimental results and analysis are presented as follows: first, the output of the hyperparameter optimization from the HGA process is illustrated. Then, the prediction results of the HGA-LSTM are provided. Finally, we compare the performance of the proposed model with other models.

In the experiment, the initial population size of the GA was set to 30, with a mutation rate of 0.1, a crossover rate of 50%, and the generations set to 20 as the stop condition. For the feed density prediction, the structure of the LSTM network and time windows were optimized using the GA and HGA, respectively. In both the GA and HGA, the hidden layer contains a fully connected layer and two LSTM layers. The other parameter configurations of the HGA-LSTM and GA-LSTM were consistent with those of the LSTM. In Table 2, the specific optimization outcomes are listed.

**Table 2.** Specific optimization outcomes based on GA and HGA.

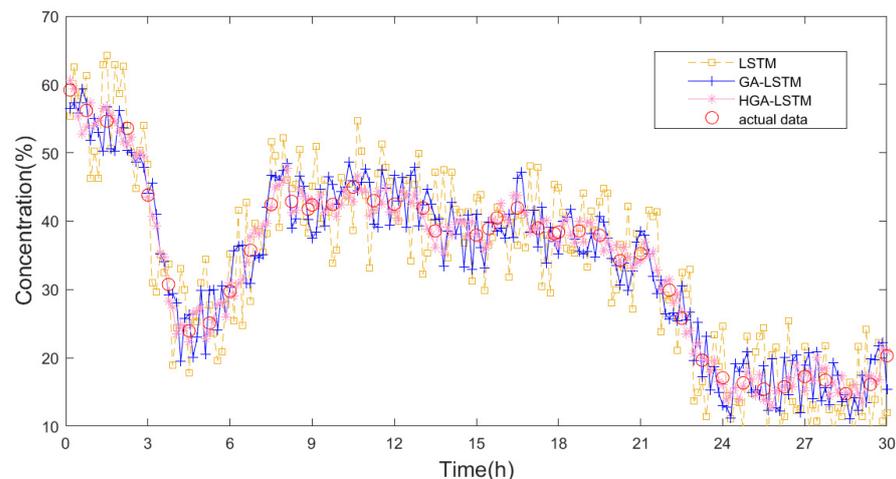
Parameter	GA-LSTM	HGA-LSTM	LSTM
Time windows	11	11	10
Number of LSTM hidden-layers	2	2	2
Number of fully connected hidden-layers	1	1	1
LSTM units on the first layer	94	88	90
LSTM units on the second layer	55	53	60
Fully connected units on the third layer	70	72	80

In addition, a dropout is necessary for reducing data overfitting effectively, and thus, it improves the model generalization ability. Selecting a befitting dropout rate is crucial because if the dropout probability is very low, it leads to an underfitting phenomenon, while an excessively high dropout probability loses the benefits of adding layers. Therefore, when the accuracy of the predicted outcome does not reach the required value, the outcome is refreshed continuously by updating the dropout to ensure the required prediction accuracy. The results are presented in Figure 8.



**Figure 8.** Auxiliary variables used in the intelligent model.

When the dropout probability reached 0.2, the *RMSE* of the predicted value of the feed density in the thickener was approximately 3.08, and there was a good match between the actual and predicted data. However, when the dropout probability increased or decreased, the *RMSE* significantly increased, and the mismatch became significant. In summary, the dropout probability was optimized in a trial-and-error manner using feedback from the predictive data. Finally, the model results of the feed ore density based on the HGA-LSTM, GA-LSTM, and LSTM were obtained, as presented in Figure 9.



**Figure 9.** The predicted value of LSTM, GA-LSTM, and HGA-LSTM.

The LSTM model was optimized through trial and error based on heuristics using the feedback of the predicted data. The performance of the LSTM, GA-LSTM, and HGA-LSTM on the testing dataset is presented in Table 3.

**Table 3.** Comparison results of LSTM, GA-LSTM, and HGA-LSTM.

Method	<i>RMSE</i>	Improvement (%)	<i>ARGE</i>	Improvement (%)
LSTM	3.83	-	0.119	-
GA-LSTM	3.21	15.45	0.0839	26.5
HGA-LSTM	3.08	19.5	0.0752	36.8

As shown in Table 3, the HGA-LSTM model is significantly better than the GA-LSTM and LSTM models. The predicted *RMSE* values of the LSTM, GA-LSTM, and HGA-LSTM models were 3.83, 3.21, and 3.08, respectively, and the predicted result was enhanced by 15.45% and 19.5% compared to those of the GA-LSTM and LSTM models, respectively. The

predicted *ARGE* values of the GA-LSTM and HGA-LSTM models were 0.0839 and 0.0752, respectively, and the predicted outcome was enhanced by 26.5% and 36.8% compared to those of the GA-LSTM and LSTM models, respectively.

The outstanding performance derived from the HGA-LSTM model was probably because the architecture of the LSTM network and time window were optimized effectively by combining the excellent global and local searching capabilities. The results demonstrate that the proper adjustment of parameters plays a critical role in achieving the desired performance. Thus, the highly effective proposed method can be used to ascertain the optimal hyperparameters for intelligent models based on deep learning algorithms, and this work expresses the potential for its application in actual industrial cases.

## 6. Conclusions

In this study, an efficient and potentially intelligent model was proposed to determine the feed ore density in a thickener. The main strategy was to employ an LSTM function optimized by an HGA combining the SQP algorithm and GA. In this method, the HGA was used to search the appropriate hyperparameters of the LSTM to improve the modeling performance. Finally, the proposed intelligent model based on the HGA-LSTM was successfully applied to an actual thickener case. This work can be extended to other scenarios where online measurement of pulp concentration is required.

**Author Contributions:** Data curation, G.Z. and J.Z.; Formal analysis, G.Z.; Funding acquisition, K.L.; Methodology, G.Z. and K.L.; Software, H.Z.; Writing—Original draft, K.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** The support from the following foundations: the National Key R\&D Program of China (2019YFE0105000), the State Key Laboratory of Process Automation in Mining, and the Metallurgy/Beijing Key Laboratory of Process Automation in Mining and Metallurgy Research: BGRIMM-KZSKL-2020-01.

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

## References

1. La Brooy, S.; Linge, H.; Walker, G. Review of gold extraction from ores. *Miner. Eng.* **1994**, *7*, 1213–1241. [[CrossRef](#)]
2. Yan, H.; Wang, F.; He, D.; Wang, Q. An operational adjustment framework for a complex industrial process based on hybrid Bayesian network. *IEEE Trans. Autom. Sci. Eng.* **2020**, *17*, 1699–1710. [[CrossRef](#)]
3. Tan, C.K.; Bao, J.; Bickert, G. A study on model predictive control in paste thickeners with rake torque constraint. *Miner. Eng.* **2017**, *105*, 52–62. [[CrossRef](#)]
4. Jia, R.; Zhang, B.; He, D.; Mao, Z.; Chu, F. Data-driven-based self-healing control of abnormal feeding conditions in thickening–dewatering process. *Miner. Eng.* **2020**, *146*, 106141. [[CrossRef](#)]
5. Jiang, Y.; Yin, S.; Dong, J.; Kaynak, O. A review on soft sensors for monitoring, control, and optimization of industrial processes. *IEEE Sens. J.* **2020**, *21*, 12868–12881. [[CrossRef](#)]
6. Kim, B.; Lee, D.E.; Preethaa, K.S.; Hu, G.; Natarajan, Y.; Kwok, K. Predicting wind flow around buildings using deep learning. *J. Wind Eng. Ind. Aerodyn.* **2021**, *219*, 104820. [[CrossRef](#)]
7. Kim, B.; Yuvaraj, N.; Sri Preethaa, K.R.; Hu, G.; Lee, D.E. Wind-Induced Pressure Prediction on Tall Buildings Using Generative Adversarial Imputation Network. *Sensors* **2021**, *21*, 2515. [[CrossRef](#)]
8. Dai, J.; Chen, N.; Yuan, X.; Gui, W.; Luo, L. Temperature prediction for roller kiln based on hybrid first-principle model and data-driven MW-DLWKPCR model. *ISA Trans.* **2020**, *98*, 403–417. [[CrossRef](#)]
9. Kim, B.; Yuvaraj, N.; Sri Preethaa, K.R.; Arun Pandian, R. Surface crack detection using deep learning with shallow CNN architecture for enhanced computation. *Neural Comput. Appl.* **2021**, *33*, 9289–9305. [[CrossRef](#)]
10. Kadlec, P.; Gabrys, B.; Strandt, S. Data-driven soft sensors in the process industry. *Comput. Chem. Eng.* **2009**, *33*, 795–814. [[CrossRef](#)]
11. Kaneko, H.; Funatsu, K. Application of online support vector regression for soft sensors. *AIChE J.* **2014**, *60*, 600–612. [[CrossRef](#)]
12. Yang, K.; Jin, H.; Chen, X.; Dai, J.; Wang, L.; Zhang, D. Soft sensor development for online quality prediction of industrial batch rubber mixing process using ensemble just-in-time Gaussian process regression models. *Chemom. Intell. Lab. Syst.* **2016**, *155*, 170–182. [[CrossRef](#)]
13. Pisa, I.; Santín, I.; Vicario, J.L.; Morell, A.; Vilanova, R. ANN-based soft sensor to predict effluent violations in wastewater treatment plants. *Sensors* **2019**, *19*, 1280. [[CrossRef](#)] [[PubMed](#)]
14. Popli, K.; Afacan, A.; Liu, Q.; Prasad, V. Development of online soft sensors and dynamic fundamental model-based process monitoring for complex sulfide ore flotation. *Miner. Eng.* **2018**, *124*, 10–27. [[CrossRef](#)]

15. Mou, L.; Ghamisi, P.; Zhu, X.X. Deep recurrent neural networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3639–3655. [[CrossRef](#)]
16. Yan, H.; Wang, F.; He, D.; Zhao, L.; Wang, Q. Bayesian network-based modeling and operational adjustment of plantwide flotation industrial process. *Ind. Eng. Chem. Res.* **2020**, *59*, 2025–2035. [[CrossRef](#)]
17. Zhou, Q.; Ooka, R. Comparison of different deep neural network architectures for isothermal indoor airflow prediction. In *Building Simulation*; Springer: Berlin/Heidelberg, Germany, 2020; Volume 13, pp. 1409–1423.
18. Yuan, X.; Qi, S.; Wang, Y. Stacked enhanced auto-encoder for data-driven soft sensing of quality variable. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 7953–7961. [[CrossRef](#)]
19. Yan, H.; Wang, F.; Yan, G.; He, D. Hybrid approach integrating case-based reasoning and Bayesian network for operational adjustment in industrial flotation process. *J. Process Control* **2021**, *103*, 34–47. [[CrossRef](#)]
20. Sherstinsky, A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Phys. D Nonlinear Phenom.* **2020**, *404*, 132306. [[CrossRef](#)]
21. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
22. Zhang, H.; Tang, Z.; Xie, Y.; Gao, X.; Chen, Q.; Gui, W. Long short-term memory-based grade monitoring in froth flotation using a froth video sequence. *Miner. Eng.* **2021**, *160*, 106677. [[CrossRef](#)]
23. Pan, H.; Su, T.; Huang, X.; Wang, Z. LSTM-based soft sensor design for oxygen content of flue gas in coal-fired power plant. *Trans. Inst. Meas. Control* **2021**, *43*, 78–87. [[CrossRef](#)]
24. Ke, W.; Huang, D.; Yang, F.; Jiang, Y. Soft sensor development and applications based on LSTM in deep neural networks. In *Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, Honolulu, HI, USA, 27 November–1 December 2017; pp. 1–6.
25. Bouktif, S.; Fiaz, A.; Ouni, A.; Serhani, M.A. Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches. *Energies* **2018**, *11*, 1636. [[CrossRef](#)]
26. Chung, H.; Shin, K.S. Genetic algorithm-optimized long short-term memory network for stock market prediction. *Sustainability* **2018**, *10*, 3765. [[CrossRef](#)]
27. Alshwaheen, T.I.; Hau, Y.W.; Ass'Ad, N.; Abualsamen, M.M. A novel and reliable framework of patient deterioration prediction in intensive care unit based on long short-term memory-recurrent neural network. *IEEE Access* **2020**, *9*, 3894–3918. [[CrossRef](#)]
28. Jahed Armaghani, D.; Hasanipanah, M.; Mahdiyari, A.; Abd Majid, M.Z.; Bakhshandeh Amnieh, H.; Tahir, M. Airblast prediction through a hybrid genetic algorithm-ANN model. *Neural Comput. Appl.* **2018**, *29*, 619–629. [[CrossRef](#)]
29. Zhang, Y.; Li, J.; Fan, X.; Liu, J.; Zhang, H. Moisture prediction of transformer oil-immersed polymer insulation by applying a support vector machine combined with a genetic algorithm. *Polymers* **2020**, *12*, 1579. [[CrossRef](#)]
30. Mandic, D.P.; Chambers, J.A. Exploiting inherent relationships in RNN architectures. *Neural Netw.* **1999**, *12*, 1341–1345. [[CrossRef](#)]
31. Felix, A.G.; Jürgen, S.; Fred, C. Learning to forget: Continual prediction with LSTM. *Neural Comput.* **2000**, *12*, 2451–2471.
32. Li, Y.C.; Zhao, L.N.; Zhou, S.J. Review of Genetic Algorithm. *Adv. Mater. Res.* **2011**, *179–180*, 365–367. [[CrossRef](#)]
33. Fletcher, R.; Leyffer, S.; Ralph, D.; Scholtes, S. Local convergence of SQP methods for mathematical programs with equilibrium constraints. *SIAM J. Optim.* **2006**, *17*, 259–286. [[CrossRef](#)]
34. Mansoornejad, B.; Mostoufi, N.; Jalali-Farahani, F. A hybrid GA–SQP optimization technique for determination of kinetic parameters of hydrogenation reactions. *Comput. Chem. Eng.* **2008**, *32*, 1447–1455. [[CrossRef](#)]
35. Li, K.; Wang, F.; He, D.; Zhang, S. A knowledge based intelligent control method for dehydration and mixing process 2017. In *Proceedings of the 2017 29th Chinese Control and Decision Conference (CCDC)*, Chongqing, China, 28–30 May 2017; pp. 477–482.