

Received December 1, 2020, accepted December 9, 2020, date of publication January 11, 2021, date of current version January 20, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3050548

Time-Series Prediction of Iron and Silicon Content in Aluminium Electrolysis Based on Machine Learning

LINSHENG CHEN^{1,3}, YONGMING WU^{1,3}, YINGBO LIU², TIANSONG LIU^{1,3},
AND XIAOJING SHENG^{1,3}

¹Sate Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, China

²Cloud Computing Center of Big Data Research Institute of Yunnan Economy and Society, Yunnan University of Finance and Economics, Kunming 650221, China

³Key Laboratory of Advanced Manufacturing Technology of Ministry of Education, Guizhou University, Guiyang 550025, China

Corresponding authors: Yongming Wu (ymwu@zu.edu.cn) and Yingbo Liu (zgliuby@hotmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 51505094, in part by the Guizhou Provincial Natural Science Foundation under Grant [2018]4706 and Grant [2016]1037, in part by the Science and Technology Support Program in Guizhou under Grant [2017]2029, in part by the Applied Basic Research Program of major projects in Guizhou under Grant JZ[2014]2001, and in part by the Talent Introduction Research Program of Guizhou University under Grant (2014)60.


ABSTRACT In analyzing dynamic characteristic of time-series data, classic prediction models rely heavily on static historical data, and tacit knowledge is difficult to be mined effectively. Therefore, a hybrid prediction model GS-GMDH is proposed based on growing neural gas (GNG) and the group method of data handling (GMDH). Firstly, a dynamic prediction mechanism, based on an incremental learning algorithm and time-series prediction, is established by GS-GMDH, by which the singularity is recognized and the prediction efficiency is improved. Secondly, to compare the performance of the proposed method, the multi-step ahead predictions with time-series data onto iron and silicon content are employed, and the new model is compared with classic machine models. Finally, the results show that the hybrid prediction model (GS-GMDH) proposed in this paper ensure an accurate and efficient prediction of time-series data for iron and silicon content.

INDEX TERMS Contents of iron and silicon, machine learning, aluminium electrolysis, GNG, GMDH.

I. INTRODUCTION

The rapid development for the aluminium electrolysis, from which waste-water and discarded aluminium are generated, has caused the destruction for the ecosystem [1]. To response the problem, people have explored aluminium electrolysis technology in terms of energy saving, emission reduction, and green production [2].

Zeng *et al.* [3] proposed liquidus model to reduce the energy consumption for aluminium production based on measuring the temperature of the electrolyte and the concentration of AlF_3 . Sun *et al.* [4] applied colorimetric method to determine the content of Fe in the anode for the electrolytic cell, which provides theoretical support to remove impurities. Vasyunina *et al.* [5] used flotation reagent produced by Clariant(Germany) to filter out Fe and Si elements of waste

The associate editor coordinating the review of this manuscript and approving it for publication was Chao Tong .

aluminium products, which is convenient for recycling of aluminium. However, most of the research is on physical and chemical properties, and there is still no efficient method for removing Fe and Si of aluminium production [6], which leads to slow progress in energy saving and emission reduction for aluminium electrolysis technology [7].

To the aluminium electrolysis technology, high-purity aluminium liquid is the product of aluminium electrolysis and is often affected by many physical indicators, which makes it difficult to mine the coupling relationship between various indicators based on traditional models. To response the above problems, machine learning is regarded as a new attempt. Khera *et al.* [8] used BP network to monitor the health for capacitors in the electrolytic cell, which prevents failures for power electronic system. Zhou *et al.* [9] proposed a singular value thresholding and extreme gradient boosting (SVT-XGBoost) model, which achieves anode effect is predicted, to improve efficiency of aluminium electrolysis.

The application of the above machine learning in industry is very successful, but there are still deficiencies. The computational capabilities of models with deep structures and complex structures have been enhanced (such as the DBN [10], DBM [11], and CNN [12], [13]), but the hyper parameters and complex structural designs lead to the need for a large amount of noise-free data [14]–[16]. To response the above problems, a deep convolutional transfer learning network (DCTLN) is developed by Guo *et al.* [17], which accurately classified faulty bearings with incomplete data information. Due to the lack of ideal samples, a simple but very practical GMDH polynomial network is used to estimation the state of battery health by Wu *et al.* [18]. However, the above models are still not stable enough, especially when working in the non-stationary and noisy environment of aluminium electrolysis [19].

The electrolytic cell is a non-stationary and noisy environment, where most chemistry takes place of aluminium electrolysis, and various impurity elements are produced. The control of Fe and Si content is a hotspot for aluminium electrolysis technology. Most Fe and Si are come from material and production equipment[20]. Excessive Fe impurities cause internal stress concentration and cracks of aluminium material. In addition, the mechanical properties and conductivity are affect by excessive Si impurities[21].

In a dynamic and noisy environment, the analysis of representative features for dynamic data, from which provide theoretical support for future work, has become a difficult problem [22], [23]. In anomaly detection and early warning of industry, it is urgent demand that clustering and classification analyses based on data-drive. Duan *et al.* [24] completes anomaly detection of breakout in continuous casting based clustering analysis, but this method relies on static data and cannot achieve true online monitoring. A K-mean dynamic clustering is proposed by Liang *et al.* [25], which overcomes the hysteresis and low accuracy of conventional overflow monitoring methods. Nooralishahi *et al.* [26] develops an online semi-supervised multi-channel classifier based on GNG learning scheme, to really complete machine learning online.

The incremental clustering model can build a balance mechanism between plasticity and stability in a noisy environment [27]–[30], which combines with a neural network is one of the effective methods to solve the above problems [31], [32]. An incremental learning clustering GNG-L is proposed by Wu *et al.* [33], which can monitor the dynamic characteristic of real-time data, and on the basis of its research, the GNG-based singularity recognition algorithm (GS) is developed in this paper. GMDH and other classic models have strong convergence. The GMDH model has been widely used in the fields of material structure [34], [35], traffic flow prediction [36], new energy technology [37]–[39], and mobile communication [40]. Then, the GS algorithm is combined with GMDH model to become a new hybrid prediction model, namely the GS-GMDH in this paper. The brief description of the hybrid prediction model is as follows:

- The singularity recognition phase: the GS algorithm is used to monitor and recognize the singularity of the time-series data, and the recognition result is adapted as the basis for triggering the prediction phase.

- The prediction phase: MLs algorithms are triggered for training and prediction.

The prediction of Fe and Si content in the electrolysis process is proposed as a new research method in this paper, the GS-GMDH model is used to predict the time-series data of iron and silicon content. Moreover, the GS-GMDH model performed better than classic models such as MLFFNN, ANFIS and GMDH. In addition, GS-GMDH can efficiently and accurately predict the content of iron and silicon in aluminum electrolysis with the less and noisy data.

This paper is organized as follows: Section 2 introduces the practical case and data source. The description of classic MLs in Section 3. Section 4 describes the hybrid prediction model proposed in this paper. Section 5 introduces the performance evaluation measures and experiments. Section 6 is the conclusion and work later.

II. CASE STUDY

The data used in the study are real-time data collected from an aluminium electrolysis plant in Guiyang, China. The research recorded the daily percentage of iron and silicon content in the electrolytic products from March 1, 2019 to August 31, 2019. The samples were collected once a day for a total of approximately 184 days. As shown in (a) and (b) of Figure 1, the changes in iron content and silicon content are similar. Therefore, the average values of iron and silicon content are used as experimental data in this paper. It is worth mentioning that this article takes the element content percentage as the experimental object. If we need to calculate the specific mass, we only need to refer to the daily production of aluminium. The raw data (Fe%, Si%) and experimental data ($(\text{Fe}\% + \text{Si}\%)/2$) are shown in Figure 1.

Figure 1 (a) shows the percentage of heavy metal iron content in the molten aluminium, (b) indicates the percentage of silicon content in the molten aluminium, and (c) indicates the average content of iron and silicon in the molten aluminium. The x-axis represents March to August, and the y-axis means the beginning of the month to the end of the month. Observations show that the content of impurities is the lowest in March and highest in the middle of June. (d) shows the change in Fe%, Si%, and $(\text{Fe}\% + \text{Si}\%)/2$ in six months, and the data used in this paper are the yellow.

Based on the practical production data for Guiyang Aluminium Plant, the change of dynamic characteristic for Fe and Si contents, doesn't rises and falls steadily, but does irregular dynamic change with seasonal change, which is obvious rise in summer and then fell slightly. For the dynamic and noisy environment, a machine learning model with strong adaptability and high stability is required to supervise and predict the content of Fe and Si.

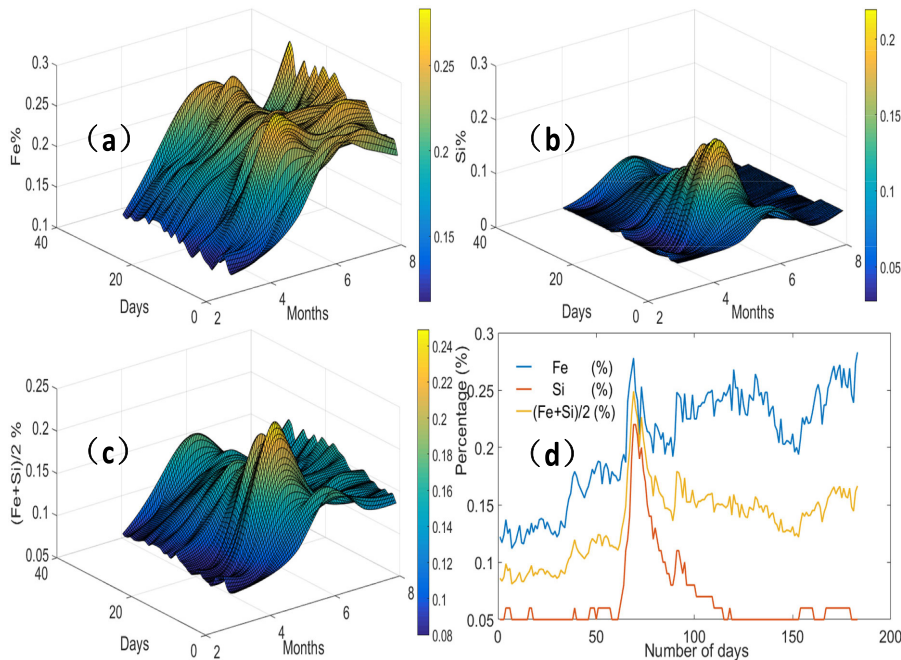


FIGURE 1. Collected data(a, b) and experimental data(c, d).

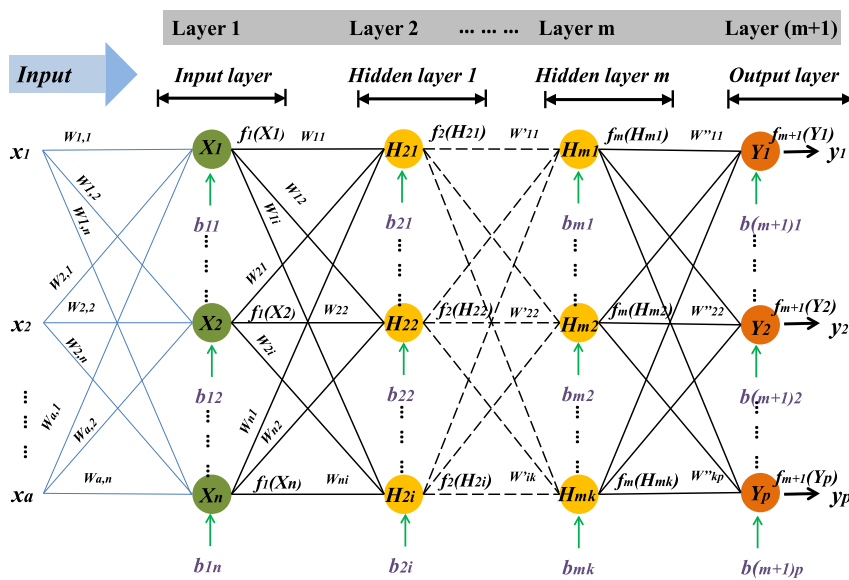


FIGURE 2. Multilayer neural network.

III. THE CLASSIC PREDICTION MODELS BASED ON MACHINE LEARNING

The classic prediction models adapted in this paper mainly include MLFFNN, ANFIS, and GMDH. Five types training function are used for comparative analysis in MLFFNN; three ANFIS models are built, including ANFIS-GP based on grid partition, ANFIS-SC based on subtractive clustering, and ANFIS-GCM based on fuzzy c-means clustering. To improve the prediction potential of the ANFIS model, GA and PSO algorithms are combined to optimize ANFIS;

A. MULTILAYER FEED-FORWARD NEURAL NETWORK (MLFFNN)

Multilayer feed-forward neural networks (MLFFNNs) are most widely used in artificial neural networks (ANN). Common MLFFNNs include perceptron networks, BP networks, and RBF networks. The BP network used in this paper is a typical application of back-propagation learning algorithm in MLFFNN [41]. The MLFFNN model is mainly includes of the input layer, the hidden layer, and an output layer. The multilayer neural network structure is shown in Figure 2.

As shown in Figure 2, the input value of each layer in the network come from the previous layer, and the values calculated by the activation function is used as the output value of this layer. x is the input value. $W_{a,n}$ is the weight between the a -th input data and the n -th neuron in the input layer. X is the weighted input of the input layer, see Eq. (1). b_{mk} is the threshold of the k -th neuron in the m -th layer. f_m is the transfer function (activation function) of layer m . W_{ni} is the weight of the n -th input layer vector in Layer 1 (input layer) and the i -th neuron in Layer 2 (hidden layer 1). H_{2i} is the weighted input of the i -th neuron in Layer 2 (hidden layer 1), see Eq. (2). W'_{ik} is the connection weight between the i -th neuron in the Layer m (hidden layer $(m-1)$) and the k -th neuron in Layer $(m+1)$ (hidden layer m). W''_{kp} is the connection weight of the k -th neuron in Layer $(m+1)$ (hidden layer m) and the p -th neuron in the output layer. Eqs. (3) and (4) show the weighted input Y_p at the output layer and the network output y .

$$\left\{ \begin{aligned} X_1 &= b_{11} + \sum_{q=1}^a x_q * W_{q,1} \\ X_2 &= b_{12} + \sum_{q=1}^a x_q * W_{q,2} \\ &M \\ X_n &= b_{1n} + \sum_{q=1}^a x_q * W_{q,n} \end{aligned} \right. \quad (1)$$

$$\left\{ \begin{aligned} H_{21} &= b_{21} + \sum_{q=1}^n f_1(X_q) * W_{q1} \\ H_{22} &= b_{22} + \sum_{q=1}^n f_1(X_q) * W_{q2} \\ &M \\ H_{2i} &= b_{2i} + \sum_{q=1}^n f_1(X_q) * W_{qi} \end{aligned} \right. \quad (2)$$

$$\left\{ \begin{aligned} Y_1 &= b_{(m+1)1} + \sum_{q=1}^k f_m(H_{mq}) * W_{q1} \\ Y_2 &= b_{(m+1)2} + \sum_{q=1}^k f_m(H_{mq}) * W_{q2} \\ &M \\ Y_p &= b_{(m+1)p} + \sum_{q=1}^k f_m(H_{mq}) * W_{qp} \end{aligned} \right. \quad (3)$$

$$\left\{ \begin{aligned} y_1 &= f_{m+1}(Y_1) \\ y_2 &= f_{m+1}(Y_2) \\ &\vdots \\ y_p &= f_{m+1}(Y_p) \end{aligned} \right. \quad (4)$$

It can be known from the above that the simple MLFFNN model does not exist in adjusting weights and thresholds. In this study, the weights and thresholds are repeatedly adjusted by the error back-propagation algorithm (BP) is

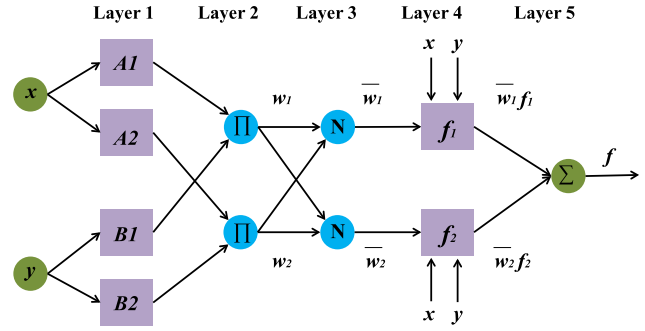


FIGURE 3. Structure of an ANFIS with two inputs, one output and two rules.

adopted to adjust, and a variety of training functions are used for comparative analysis, including the gradient descent training function (traingd), Bayesian regularization training function (trainbr), Levenberg-Marquardt algorithm (trainlm), Powell-Beale algorithm (traincgb), and conjugate gradient algorithm (trainscg).

B. ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM (ANFIS)

ANFIS is called an adaptive neural fuzzy system, which is considered to be a combination of a learning algorithm of the ANN and a concise form of fuzzy inference (FIS); namely, the training process of the model can be simplified as a process of back-propagation through errors and parameter adjustment through least squares. There are two inference methods for ANFIS, namely Mamdani and Sugeno. The fuzzy system mainly includes three parts: fuzzification, inferential decision and defuzzification [42]. Figure 3 shows a simple ANFIS model based on two inputs. The fuzzy rules base of this model combined two Sugeno “if-then” rules, which defined by Eqs. (5) and (6).

$$\text{Rule1: if } x \text{ is } A_1 \text{ and } y \text{ is } B_1, \quad \text{then } f = p_1 + q_1y + r \quad (5)$$

$$\text{Rule2: if } x \text{ is } A_2 \text{ and } y \text{ is } B_2, \quad \text{then } f = p_2 + q_2y + r \quad (6)$$

As shown in Figure 3, the structure of ANFIS has five layers. The nodes of the first and fourth layers need parameter learning, which is the core part of the model. The functions of these layers are described as follows:

Layer 1 (fuzzification layer): This layer is a fuzzification layer, which the input variables become fuzzy and the membership degree of the fuzzy set is output to the next layer. The A_i, B_i are fuzzy sets. The cells outputs (O_i^1) of this layer can be used to indicate the extent to which x and y belong to A_i and B_i , which are defined as Eqs. (7) and (8). The membership functions μ_{A_i} and μ_{B_i} are often represented by Gaussian functions, as follow:

$$O_i^1 = \mu_{A_i}(x) \quad i = 1, 2 \quad (7)$$

$$O_i^1 = \mu_{B_i}(x) \quad i = 1, 2 \quad (8)$$

$$\mu_{A_i} = \exp \left\{ - \left[\left(\frac{x - c_i}{a_i} \right)^2 \right]^{b_i} \right\} \quad (9)$$

$$\mu_{Bi} = \exp \left\{ - \left[\left(\frac{y - c_i}{a_i} \right)^2 \right]^{b_i} \right\} \quad (10)$$

where the x and y are inputs. The $\{a_i, b_i, c_i\}$ are premise parameters, which can be adjusted in training.

Layer 2 (rule layer): the layer can complete the operation of the fuzzy set of the premise part. The firing strength of the fuzzy rule is represented by the product of membership functions of each feature.

$$O_i^2 = w_i = \mu_{Ai}(x) \cdot \mu_{Bi}(y) \quad i = 1, 2 \quad (11)$$

Layer 3 (normalization layer): this layer normalizes the firing strength of each rule obtained from the previous layer and characterizes the firing proportion of the rule in the entire rule base (represented by probability).

$$O_3^i = \bar{w}_i = \frac{w_i}{w_1 + w_2} \quad i = 1, 2 \quad (12)$$

Layer 4 (defuzzification layer) is used to calculate the output of each rule.

$$O_4^i = \bar{w}_i f = \bar{w}_i (p_i x + q_i y + r_i) \quad i = 1, 2 \quad (13)$$

where $\{p_i, q_i, r_i\}$ are the consequent parameters and the \bar{w}_i is the normalized firing strength in layer 3.

Layer 5 (sum layer), the output of ANFIS model is achieved by collecting the output values for each rule that are obtained from the previous layer.

$$O_5^1 = \sum \bar{w}_i f_i = \sum w_i f_i / \sum w_i \quad i = 1, 2 \quad (14)$$

where the total output for this layer can be seen as the linear combination for the following parameters:

$$O_5^i = \bar{w}_1 f_1 + \bar{w}_2 f_2 = (\bar{w}_1 x) p_1 + (\bar{w}_1 y) q_1 + (\bar{w}_1) r_1 + (\bar{w}_2 x) p_2 + (\bar{w}_2 y) q_1 + (\bar{w}_2) r_2 \quad (15)$$

Particle swarm optimization (PSO) [43] and a genetic algorithm (GA) [44] are adopted to develop the training process of ANFIS so that the prediction strength of ANFIS is improved. ANFIS training is actually a process of reducing errors by finding the right premise (see Eqs. (9) and (10)) and consequent (see Eq. (13)) parameters. The choice membership functions refer to the Gaussian function with positive distribution, and $\{a_i, b_i, c_i\}$ are seen as the premise parameters (a_i is the peak of Gaussian curve, b_i is a trainable parameter and c_i is the root mean square of the membership function). The consequent parameters are $\{p_i, q_i, r_i\}$ and are given in Eq. (29). For example, the parameters are optimized by the GA, each chromosome has $R \times (F \times D + B)$ genes (R is the number of rules, F denotes the number of premise parameters for each membership function in Layer 1, and D represents the dimension of the input data. In addition, B denotes the number of consequent parameters for each rule in Layer 4). Based on a large number of genetic, crossover, and mutation operations, a new chromosome with the best fitness will be generated. The value of the gene on this chromosome represents the new parameters of the ANFIS

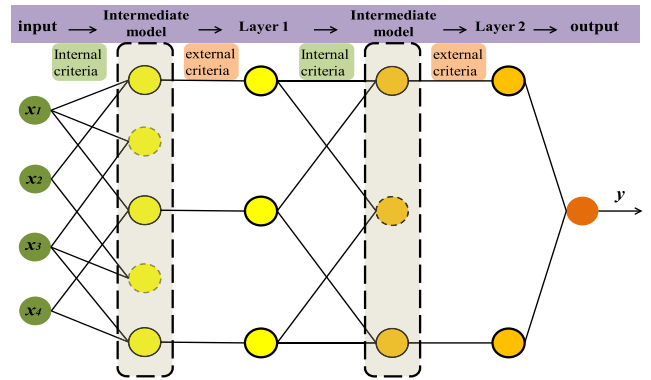


FIGURE 4. The group method of data handling.

model (the optimization process is always in the direction of reducing the output error of ANFIS). The idea of applying the PSO to develop these parameters is similar.

C. GROUP METHOD OF DATA HANDLING (GMDH)

The group method of data handling (GMDH) was discovered by Ivakhnenko [45] in 1971 and can automatically find the correlation of various indicators in the data. Additionally, the neurons in the same layer are combined in pairs to generate intermediate models (internal criteria), and then these intermediate models are filtered based on external criteria. The unfiltered models become the neurons in the next layer. The above process is repeated until the network has the best complexity [46]. The GMDH structure is shown in Figure 4.

y is the expected output and $X = (x_{i1}, x_{i2}, \dots, x_{in})$ ($i = 1, 2, \dots, m$) is the input vector. If there are m input vectors and each vector has dimension n , y and x in function f satisfy Eq. (16):

$$y_i = f(x_{i1}, x_{i2}, x_{i3}, \dots, x_{in}) \quad (i = 1, 2, 3 \dots m) \quad (16)$$

To find a suitable function f , the GMDH network is trained to find \hat{f} which is similar to f . The actual output (\hat{y}_i) of the network is as follows,

$$\hat{y}_i = \hat{f}(x_{i1}, x_{i2}, x_{i3}, \dots, x_{in}) \quad (i = 1, 2, 3 \dots m) \quad (17)$$

The goal of training is to minimize the cumulative error, as follows:

$$\sum_{i=1}^m [\hat{f}(x_{i1}, x_{i2}, x_{i3}, \dots, x_{in}) - y_i] \rightarrow \min \quad (18)$$

The input and output of neurons in the GMDH network need a reference function. The Volterra function is commonly used to represent the input-output relationship as follows:

$$y = a_0 + \sum_{i=1}^m a_i x_i + \sum_{i=1}^m \sum_{j=1}^m a_{ij} x_i x_j + \sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^m a_{ijk} x_i x_j x_k + \dots \quad (19)$$

where the a_i is the weight vector determined by the regression method to reduce the error between y and \hat{y} . Since each neuron

of the GMDH has only two inputs, Eq. (19) can be reduced to a binary quadratic polynomial, such as Eq. (20):

$$y = G(x_i, x_j) = a_0 + a_1x_1 + a_2x_2 + a_3x_1^2 + a_4x_2^2 + ax_1x_2 \quad (20)$$

where the value of G_i is calculated by the binary quadratic polynomial of Eq. (20). The value of G_i should make the mean square error of the whole (all possible pairwise combinations) small, Eq. (21). The mean square error can be used to select neurons in the intermediate model related to external criteria.

$$E = \frac{\sum_i^m (y_i - G_i)^2}{m} \rightarrow \min \quad (21)$$

If the dimension of a single sample is n and there are m samples, the matrix of the input data set and the input target value are defined, as shown in Eq. (22),

$$\begin{bmatrix} x_{1r} & x_{1s} & \cdots & y_1 \\ x_{2r} & x_{2s} & \cdots & y_2 \\ \vdots & \vdots & \ddots & \vdots \\ x_{mr} & x_{ms} & \cdots & y_m \end{bmatrix} \quad (22)$$

in which Eq. (22) satisfies $\{(y_i, x_{ir}, x_{is}), i = 1, 2, \dots, m\}$, $r, s \in \{i = 1, 2, \dots, n\}$. A matrix equation can be found based on Eq. (20),

$$Y = Xa \quad (23)$$

where X, Y are as follows:

$$X = \begin{bmatrix} 1 & x_{1r} & x_{1s} & x_{1r}^2 & x_{1s}^2 \\ 1 & x_{2r} & x_{2s} & x_{2r}^2 & x_{2s}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{mr} & x_{ms} & x_{mr}^2 & x_{ms}^2 \end{bmatrix} \quad (24)$$

$$Y = [y_1, y_2, y_3, \dots, y_m]^T \quad (25)$$

It can be known by Eq. (23) that the weight coefficient can be solved by the least square method in regression analysis as follows:

$$a = (X^T X)^{-1} X^T Y \quad (26)$$

IV. A NEW HYBRID PREDICTION MODEL GS-GMDH

The GS-GMDH model proposed in this paper is a fusion of the improved GNG and the GMDH models. The main contents are as follows:

- The singularity recognition phase. To monitor the dynamic characteristic for time-series data, the GS model proposed of singularity recognition based the GNG algorithm, which is a combination of the dynamic feature analysis [33] and actual production demand.

- The prediction phase. The prediction mechanism is triggered after the singular point is recognized, namely the time-series data is employed to complete the training and prediction of ML models. Then, the performance of each ML

model is compared based on the evaluation results. Finally, a hybrid prediction model GS-GMDH is proposed based on GS and GMDH models.

A. THE GS ALGORITHM BASED ON GNG

An improved GNG [47] was used to realize real-time monitoring for drift data onto Wu *et al.* [33], but the model does not have the ability of singularity recognition. To solve the problem, a GS model is proposed based on the study of Wu *et al.* [33], which can monitor the real-time data and distinguish the singularity. The GS algorithm is described as follows:

Step1: The number of neural nodes N is initialized; The data is normalized, and the dimension of data is obtained.

Step2: Input the sample x ; all nodes are traversed; the winning neural node J_{n^*} is calculated, and the local error $E_{J_{n^*}}$ of the node is updated.

$$E_{J_{n^*}} = E_{J_{n^*}} + \|x - J_{n^*}\|^2 \quad (27)$$

Step3: The weight of J_{n^*} and the domain node $J_{n^{**}}$ are adjusted.

$$J_{n^*} = \alpha 1 \times (x - J_{n^*}) \quad (28)$$

$$J_{n^{**}} = \alpha 2 \times (x - J_{n^{**}}) \quad (29)$$

Step4: A connection between the J_{n^*} and the second winning node J_S is establishing, and the connection age between J_{n^*} and J_S is cleared.

Step5: Neural nodes whose connection age reaches the upper limit t_{\max} are removed; the isolated nodes are removed.

Step6: The average Euclidean distance d_{aver} is calculated of interconnected nodes in the topology.

Step7: The neuron generation mechanism. If Eq. (3) is satisfied, a new node be generated.

$$J_{newnode} = 0.5 \times (J_{err \max} - J_{err \max 2}) \quad (30)$$

$$E_{newnode} = 0.5 \times E_{\max} \quad (31)$$

Step8: The neuron deletion mechanism. N_{errmin} is taken as the center, if $N_{in2} < N \times b$ is satisfied, N_{errmin} is deleted. N_{errmin}, N_{in2} , are the node with minimum local error, the number of nodes with distance from N_{errmin} is less than d_{aver} , respectively.

Step9: The global error of the neural network is reduced. E_N is the global errors. β is the global error coefficient, $\beta \in (0, 1)$.

$$E_N \leftarrow \beta \times E_N \quad (32)$$

Step10: The GS model is used for real-time monitor the singularity of the real-time data. If the preset conditions are met, it's determined that there is a singularity.

Step11: The input x is processed and repeats **Step 2**.

B. THE PREDICTION STEPS

Based on the result for clustering analysis of GS model, the value x of $[Fe\% + Si\%]/2$ is the demand of prediction

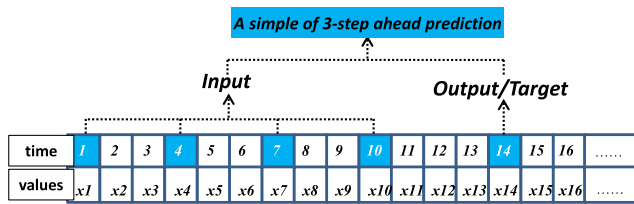


FIGURE 5. The train or test example for 3-step ahead of windows size 5.

phase by multi-step prediction, from which the past values of $[\text{Fe}\% + \text{Si}\%]/2$ to be used as shown in Figure 5.

As the Figure 5, the input sample is composed of $[x_1, x_4, x_7, x_{10}, x_{14}]$, in which $[x_1, x_4, x_7, x_{10}]$ are input data for ML models and x_{14} is output(for testing) or target(for training). Namely, sliding window operation is adopted to obtain samples [48], which the size of window is fixed for 5. The simple X of MLs models is as follows:

$$X_j = [x_i, x_{i+j}, x_{i+2j}, x_{i+2j}, x_{i+3j}] \quad j \in [1, 3], i \in [1, 183] \quad (33)$$

Time-series forecasting is widely adopted in meteorology [37]–[39], traffic management [49], finance [50], and energy source [51], etc. Especially for intelligent transportation, many achievements have been made based on time-series forecasting. The analysis of similarity and repeatability for time-series data, from which the potential periodic patterns are identify, has become a new ideas for long-term prediction [52]. In addition, deep neural networks (CNN, GMDH, ST-ResNet, etc.) may perform well for traffic flow prediction [36], [53] based on match-then-predict and prediction-after-classification. For this paper, the first is the phase of singularity recognition, the GS incremental learning model is used to monitor the dynamic characteristic of time-series data (2, 3 and 4-step ahead predictions are considered), which the monitoring results meets the condition is the premise for triggering prediction mechanism (The establishment of preset conditions should refer to the actual needs of the factory); The second is the prediction phase, five ML models are employed to predict the contents of iron and silicon, namely MLFFNN (traingd, trainbr, trainlm, traincgb and traincsg five training functions), ANFIS (ANFIS-GP, ANFIS-SC and ANFIS-FCM), ANFIS-GA, ANFIS-PSO and GMDH; The final evaluation indicators are used to select the best performance model. The flow of GS-GMDH is shown in Figure 6.

V. PERFORMANCE EVALUATION MEASURES AND EXPERIMENTS

The first part of the experiment is the singularity recognition based on the GS model, from which the 2, 3 and 4-step ahead time-series data are considered, singularities are found at different moments. The second part of the experiment is the prediction based on different MLs, including MLFFNN, ANFIS, ANFIS-GA, ANFIS-PSO and GMDH. In prediction phase, three statistical indicators, including the correlation

coefficient R (which is used to reflect the close degree of correlation between variables), mean square error MSE , and root mean square $RMSE$, are used to evaluate the proposed models. The calculation formulas are as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2} \quad (34)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2 \quad (35)$$

$$R = \frac{\sum_{i=1}^n (x_i - x^*)(y_i - y^*)}{\sqrt{\sum_{i=1}^n (x_i - x^*)^2 \sum_{i=1}^n (y_i - y^*)^2}} \quad (36)$$

where n , x_i , y_i , x^* and y^* are the number of data, the expected value (target), predicted value (output), mean of expected data and mean of predicted data, respectively.

A. THE SINGULARITY RECOGNITION PHASE-GS MODEL

The proposed model of GS is used to monitor the dynamic characteristics and discover the singularity of time-series data. The preset condition for discover the singularity (a. the number of times that a cluster wins in the GS model is higher than 90% of the total number of input data. b. the Euclidean distance from a certain neural node to the origin is greater than 0.18). The experimental results are shown in Figure 7. For the 3-step ahead data, the sample $[x_{45}, x_{47}, x_{49}, x_{51}]$ is recognized as a singularity; For the 4-step ahead data, the sample $[x_{54}, x_{57}, x_{60}, x_{63}]$ is recognized as a singularity; b. the Euclidean distance from a certain neural node to the origin is greater than 0.18). The experimental results are shown in Figure 7. For the 3-step ahead data, the sample $[x_{45}, x_{47}, x_{49}, x_{51}]$ is recognized as a singularity; For the 4-step ahead data, the sample $[x_{54}, x_{57}, x_{60}, x_{63}]$ is recognized as a singularity; For the window 5-step ahead data, the sample $[x_{42}, x_{46}, x_{50}, x_{54}]$ is recognized as a singularity.

To the 3-step ahead data, the sample $[x_{54}, x_{57}, x_{60}, x_{63}]$ is input to GS model and recognized as a singularity, then the sample S_3 used in the prediction phase (data x_1 - x_{53} are not participate involved in prediction phase, namely the data used for train and testing are reduced by about 28.96%). The data x_i is the past value of $(\text{Fe}\% + \text{Si}\%) / 2$, and the window size of time series is 5. The input sample S_3 and output O_3 are defined as follows:

$$S_3 = [x_i, x(i+3), x(i+6), x(i+9)], \quad i \in [54, 174] \quad (37)$$

$$O_3 = x^*(i+12), \quad i \in [54, 174] \quad (38)$$

where O_3 is the predicted value (output).

For 2-step ahead and 4-step ahead data, the sample S_2 , output O_2 , S_4 and O_4 are as follows:

$$S_2 = [x_i, x(i+2), x(i+4), x(i+6)], \quad i \in [45, 177] \quad (39)$$

$$O_2 = x^*(i+8), \quad i \in [45, 177] \quad (40)$$

$$S_4 = [x_i, x(i+4), x(i+8), x(i+12)], \quad i \in [42, 171] \quad (41)$$

$$O_4 = x^*(i+16), \quad i \in [42, 171] \quad (42)$$

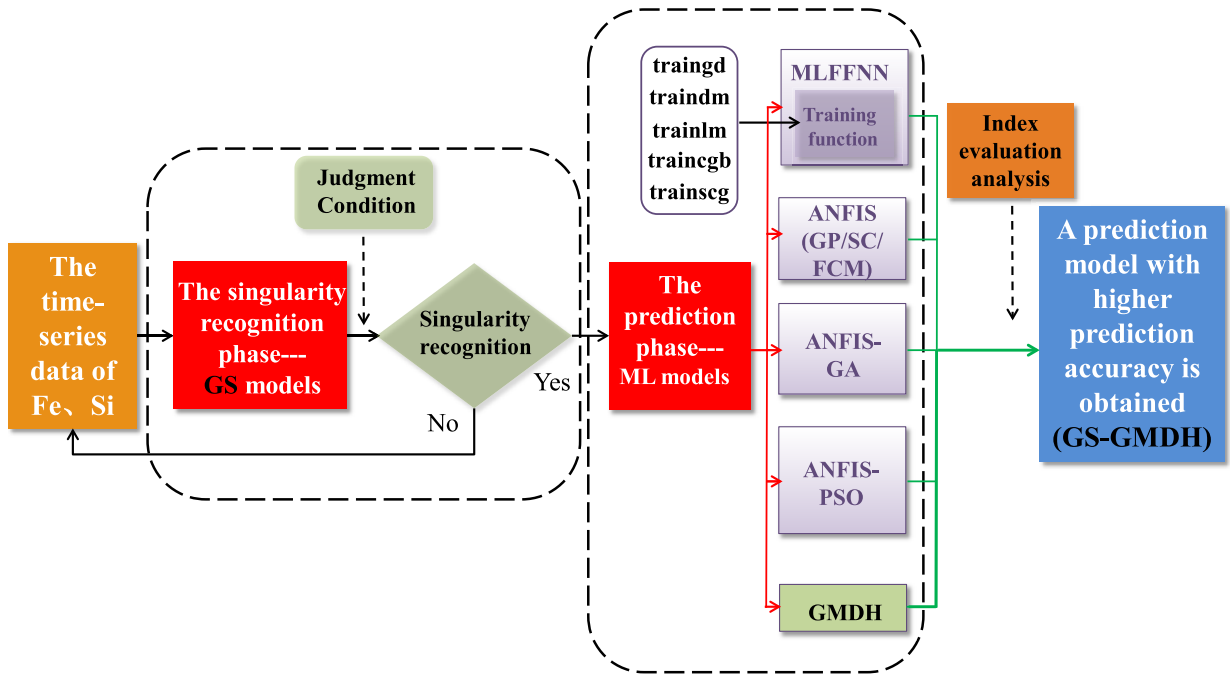


FIGURE 6. The flow of GS-GMDH.

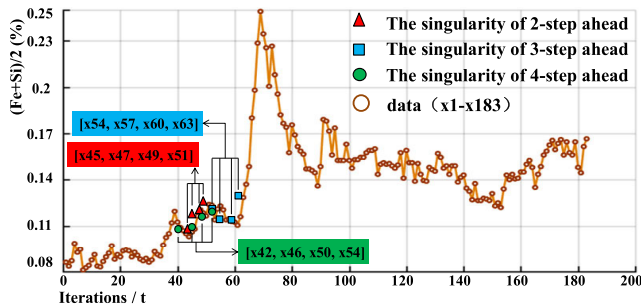


FIGURE 7. Analysis result of GS for time-series data.

TABLE 1. The impact of GS on computing cost.

Model	No GS	GS		
		2-step	3-step	4-step
<i>NRD</i> (%)	0%	24.04%	28.96%	22.40%

For multi-step ahead predictions, the number of reduced data (*NRD*) for prediction phase is shown in Table 1.

B. THE PREDICTION PHASE-ML MODELS

The experimental results of each model with multi-step ahead data in the prediction phase are shown in Table 2. To obtain the most reasonable parameters of ML models for multi-step ahead predictions, we have done a lot of prediction works based the aluminum electrolysis data. In addition, the reasonable parameters of ML models are shown in Table 2. In addition, 75% of the inputs and targets are determined as training data, 10% are as validating data, and 15% are as

testing data. To be brief, the element of “Train” represents the average result for training and verification in Table2, and the “Test” represents the test result.

For the MLFFNN model, a three-layer feedforward neural network is employed, and the number of neurons in the hidden layers is obtained to be 20. In addition, five training functions are used (traingd, trainbr, trainlm, traincgb and trainscg). For the 2-step to 4-step ahead predictions, the trainbr, traincgb, trainlm and trainscg perform better with the *R* are 0.9538, 0.8975 and 0.9717 (for the test set), respectively.

For the ANFIS model, three ANFIS (GP, SC and FCM) models are considered. In Table 2, the core parameters of ANFIS (GP), ANFIS (SC) and ANFIS (FCM) are the number of membership function *N_F*, the influence radius *I_R*, the number of clusters *N_C* and the partition matrix exponent *PME*, respectively. The ANFIS (FCM) has better performance for 4-step ahead prediction with *R* = 0.9875 (for test set).

For the ANFIS-GA model, the ANFIS is improved by GA (the ANFIS is ANFIS-FCM). The number of clusters is 2, the GA’s parameters of mutation rate, crossover percentage, mutation percentage and maximum number of iterations is set to 0.15, 0.4, 0.7 and 1000, respectively. In addition, the case of the number of population *N* is 10, 20, 50 and 200. As is shown in Table 2, ANFIS-GA models with better performance of *R*s are 0.9778, 0.9711 and 0.9878 for multi-step prediction (for test set), respectively.

Particle swarm optimization can also give ANFIS models an advantage in prediction, and the number of clusters is 2. The core parameters of the model were set as follows: the

TABLE 2. Comparisons between the proposed model and classic MLs for predicting the content of Fe and Si.

Method	Parameters	MSE		RMSE		R		NRD(%)
		Train	Test	Train	Test	Train	Test	
<i>2-step ahead</i>								
MLFFNN	<i>rainbr</i>	1.473e-04	1.227e-04	0.0121	0.0111	0.9171	0.9538	0
ANFIS(GP)	<i>N_F=4</i>	3.099e-05	1.317e-04	0.0056	0.0115	0.9849	0.9512	0
ANFIS(SC)	<i>I_R=0.5</i>	1.024e-04	6.722e-05	0.0101	0.0082	0.9642	0.9738	0
ANFIS(FCM)	<i>N_C=2,PME=2</i>	5.282e-05	9.268e-05	0.0073	0.0096	0.9748	0.9633	0
ANFIS-GA	<i>N=20</i>	5.234e-05	5.929e-05	0.0072	0.0077	0.9749	0.9778	0
ANFIS-PSO	<i>N=20</i>	5.184e-05	6.659e-05	0.0072	0.0082	0.9754	0.9745	0
GMDH	<i>N=15,L=5,P=0.5</i>	8.865e-05	7.225e-05	0.0094	0.0085	0.9648	0.9722	0
GS-GMDH	<i>I_N=2,A_m=0.5</i>	5.041e-05	5.481e-05	0.0071	0.0074	0.9787	0.9810	24.04
<i>3-step ahead</i>								
MLFFNN	<i>traincgb</i>	1.478e-04	2.329e-04	0.0122	0.0153	0.9143	0.8975	0
ANFIS(GP)	<i>N_F=4</i>	4.892e-05	1.233e-04	0.0070	0.0111	0.9765	0.9536	0
ANFIS(SC)	<i>I_R=0.8</i>	4.987e-05	8.711e-05	0.0071	0.0093	0.9764	0.9655	0
ANFIS(FCM)	<i>N_C=2,PME=2</i>	6.889e-05	8.434e-05	0.0083	0.0092	0.9686	0.9691	0
ANFIS-GA	<i>N=50</i>	7.198e-05	7.569e-05	0.0085	0.0087	0.9672	0.9711	0
ANFIS-PSO	<i>N=20</i>	4.987e-05	6.659e-05	0.0071	0.0082	0.9764	0.9697	0
GMDH	<i>N=15,L=5,P=0.5</i>	9.623e-05	4.612e-05	0.0098	0.0068	0.9647	0.9797	0
GS-GMDH	<i>I_N=2,A_m=0.5</i>	5.885e-05	2.049e-05	0.0077	0.0045	0.9727	0.9925	28.96
<i>4-step ahead</i>								
MLFFNN	<i>trainlm</i>	1.279e-04	7.413e-05	0.0113	0.0086	0.9611	0.9717	0
ANFIS(GP)	<i>N_F=4</i>	1.431e-05	1.956e-04	0.0038	0.0140	0.9932	0.9048	0
ANFIS(SC)	<i>I_R=0.8</i>	4.479e-05	1.342e-04	0.0067	0.0116	0.9785	0.9494	0
ANFIS(FCM)	<i>N_C=2,PME=2</i>	5.369e-05	3.611e-05	0.0073	0.0060	0.9745	0.9875	0
ANFIS-GA	<i>N=20</i>	5.866e-05	2.896e-05	0.0077	0.0054	0.9729	0.9878	0
ANFIS-PSO	<i>N=20</i>	5.224e-05	3.481e-05	0.0072	0.0059	0.9753	0.9876	0
GMDH	<i>N=15,L=5,P=0.5</i>	6.401e-05	4.256e-05	0.0080	0.0065	0.9698	0.9814	0
GS-GMDH	<i>I_N=2,A_m=0.5</i>	4.764e-05	2.664e-05	0.0069	0.0052	0.9774	0.9898	22.40%

number of populations N is 10, 20, 50 and 200, the inertia weight is 1, the personal learning coefficient was 1, the global learning coefficient is 2, and the maximum number of iterations is 1000. The model has the best performance for 4-step ahead prediction with $R = 0.9876$ (for test set).

For the GMDH model, the variables that need to be set manually are the number of network layers L , the maximum number of neurons in each layer N , and the selection pressure P . GMDH models with better performance of R s are 0.9722, 0.9797 and 0.9814 for multi-step ahead predictions (for test set), respectively.

For the proposed GS-GMDH hybrid prediction model, the initial number of neuron nodes I_N is 2 and the maximum connection age A_m is 10 of the singularity recognition phase.

In addition, the parameter set is the same as GMDH model of the prediction phase. For the multi-step ahead prediction, the proposed GS-GMDH model performs better than other classic model of R s are 0.9810, 0.9925 and 0.9898, respectively. The GS-GMDH model has the best prediction accuracy in the 3-step ahead of considering the prediction accuracy.

In summary, the proposed GS-GMDH model has best prediction accuracy for multi-step ahead predictions based on less aluminium electrolysis time-series data in Table 2 and Figure 8. The incremental clustering model (GS model) can build a balance mechanism between plasticity and stability for aluminium electrolysis in a noisy environment, from which the tacit knowledge of data has been mined effectively. Moreover, the processing of GS model provides theoretical

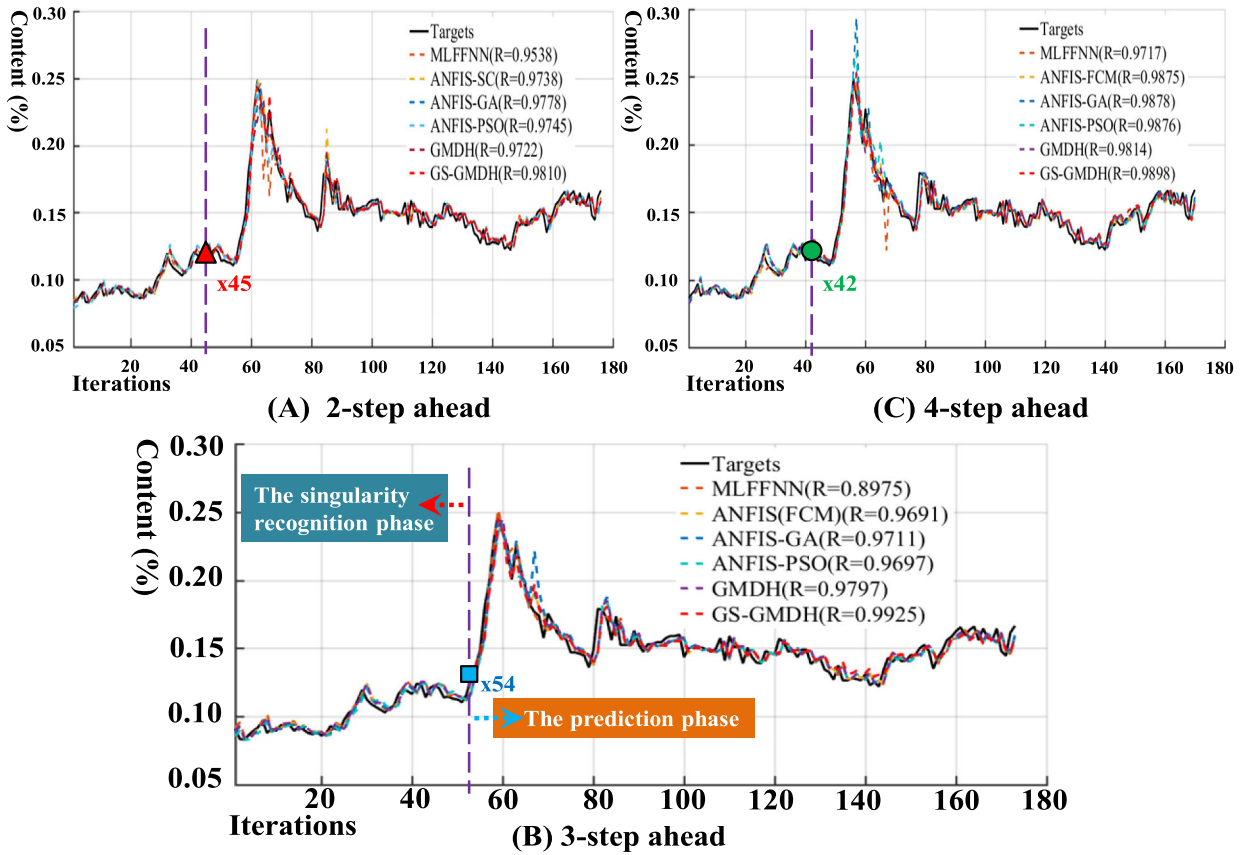


FIGURE 8. Results of predicting with multi-step ahead data.

support for subsequent prediction work. In prediction of time-series data for iron and silicon content, the GMDH model performs well for prediction work based on its strong convergence.

The results of GS-GMDH model for the singularity recognition phase and prediction phase are shown in Fig. 7. The training steps are beginning at different positions for multi-step ahead data (for the 2, 3 and 4-step ahead data, the position are x45, x42 and x54, respectively). In addition, the performance of proposed GS-GMDH is better than MLFFNN, ANFIS, ANFIS-GA, ANFIS-PSO and GMDH models (for test set) for 3-step ahead time-series data with $R = 0.9925$ and $NRD = 28.96\%$.

This find is unexpected and suggests that small-scale data sets allow for greater accuracy. It should be mentioned that, for a single purpose such as the prediction for Fe and Si contents, it is not always true that the more data we use, the more perfect result can be obtained [54]. This observation may support previous descriptions, namely the electrolytic cell is a non-stationary and noisy environment, where the distribution of data characteristic is irregular. Further leads to class-imbalance data, i.e. the imbalance classification problem, which affects both convergence for the training phase and generalization for a model in test set [55]. To solve this problem, an effective strategy is to operate on training set and change its class distribution [56]. Undersampling is seen as a

meaningful method to clean the data, i.e., to remove a small number of samples from the dataset. In addition, these deleted data often considered noise or have other characteristics, i.e. one-sided selection identifies redundant examples close to the boundary between classes [57]. In this paper, the based cluster GS model is developed to ensure that the data characteristics are extracted. Moreover, before the training of deep neural network, the clustering or classification operations are significant to reduce the errors caused by class-imbalance data [58], [59].

VI. CONCLUSION AND LATER WORK

A hybrid prediction model GS-GMDH is developed based on GNG and GMDH, which includes the singularity recognition and prediction phases. For the singularity recognition phase, we propose the GS algorithm based on the GNG, from which the monitoring of dynamic characteristics and the singularity recognition are completed of real-time data. For the prediction phase, the machine learning model is employed to predict the time-series data. Combined with the analysis for the iron and silicon content in the aluminium electrolysis, the experiment results show the overall performance of GS-GMDH proposed in this paper is better than other ML models, with NRD is 28.96% and R is 0.9925. And it is suitable for predicting the contents of iron and silicon in the aluminium electrolysis.

The future work may focus on improving our prediction methods: (1) we need to improve the adaptability and robustness of GS-GMDH model. In addition, the preset condition for discovering the singularity needs to explore for other fields. (2) the source of data sets would also be stricter, and we would adopt larger-scale data sets with more complex indicators. (3) the further analysis of the impact on imbalance data distribution for deep neural networks.

REFERENCES

- [1] K. Peng, Z. Zou, S. Wang, B. Chen, W. Wei, S. Wu, Q. Yang, and J. Li, "Interdependence between energy and metals in China: Evidence from a nexus perspective," *J. Cleaner Prod.*, vol. 214, pp. 345–355, Mar. 2019.
- [2] M. Dai, P. Wang, W.-Q. Chen, and G. Liu, "Scenario analysis of China's aluminum cycle reveals the coming scrap age and the end of primary aluminum boom," *J. Cleaner Prod.*, vol. 226, pp. 793–804, Jul. 2019.
- [3] S. Zeng, W. Shasha, and Q. Yaxing, "Control of temperature and aluminum fluoride concentration based on model prediction in aluminum electrolysis," *Adv. Mater. Sci. Eng.*, vol. 2014, Jan. 2014, Art. no. 181905.
- [4] H. Sun, D. Kocaefe, D. Bhattacharyay, Y. Kocaefe, J. Côté, and P. Coulombe, "Colorimetric methods for determining Fe, V, and Ni content in coke and anode," *Chem. Eng. Technol.*, vol. 42, no. 5, pp. 1011–1017, 2019.
- [5] N. V. Vasyunina, S. V. Belousov, I. V. Dubova, A. V. Morenko, and K. E. Druzhinin, "Recovery of silicon and iron oxides from alumina-containing sweepings of aluminum production," *Russian J. Non-Ferrous Met.*, vol. 59, no. 3, pp. 230–236, May 2018.
- [6] L. Zhang, X. Lv, A. T. Torgerson, and M. Long, "Removal of impurity elements from molten aluminum: A review," *Mineral Process. Extractive Metall. Rev.*, vol. 32, no. 3, pp. 150–228, Jul. 2011.
- [7] Y. Fang, Z. Nie, J. Yang, Q. Die, J. He, H. Yu, Q. Zhou, and Q. Huang, "Polychlorinated naphthalene emissions to the atmosphere from typical secondary aluminum smelting plants in Southwestern China: Concentrations, characterization, and risk evaluation," *Environ. Sci. Pollut. Res.*, vol. 26, no. 13, pp. 12731–12740, May 2019.
- [8] N. Khera and S. A. Khan, "Prognostics of aluminum electrolytic capacitors using artificial neural network approach," *Microelectron. Rel.*, vol. 81, pp. 328–336, Feb. 2018.
- [9] K.-B. Zhou, Z.-X. Zhang, J. Liu, Z.-X. Hu, X.-K. Duan, and Q. Xu, "Anode effect prediction based on a singular value thresholding and extreme gradient boosting approach," *Meas. Sci. Technol.*, vol. 30, no. 1, Jan. 2019, Art. no. 015104.
- [10] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.
- [11] R. Salakhutdinov and G. E. Hinton, "Deep Boltzmann machines," *J. Mach. Learn. Res.*, vol. 5, no. 2, pp. 448–455, 2009.
- [12] H. K. Aggarwal, M. P. Mani, and M. Jacob, "MoDL: Model-based deep learning architecture for inverse problems," *IEEE Trans. Med. Imag.*, vol. 38, no. 2, pp. 394–405, Feb. 2019.
- [13] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5990–5998, Jul. 2018.
- [14] C. L. P. Chen and J. Z. Wan, "A rapid learning and dynamic stepwise updating algorithm for flat neural networks and the application to time-series prediction," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 29, no. 1, pp. 62–72, Feb. 1999.
- [15] C. L. P. Chen and Z. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 10–24, Jan. 2018.
- [16] X. Wen, L. Shao, Y. Xue, and W. Fang, "A rapid learning algorithm for vehicle classification," *Inf. Sci.*, vol. 295, pp. 395–406, Feb. 2015.
- [17] L. Guo, Y. Lei, S. Xing, T. Yan, and N. Li, "Deep convolutional transfer learning network: A new method for intelligent fault diagnosis of machines with unlabeled data," *IEEE Trans. Ind. Electron.*, vol. 66, no. 9, pp. 7316–7325, Sep. 2019.
- [18] J. Wu, Y. Wang, X. Zhang, and Z. Chen, "A novel state of health estimation method of Li-ion battery using group method of data handling," *J. Power Sources*, vol. 327, pp. 457–464, Sep. 2016.
- [19] N. Masuyama, C. K. Loo, H. Ishibuchi, N. Kubota, Y. Nojima, and Y. Liu, "Topological clustering via adaptive resonance theory with information theoretic learning," *IEEE Access*, vol. 7, pp. 76920–76936, 2019.
- [20] M. A. Dewan, M. A. Rhamdhani, J. B. Mitchell, C. J. Davidson, G. A. Brooks, M. Easton, and J. F. Grandfield, "Control and removal of impurities from al melts: A review," *Mater. Sci. Forum*, vol. 693, pp. 149–160, Jul. 2011.
- [21] J. A. Al-Mejali, G. M. Haarberg, N. Bensalah, B.-A. Benkahla, and H. P. Lange, "The role of key impurity elements on the performance of aluminium electrolysis—Current efficiency and metal quality," *Light Metals*, vol. 2, no. 7, pp. 389–394, 2016.
- [22] S. Yu, M. Chen, E. Zhang, J. Wu, H. Yu, Z. Yang, L. Ma, X. Gu, and W. Lu, "Robustness study of noisy annotation in deep learning based medical image segmentation," *Phys. Med. Biol.*, vol. 65, no. 17, 2020, Art. no. 175007, doi: 10.1088/1361-6560/ab99e5.
- [23] Y.-Z. Lin, Z.-H. Nie, and H.-W. Ma, "Structural damage detection with automatic feature-extraction through deep learning," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 32, no. 12, pp. 1025–1056, 2017.
- [24] H. Duan, X. Wang, Y. Bai, M. Yao, and Q. Guo, "Application of k-means clustering for temperature timing characteristics in breakout prediction during continuous casting," *Int. J. Adv. Manuf. Technol.*, vol. 106, nos. 11–12, pp. 4777–4787, Feb. 2020.
- [25] H. Liang, G. Li, and W. Liang, "Intelligent early warning model of early-stage overflow based on dynamic clustering," *Cluster Comput.*, vol. 22, no. S1, pp. 481–492, Jan. 2019.
- [26] P. Nooralishahi, M. Seera, and C. K. Loo, "Online semi-supervised multi-channel time series classifier based on growing neural gas," *Neural Comput. Appl.*, vol. 28, no. 11, pp. 3491–3505, Nov. 2017.
- [27] H. Shen, Y. Liu, Z. Xia, and M. Zhang, "An efficient aggregation scheme resisting on malicious data mining attacks for smart grid," *Inf. Sci.*, vol. 526, pp. 289–300, Jul. 2020.
- [28] E. Lughofer, "Extensions of vector quantization for incremental clustering," *Pattern Recognit.*, vol. 41, no. 3, pp. 995–1011, Mar. 2008.
- [29] E. Lughofer and M. Sayed-Mouchaweh, "Autonomous data stream clustering implementing split-and-merge concepts—Towards a plug-and-play approach," *Inf. Sci.*, vol. 304, pp. 54–79, May 2015.
- [30] C. Wiwatcharakoses and D. Berrar, "SOINN+, a self-organizing incremental neural network for unsupervised learning from noisy data streams," *Expert Syst. Appl.*, vol. 143, Apr. 2020, Art. no. 113069.
- [31] W. Jia, D. Zhao, and L. Ding, "An optimized RBF neural network algorithm based on partial least squares and genetic algorithm for classification of small sample," *Appl. Soft Comput.*, vol. 48, pp. 373–384, Nov. 2016.
- [32] S. Ding, W. Jia, C. Su, L. Zhang, and L. Liu, "Research of neural network algorithm based on factor analysis and cluster analysis," *Neural Comput. Appl.*, vol. 20, no. 2, pp. 297–302, Mar. 2011.
- [33] Y.-M. Wu, L.-S. Chen, S.-B. Li, and J.-D. Chen, "An adaptive algorithm for dealing with data stream evolution and singularity," *Inf. Sci.*, vol. 545, pp. 312–330, Feb. 2021.
- [34] H. MolaAbasi, A. Khajeh, S. N. Semsani, and A. Kordnaeij, "Prediction of zeolite-cemented sand tensile strength by GMDH type neural network," *J. Adhes. Sci. Technol.*, vol. 33, no. 15, pp. 1611–1625, Aug. 2019.
- [35] M. H. Ahmadi, M. Sadeghzadeh, A. H. Raffiee, and K.-W. Chau, "Applying GMDH neural network to estimate the thermal resistance and thermal conductivity of pulsating heat pipes," *Eng. Appl. Comput. Fluid Mech.*, vol. 13, no. 1, pp. 327–336, Jan. 2019.
- [36] X. Song, W. Li, D. Ma, D. Wang, L. Qu, and Y. Wang, "A match-then-predict method for daily traffic flow forecasting based on group method of data handling," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 33, no. 11, pp. 982–998, Nov. 2018.
- [37] A. Khosravi, L. Machado, and R. O. Nunes, "Time-series prediction of wind speed using machine learning algorithms: A case study Osorio wind farm, Brazil," *Appl. Energy*, vol. 224, pp. 550–566, Aug. 2018.
- [38] A. Khosravi, R. O. Nunes, M. E. H. Assad, and L. Machado, "Comparison of artificial intelligence methods in estimation of daily global solar radiation," *J. Cleaner Prod.*, vol. 194, pp. 342–358, Sep. 2018.
- [39] A. Khosravi, R. N. N. Koury, L. Machado, and J. J. G. Pabon, "Prediction of hourly solar radiation in Abu Musa island using machine learning algorithms," *J. Cleaner Prod.*, vol. 176, pp. 63–75, Mar. 2018.
- [40] H. S. Hwang, "Fuzzy GMDH-type neural network model and its application to forecasting of mobile communication," *Comput. Ind. Eng.*, vol. 50, no. 4, pp. 450–457, Aug. 2006.
- [41] S. Zhou, N. Liu, C. Shen, L. Zhang, T. He, B. Yu, and J. Li, "An adaptive Kalman filtering algorithm based on back-propagation (BP) neural network applied for simultaneously detection of exhaled CO and N2O," *Spectrochim. Acta A, Mol. Biomolecular Spectrosc.*, vol. 223, Dec. 2019, Art. no. 117332.

- [42] D. Petković, N. T. Pavlović, and Ž. Čojbašić, "Wind farm efficiency by adaptive neuro-fuzzy strategy," *Int. J. Electr. Power Energy Syst.*, vol. 81, pp. 215–221, Oct. 2016.
- [43] S. Poddar and A. Kumar, "Scale-free PSO for in-run and infield inertial sensor calibration," *Measurement*, vol. 147, Dec. 2019, Art. no. 106849.
- [44] G. Qiu, Y. Gu, and J. Chen, "Selective health indicator for bearings ensemble remaining useful life prediction with genetic algorithm and weibull proportional hazards model," *Measurement*, vol. 150, Jan. 2020, Art. no. 107097.
- [45] A. G. Ivakhnenko, "Polynomial theory of complex systems," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-1, no. 4, pp. 364–378, Oct. 1971.
- [46] M. Ahmadi, M.-A. Ahmadi, M. Mehrpooya, and M. Rosen, "Using GMDH neural networks to model the power and torque of a stirling engine," *Sustainability*, vol. 7, no. 2, pp. 2243–2255, Feb. 2015.
- [47] B. Fritzke, *A Growing Neural Gas Network Learns Topologies*. Cambridge, MA, USA: MIT Press, 1995.
- [48] R. Talavera-Llames, R. Pérez-Chacón, A. Troncoso, and F. Martínez-Álvarez, "Big data time series forecasting based on nearest neighbours distributed computing with spark," *Knowl.-Based Syst.*, vol. 161, pp. 12–25, Dec. 2018.
- [49] X. Yang, Y. Zou, J. Tang, J. Liang, and M. Ijaz, "Evaluation of short-term freeway speed prediction based on periodic analysis using statistical models and machine learning models," *J. Adv. Transp.*, vol. 2020, pp. 1–16, Jan. 2020.
- [50] L. J. Cao and F. E. H. Tay, "Support vector machine with adaptive parameters in financial time series forecasting," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1506–1518, Nov. 2003.
- [51] H. Wang, Y. Liu, B. Zhou, C. Li, G. Cao, N. Voropai, and E. Barakhtenko, "Taxonomy research of artificial intelligence for deterministic solar power forecasting," *Energy Convers. Manage.*, vol. 214, Jun. 2020, Art. no. 112909.
- [52] J. Chen, K. Li, H. Rong, K. Bilal, K. Li, and P. S. Yu, "A periodicity-based parallel time series prediction algorithm in cloud computing environments," *Inf. Sci.*, vol. 496, pp. 506–537, Sep. 2019.
- [53] L. Qu, W. Li, W. Li, D. Ma, and Y. Wang, "Daily long-term traffic flow forecasting based on a deep neural network," *Expert Syst. Appl.*, vol. 121, pp. 304–312, May 2019.
- [54] Z. Hou and X. Li, "Repeatability and similarity of freeway traffic flow and long-term prediction under big data," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1786–1796, Jun. 2016.
- [55] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Netw.*, vol. 106, pp. 249–259, Oct. 2018.
- [56] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Inf. Sci.*, vol. 250, pp. 113–141, Nov. 2013.
- [57] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Syst. Appl.*, vol. 73, pp. 220–239, May 2017.
- [58] Q. Li, B. Yang, Y. Li, N. Deng, and L. Jing, "Constructing support vector machine ensemble with segmentation for imbalanced datasets," *Neural Comput. Appl.*, vol. 22, no. S1, pp. 249–256, May 2013.
- [59] Z. Sun, Q. Song, X. Zhu, H. Sun, B. Xu, and Y. Zhou, "A novel ensemble method for classifying imbalanced data," *Pattern Recognit.*, vol. 48, no. 5, pp. 1623–1637, May 2015.



YONGMING WU received the Ph.D. degree in mechanical engineering from Xiamen University, Xiamen, China, in 2014. He has been a Professor with the Key Laboratory of Advanced Manufacturing Technology, Ministry of Education. He has published more than 40 papers in major journals and international conferences. His current research interests include big data of manufacturing and intelligent manufacturing.



YINGBO LIU received the B.S. degree in automation and the M.S. degree in computer science and application from the Kunming University of Science and Technology, in 2008 and 2011, respectively, and the Ph.D. degree from the University of Chinese Academy of Sciences, in 2015. He was a Postdoctoral Researcher with the Yunnan Academy of Scientific and Technical Information, from 2015 to 2018, and a Research Fellow with SCSE of NTU, Singapore, from 2018 to 2019. He is currently an Assistant Researcher with the Yunnan University of Finance and Economics. He is also the Director of the Cloud Computing Center of Big Data Research Institute of Yunnan Economy and Society. His research interests include big data, cloud computing, and distributed systems.



TIANSONG LIU received the B.S. degree in mechanical from Zaozhuang University, Shandong, China, in 2019. He is currently pursuing the M.S. degree with the Key Laboratory of Advanced Manufacturing Technology, Ministry of Education, Guizhou University, China. His research interest includes data analysis for industry.



LINSHENG CHEN received the B.S. degree in mechanical engineering from Xihua University, Chengdu, China, in 2018. He is currently pursuing the M.S. degree with the Key Laboratory of Advanced Manufacturing Technology, Ministry of Education, Guizhou University, China. His research interests include machinery condition monitoring and data analysis for industry.



XIAOJING SHENG received the B.S. degree in computer science and technology from the Zhengzhou Normal University, Zhengzhou, China, in 2019. She is currently pursuing the M.S. degree with the Key Laboratory of Advanced Manufacturing Technology, Ministry of Education, Guizhou University, China. Her research interest includes data analysis for industry.

...