Sirte University Scientific Journal Vol. 15, No. 2 (2025) 42-49

Sirte University Scientific Journal (SUSJ)

Journal home page: http://journal.su.edu.ly/index.php/SUSJ/index
DOI: 10.37375/susj.v15i2.3722

High Speed Sintering (HSS) Modelling and prediction via Radial Basis Function Neural Network (RBFNN)

Adnan Hamad[1], George Panoutsos[2]

[1]*Omar Al-Mukhtar University, Faculty of Engineering, El-Bayda, Libya*

[2]*University of Sheffield, School of Electrical and Electronic Engineering, Sheffield S1 3JD, UK*

DOI: https://doi.org/10.37375/susj.v15i2.3722

ARTICLE INFO:

Received 14 October 2025.
Accepted 25 November 2025.
Available online 24 December 2025.

Keywords:

High speed sintering (HSS), Radial Basis Function neural network (RBFNN), Fault detection (FD)

A B S T R A C T

An additive manufacturing technique called High Speed Sintering (HSS) has enormous potential for producing intricate, superior polymer parts on a large scale. HSS process is modelled in this paper using a novel Radial Basis Function neural network (RBFNN) technique. The data gathered from the HSS process was analysed to determine the healthy/unhealthy data that could achieve a good/bad build. A powerful technique is developed for early fault detection (FD) and, consequently, to predict the quality of the parts produced using HSS. The RBFNN model was validated and tested to assess the robustness of the approach, and the simulation outcomes demonstrated that the faults could be clearly identified, and the quality of the produced parts possibly will be predicted.

1 Introduction

The new process known as High-Speed Sintering (HSS) is based on printing successive cross-sections. An infrared lamp is flashed throughout the entire building area after the HSS procedure deposits a Radiation Absorbing Material (RAM) over the relevant area. The region will absorb enough energy to raise the temperature of the underlying powder to its melting point, enabling sintering to take place. The part's un-sintered powder will be eliminated after the building's operation is complete (Majewski, C. E. et al., 2007) (Majewski, C. E. et al., 2008). Figure 1 depicts the HSS procedure (Kemnitzer, J. et al., 2024). An undesirable deviation of one or more system variables from the typical/healthy behaviour is referred to as a fault. During the build operation of HSS, Faults may arise during the process; as a result, damaged parts will be constructed, resulting in additional time and expense spent. Because reliability and safety are becoming more and more important, there is a greater focus on monitoring technical applications and processes (Isermann, R., 1984). Numerous studies on fault isolation and detection have been conducted in both academic and industrial domains within the past 30 years. The primary goal of fault isolation and detection is to enable a process system to run properly without any malfunctions that could result in subpar performance or inappropriate behaviour. A number of system malfunctions, including sensor, actuator, and component faults, can happen in a healthy

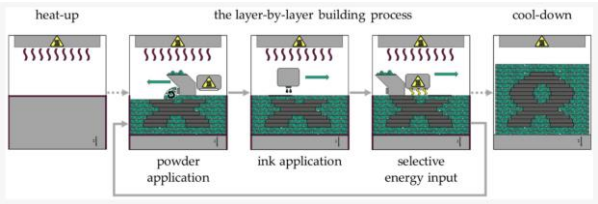

Figure 1: HSS process

system. Actuators, sensors (instruments), or process elements themselves may all have flaws. Instrument fault detection (IFD), actuator fault detection (AFD),

***Corresponding author*: E-mail: adnan.hamad@omu.edu.ly**

and component fault detection (CFD) are the terms used to describe fault detection (FD) of these defects. Globally, there is a growing interest in the fundamental studies of fault detection and isolation (Frank, P. M.,1992). Analytical (i.e., functional) redundancy, as opposed to physical redundancy, is the foundation of the FD technique. This suggests that FD takes advantage of the built-in redundancy in the static and dynamic links between the system inputs and measured outcomes. Functional redundancy uses residuals as characteristics and deals with two or more functions that describe the same process (Gertler, J. J., 1988). Stubbs, S. et al. (2012) have described a simplified monitoring-specific canonical variate analysis (CVA) state space modelling approach for FD in dynamic processes. A more straightforward CVA state space model with three matrices was presented in their paper specifically for process monitoring. Fuzzy-model-based robust FD with stochastic mixed time delays and sequential packet dropouts was introduced by Dongo, H. et al. (2012). The network-based resilient FD problem for a class of uncertain discrete-time Takagi-Sugeno fuzzy systems with sequential packet dropouts and stochastic mixed time delays is the focus of their research. In order to maintain guaranteed performance and exponentially stable FD dynamics, a fuzzy FD filter has been created. It has been demonstrated that artificial neural networks (ANN) are highly effective for modelling and controlling nonlinear systems (Narendra, K.S. and Parthasarathy, K., 1990) (Narendra, K. S., 1996). If feedback links are added, ANN can also be used to model nonlinear dynamic systems. High processing speeds, high input error tolerance, and adaptability are all features of neural networks. The artificial neural network (ANN) technique to fault identification and isolation has been proposed by numerous research and journal articles as a solution to modelling and classification issues in dynamic systems. In the last several years, RBFNNs have become incredibly popular as a substitute for the slowly convergent multi-layer perception. Like the multi-layer perception, the RBFNN can model any non-linear function (Nelles, O., 2001) (Patan, K., 2008). RBFNNs were proposed by Yu et al. (1999) for the diagnosis of process faults. It was examined how to diagnose actuator, component, and sensor failures by using the output prediction error as a residual between a neural network model and a non-linear dynamic process. Finding a potent method for FD for the HSS is the primary goal of this study. Consequently, the application of the RBF artificial neural network (ANN) for modelling and early FD in the HSS is suggested in this paper and will be used to identify various types of defects and therefore to avoid producing faulty parts. This sort of neural network has been chosen because it has a simple structure, and it is easy to train. This FD technique will be a significant contribution to the industry area if the simulation results are satisfactory and if the defects can be identified and detected.

2 Neural Network Modelling

2.1 Introduction

An outstanding mathematical technique for handling non-linear issues is artificial neural networks (ANNs). They are particularly helpful in cases where the process has not been mathematically modelled, making it impossible to apply traditional techniques like observers or parameter estimation methods (Patan, K., 2008). With minimal or no prior knowledge of the procedure, a neural network can use the learning method to extract the system attributes from past training data. This offers a tremendous deal of flexibility in modelling non-linear systems (Zhai, Y. J. and Yu, D. L., 2007). RBFNNs will be utilised in this project; they were selected due to their straightforward structure, can mimic any non-linear function and ease of training.

2.2 RBFNNs structure

Figure 2 depicts the architecture of RBFNNs. The RBFNN comprises three layers: the input layer, a single hidden layer, the non-linear layer, and the linear output layer. The input vector is denoted by x, the hidden layer output vector by h, the weight matrix by W, and the output vector by $\hat{y}$ (Patan, K., 2008) (Zhai, Y. J. and Yu, D. L., 2007) (Oliver N. O., 2001). The network's size

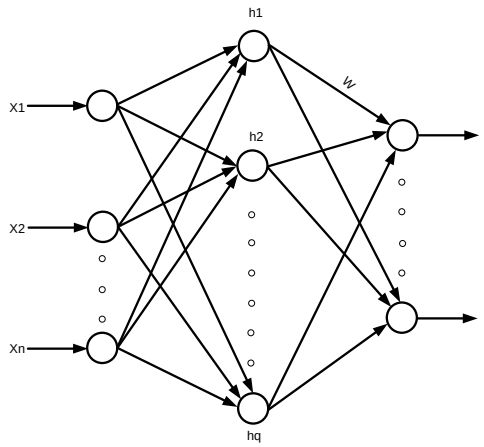

Figure 2: The RBFNN Structure.

affects training time. Consequently, the neural network's size should be modest in order to minimise training time; for this reason, the hidden layer has been selected as a single layer. In this study, using the K-means clustering algorithm, the RBF centre is chosen from a set of training data. Reducing the total squared distances between each input data point and its closest centre is the aim of the activation functions in order to adequately cover the data (Zhai, Y. J. and Yu, D. L., 2007) (Oliver N. O., 2001).

3 HSS Modelling using RBFNN

Over the past 30 years, artificial neural networks have been the subject of extensive research and have been effectively used for dynamic system modelling. Since neural networks can handle the most complex scenarios that are too poorly specified for deterministic algorithms to handle, they offer an intriguing and beneficial substitute for traditional approaches (Patan, K., 2008). This project models HSS using an RBFNN because it provides an excellent computational ability for handling non-linear challenges and training is extremely fast.

3.1 Collecting data

Getting appropriate data for training is the initial stage of modelling HSS using RBFNN. The goal of conducting an experiment scheme using such data is to manage the evaluated data as usefully as possible, subject to any available constraints, as the data will affect the network modelling implementation's accuracy. In order to ensure that the training data sufficiently covers the designated operating zone, the input signals must stimulate the process's dynamic modes at various frequencies. Lightbody, G. and Irwin, G. W., (1997) presented a mongrel irritation signal for neural network training. While identifying linear systems may merely require a constantly stimulating input signal, nonlinear system identification requires additional considerations. A neural network model's process modelling is divided into two sections: the process dynamics are captured in the first, while the basic nonlinear transmitter function is approximated in the second (Zhai, Y. J. and Yu, D. L., 2007). Five inputs were taken from the temperature data during the operation, and four inputs were the settings of the build process. A set of data was gathered for the nine input signals of the HSS. As an example, Figures 3 and 4 display the temperature data both before and after filtration. Table 1 displays the data that was gathered. Utilising Eq. (1), the temperature data (T) is separated.

$$T_{filtered}(t) = \mu_T \quad T_{filtered}(t-1) + (1-\mu_T)T(t) \qquad (1)$$

Where the value of μT, a low-pass filter coefficient, can be selected between 0 and 1. To improve the neural network's accuracy and reduce errors, all input data from HSS will be scaled to the (0, 1) range prior to training or validating the RBFNN. The output data is already 0 or 1, thus scaling is not necessary. Accordingly, a score of 0 indicates an unsuccessful build, while a score of 1 indicates a successful one. Eq. (2) makes use of the linear scale.

$$\mathcal{U}_s(t) = \frac{\mathcal{U}(t) - \mathcal{U}_{min}}{\mathcal{U}_{max} - \mathcal{U}_{min}} \qquad (2)$$

where $\mathcal{U}_s$ is the scaled input and $\mathcal{U}_{min}$ and $\mathcal{U}_{max}$ are the data set's minimum and maximum inputs, respectively. The threshold was used to calculate the inaccuracy in order to assess the modelling effects. The Eq. (3) and (4) provide the inaccuracy.

$$\text{Error} = \frac{1}{N}\sum_{t=1}^{N}|tr - \mathcal{Y}(t)| = \frac{1}{N}\sum_{t=1}^{N}|e(t)| \qquad (3)$$

$$\text{MAE\%} = \left(\frac{1}{N}\sum_{t=1}^{N}|Y(t) - Error(t)|\right) * 100 \quad (4)$$

An average of absolute errors is called the MAE. $\mathcal{Y}(t)$ is the HSS model's output (predicted output), and tr is the threshold that is selected for modelling performance. Y(t) is the HSS's actual output.

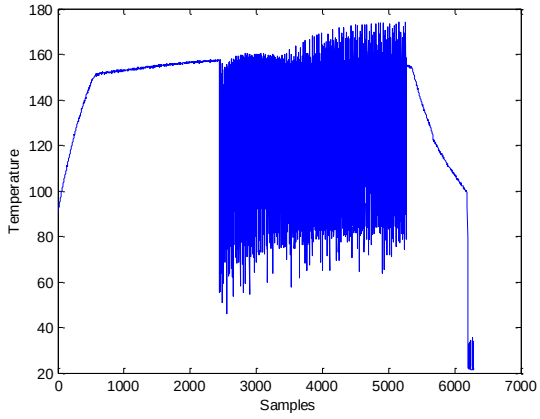

Figure 3: Temperature data before filtering

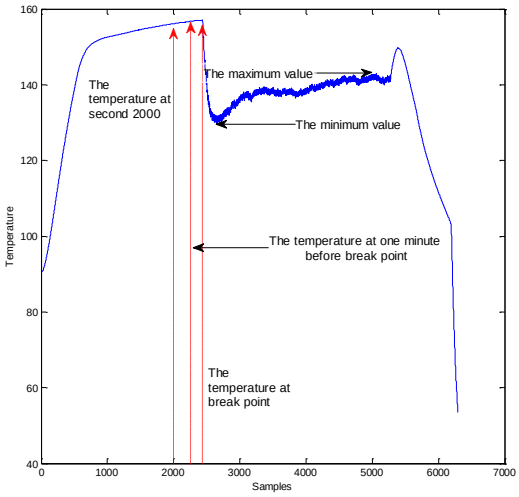

Figure 4: Temperature data after filtering

Table 1. The collected inputs data.

	The input data	Range
1	Initial power input % (ip)	47 and 53
2	Greyscale (gs)	1 and 5
3	Build overhead C˚ (bo)	142 and 152
4	Pre-head power % (pp)	35 and 70
5	Temperature at the second number 2000 (C˚)	134.4 and 156.1
6	Temperature data at one minute before the break point (C˚)	142.7 and 157
7	Temperature at the break point (C˚)	144.8 and 157.1
8	The minimum value of the temperature during the build operation (C˚)	119.1 and 133
9	The maximum value of the temperature during the build operation (C˚)	

3.2 Model structure selection

Identifying the RBFNN model's input variables is the second stage. As previously stated, the HSS to be simulated includes nine input variables, and one output vector has the values 0 and 1: (1) denotes a successful construction, whereas (0) denotes a failed build. The input from the network that caused the least oversights in modelling was chosen as shown in Figure 5. There are ten inputs and one output in the RBF model. 34 nodes have been chosen as the hidden layer nodes. Prior to training, the K-means clustering approach was used to select 34 centers, and the p-nearest-neighbors algorithm was used to determine the width σ. The width was the same for all 34 hidden layer nodes' Gaussian functions. The recursive squares algorithm's weights W were utilised for training.

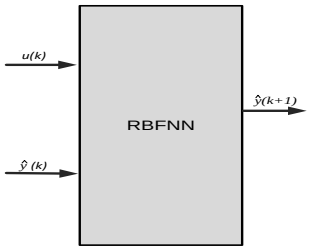

Figure 5: Structure of the RBFNN model

3.3 Training and validating the model

The HSS system produced a data set of 49 samples in total. The 34 samples in the first batch were used to train neural networks, while the 15 samples in the second set were utilised to validate the model. The model training and validation results for the 15 samples in the test data set and the 34 samples in the training data set are displayed in Figure 6. In general, it is evident that the two outputs match well with a negligible error. The set of test data has a lower modelling error than the set of training. The modelling impacts are assessed using the mean absolute error (MAE %). The MAE value for this model is 0.0206.

3.4 Simulation results

Overall, a satisfactory prediction between the output of the HSS and the output of RBFNN was obtained. Also, the RBFNN model was trained and tested using 34 hidden nodes, and the simulation results were excellent. Figure 6 shows that there is a very slight mismatch in the MAE between the RBFNN and the HSS output.

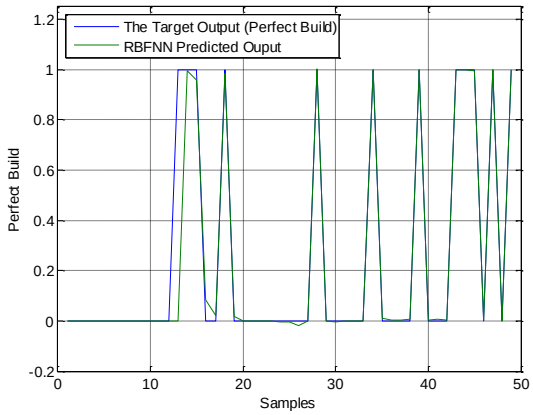

Figure 6: HSS outputs and prediction of the RBF model

3.5 Cross validation

Since the model has previously seen the data, it is generally a methodological error to use the same data set for both training and testing. Consequently, the model would just replicate the sample labels. In that scenario, the model might produce flawless test results, but once it uses unseen data, it might not be able to predict any outcomes. Applying the cross-validation procedure while excluding a portion of the available training data set is necessary to solve this issue. After the model has been trained, the deleted data will be regarded as fresh data and can be used to evaluate the model. The entire data set for this research consisted of 49 samples, and cross validation was applied four times, using different data sets for training and validation each time. Eight samples for validation and forty-one samples for training were used in the cross-validation. The model training and validation outcomes are displayed in Figures 7 - 14. Every figure has a threshold that is selected and shown in these figures as well. Additionally, the threshold for each figure is used to define the MAE%. The MAE% for each model throughout train and validation is finally determined by calculating the average value of the threshold. Each RBFNN trained model's hidden nodes and accuracy for the train and validation, respectively, are displayed in Tables 2–3. Additionally, as shown in Figures 15 -16, some of the input variables have been plotted in 3D to determine the correlation between them.

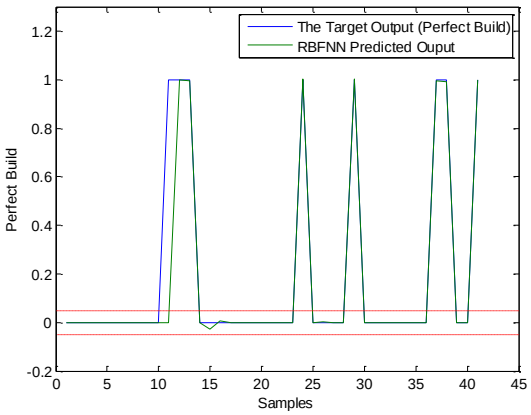

Figure 7: RBFNN prediction and HSS output of the first model training

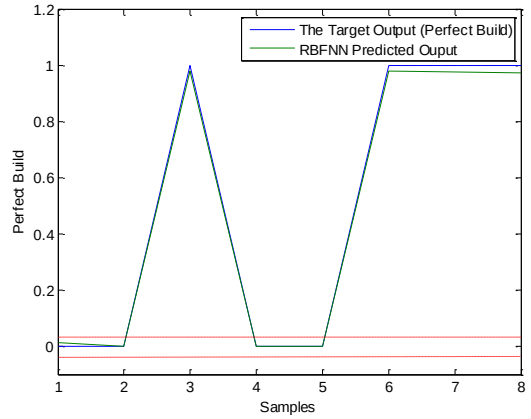

Figure 8: RBFNN prediction and HSS output of the first model validation

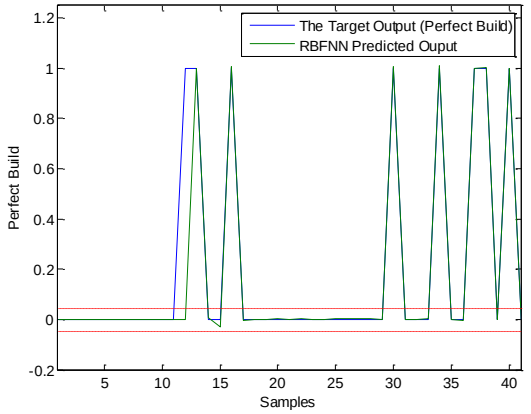

Figure 9: RBFNN prediction and HSS output of the second model training

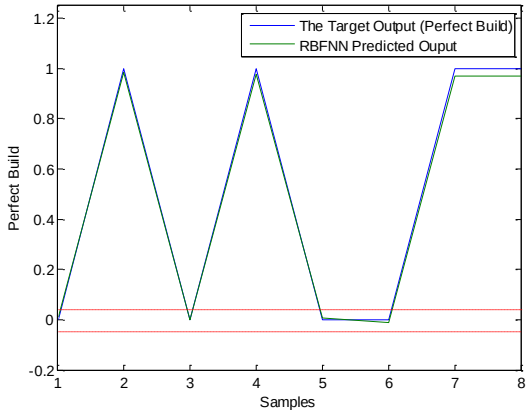

Figure 10: RBFNN prediction and HSS output of the second model validation

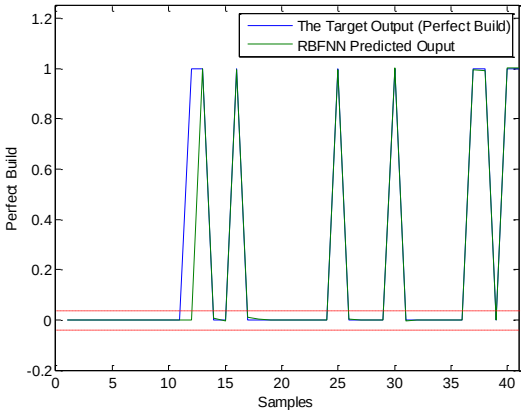

Figure 11: RBFNN prediction and HSS output of the third model training

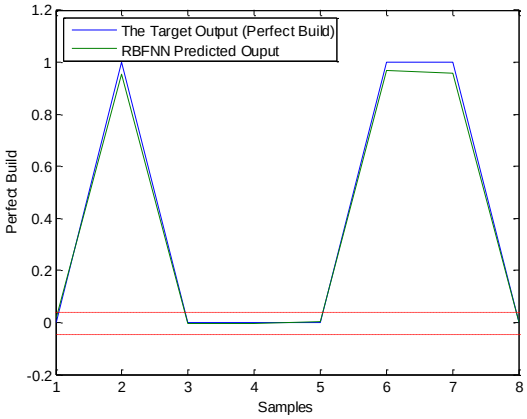

Figure 12: RBFNN prediction and HSS output of the third model validation

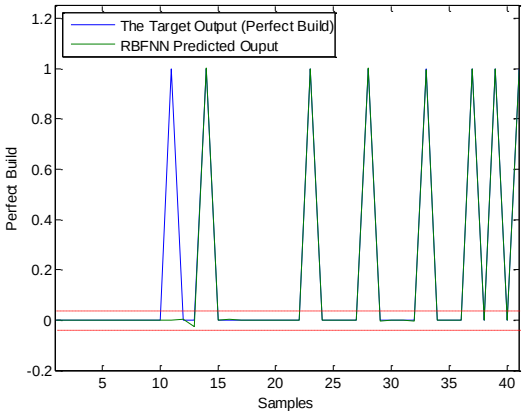

Figure 13: RBFNN prediction and HSS output of the fourth model training

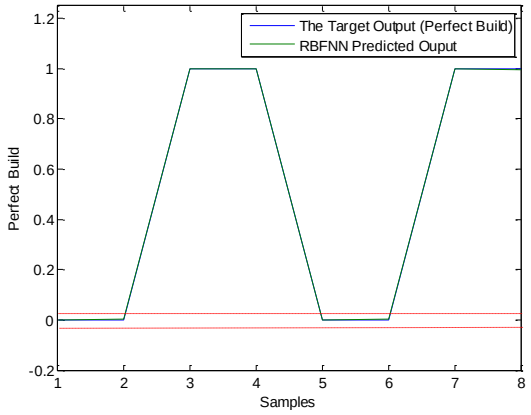

Figure 14: RBFNN prediction and HSS output of the fourth model validation

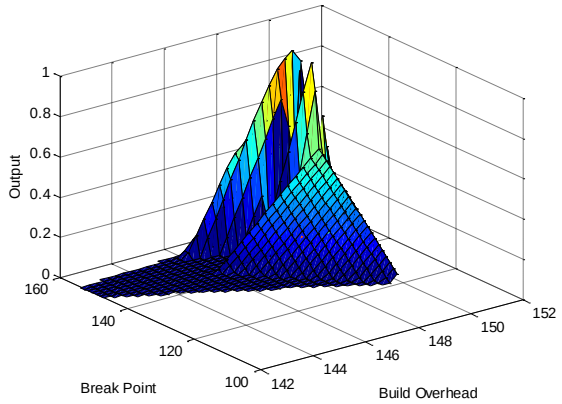

Figure 15: The correlation between break point and build overhead

Table 2. The trained models' accuracy.

RBFNN model	**Hid-den nodes**	**Threshold**	**Accuracy**	**The mean Threshold**	**Accuracy**
Model (1)	14	± 0.05	95.45 %	0.0475	97.55 %
Model (2)	13	± 0.05	92.43 %	0.0475	97.56 %
Model (3)	17	± 0.05	91.66 %	0.0475	97.53 %
Model (4)	21	± 0.04	91.63 %	0.0475	97.55 %
Average accuracy			92.79 %		97.54 %

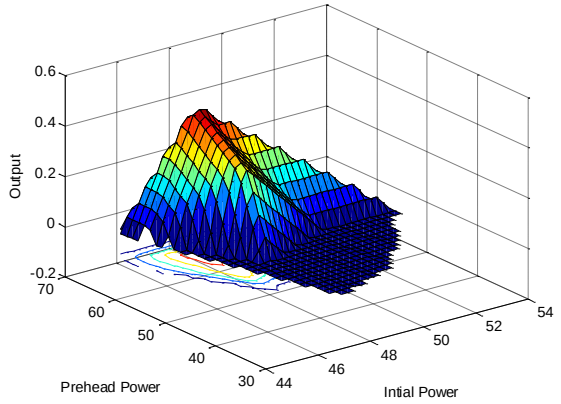

Figure 16: The correlation between pre-head power and initial power

Table 3. The validation models' accuracy.

RBFNN model	**Thres-hold**	**Accura-cy**	**The mean Thresho-ld**	**Accura-cy**
Model (1)	0.03	99.63 %	0.025	99.87 %
Model (2)	0.03	99.17 %	0.025	99.84 %
Model (3)	0.02	99.53 %	0.025	99.80 %
Model (4)	0.02	99.73 %	0.025	99.93 %
Average accuracy		99.51 %		99.86 %

4 Conclusions

This project has chosen the RBFNN to model the HSS process because it has an ability to model many non-linear systems. The RBFNN and the training model have been presented and described in this paper. The input data set has been obtained from the experiment of the HSS build operation. An RBF model was designed with 10 inputs and 1 output and the hidden layer nodes were selected for each model. RBFNN model was trained and tested using the collected data. Moreover, cross validation was applied 4 times in order to prove robust the RBFNN model. With only minor errors, the simulation results for model training and testing demonstrate good results and an excellent correspondence between the output of the HSS and RBFNN. As a result, any flawed build could be quickly

and easily identified before the process began. The accuracy and the average accuracy of train and test the proposed model are listed in Tables 2 and 3.

Conflict of interest: The authors declare that there are no conflicts of interest

References

Dongo H, Wang Z, Lam J, Gao H (2012) Fuzzy-model-based robust fault detection with stochastic mixed time delays and successive packet dropouts. IEEE Transactions on System, Man, and Cybernetics-Part B: Cybernetics, Vol. 42, No. 2.

Frank PM (1992) Principles of model-based fault detection. Control, delft the IFAC Artificial intelligence in Real-Time, Netherlands, Vol. 3 pp. 213- 220

Gertler J J (1988) Model-based failure detection and isolation in complex plants. IEEE Control System Magazine, pp. 3-11.

Isermann R (1984) Process fault detection based on modelling and estimation methods- a survey. Automatica, Vol. 20, pp. 387- 404.

Kemnitzer J, Wimmer M, Tarasova A, Döpper F (2024) High Speed Sintering of Polyamide 12: From Powder to Part Properties, Polymers 2024, 16(24), 3605; https://doi.org/10.3390/polym16243605

Lightbody G, Irwin GW (1997) Nonlinear Control Structures Based on Embedded Neural System Models. IEEE Transactions on neural networks, 8(3): 553-567.

Majewski CE, Hobbs BS, Hopkinson N (2007) Effect of bed temperature and infra-red lamp power on the mechanical properties of parts produced using high-speed sintering, Virtual physical prototyping, vol. 2, No. 2, 103-110

Majewski CE, Oduye D, Thomas HR, Hopkinson N (2008) Effect of level on the sintering behavior in the high-speed sintering process. Rapid Prototyping journal, 14/3, 155-160.

Narendra KS, Parthasarathy K (1990) Identification and control of dynamic systems using neural networks. IEEE Trans. Neural Networks , Vol. 1, No.1.

Narendra KS (1996) Neural networks for control: theory and practice. Proc. IEEE, Vol. 84, No. 10, pp. 1385-1406.

Nelles O (2001). Nonlinear system identification. Springer.

Patan K (2008) Artificial neural network for the modelling and fault diagnosis of technical process. Springer- Verlag Berlin Heidelberg.

Stubbs S, Zhang J, Morris J (2012) Fault detection in dynamic processes using a simplified monitoring-specific CVA state space modelling approach. Computer and Chemical Engineering, Elsevier, Vol. 41, pp. 77-87.

Yu DL, Gomm JB, Williams D (1999) Sensor fault diagnosis in a chemical process via RBF neural networks. Control Engineering Practice, Vol. 7, pp. 49-55.

Zhai YJ, Yu DL (2007) Radial Basis Function Based Feeddback Control for Air Fuel Ratio of Spark Ignition. Proc. Instn mech. Engrs, part D: J. automobile engineering, vol 222, pp. 415-428.