Int j simul model 22 (2023) 4, 734-745
Original scientific paper

ADAPTIVE FAULT PREDICTION AND MAINTENANCE IN PRODUCTION LINES USING DEEP LEARNING

Pang, J. L.#
College of Electronic Information, Huanghuai University, Zhumadian 463000, China
E-Mail: pangjianli@huanghuai.edu.cn (# Corresponding author)

Abstract
In the era of Industry 4.0 and intelligent manufacturing, accurately predicting and maintaining production line faults is crucial in manufacturing. This study introduces a novel deep learning-based adaptive fault prediction and maintenance strategy, overcoming limitations of traditional statistical and machine learning methods in prediction accuracy and adaptability in complex environments. A new prediction model is developed, incorporating a wide convolutional feature extraction module, a customized gating module, and a multi-layered progressive extraction module. The model's process and parameters, including fault stage division using Wasserstein distance and optimization with L2 regularization and neuron dropout, are detailed. An adaptive maintenance strategy for predictive fault detection is established, enhancing precision in fault prediction and developing more effective maintenance approaches, ultimately boosting production efficiency and reducing operational costs.
(Received in August 2023, accepted in October 2023. This paper was with the author 5 weeks for 1 revision.)

Key Words: Deep Learning, Adaptive Production Lines, Fault Prediction, Maintenance Strategies, Wasserstein Distance, L2 Regularization, Neuron Dropout

1. INTRODUCTION

The advent of Industry 4.0, integrating artificial intelligence, big data, and machine learning, has revolutionized intelligent manufacturing [1-5]. Production lines, central in this landscape, directly affect efficiency, output, and economic performance [6-9]. Thus, precise fault prediction and timely maintenance in production lines, utilizing these advanced technologies, are essential [10-12]. Research into deep learning-facilitated, adaptive fault prediction and maintenance for production lines holds significant value [13]. Deep learning allows for detailed analysis of complex data, enabling accurate fault prediction, vital for enhancing efficiency and quality, and developing proactive maintenance strategies to minimize production disruptions [14-18]. Current studies primarily use statistical and traditional machine learning methods for fault prediction, showing limitations in accuracy and adaptability in dynamic production environments [19, 20]. Existing maintenance strategies also lack flexibility to adjust to real-time conditions, limiting their effectiveness [21].

Addressing these gaps, this study presents a deep learning-based adaptive fault prediction and maintenance strategy. It includes an adaptive fault prediction model integrating a wide convolutional feature extraction module, a customized gating module, and a multi-layered progressive extraction module, enhancing fault prediction accuracy. The model's process and parameters, including fault stage division using Wasserstein distance and parameter optimization through L2 regularization and neuron dropout, are detailed. An adaptive maintenance strategy is also formulated, creating effective maintenance plans based on real-time production line status, reducing losses and improving efficiency.

2. MODULES IN PRODUCTION LINE FAULT PREDICTION

In adaptive production environments, the deployment of numerous high-precision, high-frequency sensors for fault monitoring generates a vast array of detailed operational data. The complexity and diversity of this data present significant challenges for effective fault prediction.

https://doi.org/10.2507/IJSIMM22-4-CO20

Addressing these challenges, this study introduces an adaptive model for production line fault prediction, leveraging multi-scale convolution and multi-layered progressive extraction techniques.

This model employs multi-scale convolution to capture the data's diversity and complexity across various temporal and spatial dimensions, thereby enhancing prediction accuracy. The multi-layer progressive extraction process meticulously refines and concentrates features at each level, augmenting the model's ability to understand and assimilate deeper layers of information, which in turn improves predictive efficacy. This architectural approach not only simplifies the model's complexity but also effectively prevents overfitting and strengthens its generalization capabilities, optimizing fault prediction in adaptive production lines.

The model comprises three main modules: a wide convolutional feature extraction module, a customized gating module, and a multi-layer progressive extraction module. The first-layer wide convolutional module is specifically designed to extract features from high-frequency data signals generated by sensors. It employs wide-kernel convolution to capture the breadth of data at a superficial level and narrow convolutional kernels within this window to dissect deeper semantic features of signal anomalies. This dual approach maximizes the use of sensor data, enhancing the accuracy and reliability of fault prediction.

Each module consists of five basic units, incorporating a convolution layer, batch normalization layer, activation layer, and pooling layer. These units form a layered structure for progressive feature extraction, allowing the model to better learn and interpret the inherent structure and patterns of the data. The convolution calculation formula employed within these units, involving an input feature map matrix denoted as U_u and a convolution operation represented by $\otimes$, is an integral part of this sophisticated feature extraction process.

$$D_k = \varphi\left[J_{u,k} \otimes U_u + N_k\right] \tag{1}$$

After the convolution computation, the activation function φ processes the obtained result, thereby generating a new feature map D_u.

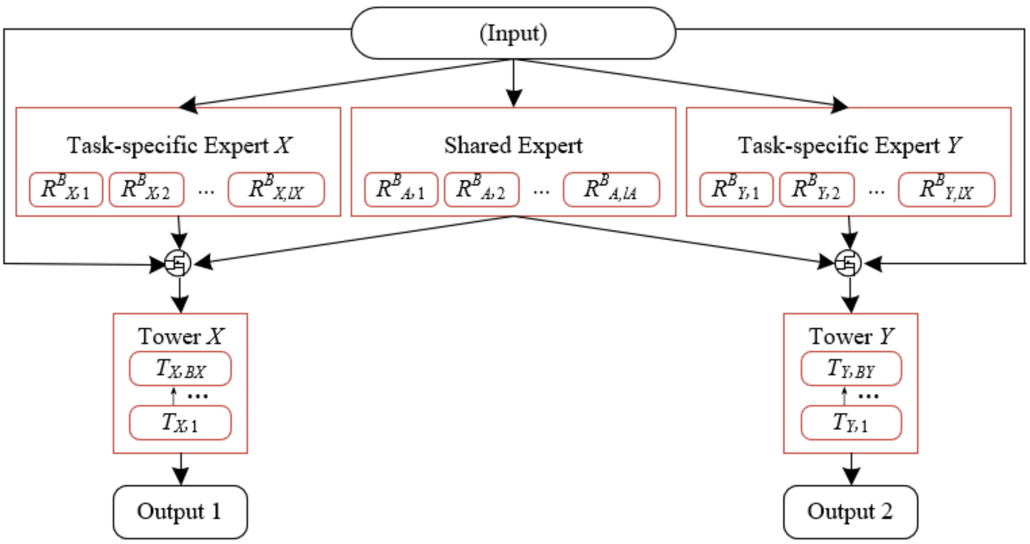


Figure 1: Structure of the customized gating module.

In adaptive production lines, handling multiple distinct fault prediction tasks is common, each with unique characteristics and shared correlations. Traditional multi-task network models often face imbalances, prioritizing some tasks over others, leading to "negative transfer" and "seesaw problems." This study introduces a customized gating module (see Fig. 1) to mitigate this issue. It features a gating mechanism and a shared expert network, combining a parameter-sharing structure with a multi-gate mixed expert network.

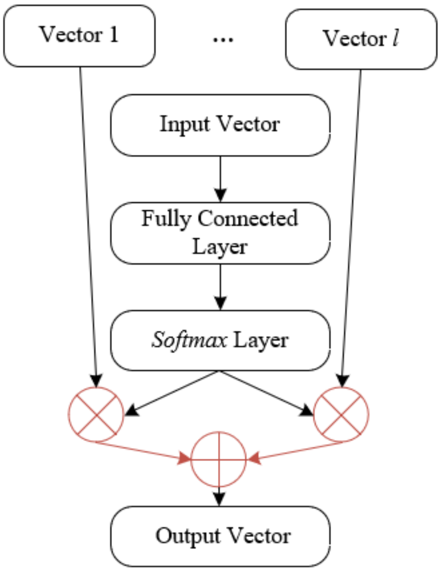


Figure 2: Gating network structure.

Fig. 2 presents the structure of the gating network. The fundamental principle is that the gating module dynamically determines how much information each task obtains from the shared expert network, thereby maintaining information sharing among tasks. Simultaneously, the shared expert network allows each task to retain a degree of independence, preventing a decrease in predictive performance due to excessive reliance on information from other tasks. This mechanism enables the model to handle multiple tasks more balanced and reasonably, enhancing the accuracy and robustness of fault prediction and better meeting the practical demands of adaptive production lines. Assuming the task-specific expert sub-network for task j is represented by (j, l_j) and the shared expert sub-network for other tasks by (a, l_a), the following formula expresses the feature processing layer $A^j(z)$ formed by the outputs of these sub-networks in the multi-gate mixed expert network:

$$A^j(z) = \left[R^Y_{j,1}, R^Y_{j,2}, \cdots, R^Y_{j,l_j}, R^Y_{a,1}, R^Y_{a,2}, \cdots, R^Y_{a,l_a}\right] \tag{2}$$

The gating module, composed of a fully connected layer and a Softmax output layer, can weight the input $h^j(z)$ of each fault prediction task with the outputs from the various sub-networks of the multi-gate mixed expert network. Assuming the weights corresponding to the sub-networks are represented by $q^j(z)$, the following calculation formula is applied:

$$h^j(z) = q^j(z) A^j(z) \tag{3}$$

By combining the initial input vector with the output vector of the expert network, the different sub-network weights $q^j(z)$ corresponding to a single fault prediction task can be determined upon completion of network training, as follows:

$$q^j(z) = \mathrm{Softmax}\left(Q^j_h z\right) \tag{4}$$

In adaptive production lines with multiple, distinct fault prediction tasks, a model architecture that balances information sharing and task independence is essential. This paper introduces a multi-layer progressive extraction multi-task module, building upon a single-layer customized gating module. It segregates the expert network into shared and task-specific networks: the shared network handles common information across tasks, while the task-specific network focuses on unique task information. An innovative progressive routing method allows for effective information extraction and representation learning across layers. This coordinated approach between shared and task-specific networks ensures a balance of information sharing and independence, enhancing the model’s predictive accuracy.

Assuming the input of task j at the k^{th} layer of the extraction network is denoted as $h^{j,k}(z)$, the weight corresponding to the expert sub-network when the input of task j in the k^{th} layer extraction network is $h^{j,k-1}(z)$ is represented by $q^{j,k}(h^{j,k-1}(z))$, and the selection matrix for task j in the k^{th} layer extraction network is denoted as $A^{j,k}(z)$. Finally, by combining the sub-networks of the multi-gate mixed expert network, the sub-network for the next layer is obtained as:

$$h^{j,k}(z) = q^{j,k}\left(h^{j,k-1}(z)\right)A^{j,k}(z) \tag{5}$$

3. SIMULATION PROCESS AND PARAMETER SETTINGS

Utilizing the three key modules discussed earlier, this study's prediction model excels in multi-task learning for diverse fault prediction tasks across various sensors and fault stages in adaptive production lines, ensuring high accuracy.

The model begins by extracting low-dimensional feature vectors from sensor-collected samples using a wide convolutional network, effectively reducing data dimensionality and mining intrinsic features. It then employs Wasserstein distance analysis for fault stage categorization, enhancing fault prediction and diagnosis accuracy under a multi-task learning framework. In testing, sample dimensions are similarly reduced, and the trained model outputs precise predictions and diagnoses of faults in adaptive production lines. Figure 3 illustrates the training and testing process of this model.

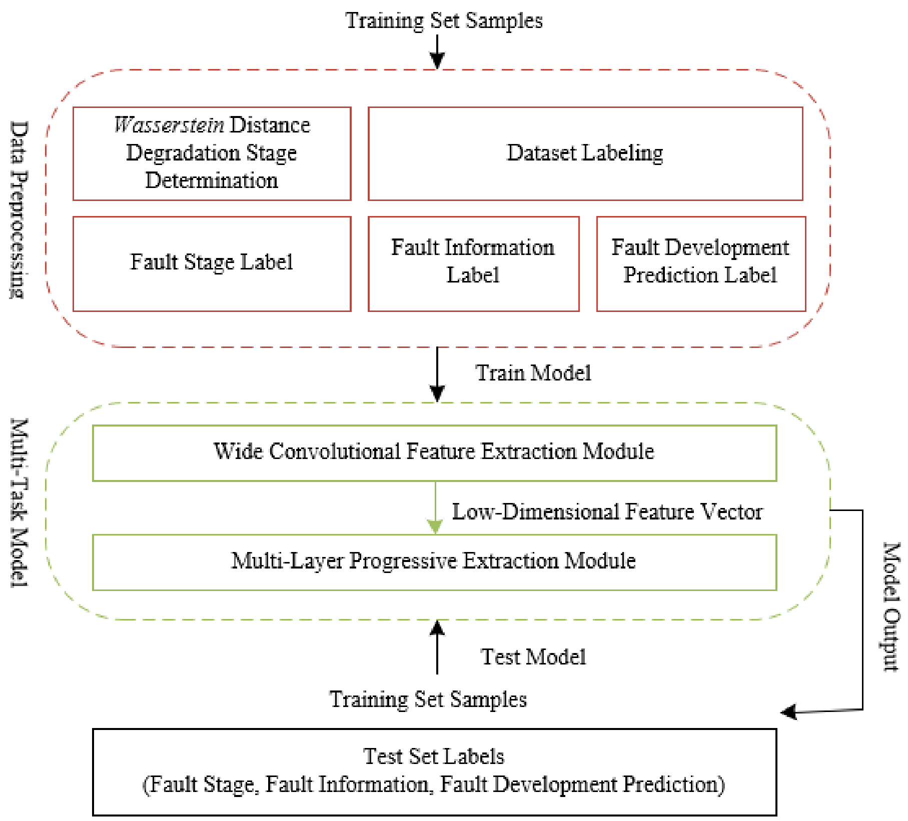


Figure 3: Training and testing process for adaptive production line fault prediction model.

In adaptive production lines, faults typically progress through multiple stages, each with distinct features. To enhance fault prediction and diagnosis, this paper employs Wasserstein distance to measure the similarity between samples from different stages, facilitating automatic fault stage identification. By integrating fault stage information into the prediction model, it gains a deeper understanding of data patterns, reducing the risk of overfitting associated with single-task learning and improving fault prediction accuracy.

Suppose the set of all possible joint distributions formed by combining probability distributions O_1 and O_2 is represented by $\Pi(O_1, O_2)$. The following formula defines the Wasserstein distance:

$$F_Q(O_1, O_2) = \underset{\varepsilon \sim \Pi(O_1, O_2)}{INF} R[\|z - t\|] \tag{6}$$

Sampling from $\Pi(O_1,O_2)$ produces $(z, t) \sim \varepsilon$, generating real samples z and synthetic samples c. The distance between z and c can be represented by $\|z\text{-}t\|$. Let the dataset sampled over the entire lifecycle of adaptive production line faults be denoted as DS, with the u^{th} sampling point represented by u, and the sampling endpoint by l, then there is:

$$DS = \{DA_1, DA_2, \cdots, DA_u, \cdots, DA_l\} \tag{7}$$

In the process of dividing the stages of adaptive production line faults, four steps are undergone:

Firstly, sample normalization. In actual production lines, data originate from multiple different sensors with varying value ranges and units. Direct utilization of this raw data for calculations might result in biases due to differences in units and ranges. Therefore, the first step involves normalizing the samples, converting all data to the same scale. This not only eliminates the impact of units but also avoids computational errors due to excessively large value ranges.

This study sampled five equal-length samples within a fixed time period, as follows:

$$DA_u = \frac{DA_u - \omega}{\delta} \tag{8}$$

$$DA_u \to \{s_1, s_2, s_3, s_4, s_5\} \tag{9}$$

Secondly, the calculation of the Wasserstein distance between samples is required to characterize the disparity in sample distributions between samples u and k. By computing the Wasserstein distance among all samples, a distance matrix can be established, which intuitively reflects the similarity between samples. Suppose the Wasserstein distance between the u^{th} and k^{th} samples is represented by q_{uk}, then there is:

$$Q = \begin{bmatrix} q_{11} & \cdots & q_{1b} \\ \vdots & \ddots & \vdots \\ q_{b1} & \cdots & q_{bb} \end{bmatrix} = [q_1, q_2, \cdots, q_b] \tag{10}$$

Subsequently, the calculation of the Pearson correlation coefficient is essential. This coefficient is a statistical metric that indicates the linear relationship between two variables. By determining the Pearson correlation coefficient for each pair of samples, a correlation matrix is constructed. This matrix aids in deepening the understanding of the interrelationships among the samples. If the average distance of sample u from other samples is represented by q_u^-, and that of sample k by q_k^-, the corresponding formula is as follows:

$$\vartheta_{uk} = \frac{\sum (q_u - \bar{q}_u)(q_k - \bar{q}_k)}{\sqrt{\sum (q_u - \bar{q}_u)^2 \sum (q_k - \bar{q}_k)^2}} \tag{11}$$

The correlation coefficient for all samples can be obtained through the following expression:

$$\vartheta = \{\vartheta_{12}, \vartheta_{23}, \cdots, \vartheta_{b-1,b}\} \tag{12}$$

Finally, the minimum value of the correlation coefficient curve must be determined. In this step, the point on the correlation coefficient curve with the smallest value is identified. The coordinate of this point represents the fault stage to be divided, namely the initial fault occurrence time y. As the smaller the value of the correlation coefficient, the lower the correlation between the two variables, the point corresponding to this minimum value represents the division point of the fault stage.

$$y = \arg\min(\vartheta) \tag{13}$$

In the variable and complex conditions of adaptive production lines, where data is often noisy and contains outliers, the model's generalization ability and stability are crucial. To address this, the study implements L2 regularization and neuron dropout mechanisms to enhance stability and prevent overfitting. L2 regularization adds the L2 norm of parameter weights as a penalty to the loss function, encouraging simpler models and reducing complexity. This is particularly effective in fault prediction and diagnosis for adaptive production lines, where data complexity can lead to overfitting. Consequently, L2 regularization aids in improving the model's generalization capabilities.

$$K(\phi) = \frac{1}{2l}\left[\sum_{u=1}^{l}\left(g_\phi(z_u) - t_u\right)^2 + \eta\sum_{k=1}^{l}\phi_k^2\right] \tag{14}$$

Assuming the number of samples is represented by l, the loss function by $K(\cdot)$, the samples and corresponding labels by z and t, and the L2 regularization factor by η. The partial derivative of the loss function with respect to the weight parameters is as follows:

$$\frac{\sigma K(\phi)}{\sigma \phi_k} = \frac{1}{l}\sum_{u=1}^{l}\left(g_\phi(z_u) - t_u\right)z_{uk} + \frac{\eta}{l}\phi_k \tag{15}$$

Suppose the learning rate is denoted by λ. Further, for the optimization of parameters, gradient descent updating is performed:

$$\phi_k \to \phi_k - \lambda \cdot \frac{\sigma K(\phi)}{\sigma \phi_k} = \left(1 - \frac{l\eta}{l}\right)\phi_k - \frac{\lambda}{l}\sum_{u=1}^{l}\left(g_\phi(z_u) - t_u\right)z_{uk} \tag{16}$$

It can be observed that with the addition of the regularization term, the rate of parameter update becomes slower, as follows:

$$1 - \frac{\beta\eta}{l} < 1 \tag{17}$$

The neuron dropout mechanism is a key method to boost a model's stability, particularly useful in fault prediction and diagnosis for adaptive production lines with complex and variable conditions. By randomly deactivating a portion of neurons during training, it reduces the model's dependency on specific neurons, enhancing stability and robustness. This is crucial in adaptive production lines, where data distribution can change, and over-reliance on certain neurons could diminish predictive performance. The neuron dropout mechanism thus effectively improves the model's robustness.

$$x_u^{(m+1)} = q_u^{(m+1)} t^m + n_u^{(m+1)} \tag{18}$$

$$t_u^{(m+1)} = d\left(x_u^{(m+1)}\right) \tag{19}$$

The structural changes of the network after introducing the neuron dropout mechanism can be characterized by the following expressions:

$$e_k^{(m)} \sim \mathrm{Bernoulli}(o) \tag{20}$$

$$\tilde{t}^{(m+1)} = e^{(m)} * t^{(m)} \tag{21}$$

$$x_u^{(m+1)} = q_u^{(m+1)} \tilde{t}^m + n_u^{(m+1)} \tag{22}$$

$$t_u^{(m+1)} = d\left(x_u^{(m+1)}\right) \tag{23}$$

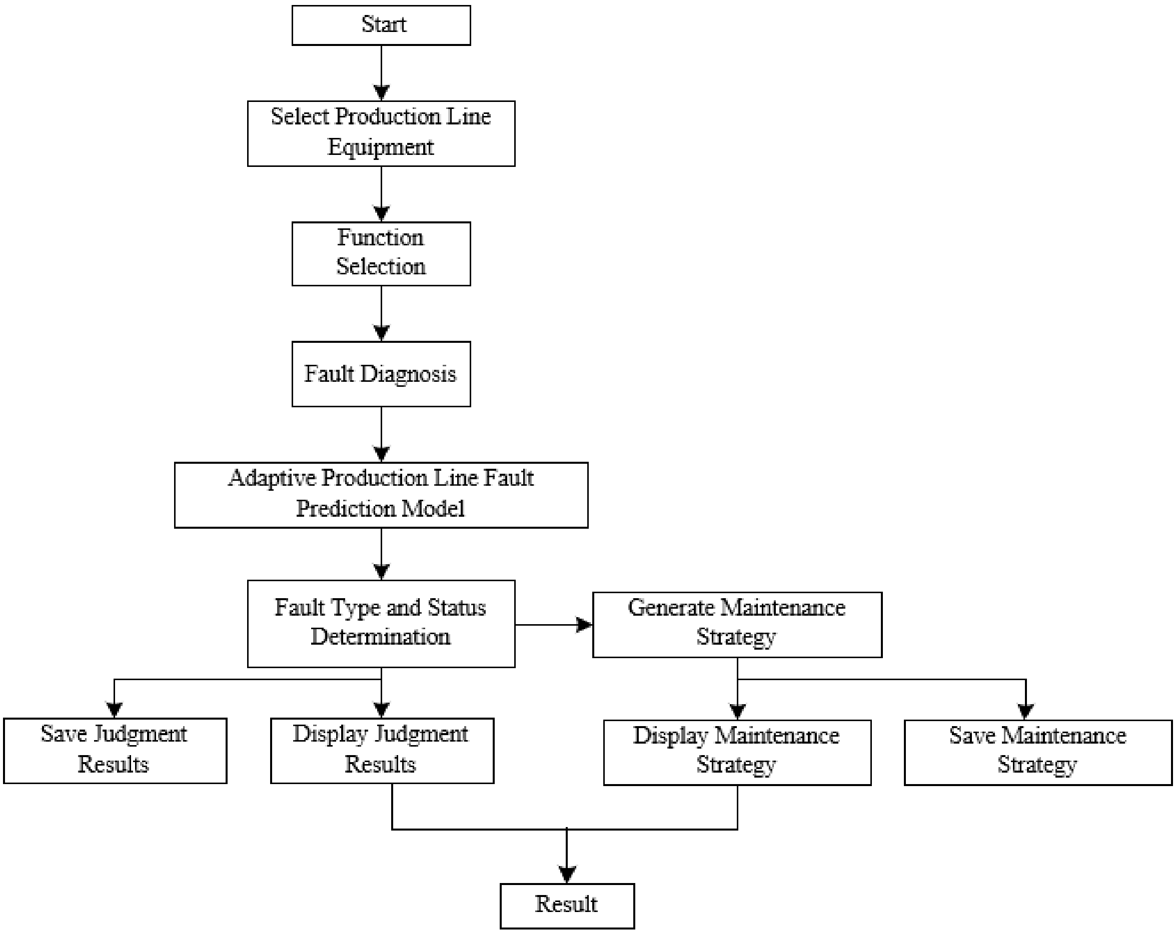


Figure 4: Functional process flowchart of the adaptive production line fault predictive maintenance system.

It can be observed that some weights are deactivated during the current round of training after undergoing processing.

Fig. 4 illustrates the process flowchart of the adaptive production line fault predictive maintenance system, which generates a strategy based on the line's real-time status. The steps include:

(1) Using the fault prediction model to identify potential fault locations and equipment.
(2) Prioritizing these locations and equipment based on fault impact and occurrence probability.
(3) Developing a maintenance plan detailing timing, location, and method.
(4) Executing real-time maintenance according to the plan and priorities.
(5) Feeding maintenance results back into the model to refine its accuracy.

This adaptive maintenance strategy not only minimizes downtime and boosts production efficiency but also continually enhances the fault prediction model's accuracy, stabilizing production line operations.

4. EXPERIMENTAL RESULTS AND ANALYSIS

Fig. 5 shows a schematic of fault stages in adaptive production lines, with the horizontal axis representing sample number and the vertical axis indicating the Wasserstein distance correlation coefficient. A significant change at the 82nd sample point marks a fault occurrence, demonstrating the model's effectiveness in distinguishing between normal and faulty states using Wasserstein distance.

Fig. 6 displays the *RMS* fault stage, where *RMS* values remain low until a notable increase after the 82nd sample point, indicating a fault or anomaly. This increase often signals a negative shift in the equipment's operational state, such as wear or faults, and serves as an early warning for maintenance. The model's ability to provide early alerts helps reduce downtime and maintenance costs, boosting production efficiency and equipment reliability.

Table I compares the performance of various fault prediction models across three production lines. The deep learning-based adaptive model proposed in this paper outperforms others (MMOE Model, CNN, First-Layer CNN, and BP Neural Network) in predictive *RMSE*, with values of 5.48, 7.23, and 9.64, and classification accuracy rates of 97.25 %, 97.85 %, and 96.89 % on the respective lines. These results demonstrate the model's superior accuracy and consistent reliability over 96 % across all lines. Thus, this model excels in both predictive *RMSE* and classification accuracy, offering more precise fault predictions, reducing downtime, saving repair costs, and enhancing production efficiency.

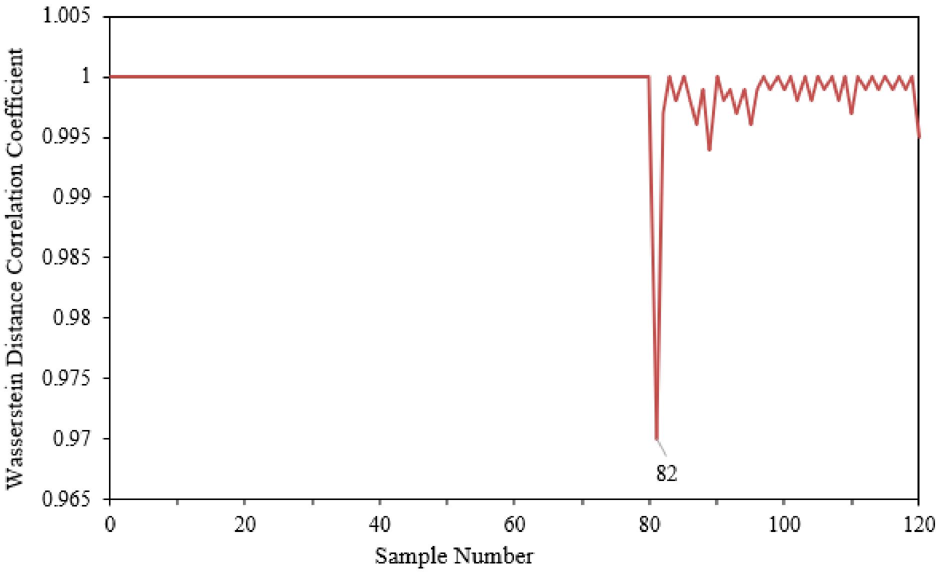


Figure 5: Schematic diagram of fault stages in adaptive production lines.

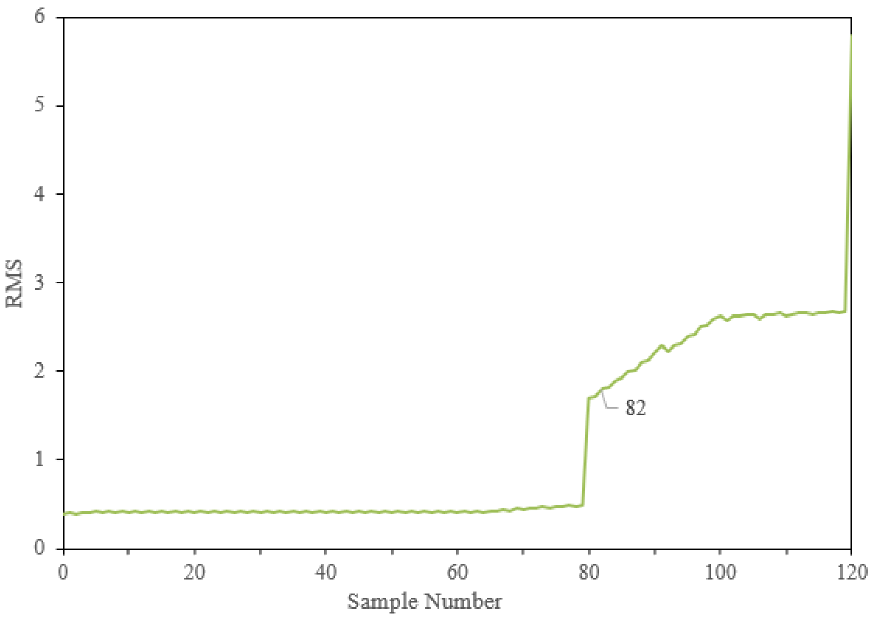


Figure 6: *RMS* fault stage in adaptive production lines.

Table I: Comparative performance of equipment fault prediction of three production lines in the dataset.

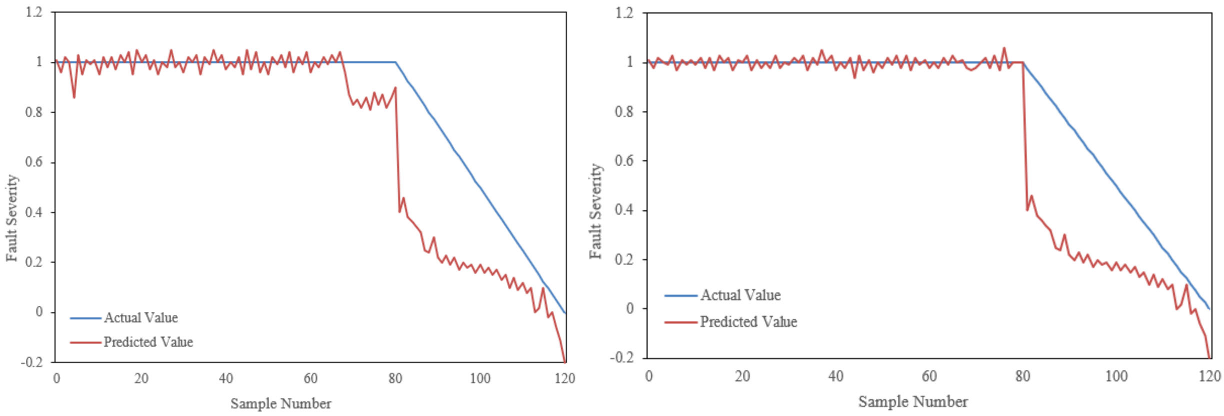
		Production line 1	Production line 2	Production line 3
The proposed model	Predictive *RMSE*	5.48	7.23	9.64
	Classification accuracy	97.25 %	97.85 %	96.89 %
MMOE model	Predictive *RMSE*	5.69	7.26	13.25
	Classification accuracy	95.36 %	95.86 %	95.24 %
CNN	Predictive *RMSE*	7.89	14.25	11.26
	Classification accuracy	94.52 %	96.32 %	92.47 %
First-Layer CNN	Predictive *RMSE*	7.68	13.28	21.47
	Classification accuracy	94.56 %	96.38 %	94.58 %
BP Neural Network	Predictive *RMSE*	12.36	22.58	25.36
	Classification accuracy	92.33 %	93.44 %	88.56 %

Table II: Comparative experimental results for different prediction tasks.

Task	Production line number	Average *RMSE*	*RMSE* standard deviation
Fault prediction	1	5.48	1.12
	2	7.23	2.61
	3	9.68	1.85
Fault stage determination	1	6.23	0.89
	2	8.12	2.63
	3	12.36	1.95
Fault diagnosis + Fault stage determination	1	4.98	2.31
	2	12.63	4.95
	3	16.32	4.51

The data provided in Table II displays the experimental results for different prediction tasks-fault prediction, fault stage determination, and a combination of fault diagnosis and fault stage determination-across three different production lines. The effectiveness of the model can be evaluated through an analysis of the average *RMSE* and *RMSE* standard deviation. In the fault prediction task for the three production lines, the model's average *RMSE* values were 5.48, 7.23, and 9.68, respectively, indicating a high level of accuracy in fault prediction. Simultaneously, the *RMSE* standard deviations were 1.12, 2.61, and 1.85, indicating good stability in prediction results for Production lines 1 and 3, while line 2 was somewhat less stable. In determining different stages of faults, the model's average *RMSE* on each production line increased to 6.23, 8.12, and 12.36, respectively, reflecting that the increased complexity of the task may lead to a decrease in predictive accuracy. The *RMSE* standard deviations remained broadly consistent with those in the fault prediction task, suggesting that the stability of prediction results was not significantly impacted by the increased complexity of the task. In the combined task of fault diagnosis and fault stage determination, Production line 1 showed improved model performance, with the average *RMSE* decreasing to 4.98, but the *RMSE* standard deviation increased to 2.31, indicating that although average predictive accuracy improved, the variability of results also increased. For Production lines 2 and 3, the average *RMSE* values significantly increased to 12.63 and 16.32, respectively, and the *RMSE* standard deviations also substantially rose, highlighting challenges in predictive stability and accuracy in more complex tasks. From this data analysis, it can be concluded that the deep learning-based adaptive production line fault prediction model proposed in this paper performs well in the sole task of fault prediction, demonstrating high accuracy and stability.

From the above figure, a comparison can be made between the prediction results under two different task settings: "Fault prediction" and "Fault diagnosis + Fault stage determination." In the figure, two curves can be observed: one representing the actual data of fault severity, and the other representing the fault severity values predicted by the model. It is evident that, in most of the time series, the two curves closely follow each other, indicating that the model is able to predict the actual events with high accuracy most of the time. At the fault maintenance points, the two curves show some deviations. The conclusion can be drawn that the prediction model can consistently match the actual data accurately prior to the occurrence of faults, demonstrating the model’s good understanding and predictive ability under normal operational conditions. The fault maintenance strategy has been effective in mitigating the severity of faults. The maintenance strategy of the proposed model provides timely intervention in fault maintenance for adaptive production lines, helping to prevent production losses and reduce downtime.



a) One auxiliary task b) Two auxiliary tasks

Figure 7: Comparative prediction results under two different task settings.

5. CONCLUSION

This paper has successfully designed a deep learning-based adaptive production line fault prediction model. This model utilizes a wide convolutional feature extraction module, a customized gating module, and a multi-layer progressive extraction module. These modules combined enable more accurate capture of the characteristics of equipment operational data and prediction of potential faults. To segment fault stages, the model employs a method based on Wasserstein distance, and it also uses L2 regularization and neuron dropout mechanism to enhance the accuracy and stability of predictions. Finally, an adaptive maintenance strategy for predictive fault detection in production lines is proposed, formulating maintenance plans based on real-time status to strategically reduce production losses and increase efficiency.

Experimental results indicate that the model proposed in this paper achieved lower *RMSE* values and higher classification accuracy in fault prediction tasks across different production lines, outperforming MMOE models, CNN, first-layer CNN, and BP neural networks. In tasks involving fault stage determination and the more complex fault diagnosis plus fault stage determination, the model faced greater challenges but still demonstrated good predictive accuracy on some production lines, indicating its adaptability. Comparative analysis of model predictions and actual fault data showed that the model closely matched actual data across most time series, while rapidly adjusting its output during faults to align with new data patterns.

The deep learning-based adaptive production line fault prediction model introduced in this paper demonstrates high accuracy and robustness, surpassing several existing models in fault prediction tasks and showing feasibility in complex tasks. By incorporating advanced feature extraction modules, fault stage division methods, and model optimization techniques, an effective strategy for fault prediction and preventive maintenance was successfully proposed. This strategy not only reduces potential downtime and losses in production lines but also provides a theoretical and practical basis for intelligent maintenance in adaptive production lines, holding significant importance for the development of intelligent manufacturing systems in the context of Industry 4.0.

ACKNOWLEDGEMENT

This work was supported by the Henan Provincial Higher Education Teaching Reform Research and Practice Project (Grant No.: 2021SJGLX536).

REFERENCES

[1] Lindegren, M. L.; Lunau, M. R.; Mafia, M. M. P.; Ribeiro da Silva, E. (2022). Combining simulation and data analytics for OEE improvement, *International Journal of Simulation Modelling*, Vol. 21, No. 1, 29-40, doi:10.2507/IJSIMM21-1-584

[2] Nguyen, H. T.; Lam, B. Q.; Nguyen, Q. L. H. T. T.; Nguyen, P. T.; Huynh, V. D. B. (2022). Career path choice in the Industry 4.0 using grey decision making, *Journal of System & Management Sciences*, Vol. 5, No. 12, 268-281, doi:10.33168/JSMS.2022.0516

[3] Simon, J.; Trojanová, M.; Hošovský, A.; Sárosi, J. (2021). Neural network driven automated guided vehicle platform development for Industry 4.0 environment, *Technical Gazette*, Vol. 28, No. 6, 1936-1942, doi:10.17559/TV-20200727095821

[4] Stojanović, D.; Joković, J.; Tomašević, I.; Simeunović, B.; Slović, D. (2023). Algorithmic approach for the confluence of lean methodology and Industry 4.0 technologies: challenges, benefits, and practical applications, *Journal of Industrial Intelligence*, Vol. 1, No. 2, 125-135, doi:10.56578/jii010205

[5] El Abbadi, L.; Elrhanimi, S.; el Manti, S. (2020). A literature review on the evolution of lean manufacturing, *Journal of System and Management Sciences*, Vol. 10, No. 4, 13-30, doi:10.33168/JSMS.2020.0402

[6] Wang, Y.; Zhang, S. Y.; Zhang, Q. P.; Lin, S. M.; Pang, G. S. (2022). IoT-based distributed simulation of industrial automation production line management, *International Journal of Simulation Modelling*, Vol. 21, No. 4, 696-707, doi:10.2507/IJSIMM21-4-CO18

[7] Sinay, J.; Markulik, S.; Pacaiova, H. (2017). Quality as a part of modern technology in the automotive industry, *Proceedings of the* 2nd *EAI International Summit, Smart City 360°*, 9 pages, doi:10.4108/eai.14-2-2017.152167

[8] Liu, M.; Lü, X.; Qi, X. (2012). Design of automatic seedling production line based on AT89S52, *Transactions of the Chinese Society of Agricultural Engineering*, Vol. 28, No. 2, 229-232

[9] Dabic-Miletic, S. (2023). Advanced technologies in smart factories: a cornerstone of Industry 4.0, *Journal of Industrial Intelligence*, Vol. 1, No. 3, 148-157, doi:10.56578/jii010302

[10] Jia, L. M.; Wang, Y. (2022). Research on industrial production defect detection method based on machine vision technology in industrial Internet of Things, *Traitement du Signal*, Vol. 39, No. 6, 2061-2068, doi:10.18280/ts.390618

[11] Mbilong, P. M.; Aarab, Z.; Belouadha, F.-Z.; Kabbaj, M. I. (2023). Enhancing fault detection in CNC machinery: a deep learning and genetic algorithm approach, *Ingénierie des Systèmes d'Information*, Vol. 28, No. 5, 1361-1375, doi:10.18280/isi.280525

[12] Padhee, S.; Nandan, D. (2021). Design of automated visual inspection system for beverage industry production line, *Traitement du Signal*, Vol. 38, No. 2, 461-466, doi:10.18280/ts.380225

[13] Wei, W.; Liu, L.; Yang, M.; Li, J.; Wu, F. (2021). Predictive maintenance system for production line equipment based on digital twin and augmented reality, Wang, Y.; Martinsen, K.; Yu, T.; Wang, K. (Eds.), *Advanced Manufacturing and Automation XI. IWAMA 2021*, Lecture Notes in Electrical Engineering, Vol. 880, 479-486, doi:10.1007/978-981-19-0572-8_61

[14] Gao, F.; Dai, Y.; Cao, S. (2022). Failure prediction and predictive maintenance of key equipment in seamless steel tubes & pipes production line, *Proceedings of the 6th International Conference on ThermoMechanical Processing*, 181-188

[15] Garcia, E.; Montés, N.; Llopis, J.; Lacasa, A. (2022). Miniterm, a novel virtual sensor for predictive maintenance for the Industry 4.0 era, *Sensors*, Vol. 22, No. 16, Paper 6222, 15 pages, doi:10.3390/s22166222
[16] Carvalho, T. P.; Soares, F. A.; Vita, R.; Francisco, R. D. P.; Basto, J. P.; Alcalá, S. G. (2019). A systematic literature review of machine learning methods applied to predictive maintenance, *Computers & Industrial Engineering*, Vol. 137, Paper 106024, 10 pages, doi:10.1016/j.cie.2019.106024
[17] Schwendemann, S.; Amjad, Z.; Sikora, A. (2021). A survey of machine-learning techniques for condition monitoring and predictive maintenance of bearings in grinding machines, *Computers in Industry*, Vol. 125, Paper 103380, 17 pages, doi:10.1016/j.compind.2020.103380
[18] Hülsmann, M.; Förstner, R. (2020). Predictive maintenance as enhancement of fault detection, isolation and recovery strategies of spacecraft, *Proceedings of the 71st International Astronautical Congress – The Cyber Space Edition*, *IAF Space Operations Symposium 2020* (at IAC 2020), Paper 56580, 74-83
[19] Raabe, H.; Myklebust, O.; Eleftheriadis, R. (2018). Vision based quality control and maintenance in high volume production by use of zero defect strategies, Wang, K.; Wang, Y.; Strandhagen, J.; Yu, T. (Eds.), *Advanced Manufacturing and Automation VII. IWAMA 2017*, Lecture Notes in Electrical Engineering, Vol. 451, 405-412, doi:10.1007/978-981-10-5768-7_43
[20] Zhang, Z.; Li, Z.; Zhao, C. (2022). Research on condition monitoring and fault diagnosis of intelligent copper ball production lines based on big data, *IET Collaborative Intelligent Manufacturing*, Vol. 4, No. 1, 45-57, doi:10.1049/cim2.12043
[21] Kapadia, J.; Tabibzadeh, M. (2017). An integrated system-oriented risk and reliability analysis methodology to improve maintenance strategies of beverage filling and packing line equipment, *Proceedings of the 2017 Industrial and Systems Engineering Conference*, 824-829