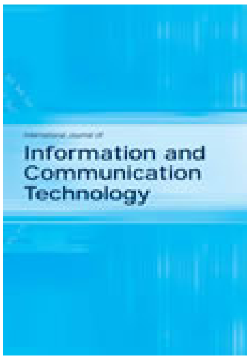


International Journal of Information and Communication Technology

ISSN online: 1741-8070 - ISSN print: 1466-6642
https://www.inderscience.com/ijict

Optimisation of PID control for tobacco feeding rail based on convolutional neural network

Guoqing Xu, Bin Peng, Yufei Li

DOI: 10.1504/IJICT.2026.10075849

Article History:

Received:	02 August 2025
Last revised:	22 November 2025
Accepted:	23 November 2025
Published online:	04 February 2026

Optimisation of PID control for tobacco feeding rail based on convolutional neural network

Guoqing Xu

Liuzhou Cigarette Factory,
China Tobacco Guangxi Industrial Co., Ltd.,
Liuzhou, 545005, China
Email: 13078069270@163.com

Bin Peng

Qingdao Vic Powder Metallurgy Co., Ltd.,
Qingdao, 266000, China
Email: 110615254@qq.com

Yufei Li*

China Tobacco Guangxi Industrial Co., Ltd.,
Nanning, 530001, China
Email: lzhlyfei@163.com
*Corresponding author

Abstract: The tobacco wire guide system is a key component in cigarette production equipment. This paper proposes a tobacco wire guide PID control optimisation model based on convolutional neural network (CNN-PID). The tobacco wire guide speed, guide position, guide temperature, ambient temperature, ambient humidity, and PID parameters at the previous moment are selected as model dependent variables. After normalisation, they are input into the lightweight convolutional neural network. After model parameter adjustment, the predicted proportional gain, integral gain and derivative gain are finally output. After training, the R^2-score values of CNN-PID on proportional gain, integral gain and derivative gain are 0.992, 0.984, and 0.982, respectively. In addition, the R^2-score values of the CNN-PID model are better than those of traditional PID control, BP neural network PID, and PSO optimised PID.

Keywords: PID control optimisation; convolutional neural network; CNN; intelligent control algorithm.

Reference to this paper should be made as follows: Xu, G., Peng, B. and Li, Y. (2026) 'Optimisation of PID control for tobacco feeding rail based on convolutional neural network', *Int. J. Information and Communication Technology*, Vol. 27, No. 4, pp.1–15.

Biographical notes: Guoqing Xu received his Bachelor's degree from Zhengzhou University of Light Industry in July 2002. He is currently working at Liuzhou Cigarette Factory of China Tobacco Guangxi Industrial Co., Ltd. His research interests include cigarette-making units equipment maintenance management.

Bin Peng received his Bachelor's degree from Qingdao Feiyang College in 2007. He is currently working at Qingdao Vic Powder Metallurgy Co., Ltd. His research interests include manufacturing of tobacco machinery and its applications.

Yufei Li holds a Master's degree and the title of Engineer. He is currently employed at China Tobacco Guangxi Industrial Co., Ltd. His research interests include cigarette processing technology and process management.

1 Introduction

In recent years, deep learning methods have developed rapidly. Schmidhuber (2015) sorted out the evolution of neural networks from perceptrons to LSTMs, explained the biological inspiration of the backpropagation algorithm, demonstrated the superiority of deep structures in speech recognition and natural language processing, and proposed the framework of 'neural network computability theory'. Golovko (2017) analysed the mathematical principles of the three major paradigms of deep learning, focusing on the differences between CNN and RNN in feature extraction and time series modelling; explored the impact of hardware acceleration (GPU/TPU) on training efficiency, and defense strategies against adversarial sample attacks. Xia (2019) reviewed the key technologies of deep learning in target detection, from traditional HOG features to end-to-end models such as YOLO/SSD, analysed the limitations of multi-scale detection and occlusion processing, and explored transfer learning solutions in small sample scenarios. Wang et al. (2022) summarised lightweight CNN technology, quantised MobileNetv3 to 8-bit integers, improved accuracy through knowledge distillation, and realised real-time inference on edge devices.

As a mature method of deep learning, convolutional neural networks have also developed rapidly. Chua (1997) proposed the convolutional neural network (CNN) theory, providing a mathematical foundation for brain-like computing hardware. Zhang et al. (2019) systematically analysed the spectral domain/spatial domain theoretical framework of GCN, compared the generalisation capabilities of variants such as GAT and GraphSAGE in social network analysis and molecular property prediction, and pointed out the computational bottleneck of heterogeneous graph learning. Alzubaidi et al. (2021) systematically reviewed the theoretical basis of deep learning and the design evolution of mainstream CNN architectures, analysed the challenges of gradient vanishing and overfitting in training, and summarised its application bottlenecks in medical imaging, autonomous driving and other fields; proposed future breakthroughs in model interpretability, lightweight deployment and cross-modal learning. Lu et al. (2021) constructed a CNN-BiLSTM-AM stock price prediction model: CNN extracts local patterns of K-line, BiLSTM models long-term trends, and the attention mechanism focuses on key fluctuation nodes. Cao et al. (2022) reviewed the breakthroughs of graph convolutional networks (GCN) in point cloud processing: dynamic graph construction algorithms (such as DGCNN) were designed to capture local geometric structures, and mIoU reached 89.7% in point cloud segmentation tasks. Yue et al. (2022) proposed a dynamic region graph convolutional network (DRGCNN) to divide local regions of point clouds by adaptive clustering.

Deep learning and convolutional networks have been widely used in the tobacco industry. Bi et al. (2020) used reinforcement learning (DQN) to dynamically control tobacco drying temperature and humidity, reducing energy consumption by 12% and the standard deviation of moisture content ≤ 0.8%. Jingyuan et al. (2023) proposed a temporal convolutional network (FATCN) with fusion attention mechanism to solve the problem of strongly coupled nonlinearity in the drying process, and shorten the control delay to within 5 seconds. Kusiak (2020) explored the application of convolution and generative adversarial networks (GAN) in digital twins, generated synthetic data to optimise manufacturing parameters, predicted tool wear error ≤ 0.05 mm, and promoted closed-loop control of smart factories. Tulbure et al. (2022) deepened the research on DCNNs in micro-defect detection and proposed a multi-spectral fusion method to enhance the ability to identify surface scratches, reducing the missed detection rate to 0.8% in aluminium alloy plate detection. Khanam et al. (2024) comprehensively reviewed the innovative application of CNN in industrial defect detection, compared the accuracy and real-time performance of Faster R-CNN, Mask R-CNN and other models in the quality inspection of aerospace parts and electronic components, and proposed a generative adversarial network (GAN) enhancement strategy to solve data scarcity. Vashishtha et al. (2024) detailed the engineering practice of CNN in rotating machinery fault diagnosis, developed 1D-CNN to directly analyse the vibration signal spectrum, replaced the traditional time-frequency analysis method, and achieved early fault warning in the case of wind power gearbox (accuracy 96.5%). Farkh et al. (2021) proposed to capture the posture deviation of the mobile robot in real time through visual sensors, extract spatial features through CNN and output PID parameter adjustment signals, and reduce the trajectory tracking error by 42% in dynamic obstacle scenes. Li et al. (2022) conducted an in-depth study on the deep optimisation method of workpiece surface defect classification, designed a multi-branch CNN to fuse local texture and global deformation features, combined with transfer learning to reduce labelling dependence, and achieved a recall rate of 98.2% in rail crack detection. Kim et al. (2023) designed an AI-driven hybrid control algorithm, integrated CNN environmental perception and reinforcement learning decision modules, optimised the torque distribution strategy of robot vehicles on slippery roads, and achieved zero collision obstacle avoidance at a speed of 80 km/h. Perišić and Jovanović (2023) reviewed the current status of CNN integration in automatic control systems, covering vision-based PLC fault diagnosis (false alarm rate < 3%) and real-time dynamic planning (response delay ≤ 10 ms), and pointed out the energy consumption optimisation path for FPGA deployment. Wang et al. (2023) developed a hierarchical CNNPID active steering model: the upper-layer CNN predicts the emergency lane change trajectory, and the lower-layer PID dynamically adjusts the steering angle. The joint simulation shows that the lateral acceleration fluctuation is reduced by 35%, avoiding the risk of loss of control. Xie et al. (2023) developed a CNN-LSTM-PID battery thermal management virtual sensor, combined the electrochemical model with the operating data to predict the temperature distribution (error < 1.5℃), and solved the problem of on-board BMS sampling delay. Xie et al. (2024) verified the feasibility of scaled robots in autonomous driving research, designed a 1:10 simulation platform to reproduce the intersection conflict scene, and accelerated the iteration efficiency of the collision test algorithm by 300%.

In the past few years, more and more deep learning technologies have been integrated into the industrial machinery industry, making great contributions to parameter adjustment. However, there are still the following drawbacks that need to be further optimised:

1. the input data features used are single and the features proposed by the network are not comprehensive enough
2. the model is complex, the training cycle is long, and the training difficulty is high. In response to the above problems, this paper proposes a model named CNN-PID.

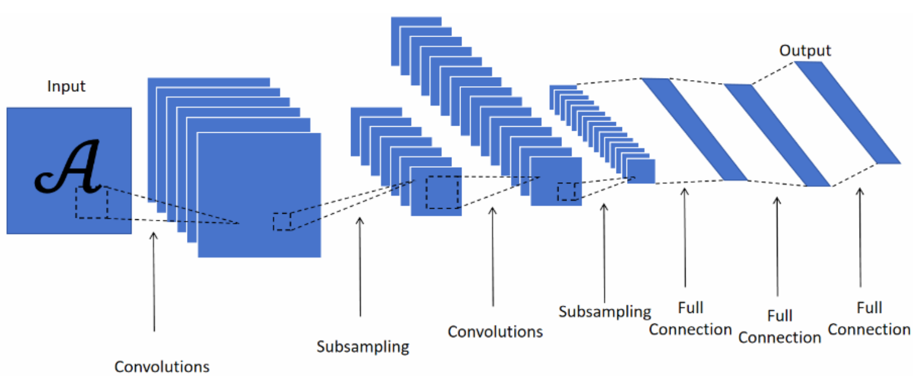
The tobacco wire feed rail speed, rail position, rail temperature, ambient temperature, ambient humidity, and PID parameters of the previous moment are used as the input of the lightweight one-dimensional convolutional neural network. Through continuous iterative optimisation of the model, the optimised PID parameters are finally output. In terms of computational efficiency, lightweight CNNs hold a distinct advantage. While CNN-LSTM hybrid models can capture long-term temporal dependencies, their high computational complexity may result in inference times that fail to meet real-time control requirements on edge devices.

2 Relevant technologies

2.1 Convolutional neural network

CNN is a mature model, has the structural advantages of parameter sharing and local connection, and can efficiently process grid-like data (such as images and speech) without relying on additional feature engineering. Compared with traditional fully connected networks, CNN has fewer parameters and stronger expressiveness when processing high-dimensional data, and can be used for both supervised learning and unsupervised learning. Its feature extraction is shift-invariant, so it is also called 'shift-invariant neural networks'. The core idea of CNN is to extract features and hierarchical representation of input data through a multi-layer structure, so as to achieve tasks such as classification, regression, and segmentation. The basic principles of CNN include its network structure design, forward propagation mechanism, parameter update process, and model training and reasoning methods.

Figure 1 Convolutional neural network architecture (see online version for colours)



The input layer of CNN is used to receive raw data, and its form depends on the type of task:

1 one-dimensional convolutional network (1D CNN) is often used to process time series or audio signals, and the input is a one-dimensional or two-dimensional array

2 two-dimensional convolutional network (2D CNN) is widely used for image processing, and the input is a two-dimensional pixel matrix (greyscale image) or a three-dimensional array (RGB image)

3 three-dimensional convolutional network (3D CNN) can process video frame sequences or volumetric medical images, and the input is a four-dimensional array.

Similar to other neural networks, CNN usually normalises the input data before training. For example, normalise pixel values from 0–255 to the interval [0, 1], or perform zero mean unit variance normalisation on the channel dimension. Such preprocessing helps improve the training efficiency and convergence speed of the model.

The convolutional kernel performs weighted sum operations with a fixed-size window by sliding on the input feature map, as follows:

$$y(i, j) = \sum_{m=1}^{k} \sum_{n=1}^{k} \omega_{m,n} \cdot x(i+m, j+n) + b \tag{1}$$

where $\omega_{m,n}$ is the weight of the convolution kernel, b is the bias term, x is the input feature map, and y is the output feature map.

The pooling layer is used to reduce the number of parameters, and improve feature invariance. The common pooling method is max pooling. Normalisation layers and regularisation layers can help speed up training and reduce overfitting. Common normalisation techniques include batch normalisation and layer normalisation, while dropout is a regularisation method that randomly suppresses some neurons. The output layer of the convolutional neural network is located at the very end of the network. Its function is to convert the deep feature map extracted by the model into task-related prediction results, such as category labels, numerical estimates, or pixel-level classification. The structure and behaviour of the output layer should be reasonably designed according to the specific task type to ensure that the dimension, numerical range, and semantics of the output match the target. In regression tasks, the output layer directly outputs one or more continuous values, generally without using an activation function or using linear activation. This structure is suitable for predicting actual quantitative values, such as temperature, distance, or ratings. The activation function of the output layer should be adapted to the selected loss function. For example, cross entropy loss is often combined in classification tasks, while mean square error or mean absolute error is usually used in regression tasks. The mathematical definitions and performance characteristics of different activation functions have been explained in detail in the previous article.

Training CNN mainly includes the following steps:

1 send the input sample to the model to perform forward propagation to obtain the prediction result

2 calculate the loss value based on the prediction result and the true label

3 perform backpropagation to calculate the gradient

4 use the optimiser to update the model parameters.

Repeat the above process, traverse the entire training set, and perform multiple rounds of training until the loss function converges or reaches the predetermined stopping condition.

In image classification tasks, the typical CNN workflow involves inputting an image into the network, performing multiple layers of convolution operations to extract features. These features are then combined layer by layer into higher-level representations. Finally, a FC layer is used to classify the image. This layered structure enables CNNs to capture multi-level features within an image, enabling accurate classification of complex images. The application of CNNs in image classification has benefited from the continuous optimisation of classic models. For example, AlexNet, introduced in 2012, significantly improved ImageNet classification accuracy through its deep networks and ReLU activations for accelerated training. CNNs play a vital role in object detection and recognition tasks and are one of the key technologies driving the practical application of computer vision. Unlike image classification, object detection requires not only that the model identify the objects in an image but also accurately locate each object within the image, namely, regressing its bounding box. This task places higher demands on the model, prompting researchers to develop a series of network designs specifically for detection based on the classic CNN architecture. Image segmentation, a further development of object detection, is another important application direction of convolutional neural networks in visual understanding. Image segmentation aims to classify each pixel in an image as a semantic category or instance object, endowing the model with a more refined understanding of the image content. Their multi-layered, stacked convolutional architecture effectively captures syntactic and semantic features in language sequences and offers inherent advantages in parallel computing, improving model training efficiency. With the introduction of deep convolutional architectures and residual connections, CNNs have become an increasingly popular alternative for sequence modelling. In summary, CNNs excel at extracting data features. In this paper, they are used to extract features from tobacco feed rail data and, through training, output optimised PID data.

2.2 PID controller

PID controller is a commonly used regulator in industrial control systems.

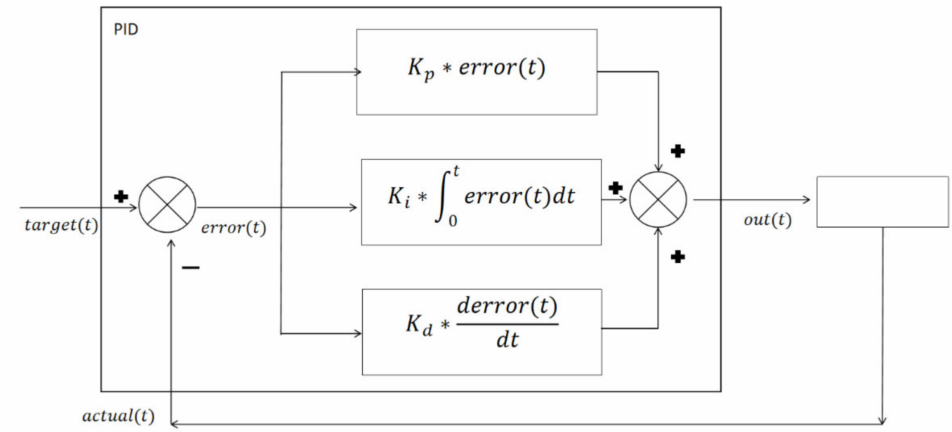
1 K_p (proportional gain) is to speed up the response speed of the system and improve the regulation accuracy of the system. It is directly proportional to the current error, thereby quickly reducing the error.

2 K_i (integral gain) is to eliminate the steady-state error of the system. Through the integral action, it ensures that the long-standing error can get enough response and finally achieve an error-free state.

3 K_d (differential gain) is to improve the dynamic performance of the system, especially the response to rapid changes. K_d adjusts the control action by predicting the trend of error changes, reducing or avoiding overshoot, and making the system more stable.

In PID control, the reasonable combination of these three parameters can make the system achieve a balance between response speed, steady-state performance and dynamic performance. Different control occasions and different control objectives will put forward different requirements for K_p, K_i and K_d. Therefore, designers and engineers of the control system need to adjust parameters according to specific circumstances to achieve the best control effect.

Figure 2 PID controller



In Figure 2,

$$error(t) = target(t) - actual(t) \tag{2}$$

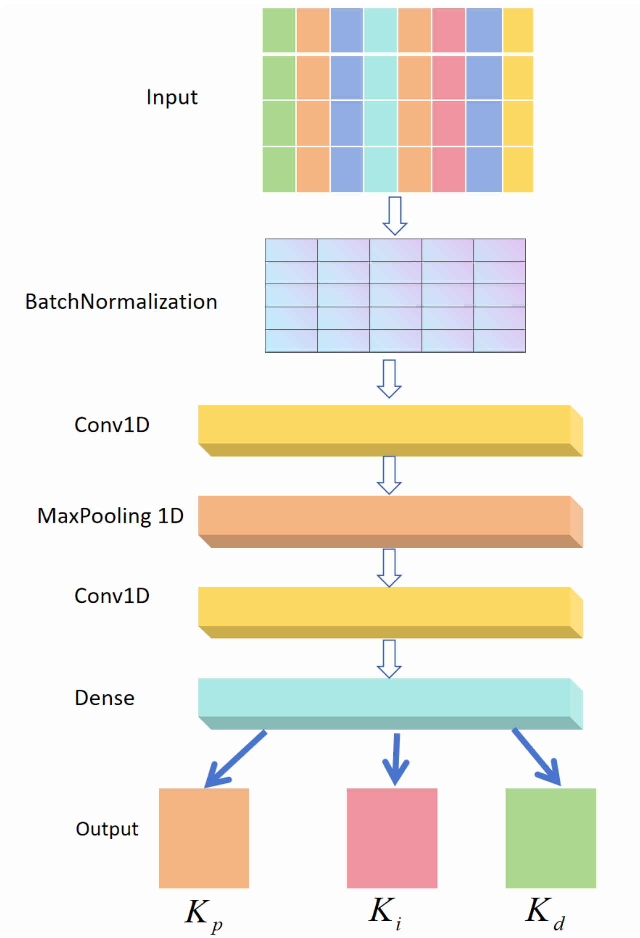
The PID output value is defined as:

$$out(t) = K_p * error(t) + K_i * \int_0^t error(t)dt + K_d * \frac{derror(t)}{dt} \tag{3}$$

3 Introduction to CNN-PID

This paper proposes CNN-PID, which uses tobacco feed rail speed, rail position, rail temperature, ambient temperature, ambient humidity, and PID parameters at the previous moment as model inputs. Among them, the rail speed directly affects the efficiency and uniformity of tobacco delivery, and speed fluctuations can cause tobacco accumulation or material breakage. As a key dynamic parameter, its rate of change can reflect the inertia characteristics of the system and is used to predict the direction of PID parameter adjustment. The rail position signal characterises the uniformity of tobacco distribution, and the offset reflects the deformation of the mechanical structure. This parameter is used as a spatial domain feature input to help CNN establish a position-speed coupling control model. The change in rail temperature affects the expansion coefficient of the rail material, and the temperature gradient distribution can reflect the wear of the bearing. The ambient temperature and humidity parameters affect the stability of tobacco moisture content. Sudden changes in temperature and humidity will cause changes in the friction coefficient of tobacco, which in turn affects the uniformity of delivery. After training, the model output is the optimised PID parameter value. The overall process of CNN-PID is shown in Figure 3.

Figure 3 CNN-PID model (see online version for colours)



As shown in Figure 3, the CNN-PID model primarily consists of one-dimensional convolution, batch normalisation, pooling, fully connected, and dropout layers. The sensor data from tobacco supply guides is inherently a multivariate time series. 1D-CNNs are specifically designed to process such data, with their convolutional kernels sliding along the time dimension to automatically extract temporal local patterns of parameters like wind speed, flow rate, and valve opening. Compared to RNN/LSTM, 1D-CNNs offer higher computational efficiency.

By progressively extracting features and performing classification or regression tasks, the model ultimately outputs a dimension of 3, outputting predicted values of K_p, K_i and K_d. Batch normalisation normalises the input data. During training, it normalises the data within each batch, aligning the mean to near 0 and the variance to near 1. This helps accelerate model convergence, alleviate the vanishing gradient problem, enhance the model's generalisation capabilities, and reduce training fluctuations caused by differences in data distribution across batches. The first Conv1D layer serves as the model's initial feature extraction layer, using a one-dimensional convolutional kernel to slide across the input sequence data to mine local features. The convolution operation, through parameter sharing and a sliding window mechanism, efficiently extracts data features such as tobacco feed rail speed, rail position, rail temperature, ambient temperature, ambient humidity, and the previous PID parameter data. MaxPooling1D performs one-dimensional max pooling to compress data along the sequence length dimension. By selecting the maximum value within each sliding window, the most significant features are retained, while reducing data dimensionality and computational complexity. It also achieves a certain degree of feature downsampling and local invariance. The second Conv1D layer performs another one-dimensional convolution, building on the pooled features to extract more complex and abstract features. Compared to the first Conv1D layer, it mines deeper data features based on the existing local features. The dropout layer is used for regularisation by randomly 'dropping out' some neurons during training, setting their outputs to 0. This effectively prevents model overfitting, forcing the model to learn more robust and generalisable features and avoiding over-reliance on specific neuron combinations. Finally, the dense (output layer) maps the features into three dimensions and outputs K_p, K_i and K_d. The lightweight convolutional neural network model employed in this paper performs feature extraction through two layers of one-dimensional convolutions: the first convolution layer utilises 16 3 × 1 convolution kernels; following dimension reduction via a pooling layer, the second convolution layer employs 32 2 × 1 convolution kernels.

In the tobacco supply system, the accurate, stable and reliable operation of the guide rail is crucial to the uniform transportation of tobacco, flow control and the final cigarette quality. The PID controller is the core component to maintain the precise position or speed of the guide rail. It is an advanced adaptive control strategy to use the guide rail speed, guide rail position, guide rail temperature, ambient temperature, ambient humidity and the PID parameters of the previous moment to predict and optimise the PID parameters of the next moment.

4 Experimental results and analyses

The data of CNN-PID comes from the tobacco silk production line, and a total of 10,000 data are collected in order to build a more robust CNN-PID model.

4.1 CNN-PID evaluation indicators

The evaluation indicators used in this paper are MSE, MAE and R^2-score (coefficient of determination).

1 Mean square error (MSE): a commonly used indicator for regression tasks.

$$MSE = \frac{1}{m}\sum_{i=1}^{m}\left(y_i - \hat{y}_i\right)^2 \tag{4}$$

2 Mean absolute error (MAE): a more robust regression indicator, which calculates the mean absolute error and is suitable for scenarios with many outliers.

$$MAE = \frac{1}{m}\sum_{i=1}^{m}\left|\left(y_i - \hat{y}_i\right)\right| \tag{5}$$

3 R^2-score: measures the model's ability to explain the variance of the target PID.

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{6}$$

SS_{res} is the residual sum of squares, which represents the sum of squares of the difference between the actual value and the predicted value.

$$SS_{res} = \sum_{i=1}^{m}\left(y_i - \hat{y}_i\right)^2 \tag{7}$$

$$SS_{tot} = \sum_{i=1}^{m}\left(y_i - \bar{y}\right)^2 \tag{8}$$

where y_i represents the true value, $\hat{y}_i$ represents the predicted value. When R^2 is greater than or equal to 0.9, it means that the model has excellent fitting ability and can highly explain data variation. Between 0.7 and 0.89, it means that the model has good fitting ability and can significantly explain data variation. When it is less than 0.5, it means that the model has poor fitting ability. In the CNN-PID model, the performance of MSE, MAE and R^2-score is comprehensively evaluated. When the three evaluation indicators K_p, K_i and K_d are optimal, it indicates that the values of predicted by the model are optimal.

4.2 CNN-PID experimental results

4.2.1 Comparison of MAE and Huber loss performance in CNN-PID

Finally, Huber loss is used, which combines MSE and MAE and can avoid the loss caused by MSE and MAE as much as possible. It is defined as follows:

$$L_\delta\left(y, \hat{y}\right) = \begin{cases} \frac{1}{2}\left(y - \hat{y}\right)^2, & \text{if } \left|y - \hat{y}\right| \leq \delta \\ \delta\left|y - \hat{y}\right| - \frac{1}{2}\delta^2, & \text{otherwise} \end{cases} \tag{9}$$

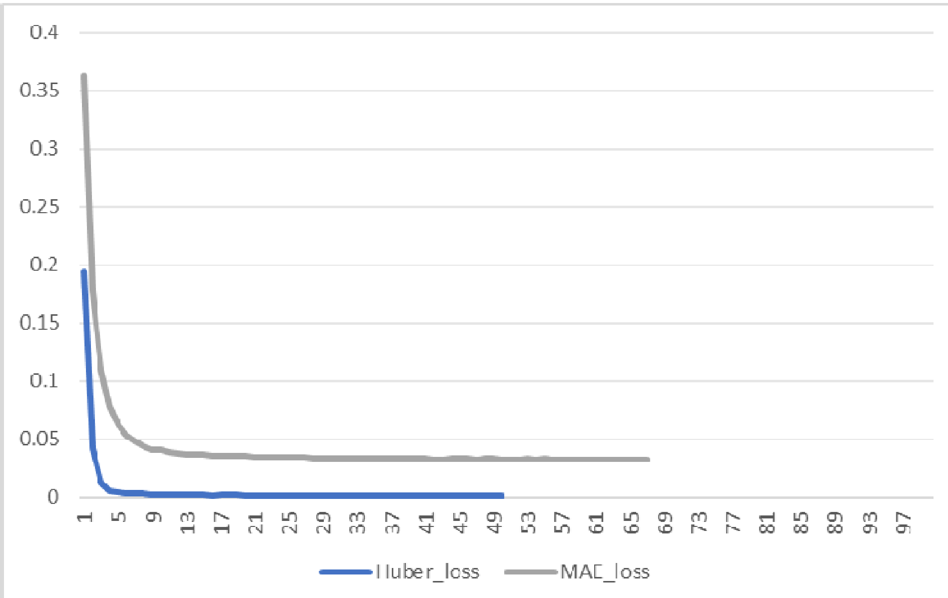
Among them, y represents the true value, $\hat{y}$ represents the predicted value, δ represents the threshold, and controls the limit of the error size. When the error is less than δ, MSE is used, and when the error is greater than δ, MAE is used. Huber loss features:

- Smoothness: when the error is less than δ, Huber loss is the same as MSE loss. The square error makes the gradient smoother, which helps to converge quickly.
- Robustness to outliers: when the error is greater than δ, MAE is used to reduce the impact of outliers on the model, making Huber loss less sensitive to large deviations.

- Adjustability: δ is a hyperparameter that can be adjusted according to needs to balance the effects of MSE and MAE.

In addition, an early stopping mechanism is set during the training process. If the verification loss does not improve for 10 consecutive rounds, the training is terminated. This can reduce the training time and save training resources as much as possible.

Figure 4 CNN-PID training process using different loss functions (see online version for colours)



4.2.2 Comparison results of different models

This paper compares the prediction performance of different models for K_p, K_i and K_d on R^2-score. This article compares CNN-PID with traditional PID control, BP neural network PID, and PSO optimised PID.

Traditional PID control is the most classic algorithm in industrial control. Its core is to achieve closed-loop control through a combination of three steps: proportional (P), integral (I), and derivative (D). The proportional step outputs the controlled variable proportional to the current error, rapidly reducing the error. However, using it alone can result in steady-state error. The integral step accumulates historical errors, gradually eliminating steady-state error, but this can increase system overshoot and response time. The derivative step makes advance adjustments based on the rate of error change to suppress overshoot and enhance stability, but is sensitive to noise. The advantages of traditional PID are its simple structure, ease of implementation, and strong robustness. It is suitable for linear systems with small time delays.

BP neural network PID is an intelligent control algorithm that combines a backpropagation neural network with PID. Its core is to use a neural network to adaptively optimise PID parameters. Its operating principle is that the system error is used as input to the neural network. Through self-learning (using the backpropagation algorithm to adjust the weights), the neural network outputs the optimal PID parameters in real time, and the PID controller then outputs the controlled variable. Leveraging the nonlinear mapping and self-learning capabilities of BP networks, this approach addresses the difficulty of dynamic tuning of traditional PID parameters and adapts to changing system characteristics. PSO-optimised PID is a control strategy that uses a particle swarm optimisation algorithm to search for optimal PID parameters. Its core approach is to replace manual tuning with swarm intelligence. PID parameters are considered 'particles', and a swarm of particles is initialised within the parameter space. By simulating the foraging behaviour of a flock of birds, the particles dynamically adjust their search direction based on their own historical optimal positions and the swarm's optimal position, ultimately finding the parameter combination that optimises system performance.

Figure 5 Performance of different models (see online version for colours)

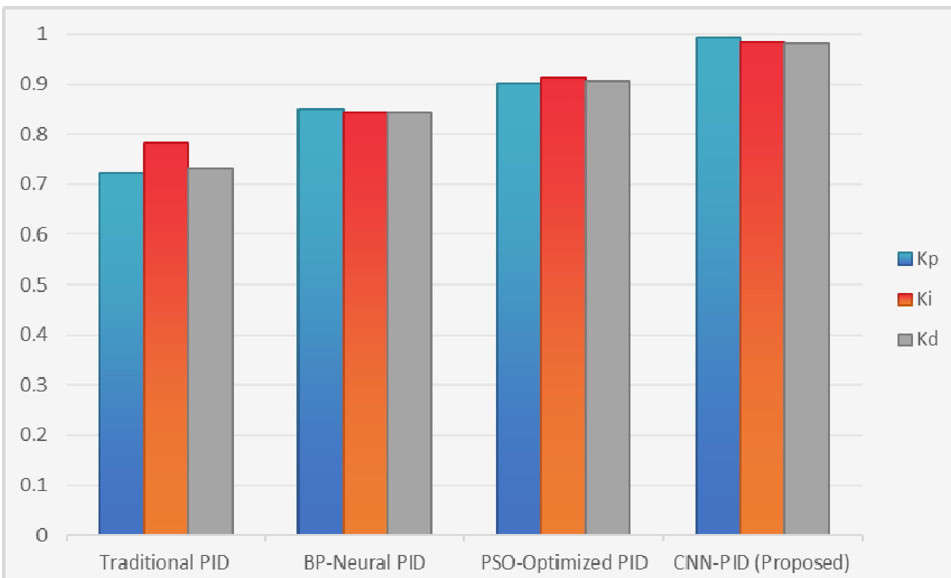
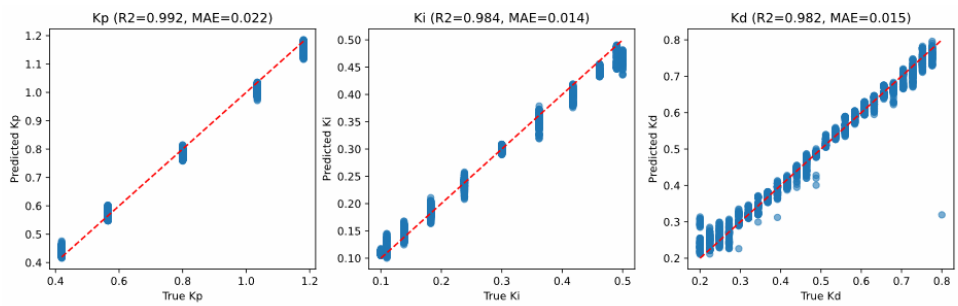


Figure 5 shows the comparison results of CNN-PID with traditional PID control, BP neural network PID, and PSO optimised PID. It can be seen that the CNN-PID proposed in this paper is better than other models in terms of R^2-score, indicating that the model has optimisation performance. In general, the traditional PID control is limited by linear characteristics and has poor adaptability to nonlinear working conditions. The BP neural network PID is prone to fall into local optimality and has a slow convergence speed. The global search ability of the PSO optimised PID is better than that of the BP network, but the real-time performance is insufficient. The CNN-PID control benefits from the dynamic extraction ability of the convolution layer for time series features, which is significantly better than the traditional method.

Compared to traditional PID, BP-PID, and PSO-PID, the core advantage of CNN-PID lies in its unique hierarchical feature extraction mechanism, a convolutional neural network. Leveraging the local perception and weight sharing properties of multi-layer convolutional kernels, CNNs automatically extract multi-scale spatiotemporal features from the system's dynamic response. Convolutional kernels extract dynamic characteristics and global trends in the data, while BP neural networks can only flatten features through fully connected layers. Compared to PSO-optimised PID controllers, CNN-PID achieves end-to-end joint optimisation. PSO-PID only optimises fixed-structure PID parameters and cannot adjust the control architecture itself. CNN-PID simultaneously optimises both the feature extraction network and the PID parameter generation network through backpropagation, forming a closed-loop 'feature-control' optimisation. CNN's translational invariance and pooling operations give it unique robustness. The maximum pooling layer effectively suppresses high-frequency noise, and the local receptive field of the convolution kernels makes the model insensitive to local parameter variations.

Figure 6 (a) K_p, (b) K_i and (c) K_d regression results of CNN-PID (see online version for colours)



(a) (b) (c)

As can be seen from Figure 6, the X-axis represents the true value of the model, and the Y-axis represents the predicted value of the model. When Y = X, it means that the predicted value is equal to the true value, and the model has the strongest regression ability. It can be seen from the figure that the model's prediction results for K_p, K_i and K_d are on both sides of the Y = X straight line, indicating that the model has good prediction performance. Early stopping and learning rate decay strategies were employed during model training to prevent overfitting. Simultaneously, data augmentation techniques (adding noise and time shifts) enhanced the model's generalisation capabilities. Consequently, the trained model ensures consistency between predicted outcomes and actual results.

Table 1 CNN-PID evaluation results

Metrics	K_p	K_i	K_d
R^2-score	0.992	0.984	0.982
MAE	0.022	0.014	0.015
MSE	0.007	0.003	0.005
RMSE	0.026	0.017	0.023

Finally, the CNN-PID predictions of K_p, K_i and K_d reached 0.992, 0.984, and 0.982, respectively, which shows that the CNN-PID model proposed in this paper has good regression characteristics and performs well in predicting PID parameters.

5 Conclusions

This paper proposes a convolutional neural network-based PID control optimisation for tobacco feed rails (CNN-PID). The model uses the tobacco feed rail speed, position, temperature, ambient temperature, humidity, and the previous PID parameters as input. A lightweight convolutional neural network extracts features and outputs predicted values for proportional gain, integral gain and derivative gain.

To minimise model complexity and improve training efficiency, the lightweight convolutional neural network consists of two layers: the first with 128 neurons and the second with 256 neurons. Each layer uses the rectified linear unit (ReLU) activation function, and is finally connected to a fully connected layer to output predicted values for proportional gain, integral gain and derivative gain. To find a better loss function, MAE was compared with Huber loss. To achieve faster model convergence and shorten training time, an early stopping mechanism was incorporated into the training process. Loss curves were plotted for comparison. Huber loss converges faster and produces lower post-training loss values. Comparing the model with traditional PID control, BP neural network PID, and PSO-optimised PID, the proposed CNN-PID model achieved the best performance. This paper also visualised the predicted and actual results through regression. The regression results showed that both the predicted and actual results fluctuated along the Y = X regression line, demonstrating the model's strong regression capabilities.

In summary, CNN-PID can accurately predict and regress PID parameters, exhibiting good robustness and suitable for training and application in actual tobacco supply systems. During actual deployment, high real-time performance is required. Therefore, this paper optimises the network architecture by avoiding complex recurrent connections and replacing two-dimensional convolutional neural networks with one-dimensional convolutional neural networks to reduce computational load. To address noise issues in data quality, a sliding average filter is added to suppress high-frequency noise.

Declarations

All authors declare that they have no conflicts of interest.

References

Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M. and Farhan, L. (2021) 'Review of deep learning: concepts, CNN architectures, challenges, applications, future directions', *Journal of Big Data*, Vol. 8, No. 1, p.53.

Bi, S., Zhang, B., Mu, L., Ding, X. and Wang, J. (2020) 'Optimization of tobacco drying process control based on reinforcement learning', *Drying Technology*, Vol. 38, No. 10, pp.1291–1299.

Cao, P., Zhu, Z., Wang, Z., Zhu, Y. and Niu, Q. (2022) 'Applications of graph convolutional networks in computer vision', *Neural computing and Applications*, Vol. 34, No. 16, pp.13387–13405.

Chua, L.O. (1997) 'CNN: a vision of complexity', *International Journal of Bifurcation and Chaos*, Vol. 7, No. 10, pp.2219–2425.

Farkh, R., Quasim, M.T., Al Jaloud, K., Alhuwaimel, S. and Siddiqui, S.T. (2021) 'Computer vision-control-based CNN-PID for mobile robot', *Computers, Materials & Continua*, Vol. 68, No. 1, pp.32–46.

Golovko, V. (2017) 'Deep learning: an overview and main paradigms', *Optical Memory and Neural Networks*, Vol. 26, No. 1, pp.1–17.

Jingyuan, L., Xin, L., Jie, L., Yixin, C. and Xian, H. (2023) 'Control method for moisture content in dried tobacco based on fused attention time convolution network', *Tobacco Science & Technology*, Vol. 56, No. 8, p.1206.

Khanam, R., Hussain, M., Hill, R. and Allen, P. (2024) 'A comprehensive review of convolutional neural networks for defect detection in industrial applications', *IEEE Access,* Vol. 18, No. 12, pp.36–38.

Kim, H-G., Myoung, J., Lee, S., Park, K-M. and Shin, D. (2023) 'Design of ai-powered hybrid control algorithm of robot vehicle for enhanced driving performance', *IEEE Embedded Systems Letters*, Vol. 16, No. 2, pp.94–97.

Kusiak, A. (2020) 'Convolutional and generative adversarial neural networks in manufacturing', *International Journal of Production Research*, Vol. 58, No. 5, pp.1594–1604.

Li, Q., Luo, Z., Chen, H. and Li, C. (2022) 'An overview of deeply optimized convolutional neural networks and research in surface defect classification of workpieces', *IEEE Access*, Vol. 10, pp.26443–26462.

Lu, W., Li, J., Wang, J. and Qin, L. (2021) 'A CNN-BiLSTM-AM method for stock price prediction', *Neural Computing and Applications*, Vol. 33, No. 10, pp.4741–4753.

Perišić, N. and Jovanović, R. (2023) 'Konvolucione neuronske mreže u sistemima automatskog upravljanja-pregled stanja u oblasti istraživanja', *Tehnika*, Vol. 78, No. 4, pp.433–441.

Schmidhuber, J. (2015) 'Deep learning in neural networks: an overview', *Neural Networks*, Vol. 61, pp.85–117.

Tulbure, A-A., Tulbure, A-A. and Dulf, E-H. (2022) 'A review on modern defect detection models using DCNNs-deep convolutional neural networks', *Journal of Advanced Research*, Vol. 35, pp.33–48.

Vashishtha, G., Chauhan, S., Sehri, M., Hebda-Sobkowicz, J., Zimroz, R., Dumond, P. and Kumar, R. (2024) 'Advancing machine fault diagnosis: a detailed examination of convolutional neural networks', *Measurement Science and Technology*, Vol. 36, No. 2, p.022001.

Wang, C-H., Huang, K-Y., Yao, Y., Chen, J-C., Shuai, H-H. and Cheng, W-H. (2022) 'Lightweight deep learning: an overview', *IEEE Consumer Electronics Magazine*, Vol. 13, No. 4, pp.51–64.

Wang, W., Liang, J., Pan, C. and Chen, L. (2023) 'Hierarchical CNNPID based active steering control method for intelligent vehicle facing emergency lane-changing', *Chinese Journal of Mechanical Engineering*, Vol. 36, No. 1, p.100.

Xia, Z. (2019) 'An overview of deep learning', *Deep Learning in Object Detection and Recognition*, pp.1–18, Wiley, Los Angeles, USA.

Xie, J., Yang, R., Gooi, H.B. and Nguyen, H.D. (2023) 'PID-based CNN-LSTM for accuracy-boosted virtual sensor in battery thermal management system', *Applied Energy*, Vol. 331, p.120424.

Xie, Z., Ramezani, M. and Levinson, D. (2024) 'Reduced-scale mobile robots for autonomous driving research', *IEEE Transactions on Intelligent Transportation Systems*, Vol. 2, No. 2, pp.32–36.

Yue, C., Wang, Y., Tang, X. and Chen, Q. (2022) 'DRGCNN: dynamic region graph convolutional neural network for point clouds', *Expert Systems with Applications*, Vol. 205, p.117663.

Zhang, S., Tong, H., Xu, J. and Maciejewski, R. (2019) 'Graph convolutional networks: a comprehensive review', *Computational Social Networks*, Vol. 6, No. 1, pp.1–23.