Journal of Computers Vol. 36 No. 5, October 2025, pp. 77-88
https://doi.org/10.63367/19911599202510360500 5

A Screw Fastening State Detection Method Based on Multi-Channel Information Fusion

Yanjun Zhu[1*], Jianfeng Luo[2], Wenliang Cao[1],
Yong Peng[1], Jianxin Li[1], Gang Ke[1], and Jia Wang[1]

[1] School of Electronic Information, Dongguan Polytechnic,
Dongguan, Guangdong, China
{26307320, 253925564, 289593848, 279149042, 403513673, 1765201668}@qq.com

[2] School of Artificial Intelligence, Dongguan Polytechnic,
Dongguan, Guangdong, China
251048547@qq.com

Received 21 July 2024; Revised 6 August 2024; Accepted 6 August 2024

Abstract. To accurately identify whether a screw fastening operation has encountered a fault and determine the specific type of fault, a screw fastening state detection method based on multi-channel information fusion is proposed. Firstly, the time-series data of torque, angle, and rotational speed during the screw fastening process are divided into different channels. The GRU (Gated Recurrent Unit) is then used to extract the temporal features of each channel's information. Subsequently, a self-attention mechanism is used to further distinguish the important relationships between the information. Finally, screw fastening state detection is achieved through information fusion and a classifier. Experimental results show that the proposed method can effectively identify fastening faults. Compared with traditional methods that combine manually designed features with classifiers, the method designed in the paper achieves better detection performance.

Keywords: screw fastening fault, GRU, multi-source signal fusion, self-attention mechanism

1 Introduction

With the advancement of industrial automation, automatic screw fastening machines are widely used in the assembly lines of precision electronic products to reduce labor costs and improve production efficiency. These devices have the capability to identify screw fastening faults, playing a crucial role in ensuring product quality. Nowadays, the use of artificial intelligence technology to solve practical problems has become a hot topic. There are two approaches to addressing the issue of screw fastening fault detection. One approach is to use industrial cameras to take photos of the completed screw fastening and then judge whether the screws have been tightened based on the images. Although the method can effectively discern the fastening result, it is sensitive to external lighting conditions, requires a good imaging environment, and demands high-quality cameras, leading to higher costs. The other approach involves collecting data through sensors inside the electric screwdriver, analyzing the time-series data, and establishing models to determine the fastening result. This method eliminates reliance on external environments, controls production costs, and holds significant practical value, having already achieved certain results. In the analysis of time-series data for screw fastening, most current methods use machine learning algorithms to analyze and model individual time-series data. However, useful sequential data such as angle, torque, and speed are generated during the screw fastening process. Therefore, the paper proposes a screw fastening state detection method based on multi-channel information fusion.

* Corresponding Author

2 Related Work

2.1 Screw Tightening

The sequence data generated during the screw tightening process has the characteristics of high dimensionality and unequal lengths. Traditional classification algorithms are generally not directly applicable to the problem of unequal-length sequences. To address this issue, researchers both domestically and internationally have proposed methods based on distance metrics for classifying sequences based on their similarity. Among these methods, Dynamic Time Warping (DTW) is one of the most widely used approaches [1-3]. Although DTW performs well in terms of classification accuracy, it suffers from high computational costs. To reduce the computational time required by DTW, K. Y. Zhou [4] proposed a classifier that integrates subsequence morphological information, while Batista et al. [5] introduced a complexity-invariant distance metric. However, distance metric methods often require prior expertise in selecting an appropriate distance measure, posing a challenge in analyzing time series data. Due to the typically long and large volume of data in screw tightening sequences, distance metric-based methods often require significant computation time. Therefore, feature-based time series classification methods offer a new approach to address this issue. These methods extract features that can describe the characteristics of time series data, rather than directly using the entire sequence. Subsequently, classification algorithms such as Support Vector Machine (SVM) [6], Logistic Regression (LR) [7], Neural Networks [8], and Random Forest (RF) [9] are employed for classification. Feature-based time series classification methods can reduce the complexity of time series and computational workload, making them suitable for large-scale datasets. However, the selection of key features often relies heavily on expert knowledge, and shallow models exhibit limited performance in the presence of noise interference and dynamic operating conditions. Therefore, further research is needed to advance the intelligent development of fault diagnosis.

2.2 GRU

The Gated Recurrent Unit (GRU) is an improved structure derived from the Long Short-Term Memory (LSTM) neural network [10, 11]. GRU simplifies the complex internal structure of LSTM [12] while maintaining accuracy and improving computational efficiency. It possesses strong non-linear fitting capabilities and can capture long-term dependencies of features. When handling long time-series data, GRU demonstrates promising results, the structure of GRU is shown in Fig. 1.

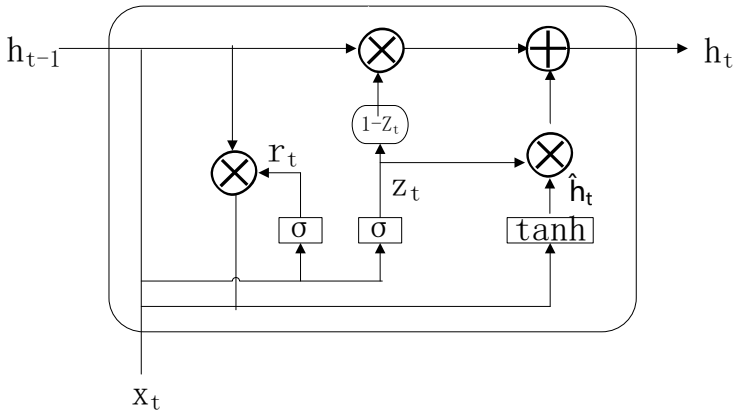

Fig. 1. Schematic diagram of the GRU structure

h_t represents the hidden state at the current time step, h_{t-1} represents the hidden state at the previous time step, $\hat{h}_t$ represents the candidate hidden state, r_t and z_t represent the reset gate and update gate respectively, σ denotes the sigmoid function, and tanh() is the hyperbolic tangent function,the calculation formulas are as follows.

$$r_t = \sigma\left(W_r \cdot \left[h_{t-1}, x_t\right]\right) \tag{1}$$

$$z_t = \sigma\left(W_z \cdot \left[h_{t-1}, x_t\right]\right) \tag{2}$$

$$\hat{h}_t = \tanh\left(W \cdot \left[r_t^{\circ h}{}_{t-1}, x_t\right]\right) \tag{3}$$

$$h_t = \left(1 - z_t\right)^{\circ} h_{t-1} + z_t^{\circ} \hat{h}_t \tag{4}$$

x_t is the input vector at time step t, [] denotes the shared chain, W_r,W_z and W are weight matrices, " ∘ " represents the Hadamard product of matrices. The hidden state h_t encodes information from all infected nodes at time t.

2.3 Self-Attention Mechanism

The self-attention mechanism is a key component of Transformer [13, 14]. The basic idea is to compute a weight for each element in a given input sequence to represent its importance, and then apply these weights to the representation of the element for weighted summation [15]. The computation of attention weights between elements is obtained by calculating the similarity between the representation of the element and that of all other elements. Typically, dot product or additive operations are used to compute the similarity, and the softmax function is applied to convert the similarity into a probability distribution. The resulting attention weights represent the importance of each element to the others. The principle is shown in Fig. 2.

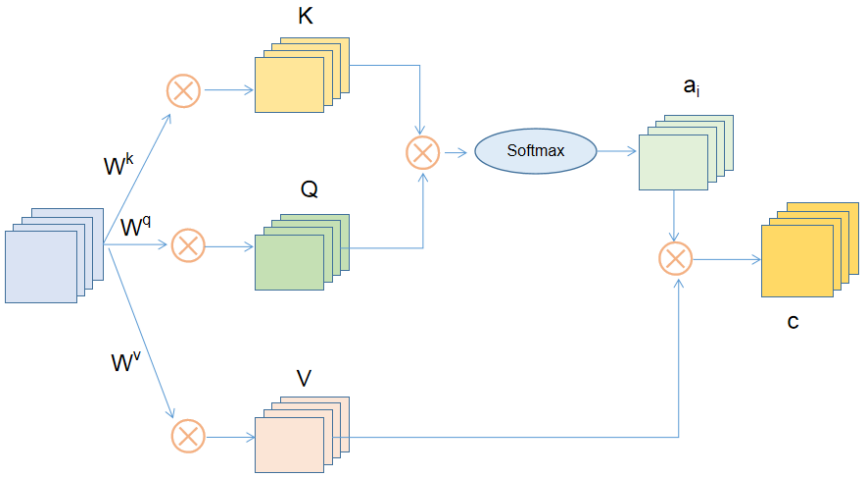

Fig. 2. Self-attention principle

The formula for self-attention mechanism shown in Equation (5):

$$\begin{cases} Q = W^q X \\ K = W^k X \\ V = W^v X \\ Attention(Q, K, V) = softmax\left(\dfrac{QK^T}{\sqrt{d_k}}\right)V \end{cases} \tag{5}$$

In the Equation (5), X represents the input information to the attention mechanism; Q, K, and V represent the matrices obtained by linearly mapping; W_q, W_k, and W_v represent the transformation matrices of different linear mappings; d_k denotes the dimension of K.

3 Detection Architecture

3.1 Model Design

Regarding the time series data of the screw fastening, the paper proposes an online detection model for screw tightening based on multi-channel information fusion and deep learning. The model utilizes a combination of Gated Recurrent Unit (GRU) and multi-head self-attention mechanism to fuse temporal data information through multiple channels for detecting the screw tightening state. The model consists of four modules: data preprocessing module, multi-channel network feature extraction module, multi-head self-attention module, and detection classification module, the model structure is shown in Fig. 3.

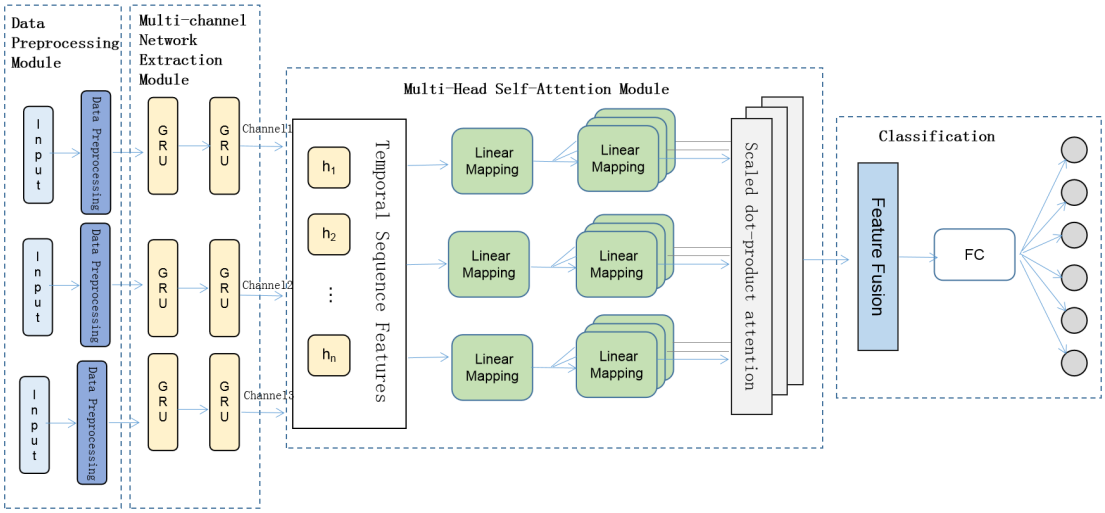

Fig. 3. Diagram of the screw tightening status detection model

(1) Data Preprocessing Module

The length of the temporal data is correlated with the lock fastening result. Lock fastening success and lock fastening misalignment are associated with shorter lock fastening time, with a minimum temporal data data length of 1532. On the other hand, lock fastening states like sliding teeth have longer lock fastening time, with a maximum temporal data data length of 2033. However, deep learning models require input data to have the same resolution. To address this issue, the paper standardizes all input data to a length of 2048. For data shorter than 2048, zeros are appended at the end to complete the input data. Additionally, since the units for torque, angle, and speed in the lock fastening process differ and their value ranges vary significantly, it is necessary to normalize the data for each time point. The normalization formula for the data is shown in Equation (6).

$$X = \frac{X' - \mu}{\sigma} \tag{6}$$

Where X represents the normalized data, X' represents the specific values corresponding to the dimensions of torque, angle, and speed before normalization, μ represents the mean, and σ represents the standard deviation.

(2) Multi-channel Network Feature Extraction Module

The multi-channel network feature extraction module utilizes preprocessed torque, angle, and speed data to extract temporal features through three separate channels, with each channel employing GRU for automatic feature extraction.

(3) Multi-head Self-attention Module

The multi-head self-attention module is implemented using the Multi-Head Self-Attention (MHSA) mechanism, which is a variant of the self-attention mechanism. This mechanism allows learning multiple attention distributions in different feature subspaces and combines them. The basic idea behind the multi-head self-attention mechanism is to use multiple attention heads to learn different attention distributions. Each attention head is generated by an independent weight matrix and can capture the dependencies between different parts of the input sequence. The schematic diagram of this mechanism is shown Fig. 4.

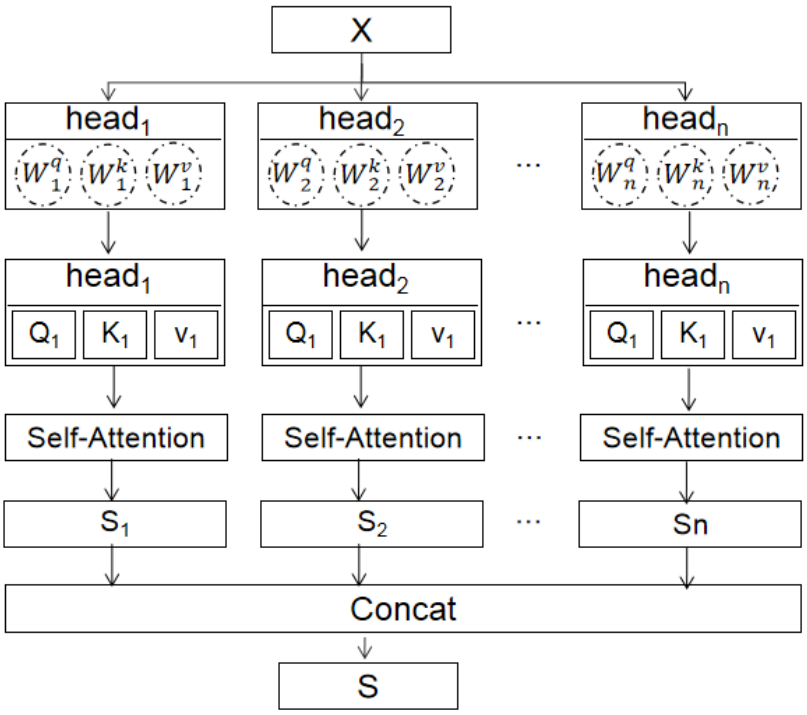

Fig. 4. The schematic diagram of the multi-head self-attention mechanism

First, the input information is multiplied by the weight coefficients W_i^q, W_i^k, W_i^v to obtain the vectors Q_i, K_i, V_i, which are then divided into multiple attention heads, denoted as $head_i$, where i represents the sequence number of different heads, as shown in Equation (7).

$$head_i = Attention\left(QW_i^q, kW_i^k, VW_i^v\right) \tag{7}$$

Then, each head performs different self-attention calculations, extracting features from different positions, as shown in Equation (8).

$$Attention\left(Q,K,V\right) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{8}$$

Finally, the results calculated by the n heads are concatenated and input into a fully connected layer for linear transformation to obtain the final result, as shown in Equation (9).

$$S = Concat(S_1, S_2, ...S_n) \quad (9)$$

The GRU's output temporal features are fed into the MHSA self-attention mechanism to analyze the interrelations between different time points, as well as the complex relationships of temporal features in different subspaces, aiming to enhance the model's capacity to analyze the dependencies and global relationships of temporal features.

(4) Detection Classification Module

The detection classification module consists of a feature fusion layer and a fully connected layer. The feature fusion layer is used to flatten the features from 3 channels, and then the fully connected layer integrates the global feature information and outputs the detection results.

3.2 Detection Process

First, construct a multi-channel one-dimensional screw fastening dataset, input the multi-channel dataset into the model for training, and finally use the trained model to detect the screw fastening status. The detection process is shown in the Fig. 5.

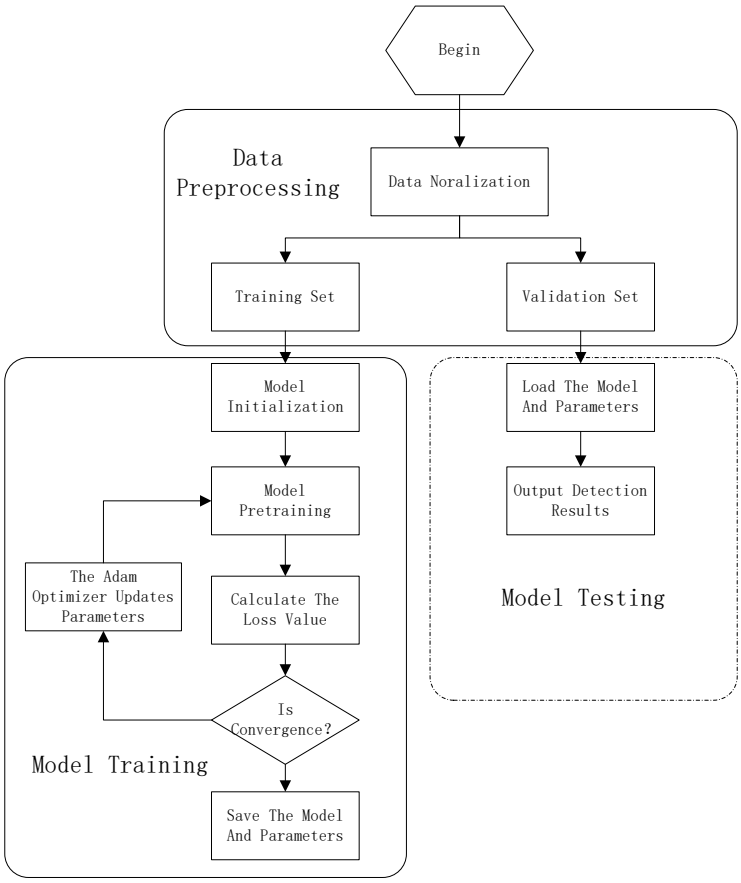

Fig. 5. Flowchart of the detection process

(1) Data noralization. Normalize the data, including padding the temporal data of the screw fastening and normalizing the torque, angle, and speed sensor data.

(2) Build the dataset. Evaluate the performance of screw fastening fault classification using 10-fold cross-validation. Divide the dataset into 10 mutually exclusive subsets of similar size, where in each experiment, the union of 9 subsets is used as the training set and the remaining 1 subset is used as the test

set. Then, there are 10 sets of training and test datasets.
(3) Initialize. Initialize the GRU model and the attention mechanism module.
(4) Compute the loss function. Input the training set into the model and calculate the error between the model's output and the truth.
(5) Update parameters. Update the parameters of each layer in the network using backpropagation algorithm.
(6) Check for convergence. After each training iteration, determine if the model has converged. If the model has converged, then the model training is complete. If the model has not converged, repeat steps (4) to (6).
(7) Save the model. Save the trained model parameters, then test the model using the test set to check the model's detection rate.

4 Experimental Dataset and Evaluation Criteria

4.1 Experiment Dataset

The screw fastening status detection system can be divided into two parts: the fastening module and the detection module. The fastening module mainly uses a torque sensor to collect the angle, torque, and speed sequence data generated during the fastening process and stores it. After each fastening operation is completed, experts judge the fastening result based on experience and write it into the storage system as the true label of the sequence. The fastening results are divided into two categories: successful and unsuccessful. Unsuccessful cases include misalignment, free spinning, stripped threads, not fully inserted, and not tightened. In the case of a successful fastening, no further processing is required. If misalignment occurs, the skewed screw needs to be manually removed and the fastening process repeated. If free spinning occurs, it means that the electric screwdriver has not properly attached the screw, and production line personnel need to check if the screw feeder is malfunctioning. If stripped threads occur, the electric screwdriver needs to reverse to remove the stripped screw. If the screw is not fully inserted, an attempt should be made to insert the screw again. If it is not tightened, the screw needs to be removed, and the hole inspected for foreign objects.

The data in the paper was collected by customized automatic screw fastening machines used in the manufacturing of precision electronic products. The machine is capable of automatically picking up screws, positioning them in holes, and completing the fastening process. The screwdriver's bit is driven to rotate by a motor, and torque, angle, and speed data are collected by sensors in the motor every 1 millisecond and recorded in the host computer. After each screw is locked, the fastening results are manually inspected. The dataset used in the paper includes the following quantities for each category of screw fastening results: successful - 2135, not tightened - 578, not inserted - 269, cross-threaded - 287, free spinning - 52, skewed - 357, totaling 3678.

4.2 Parameter Selection

The parameters of deep learning neural networks are numerous, and each parameter has a different impact on the final result. Some parameters may affect the accuracy, while others may influence the convergence speed of the network. The main model parameters are shown in Table 1.

Table 1. Key model parameters

Model parameters	Parameter details
Number of heads in MHSA self-attention mechanism	3
Loss function	categorical_crossentropy
Optimizers	adam
Optimizer learning rate	0.001
dropout	0.2
Epoch	20
Batch	64
Iteration	50

4.3 Evaluation Criteria

The screw fastening sequence contains multiple categories. When dealing with supervised learning problems, the use of a confusion matrix can accurately reflect the misclassification distribution of each category. For the six-class problem in the paper, its confusion matrix is shown below.

$$\begin{bmatrix} n_{11}^T & n_{12}^F & \cdots & n_{16}^F \\ n_{21}^F & n_{22}^T & \cdots & n_{26}^F \\ \vdots & \vdots & \ddots & \vdots \\ n_{61}^F & n_{62}^F & \cdots & n_{66}^T \end{bmatrix} \tag{10}$$

In the equation (10), the diagonal elements n_{ii}^T represent the amount of correctly classified data; n_{ij}^F represents the amount of data from category i being classified as category j.

The accuracy (A) is the ratio of the number of correctly classified data to the total amount of data, and the formula is shown in Equation (11).

$$A = \frac{\sum_i n_{ii}^T}{\sum_{ij} n_{ij}} \tag{11}$$

The calculation formulas for precision (P_i) and recall (R_i) of each category are as follows.

$$P_i = \frac{n_{ii}^T}{n_{ii}^T + \sum_j n_{ij}^F} \tag{12}$$

$$R_i = \frac{n_{ii}^T}{n_{ii}^T + \sum_j n_{ji}^F} \tag{13}$$

The calculation formulas for macro precision (M_p) and macro recall (M_R), which are the average values of precision and recall across all categories, are as follows.

$$M_p = \frac{1}{N}\sum_{i=1}^{N} P_i \tag{14}$$

$$M_R = \frac{1}{N}\sum_{i=1}^{N} R_i \tag{15}$$

M_{F1} is the harmonic mean of MP and MR, and the calculation formula is as follows.

$$M_{F1} = \frac{2M_P M_R}{M_P + M_R} \tag{16}$$

5 Experimental Study

The screw lock fastening state detection experiment was conducted on the following hardware environment: CPU - Intel Xeon i7-14700F, GPU - Nvidia Tesla P100, 16GB memory, running on a 64-bit Ubuntu PC. The experiment was implemented using the PyTorch deep learning framework based on the Python language.

5.1 Parameter Experiment

The main structure of the network includes GRU and Multi-Head Self-Attention (MHSA) mechanism. Among these two components, the number of layers in GRU has a significant impact on the results. When the number of GRU layers is small, the detection performance is poor but the training time is shorter. On the other hand, when the number of GRU layers is large, the detection performance is better but the training time is longer. The impact of different layer numbers on the results is shown in Table 2.

Table 2. Impact of GRU layers on accuracy and training time

Layers	1	2	3	4
A/%	93.39	96.56	96.87	97.12
Training time	67.12	78.05	99.87	115.73

From Table 2, it can be seen that selecting 2 layers for GRU achieves a high detection rate with the shortest training time. Therefore, this paper chooses a 2-layer GRU for feature extraction.

To determine the optimal number of GRU units, we initially selected 1 layer of GRU and conducted multiple simulation experiments by gradually increasing the number of GRU units in powers of 2 within the range of 1 to 256. The experimental results are shown in Table 3. As the number of GRU units increased, the experimental accuracy initially increased and then decreased. The peak accuracy was achieved with 64 GRU units. Therefore, we chose to use 128 and 64 GRU units for the first and second layers, respectively.

Table 3. Experimental results for different number of GRU units

GRU units	A/%	M_{F1}/%
1	80.12	71.02
2	81.23	71.98
4	79.34	69.13
8	82.36	73.56
16	82.56	74.25
32	78.01	69.45
64	85.19	82.35
128	84.23	79.73
256	82.63	77.92

5.2 Dissociation Experiments

Dissociation experiments were conducted on the multi-channel network feature extraction layer and the MHSA layer by individually removing each layer and comparing the results, as shown in Table 4.

Table 4. Results of dissociation experiments

Removing layer	A/%	M_{F1}/%
GPU	90.12	89.67
MHSA	92.39	91.98
None	96.56	94.24

The experimental results indicate that removing either layer adversely affects the model's detection performance.

5.3 Experiment Comparison and Results Analysis

The results of screw fastening state are categorized into two classes: success and failure, where failures include misalignment, slipping, stripped threads, incomplete engagement, and inadequate tightening. We conducted 10 rounds of training and testing. Fig. 6 illustrates the confusion matrix of the detection results of our model on the

screw tightening dataset. Calculated using the confusion matrix and Formula 11, the model achieves an overall accuracy of 96.56%, indicating its high precision.

	Success	Loose	Not fully engaged	Stripped threads	Slipping	Misaligned
Success	2129	6	0	0	0	0
Loose	9	539	2	3	0	25
Not fully engaged	0	2	261	6	0	0
Stripped threads	1	1	21	262	2	0
Slipping	0	0	0	4	48	0
Misaligned	7	22	5	2	0	321

Fig. 6. Confusion matrix of the model's detection results

According to Formulas 12 to 16, the precision, recall, and macro M_{F1} score of each category in the screw tightening detection results are shown in the Table 5.

Table 5. Classification results of the proposed model

Classification results	Precision/%	Recall/%	M_{F1}/%
Success	99.72	99.21	99.46
Loose	93.25	94.56	93.91
Not fully engaged	97.03	90.31	93.55
Stripped threads	91.29	94.58	92.91
Slipping	92.31	96.01	94.12
Misaligned	89.92	92.77	91.32
Macro-average	93.92	94.57	94.24

Currently, the detection results of screw fastening mainly rely on machine learning algorithms such as Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Gradient Boosting Decision Trees (GBDT). The paper compares the proposed model with these algorithms, and the specific results are shown in Table 6.

Table 6. Comparison of the proposed model with machine learning algorithm

Model	M_{F1}/%	A/%
kNN	69.13	75.89
SVM	86.35	81.87
RF	89.02	92.45
XGBoost	89.39	92.68
GBDT	89.97	93.02
The proposed model	94.24	96.56

Due to the limitations of manually designed features in traditional approaches, detailed information in screw fastening data is often overlooked, resulting in poorer classification performance. The proposed network structure in this paper extracts feature vectors from three different sensor inputs simultaneously and integrates them through a fusion layer. This method sensitively captures the detailed sequence information of screw fastening while also grasping the overall sequence characteristics, thus achieving excellent recognition performance.

6 Conclusions

In the process of screw tightening, time-series data such as torque, angle, and speed contain crucial information for determining the attachment status. Therefore, this paper proposes a screw tightening state detection model based on multi-channel information fusion. Firstly, the time-series information from different sensors is divided into separate channels, and GRU is used to extract temporal features from each channel. Then, a self-attention mechanism is introduced to further recognize the important relationships between the information. Finally, screw tightening state detection is achieved through information fusion and a classifier. Experimental results demonstrate that compared to the traditional approach of combining manually designed features with a classifier, the proposed model in this paper achieves better accuracy in detection and classification.

7 Acknowledgement

The work of Yanjun Zhu was supported by Key Fields Special Project of Guangdong Provincial Department of Education in 2023 (No.2023ZDZX1088), the special fund for electronic information engineering technology specialty group of the national double high program of Dongguang Polytechnic in 2023 (No. ZXD202308).

The work of Jianfeng Luo was supported by Songshan Lake Technology Commissioner Project (No. 20234434-01KCJ-G).

The work of Wenliang Cao was supported by the Dongguan Science and Technology of Social Development Program under Grant No. 20221800905842.

The work of Yong Peng was supported by the project for the innovative platform in industry and education integration for higher vocational college of guangdong province-the innovative platform in industry and education integration of testing technology for intelligent terminal product under Grant No. 2020CJPT014.

The work of Jianxin Li was supported by Dongguan Science and Technology Ombudsman Project in 2023 (NO. 20231800500282), Dongguan Science and Technology of Social Development Program (NO. 20231800903592), Special fund for electronic information engineering technology specialty group of national double high program of Dongguang Polytechnic (NO. ZXD202303).

The work of Gang Ke was supported by Songshan Lake Technology Commissioner Project (No. 20234437-01KCJ-G).

The work of Jia Wang was supported by Special project fund for electronic information engineering technology specialty group of the national double high program of Dongguang Polytechnic in 2022 (No. ZXD202205).

References

[1] H. He, H. Sun, Y. Chen, Analysis and assessment of ground motion characteristics and similarity using dynamic time warping distance, Journal of Seismology 27(6)(2023) 1013-1033.
https://doi.org/10.1007/s10950-023-10176-2

[2] X. Chen, X. Ye, X. Xiong, Y. Zhang, Y. Li, Improving the accuracy of wind speed spatial interpolation: A pre-processing algorithm for wind speed dynamic time warping interpolation, Energy 295(2024) 130876.
https://doi.org/10.1016/j.energy.2024.130876

[3] L. Cui, Q. Zhang, Y. Shi, L. Yang, Y. Wang, J. Wang, C. Bai, A method for satellite time series anomaly detection based on fast-DTW and improved-KNN, Chinese Journal of Aeronautics 36(2)(2023) 149-159.
https://doi.org/10.1016/j.cja.2022.05.001

[4] K.-Y. Zhou, S. Hu, An improved morphological weighted dynamic similarity measurement algorithm for time series data, International Journal of Intelligent Computing and Cybernetics 11(4)(2018) 486-495.
https://doi.org/10.1108/IJICC-12-2016-0059

[5] G.-A. Batista, X. Wang, E.-J. Keogh, A complexity-invariant distance measure for time series, in: Proc. 2011 SIAM International Conference on Data Mining, 2011.
https://doi.org/10.1137/1.9781611972818.60

[6] A. Dhakar, B. Singh, P. Gupta, Diagnosing faults in rolling bearings of an air compressor set up using local mean decomposition and support vector machine algorithm, Journal of Vibration Engineering & Technologies 12(4)(2024) 6635-6648.
https://doi.org/10.1007/s42417-024-01275-6

[7] C. Rakshe, S. Kunneth, S. Sundaram, J.-A. Ronickom, Diagnostic classification of ASD using fractal functional connec-

tivity of fMRI and logistic regression, Studies in Health Technology and Informatics 305(2023) 60-63.
https://doi.org/10.3233/shti230424
[8] A. Yilmaz, U. Yolcu, A robust training of dendritic neuron model neural network for time series prediction, Neural Computing and Applications 35(14)(2023) 10387-10406.
https://doi.org/10.1007/s00521-023-08240-6
[9] L. Breiman, Random forests, Machine Learning 45(1)(2001) 5-32.
https://doi.org/10.1023/A:1010933404324
[10] S. Li, J. Xie, X. Wang, F. Ge, Prediction of the creeping of AFC based on fuzzy reasoning and Bi-LSTM fusion iteration, Artificial Intelligence Review 57(6)(2024) 142.
https://doi.org/10.1007/s10462-024-10773-8
[11] Z. Liu, G. Chen, B. An, Y. Liu, Modeling of stepper motor fault diagnosis based on GRU time series analysis, in: Proc. 2022 Asian Simulation Conference, 2022.
https://doi.org/10.1007/978-981-19-9198-1_48
[12] R.-G. Geng, Y.-Q. Wu, Q.-Q. Xiao, S. Xu, Research on resource prediction of space-based information network based on improved GRU algorithm, Journal of Northeastern University (Natural Science) 44(3)(2023) 305-314.
https://doi.org/10.12068/j.issn.1005-3026.2023.03.001
[13] A.-H. Shin, S.-T. Kim, G.-M. Park, Time series anomaly detection using transformer-based GAN with two-step masking, IEEE Access 11(2023) 74035-74047.
https://doi.org/10.1109/ACCESS.2023.3289921
[14] J. Xia, Y. Feng, C. Lu, C. Fei, X. Xue, LSTM-based multi-layer self-attention method for remaining useful life estimation of mechanical systems, Engineering Failure Analysis 125(2021) 105385.
https://doi.org/10.1016/j.engfailanal.2021.105385
[15] L. Ding, Q. Li, Fault diagnosis of rotating machinery using novel self-attention mechanism TCN with soft thresholding method, Measurement Science and Technology 35(4)(2024) 047001.
https://doi.org/10.1088/1361-6501/ad1eb3