Article

The Temperature Prediction of Permanent Magnet Synchronous Machines Based on Proximal Policy Optimization

Yuefeng Cen [1,*], Chenguang Zhang [1], Gang Cen [1], Yulai Zhang [1] and Cheng Zhao [2]

[1] School of Information and Electronic Engineering, ZheJiang University of Seience and Technology, Hangzhou 310023, China; zcgicom@163.com (C.Z.); gcen@163.com (G.C.); zhangyulaai@zust.edu.cn (Y.Z.)

[2] School of Economics, Zhejiang University of Technology, Hangzhou 310014, China; zhaoc@zjut.edu.cn

* Correspondence: cyf@zust.edu.cn

Abstract: Accurate temperature prediction plays an important role in the thermal protection of permanent magnet synchronous motors. A temperature prediction method of permanent magnet synchronous machines (PMSMs) based on proximal policy optimization is proposed. In the proposed method, the actor-critic framework of reinforcement learning is introduced to model the effective temperature prediction mechanism, and the correlations between the input features are then analyzed to select the appropriate input features. Finally, the simplified proximal policy optimization algorithm is introduced to optimize the value of the prediction temperature of PMSMs. Experimental results reveal the high accuracy and reliability of the proposed method compared with an exponential weighted moving average method (EWMA), a recurrent neural network (RNN), and long short-term memory (LSTM).

1. Introduction

Temperature prediction of permanent magnet synchronous machines (PMSMs) has been a research focus in the field of motor protection. In recent years, researchers have made many attempts to predict the temperature of PMSMs [1], since temperature is an important factor for PMSMs to work. Most researchers have focused on the thermal model of the motor. For example, the temperature equivalent model based on hardware-in-loop (HIL) was proposed to effectively predict the motor temperature [2], but this method required high calculation complexity. An equivalent thermal transfer model with two heat nodes for a permanent magnet synchronous motor was also proposed [3]. The thermal effect of the current and stator frequency was considered. The predicted results verified the rationality of this transfer model. Mohamed et al. [4] constructed a Lumped Parameter Thermal Network (LPTN) to calculate important component temperatures inside PMSMs. The air temperature between permanent magnets was considered in this model. However, the computational complexity of the model is high. Wallscheid et al. [5] proposed a dynamic measurement method by introducing the magnetic flux observer into the time-dependent dispersion model of PMSMs. However, this approach is not universal because it is strongly correlated to machine speed. Wallscheid et al. [6] examined the prediction performance of flux observers in PMSMs, and the results illustrated that the worst case of the Euclidean norm is less than 10 K. Lan et al. [7] established a temperature thermal network with 38 nodes by analyzing the temperature fields of PMSMs, which accurately described the temperature values of each component inside the motor. However, the acquisition of overheat spots lacked optimization. Sciascera et al. [8]

built a variable heat model of an LPTN to improve the prediction accuracy of the traditional LPTN, which requires low computational complexity. In addition, this model provides an effective fine-tuning experience of model parameters. Liu et al. [9] investigated the signal injection method for estimating the temperature of the motor stator windings, but the temperature estimation results under motor overload were not given. Du et al. [10] established a finite element model of the electromagnetic fields of the motor using finite element analysis. The model obtained a temperature distribution of major components inside the motor under a rated working condition by calculating motor loss and a coefficient of thermal conductivity. In conclusion, the above models aimed to establish the empirical formulas of motor temperature. However, these processes of modeling design and the factors adopted depend on prior experience. In this work, temperature prediction is seen as a time series problem, and the temperature change of motor components can be fitted dynamically with additional degrees of freedom due to the capability of the dynamic tuning in PPO-RL.

The development of artificial intelligence technology has shown great potential in the field of temperature prediction. Xu et al. [11] proposed a novel deep-learning-based indoor temperature prediction method for public buildings, which verified the prediction accuracy in the direction of indoor temperature change and its disadvantage in the horizontal direction. Liu et al. [12] analyzed the time dependence of ocean temperatures at multiple depths and proposed a time-dependent ocean temperature prediction method, and the test results showed a better predictive performance than both support vector regression (SVR) and a multilayer perceptron regressor (MLPR). Wallschied et al. [13] verified the feasibility of LSTM on temperature prediction. However, the introduction of memory blocks in LSTM made the topological relationships complex, thus increasing the computing complexity.

In order to provide an accurate prediction method, we propose a method based on correlation analysis (CA) and proximal policy optimization (PPO) [14]. It selects the input features by correlation analysis and optimizes the model training process with a PPO algorithm. The remainder of the paper is organized as follows: The dataset and the correlation analysis process are described in Section 2. The rationale of our proposed method is presented in Section 3. The predictive model is validated and compared with other predictive networks in Section 4. Finally, a conclusion is given in Section 5.

2. Dataset and Correlation Analysis

In order to improve the prediction accuracy, an effective data processing method is proposed in this paper. The specific process is shown in Figure 1.
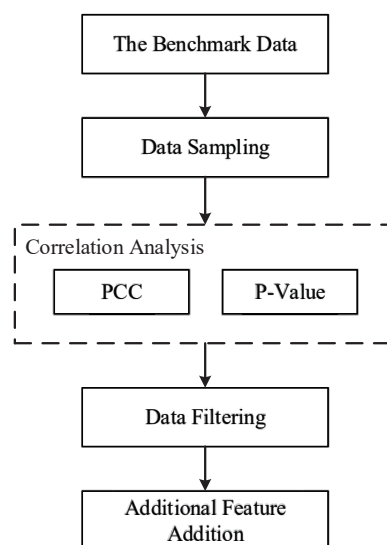

Figure 1. The process of data correlation analysis.

The benchmark data are firstly sampled, and the correlation analysis is then conducted on the sampled data using the Pearson correlation coefficient (PCC) and the *p*-value. After the correlation analysis, the data features that are significantly negatively correlated with the predicted target are discarded. Meanwhile, some additional features such as the voltage magnitude u_s, the current magnitude i_s, and the electric apparent power S_{el} are added to the processed sampled data to enrich the dataset and improve prediction accuracy.

2.1. Data Description

The benchmark data used in the experiment came from the Kaggle data science competition platform. The measurement and collection of the data were conducted by the University of Paderborn in Germany, and the benchmark data were normalized. The definitions of parameters and symbols of the column labels in the benchmark data are shown in Table 1. ϑ_{sy}, ϑ_{st} and ϑ_{sw} were chosen as the test objectives for the experiment. The data contain 990,000 pieces. The experiment consisted of 52 measurement sessions, and each measurement session can be distinguished by S_{id}. All measurement records were measured at a sampling frequency of 2 Hz on a test bench equipped with a three-phase permanent magnet synchronous motor.

Table 1. The benchmark data column label.

Parameters	Symbols
ambient temperature	ϑ_a
coolant temperature	ϑ_c
voltage d-component	u_d
voltage q-component	u_q
current d-component	i_d
current q-component	i_q
motor speed	n_{mech}
torque	T_m
Permanent Magnet temperature	ϑ_{pm}
stator yoke temperature	ϑ_{sy}
stator tooth temperature	ϑ_{st}
stator winding temperature	ϑ_{sw}
unique id	S_{id}

The benchmark data involve the main thermal process of a PMSM. In our work, 30,000 samples were taken from the benchmark data at random. The experiment selected 300 test samples as the test set, and the rest of the samples were used as the training set.

2.2. Correlation Analysis

Equipment failure often occurs in the process of continuous data acquisition, which will cause partial distortion in the benchmark data and become interference factors of the prediction. Therefore, Pearson correlation coefficient analysis [15] was adopted to observe the correlation between different features, and the *p*-value [16] was used to measure the related level.

The general expression of the Pearson correlation coefficient is as follows:

$$r_{xy} = \frac{\text{cov}(x,y)}{\sigma_x \sigma_y} = \frac{E\left[(x - \mu_x)(y - \mu_y)\right]}{\sigma_x \sigma_y} \tag{1}$$

where σ_x and σ_y are the standard deviation of variables x and y, respectively. Additionally, $\text{cov}(x,y)$ is the covariance of the two variables, and μ_x and μ_y are the average values of variables x and y, respectively. In general, if the covariance of x and y is larger than 0, the variables x and y are positively correlated. If the covariance of x and y is equal to 0, the variable x the variable y are independent. Otherwise, the variable x and the variable y are negatively correlated.

The correlations between data features are discussed, and the significance level *p*-value is also evaluated. It is generally acknowledged that there is a significant difference between the two groups of data characteristics when the *p*-value is less than 0.05, and the difference is of particular significance when the *p*-value is less than 0.01.

In order to evaluate the correlation between the monitoring target and the benchmark data, the Pearson correlation values of each feature through the thermal diagram of the sampling data are analyzed in Figure 2. The values of the correlation coefficients between ϑ_{sy} and T_m, the current d-component i_d, and the current q-component i_q of PMSMs are all negatively correlated. The joint distribution density diagrams between ϑ_{sy} and the above three features are shown in Figure 3.
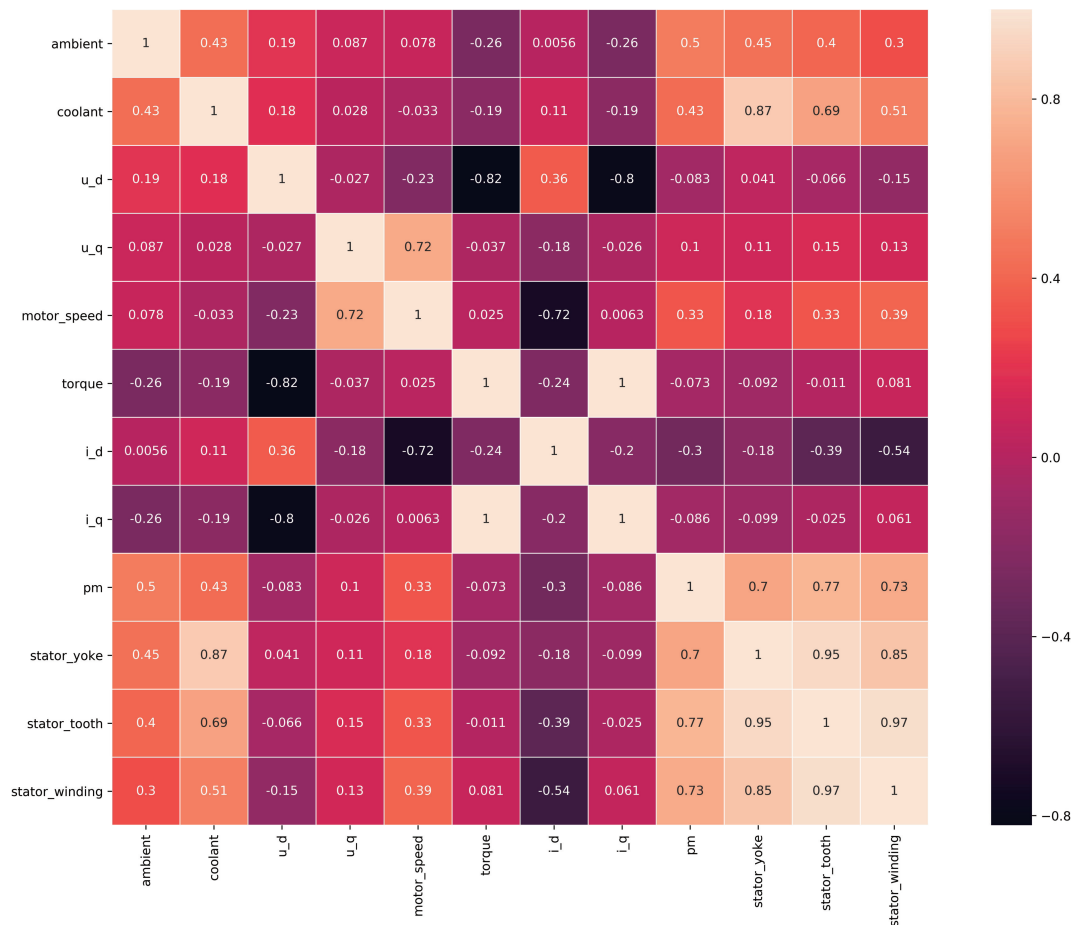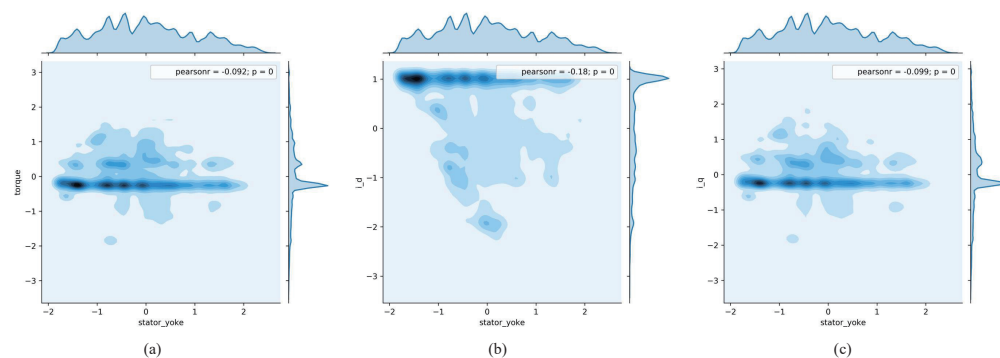

Figure 2. The feature thermal diagram of sampled data.

Figure 3. The kernel density estimation of ϑ_{sy}.

The correlation coefficients between ϑ_{st} and the voltage d-component u_d, the motor T_m, the current d-component i_d, and the current q-component i_q are shown in Figure 2, respectively. The values of the above correlation coefficients are all less than 0. Figure 4 shows the joint distribution of ϑ_{st} with the above features to further show the correlation degree of the features.
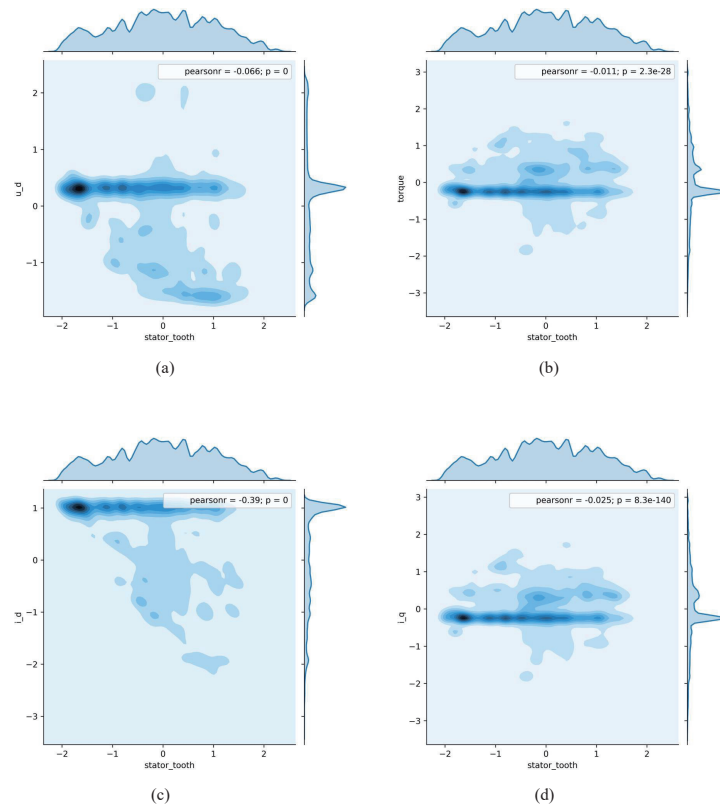

Figure 4. The kernel density estimation of ϑ_{st}.

In the same way, it can be seen in Figure 2 that the correlation coefficients of ϑ_{sw} with the voltage d-component ud and the current d-component i_d are negative, respectively, so there are negative correlations between the features. Meanwhile, the joint distribution density diagrams of the target feature ϑ_{sw} are shown in Figure 5.
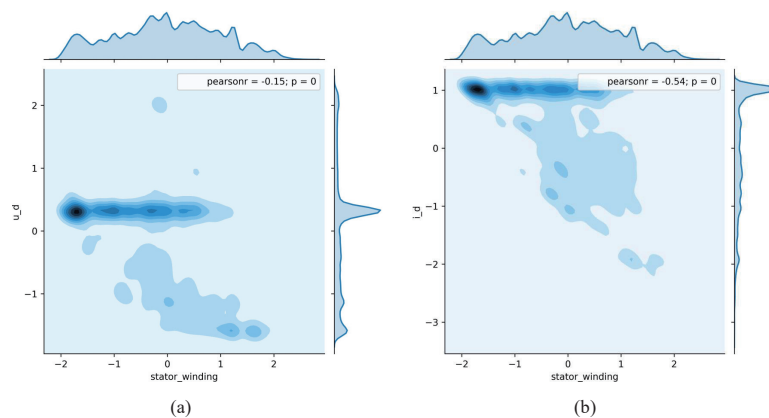

Figure 5. The kernel density estimation of ϑ_{sw}.

On the basis of the sampled data set, some additional feature quantities are considered in this paper. These features include the voltage magnitude u_s based on their dq-components, the current magnitude i_s based on their dq-components, and the electric apparent power S_{el}, respectively. The specific calculation methods are defined as follows:

$$u_s = \sqrt{u_d^2 + u_q^2} \tag{2}$$

$$i_s = \sqrt{i_d^2 + i_q^2} \tag{3}$$

$$S_{el} = u_s * i_s \tag{4}$$

where u_d and u_q are the components of voltage on d-component and q-component respectively, i_d and i_q are the components of the current on d-component and q-component, respectively, and $*$ represents dot product operation.

3. The Proposed Method

3.1. Reinforcement Learning

In order to accurately predict the temperature of the main components of the PMSMs, the Actor-Critic framework of reinforcement learning (RL) [17] is introduced into the predictive network. The general structure of the Actor-Critic learning framework is shown in Figure 6.
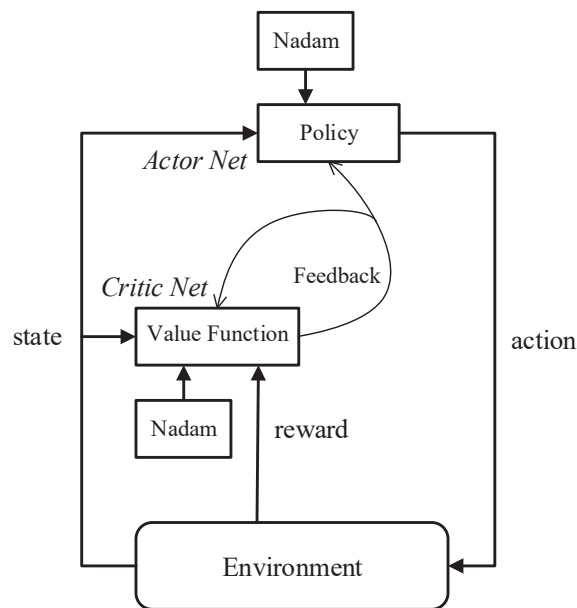

Figure 6. The Actor-Critic learning framework.

The Actor network and the Critic network are the main parts of this framework. The interactive state values come from the record datasets of PMSMs in the environment, and the dynamic selection process of state values are the basis of model training.

The target function of actor training can be dynamically adjusted by the feedback function. Therefore, the feedbacks of the Critic network to the Actor network are particularly important in the prediction process. In addition, the Nadam algorithm is used in the gradient optimization process.

3.2. Proximal Policy Optimization

The PPO algorithm is one of the policy gradient methods for RL proposed by OpenAI in 2017, and this algorithm is often applied in the control process of intelligent agents. The algorithm can easily achieve adjustments of hyper-parameters during the training of agents. In each iteration, it will attempt to minimize the objective function and recalculate the new update strategy. The objective function of the PPO algorithm can be defined by Formula (5):

$$L^{CLIP}(\theta) = \hat{E}[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t)] \tag{5}$$

where ε is a constant, and $\hat{A}_t$ is the feedback of the Critic network. Furthermore, $r_t(\theta)$ is the ratio of the new strategy and the old strategy, and its calculation method is represented by Formula (6):

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \tag{6}$$

where $\pi_\theta(a_t|s_t)$ is the updated new policy, $\pi_{\theta_{old}}(a_t|s_t)$ is the corresponding old policy, a_t and s_t are the action and state values at time t, respectively.

As shown in Formula (5), the objective function $L^{CLIP}(\theta)$ includes two main parts: The first part is a product of the strategy ratio $r_t(\theta)$ and the feedback value $\hat{A}_t$. The second part is a product about $r_t(\theta)$ and the feedback value $\hat{A}_t$ after clipping in the interval $[1-\varepsilon, 1+\varepsilon]$. Finally, the minimum value of the two parts can be obtained by Formula (5).

The definition of strategy ratio $r_t(\theta)$ is given in Formula (7), where out_t represents the output value at time t, and y_t represents the real output value. Additionally, the output out_t is given by the Actor network, and the loss function of the critic network is selected as the feedback $\hat{A}_t$. The strategy ratio $r_t(\theta)$ and $\hat{A}_t$ are defined as follows:

$$r_t(\theta) = \frac{out_t}{y_t} \tag{7}$$

$$\hat{A}_t = \frac{1}{N}\sum(out_t - y_t)^2 \tag{8}$$

where N denotes the number of all predicted values.

3.3. Model Construction and Prediction

The temperature prediction model of the PMSM is shown in Figure 7. The Actor network and the Critic network include an input layer and an output layer, respectively, and h_i $(i = 1, 2, ..., 5)$ is the hidden layer.

The definition methods of hidden layers in the model are as follows:

$$h_1 = \text{relu}(x_t * w_1 + b_1) \tag{9}$$

$$h_i = \text{relu}(h_{i-1} * w_i + b_i) \tag{10}$$

$$out_t = \text{relu}(h_5 * w_{out} + b_{out}) \tag{11}$$

where x_t is the input data matrix at time t, and $*$ is an element multiplication sign. w_i, b_i and h_i, respectively, represent the weight, the bias, and the output of each hidden layer for the network, $(i = 1, 2, ..., 5)$. The corresponding weights and bias of the network output layer are w_{out} and b_{out}, respectively, and the final predicted value of the network at time t is out_i. Further, θ and θ_{out}, respectively, represent the parameter vectors before and after the policy update.
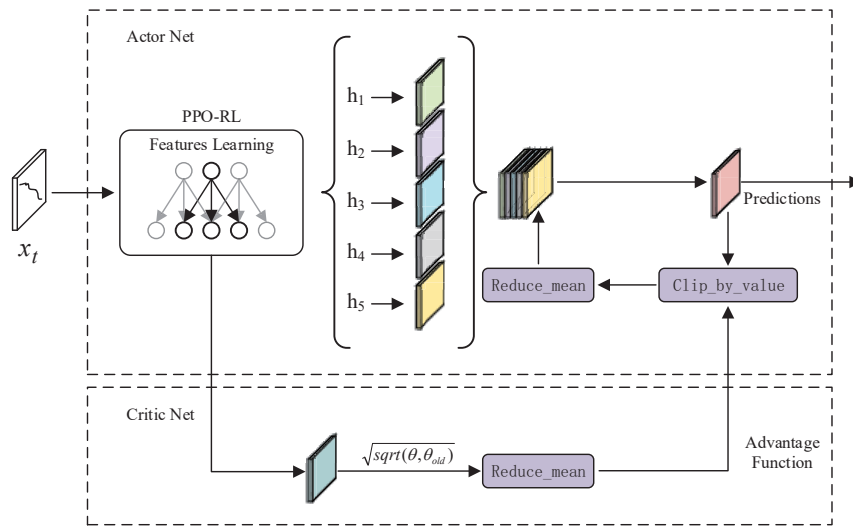
Figure 7. A prediction model of motor temperature based on proximal policy optimization-reinforcement learning (PPO-RL) .

After the completion of data processing and model construction, the loss objective function of the training model is determined by Formula (5). The Actor network and the Critic network share five hidden layers in this model, and the numbers of these network neurons are 512, 256, 128, 64, and 32, respectively. Moreover, the relu function is used as an activation function in each hidden layer.

The model chooses the input sequence with step size 5 as the input data. In the process of iteration training, the target $L^{CLIP}(\theta)$ of model training is calculated according to the $\hat{A}_t$ value, and $r_t(\theta)$ is updated at each step.

In order to accelerate the convergence of the objective function and make the gradient reach the global minimum more quickly, the Nadam algorithm is used to optimize the training process. The correction value $\hat{g}_t$ of gradient g_t is introduced into the Nadam algorithm and compared with Adam at time t, and the gradient $\hat{g}_t$ is defined by Formula (12). In addition, the updated gradient $\Delta\theta_t$ is calculated by Formula (13). Finally, the predicted output values can be obtained by the trained model.

$$\hat{g}_t = \frac{g_t}{1 - \prod_{i-1}^{t} u_i} \tag{12}$$

$$\Delta\theta_t = -\eta * \frac{\bar{m}_t}{\sqrt{\hat{n}_t + x}} \tag{13}$$

Here, u_i is the momentum factor of the first moment estimation at time i, η is the learning rate of the Nadam algorithm, $\hat{n}_t$ is the correction value of the second raw moment estimation of gradient at time t, and ξ is a positive number close to but not equal to zero.

4. Experiments and Results Analysis

4.1. Experimental Environment and Parameter Definition

The experimental environment in this experiment consisted of an Intel(R) Core(TM) i5-8250U 3.4 GHz quad-core processor with a 16 GB memory. The operating system was 64 bit Windows 10, the programming language version was Python3.7.5, and the deep learning framework version was Tensorflow1.13.1. The hyper-parameters considered during the experiment are in Table 2.

Table 2. Hyper-parameter sets of the optima found in experiments.

Hyper-Parameter	Models			
architecture	RNN	LSTM	PPO-RL	EWMA
hidden layers	2	3	5	-
units per layer	[40,20]	[100,50,1]	[512,256,128,64,32]	-
weight	normal	normal	normal	-
optimizer	adam	rmsprop	nadam	-
learn rate	0.01	0.01	$[2.10^{-5}, 1.10^{-5}]$	-
gaussian noise	1.10^{-3}	1.10^{-3}	1.10^{-3}	1.10^{-3}
epsilon(ε)	-	1.10^{-6}	0.02	-

In addition to the types of parameters in the table that are self-explanatory, some of the hyper-parameters not specifically mentioned should be interpreted as follows: When initializing the weights of the prediction network, the simplest method would be to assign random values from the interval $[-1, 1]$. In addition, more complex and efficient initializing schemes of weights can be considered, such as unit normal distribution or uniform distribution.

4.2. Model Evaluation

The goal of this paper is to predict the temperature of the PMSMs at the next moment. Therefore, the most effective evaluation methods for the above PMSM temperature prediction are the root mean square error (RMSE) [18] and the mean absolute error (MAE) [19]. As shown in Equations (14) and (15), the RMSE and MAE are calculated as follows:

$$RMSE = \sqrt{\frac{1}{N} \sum_{j=0}^{N-1} (R_j - P_j)^2} \tag{14}$$

$$MAE = \frac{1}{N} \sum_{i=0}^{N-1} |R_i - P_i| \tag{15}$$

where R_j represents the measured temperature of the target, P_j represents the predicted temperature of the target, and N denotes the number of test data.

In order to comprehensively evaluate the prediction performance of different methods, the Euclidean norm L_2 [20] and worst-case error L_∞ [21] are introduced to measure the approximation degree of the prediction target. The specific evaluation indexes are defined as follows:

$$L_2 = \sqrt{\sum_{j=0}^{N-1} (R_j - P_j)^2} \tag{16}$$

$$L_\infty = \max_{i} \sum_{j=0}^{N-1} |e_{ij}| \tag{17}$$

where R_j, P_j and N represents the same elements as in RMSE, and $|e_{ij}|$ indicates the sum of absolute values for all error in row i.

4.3. Experimental Results and Analysis

In order to evaluate the overall performance of our proposal and the comparative methods on the sampled dataset, the trend prediction results for ϑ_{sy}, ϑ_{st} and ϑ_{sw} are demonstrated, respectively. As shown in Figures 8–10, the prediction curves that we proposed fit the real curves in the prediction period best. Although the curves of LSTM and the RNN conformed to the real target curves at first, they largely deviate at the end. Moreover, the fitted curves given by the EWMA method have a large delay characteristic. The x-coordinates of the curve represent the prediction period of the test data, and the y-coordinates are the prediction targets in Figures 8–10.
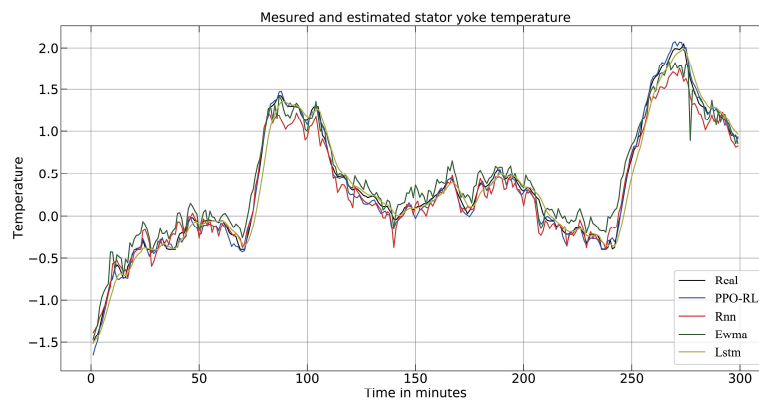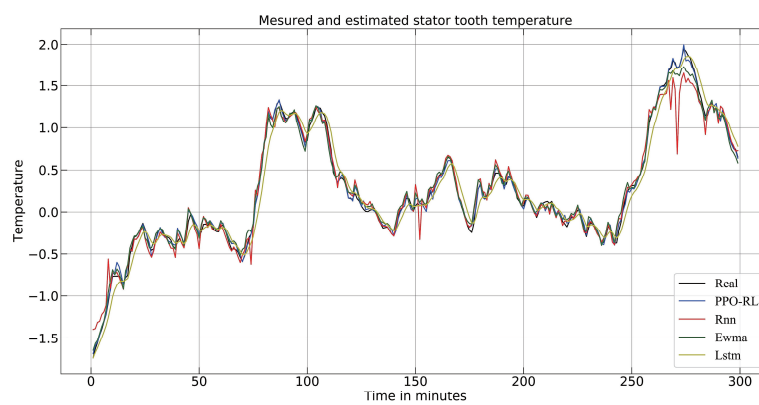
Figure 8. Fit curve of stator yoke temperature.
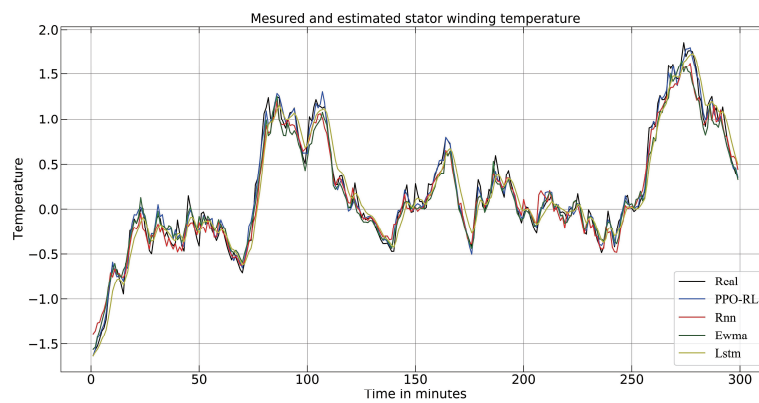

Figure 9. Fit curve of stator tooth temperature.

Figure 10. Fit curve of stator winding temperature.

The relating evaluation indicators for ϑ_{sy}, ϑ_{st} and ϑ_{sw} are provided on Table 3, Table 5 and Table 7 respectively, including RMSE, MAE, Euclidean norm L_2 and Infinite norm L_∞.

The quantitative evaluation indicators of the temperature prediction of ϑsy with four prediction methods are given in Table 3. According to Table 3, the prediction error values of the prediction model proposed in this paper are the lowest compared with the other three methods. In the optimal case, the RMSE value and the L_2 of PPO-RL decreased by 0.1540 and 2.6624, compared with the LSTM network.

Table 3. Evaluation results for different predictive methods for ϑ_{sy}.

Methods	RMSE	MAE	L_2	L_∞
RNN	0.1867	0.1480	3.2288	44.2432
LSTM	0.1732	0.1570	2.9943	46.9501
PPO-RL	0.2293	0.2145	3.9644	64.1299
EWMA	0.0753	0.0612	1.3020	18.3111

In order to compare the computational complexity of the four methods, the calculation time of each method on the training set and test set was given after 30 iterations. It can be seen in Table 4 that the computational complexity of the PPO-RL is relatively high for ϑ_{sy} on the training set. By contrast, it shows a low complexity for ϑ_{sy} on the test set, which is 0.38 min lower than that of the LSTM.

Table 4. Computing time analysis of train learning and test learning for ϑ_{sy}.

	EWMA	RNN	LSTM	PPO-RL
train learning (min)	-	725.28	269.74	827.56
test learning (min)	0.32	0.41	0.62	0.24

The quantitative evaluation indicators of temperature prediction of stator tooth with four prediction methods are given in Table 5. As shown, the PPO-RL method proposed in this paper has achieved an excellent performance. The RMSE value and MAE value of PPO-RL decreased by 0.0117 and 0.0424, respectively, and its Euclidean norm L_2 reached the minimum value.

Table 5. Evaluation results for different predictive methods for ϑ_{st}.

Methods	RMSE	MAE	L_2	L_∞
RNN	0.3630	0.2843	6.2768	85.0010
LSTM	0.2674	0.2222	4.6240	66.4279
PPO-RL	0.1277	0.1097	2.2075	32.8097
EWMA	0.1160	0.0673	2.0057	20.1257

As can be seen in Table 6, LSTM has the lowest computational complexity for ϑ_{st} on the training set, while compared with LSTM, the RNN, and EWMA, PPO-RL has the lowest computational complexity of ϑ_{st} on the test set.

Table 6. Computing time analysis of train learning and test learning for ϑ_{st}.

	EWMA	RNN	LSTM	PPO-RL
train learning (min)	-	779.93	248.00	830.47
test learning (min)	0.36	0.39	0.74	0.19

The quantitative evaluation indicators of temperature prediction of ϑ_{sw} with four prediction methods are given in Table 7. It can be seen in the table that the PPO-RL model has a lower prediction error and can obtain a higher prediction accuracy. It is worth noting that the LSTM network has a lower error than the RNN network and the EWMA method in the prediction experiment of the ϑ_{st}. The errors in the prediction of the ϑ_{sy} and $\vartheta_{s}w$ are high.

Table 8 shows the computing time analysis of the four methods on the training set and test set for ϑ_{sw}. The PPO-RL shows the optimal computational complexity on the test set for ϑ_{sw}, while the LSTM has the greatest computational complexity on the test set, which is 0.79 min higher than PPO-RL.

Table 7. Evaluation results for different predictive methods for ϑ_{sw}.

Methods	RMSE	MAE	L_2	L_∞
RNN	0.2929	0.2348	5.0639	70.2100
LSTM	0.1293	0.1085	2.2364	32.4482
PPO-RL	0.1997	0.1545	3.4537	46.1984
EWMA	0.1006	0.0637	1.7404	19.0599

Table 8. Computing time analysis of train learning and test learning for ϑ_{st}.

	EWMA	RNN	LSTM	PPO-RL
train learning (min)	-	706.23	234.53	831.98
test learning (min)	0.32	0.25	1.00	0.21

5. Conclusions

This paper systematically elaborates on the research status and shortcomings of traditional thermal network and machine learning methods on PMSM temperature prediction. Based on the problems found in the literature review, a temperature prediction method of PMSM based on proximal optimization is proposed. This method can obtain a better performance by adjusting the network structure and minimizing the objective function of PPO.

The prediction performance of the proposed method as well as three other classical machine learning networks were explored to validate the applicability and validity of this method. The results further show that the performance of an LSTM neural network is uncertain with regard to the test samples, which increases the difficulty of solving the global optimal values in the training process. In future research work, the improvement of the real-time performance of this method should be considered.

Author Contributions: Y.C.: data reduction, project administration, and experimentation; C.Z. (Chenguang Zhang): methodology, validation, and writing—original draft preparation; G.C.: review and editing; Yulai Zhang: conceptualization and structuration; C.Z. (Cheng Zhao): translation and editing. All authors have read and agreed to the published version of the manuscript.

References

1. Zhu, Y.; Xiao, M.; Lu, K.; Wu, Z.H.; Tao, B. A Simplified Thermal Model and Online Temperature Estimation Method of Permanent Magnet Synchronous Motors. *Appl. Sci.* **2019**, *9*, 3158. [CrossRef]
2. Li, Y.Z.; Zhu, S.J.; Li, Y.H.; Lu, Q. Temperature Prediction and Thermal Boundary Simulation Using Hardware-in-Loop Method for Permanent Magnet Synchronous Motors. *IEEE/ASME Trans. Mechatron.* **2016**, *21*, 276–287. [CrossRef]
3. Kral, C.; Haumer, A.; Lee, S.B. A Practical Thermal Model for the Estimation of Permanent Magnet and Stator Winding Temperatures. *IEEE Trans. Power Electron.* **2014**, *29*, 455–464. [CrossRef]
4. Mohamed, A.H.; Hemeida, A.; Rashekh, A.; Vansompel, H.; Arkkio, A.; Sergeant, P. A 3D Dynamic Lumped Parameter Thermal Network of Air-Cooled YASA Axial Flux Permanent Magnet Synchronous Machine. *Energies* **2018**, *11*, 16. [CrossRef]

5. Wallscheid, O.; Specht, A.; Böecker, J. Determination of Rotor Temperature for an Interior Permanent Magnet Synchronous Machine Using a Precise Flux Observer. In Proceedings of the 2014 International Power Electronics Conference (IPEC-Hiroshima 2014—ECCE ASIA), Hiroshima, Japan, 18–21 May 2014; pp. 1501–1507.

6. Wallscheid, O.; Kirchgäessner, W.; Böecker, J. Investigation of Long Short-Term Memory Networks to Temperature Prediction for Permanent Magnet Synchronous Motors. In Proceedings of the 2017 International Joint Conference On Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 1940–1947.

7. Lan, Z.Y.; Wei, X.H.; Li, H.R.; Liao, K.L.; Chen, L.H. Thermal Analysis of PMSM Based on Lumped Parameter Thermal Network Method. *J. Electr. Eng.* **2017**, *12*, 13–16.

8. Sciascera, C.; Giangrande, P.; Papini, L.; Gerada, C.; Galea, M. Analytical Thermal Model for Fast Stator Winding Temperature Prediction. *IEEE Trans. Ind. Electron.* **2017**, *64*, 6116–6126. [CrossRef]

9. Liu, P.; Wang, X.; Sun, Q.Z.; Huang, S.D.; Tu, C.M.; Yang, W.L. Signal injection strategy optimization of stator winding temperature estimation for permanent magnet synchronous motor. *Electr. Mach. Control* **2019**, *23*, 18–26.

10. Du, A.M.; Zhang, D.X.; Sun, M.M.; Yun, Z.Z. Research on Temperature Field of the Permanent Magnet Synchronous Motors for Hybrid Vehicles Cooled by Oil. *Automob. Technol.* **2019**, *4*, 34–39.

11. Xu, C.L.; Chen, H.X.; Wang, J.Y.; Guo, Y.B.; Yuan, Y. Improving prediction performance for indoor temperature in public buildings based on a novel deep learning method. *Build. Environ.* **2019**, *148*, 128–135. [CrossRef]

12. Liu, J.; Zhang, T.; Han, G.J.; Gou, Y. TD-LSTM: Temporal Dependence-Based LSTM Networks for Marine Temperature Prediction. *Sensors* **2018**, *18*, 3797. [CrossRef] [PubMed]

13. Wallscheid, O.; Specht, A.; Böecker, J. Observing the Permanent-Magnet Temperature of Synchronous Motors Based on Electrical Fundamental Wave Model Quantities. *IEEE Trans. Ind. Electron.* **2017**, *64*, 3921–3929. [CrossRef]

14. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization algorithms. *arXiv* **2017**, arXiv:1707. 06347.

15. Berthold, M.; Höppner, F. On Clustering Time Series Using Euclidean Distance and Pearson Correlation. *arXiv* **2016**, arXiv:1601. 02213.

16. Wasserstein, R.L.; Lazar, N.A. The ASA's Statement on *p*-values: Context, Process, and Purpose. *Am. Stat.* **2016**, *70*, 129–133. [CrossRef]

17. Wang, Z.Y.; Bapst, V.; Heess, N.; Mnih, V.; Munos, R.; Kavukcuoglu, K.; Freitas, N.D. Sample Efficient Actor-Critic with Experience Replay. *arXiv* **2016**, arXiv:1611.01224.

18. Hietaharju, P.; Ruusunen, M.; Leiviskä, K. A Dynamic Model for Indoor Temperature Prediction in Buildings. *Energies* **2018**, *11*, 1477. [CrossRef]

19. Wu, D.; Wang, H.; Seidu, R. Smart data driven quality prediction for urban water source management. *Future Gener. Comput. Syst.* **2020**, *107*, 418–432. [CrossRef]

20. Carlini, N.; Wagner, D. Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods. *arXiv* **2017**, arXiv:1705. 07263.

21. Carlini, N.; Wagner, D. Towards Evaluating the Robustness of Neural Networks. *arXiv* **2016**, arXiv:1608.04644.