



Article

Temperature Prediction of PMSMs Using Pseudo-Siamese Nested LSTM

Yongping Cai ¹, Yuefeng Cen ^{1,*}, Gang Cen ¹, Xiaomin Yao ², Cheng Zhao ³ and Yulai Zhang ¹

- ¹ School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China; ypchoi97@163.com (Y.C.); gcen@163.com (G.C.); zhangyulai@zust.edu.cn (Y.Z.)
² College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310014, China; yaoxm@zjut.edu.cn
³ School of Economics, Zhejiang University of Technology, Hangzhou 310014, China; zhaoc@zjut.edu.cn
* Correspondence: cyf@zust.edu.cn

Abstract: Permanent Magnet Synchronous Motors (PMSMs) are widely used in electric vehicles due to their simple structure, small size, and high power-density. The research on the temperature monitoring of the PMSMs, which is one of the critical technologies to ensure the operation of PMSMs, has been the focus. A Pseudo-Siamese Nested LSTM (PSNLSTM) model is proposed to predict the temperature of the PMSMs. It takes the features closely related to the temperature of PMSMs as input and realizes the temperature prediction of stator yoke, stator tooth, and stator winding. An optimization algorithm of learning rate combined with gradual warmup and decay is proposed to accelerate the convergence during the training and improve the training performance of the model. Experimental results reveal the proposed method and Nested LSTM (NLSTM) achieves high accuracy by comparing with other intelligent prediction methods. Moreover, the proposed method is slightly better than NLSTM in temperature prediction of PMSMs.

Keywords: PMSMs; temperature prediction; NLSTM



Citation: Cai, Y.; Cen, Y.; Cen, G.; Yao, X.; Zhao, C.; Zhang, Y. Temperature Prediction of PMSMs Using Pseudo-Siamese Nested LSTM. *World Electr. Veh. J.* **2021**, *12*, 57. <https://doi.org/10.3390/wevj12020057>

Academic Editors: Ming Cheng and Joeri Van Mierlo

Received: 17 February 2021
Accepted: 30 March 2021
Published: 2 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Permanent magnet synchronous motors (PMSMs) are the core components of electric vehicles due to their excellent power density, efficiency, and prime torque [1]. However, the high power density will cause a serious temperature increase, which may affect the working efficiency and even damage the core components of motors [2]. Therefore, an enormous amount of research effort goes into the temperature prediction of the PMSMs to ensure the safe running of the motors [3,4].

There were three main categories of methods proposed by previous researchers to predict the temperature of PMSMs as follows: temperature formula, parameter identification, and thermal networks. Methods of the first category mainly included finite element analysis (FEA) and computational fluid dynamics (CFD) [5,6]. The advantages of these methods are convenient to obtain the temperature of arbitrarily shaped devices. However, the modeling process of these methods requires high computing complexity [7]. Methods of parameter identification were mainly realized by flux observation and signal injection [8,9], which required high precision of the measuring instruments [10]. Lumped parameter thermal networks (LTPNs) were widely used when methods of thermal networks were considered [11–13]. Based on the idea of the thermal circuit method, the LTPN method makes a more detailed partition of motor structure. Depending on the degree of dispersion of the topology, the thermal network can be composed of several or even hundreds of nodes. The more discrete the LTPNs are, the more accurate the result will be. However, it will also bring additional calculations.

With the rapid development of artificial intelligence technology, deep learning models have been widely applied to temperature prediction in industrial fields. Several models

based on deep learning were applied to predict the environmental changes such as the greenhouse and the temperature of sea surface [14]. A long short-term memory (LSTM) network [15] was first introduced to predict the temperature of PMSMs, and it performed well in the prediction accuracy [16]. The work presented in [17] showed the feasibility of the deep residual convolutional loop network in the application of temperature prediction of PMSMs. It confirmed the feasibility and accuracy of temperature prediction of PMSMs.

The NLSTM network was another deep learning model based on the LSTM network, which owned a more efficient time hierarchy and flexible processing of internal memory [18]. It was applied to realize seizure detection combined with convolution neural networks. The advanced architecture effectively explored the inherent time dependence hidden in electroencephalogram (EEG) signals and revealed the performance as superior [19]. The ability of NLSTM to dynamically capture hierarchical time dependencies in traffic data was verified by using this characteristic. NLSTM could also efficiently access internal memory when constructing the time layer structure [20].

In this paper, a novel model based on the NLSTM called the PSNLSTM is proposed to predict the future temperature of PMSMs. Two NLSTM networks with different time steps are used to capture the time dependence and abstract features of temperature changes. Abstract features refer to higher-level features, which can better describe the temperature change characteristics of PMSMs. They are not obtained through simple linear transformations, and can be more easily learned and calculated by neural networks. Adjustment of learning rate plays a vital role in the training of the deep learning networks. Thus, an optimization algorithm of the learning rate is proposed to accelerate the convergence and improve training performance.

The remainder of this paper is organized as follows: the model of the Pseudo-Siamese NLSTM network is introduced in Section 2. The temperature benchmark of the PMSMs and evaluation indicators are introduced in Section 3. Furthermore, the optimization algorithm of the learning rate is demonstrated in Section 4. The experimental results and assessment are demonstrated in Section 5. The paper is concluded in Section 6.

2. Pseudo-Siamese Nested LSTM Network

2.1. Nested LSTM Network

The NLSTM network is a novel RNN architecture with multiple levels of memories, which add depth to LSTM via nesting as opposed to stacking [18]. The architecture of the NLSTM memory block is shown in Figure 1.

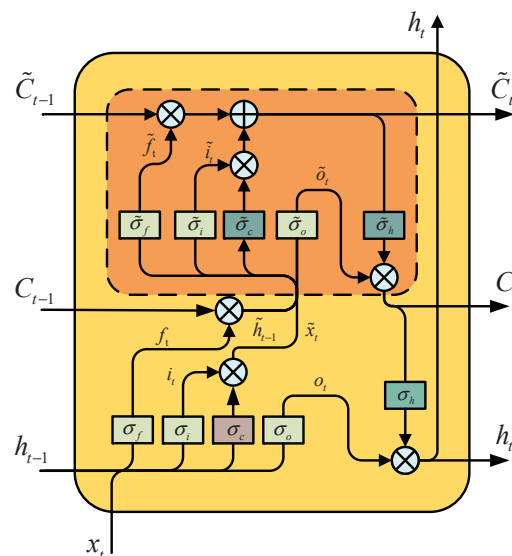


Figure 1. Structure of the NLSTM memory block.

The input and output of the NLSTM network are the same as the LSTM network. Another temporary cell state is added in NLSTM, which is used to transfer the memory state of the internal memory block. An NLSTM memory block is equivalent to two LSTM memory blocks in the form of a nested structure. The inner LSTM block, which is surrounded by dotted lines in Figure 1, becomes the memory function of the external LSTM block. The memory function is dedicated to managing the long-term information between the memory blocks.

The external LSTM calculates \tilde{h}_{t-1} and \tilde{x}_t through the input x_t of the current time and the output h_{t-1} of the previous time:

$$f_t = \sigma_f(W_f[h_{t-1}, x_t] + b_f) \quad (1)$$

$$i_t = \sigma_i(W_i[h_{t-1}, x_t] + b_i) \quad (2)$$

$$o_t = \sigma_o(W_o[h_{t-1}, x_t] + b_o) \quad (3)$$

$$\tilde{h}_{t-1} = f_t \odot C_{t-1} \quad (4)$$

$$\tilde{x}_t = i_t \odot \sigma_c(W_c[x_t, h_{t-1}] + b_c) \quad (5)$$

where f_t , i_t and o_t are the three states of the gates, σ_i , σ_ρ , σ_f are the sigmoid activation functions, which realize selective memory and forgetting. They achieve long-term memory without causing a gradient explosion. σ_c is the linear activation function. In addition, \tilde{h}_{t-1} and \tilde{x}_t obtained from external LSTM are input and the hidden state of NLSTM internal memory function, respectively.

For the internal memory function, the internal operation mode is controlled by the following equation:

$$\tilde{i}_t = \tilde{\sigma}_i(\tilde{W}_i[\tilde{h}_{t-1}, \tilde{x}_t] + \tilde{b}_i) \quad (6)$$

$$\tilde{f}_t = \tilde{\sigma}_f(\tilde{W}_f[\tilde{h}_{t-1}, \tilde{x}_t] + \tilde{b}_f) \quad (7)$$

$$\tilde{C}_t = \tilde{f}_t \odot \tilde{C}_{t-1} + \tilde{i}_t \odot \tilde{\sigma}_c(\tilde{W}_c[\tilde{h}_{t-1}, \tilde{x}_t] + \tilde{b}_c) \quad (8)$$

$$\tilde{o}_t = \tilde{\sigma}_o(\tilde{W}_o[\tilde{h}_{t-1}, \tilde{x}_t] + \tilde{b}_o) \quad (9)$$

$$\tilde{h}_t = \tilde{o}_t \odot \tilde{\sigma}_h(\tilde{C}_t) \quad (10)$$

where $\tilde{\sigma}_i$, $\tilde{\sigma}_o$, $\tilde{\sigma}_f$ all are the sigmoid activation function, which are consistent with external LSTM. $\tilde{\sigma}_c$, $\tilde{\sigma}_h$ are the tanh activation function. After a series of operations of the internal LSTM, the state update of the external LSTM unit is obtained:

$$C_t = \tilde{h}_t \quad (11)$$

Finally, the following equation is used to obtain the last output of NLSTM memory block:

$$h_t = o_t \odot \sigma_h(C_t) \quad (12)$$

where σ_h is the tanh activation function.

2.2. Model Architecture Proposed

The Siamese Network is a conjoined neural network architecture shown in Figure 2. It realizes "Siamese" through weight sharing [21]. If the weights of the neural networks on the left and right are not shared or the neural networks are different, the architecture is defined as the Pseudo-Siamese networks. This architecture is widely used in the fields of information similarity matching and comparison [22,23].

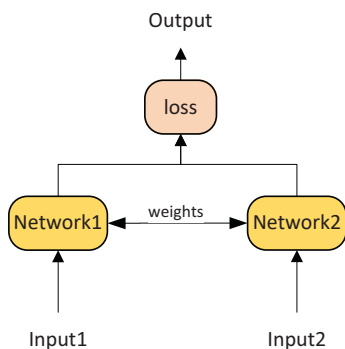


Figure 2. The architecture of the Siamese network.

A novel architecture based on NLSTM and Pseudo-Siamese network calling PSQLSTM is proposed in this paper. Two NLSTM networks with different time steps are adopted as the neural networks in the Pseudo-Siamese network. The architecture of the model is shown in Figure 3. After the temperature benchmark data set of PMSMs is preprocessed, it is sent to the recurrent layer. In the recurrent layer, one NLSTM network with long time steps is used to obtain the trend of temperature series changes of PMSMs. The other with short time steps is used to obtain the detail of the temperature series changes. Through the above two NLSTM networks, the higher level temperature features of the next moment are obtained. Then, a full connection layer is added to extract the temperature features for each NLSTM network. Finally, another full connection layer is used to fuse the temperature features of different neural networks to get the predicted temperature.

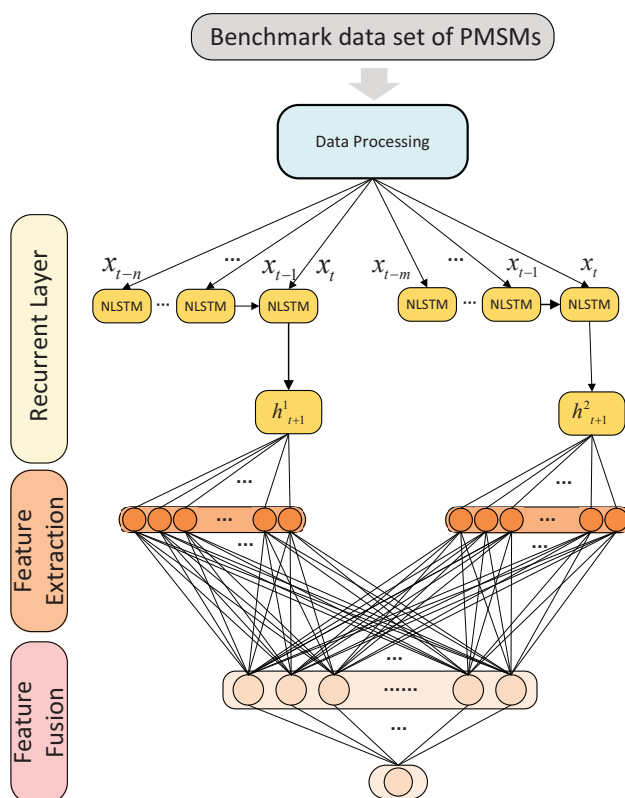


Figure 3. The architecture of PSQLSTM. Where $x_{t-m}, \dots, x_{t-n}, \dots, x_t$ are one-dimensional tensors as inputs, $n < m$. h_{t+1}^1 is the higher level temperature characteristics obtained by NLSTM network with short time steps, h_{t+1}^2 is the higher level temperature characteristics obtained by NLSTM network with long time steps, and these are also one-dimensional tensors.

3. Temperature Benchmark Data Set and Valuation Indicators

3.1. Temperature Benchmark Data Set

The data set used in this experiment is from the Kaggle Data Science online competition platform. In addition, it is based on a test bench having a three-phase PMSM mounted (for detailed information of PMSM, see [24]). The data measurement and collection are provided by the Department of Power Electronics and Electrical Drive of Paderborn University in Germany. Table 1 shows the column labels for the benchmark dataset. It includes more than 990,000 pieces of data.

Table 1. The column labels of the benchmark dataset.

Parameter Name	Symbol
Ambient temperature	ϑ_a
Coolant temperature	ϑ_c
Voltage d-component	u_d
Voltage q-component	u_q
Motor speed	n_{mech}
Actual torque	T_m
Current d-component	i_d
Current q-component	i_q
Permanent Magnet temperature	ϑ_{PM}
Stator yoke temperature	ϑ_{SY}
Stator tooth temperature	ϑ_{ST}
Stator winding temperature	ϑ_{SW}
unique ID	id

All recordings are sampled at 2 Hz. Each measurement session in the data set can represent the entire electrothermal characteristics of the PMSM well. In addition, this data set is mildly anonymized, and each set of parameters has been standardized.

In the experiment, 51 measurement sessions in the entire data were set as the training set, and the one remaining session ($id = 32$) was set as the test set. The data set was down-sampled at an appropriate frequency and cleaned. Since each measurement session is independent of each other, the down-sampling operation is performed separately, retaining only the same frequency. In addition, from time to time in the training model, data from different measurement session is not continuous data as input to the model. Finally, the number of data in the training sets is 32,263, and the number of data in the test sets is 412.

The temperatures of ϑ_{SY} , ϑ_{ST} and ϑ_{SW} are chosen as the prediction object of the PMSMs. Since the temperature characteristics of these core components are different, we have separately trained PSNLSTM and the comparative models. In addition, the input of the model does not include the predicted target temperature feature.

3.2. Evaluation Indicators

There are several common evaluation indicators adopted to evaluate the prediction accuracy as follows: mean square error (MSE), root mean square error (RMSE), and mean absolute error (MAE). The definition formulas are as follows:

$$MAE = \frac{1}{n} \sum_{t=0}^{n-1} |a_t - p_t| \quad (13)$$

$$MSE = \frac{1}{n} \sum_{t=0}^{n-1} (a_t - p_t)^2 \quad (14)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=0}^{n-1} (a_t - p_t)^2} \quad (15)$$

Among them, a_t and p_t respectively represent the true value and predicted value at the time t .

To assess the volatility of the predicted results, the standard deviation of the prediction error (STDPE) is introduced, which is defined using the following formula:

$$d_t = p_t - a_t \quad (16)$$

$$STDPE = \sqrt{\frac{\sum_{t=0}^{n-1} (d_t - \bar{d})^2}{n-1}} \quad (17)$$

where d_t is the prediction error at time t , and the definitions of the other parameters are the same as those mentioned above.

Another evaluation indicator, coefficient of determination is represented as R^2 . It can reflect the proportion of the variation of dependent variables that can be explained by independent variables, and its definition formula is as follows:

$$R^2 = 1 - \frac{\sum_{t=0}^{n-1} (a_t - p_t)^2}{\sum_{t=0}^{n-1} (a_t - \bar{a})^2} \quad (18)$$

It can be used to observe the prediction error compared with the mean reference error. The value region of R^2 is between 0 and 1.

4. Learning Rate Optimization

During the training process of the deep learning models, the optimization of hyper-parameters plays an important role. As one of the essential hyper-parameters in deep learning models, the learning rate determines whether and when the objective function converges to the local optimum. A proper learning rate can help the objective function converge to an optimum in a proper period. If the value of the learning rate is too small, the loss of models will decline very slowly. Otherwise, a large learning rate will cause a considerable variation of parameters when the parameters update, which will lead to the missing of the optimal point or lead to the rise of the model loss. It is worth noting that the model requires a different learning rate in each stage of training. When the parameters fall into the local optimum, a larger learning rate is needed to escape from this point. Correspondingly, a smaller learning rate is required to approximate the global optimum.

Therefore, it is necessary to adjust the learning rate dynamically during the training of the model. Thus, gradual warmup is introduced to accelerate the convergence of deep learning models, which was first mentioned in the literature [25]. After that, the attenuation of the learning rate should be considered to get the optimal result after several epochs, instead of keeping the learning rate constant until the end. In this experiment, a novel optimization algorithm of learning rate is proposed, which combines a gradual warmup algorithm, cosine annealing algorithm, and Nadam optimizer to realize the update of the learning rate effectively. This algorithm is designed to accelerate the convergence of the NLSTM on the PMSMs temperature data set and improve its training performance.

In this optimization algorithm, the learning rate changes can be divided into the following three stages: gradual warmup, keeping constant, and annealing. First, within a certain number of training steps, the learning rate gradually increases from a small value to a preset value. Thus, the overfitting of the network can be avoided. Using a smaller learning rate can make the model gradually stabilize. After a gradual warmup, the model has achieved a relatively stable state. Then, using a larger learning rate of training, be able to make the model more quickly converge to achieve a better training effect. Second, after the gradual warmup is completed, the model will continue training at the set learning rate. It also ensures that the model can jump out of the local optimum. After that, the model

result is close to the global optimum. The learning rate should be attenuated to avoid the miss of the global optimum. The cosine function is widely adopted to anneal the learning rate.

In practical applications, the combination of cosine annealing and stochastic gradient descent (SGD) can speed up model fitting to a certain extent and achieve better fitting results. Moreover, it is suggested in [26] that learning rate annealing can also be added when Adam is used. In addition, Adam [27] is different from SGD, which is an optimizer with first-order and second-order momentum. In Adam, the main parameter update formula is as follows:

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\hat{V}_t + \varepsilon}} \hat{m}_t = \theta_{t-1} - \frac{\eta}{\sqrt{\hat{V}_t + \varepsilon}} \left(\frac{\beta_1 m_{t-1}}{1 - \beta_1^t} + \frac{(1 - \beta_1) g_t}{1 - \beta_1^t} \right) \quad (19)$$

Among them, t is the time step of updating model parameters, θ is the parameter to be updated, β_1 is the exponential decay rate of the first-order moment, η is the learning rate, ε is the constant term, m is the first-order moment estimation of the gradient, \hat{m} is the correction of m , and \hat{V} is the correction of the second-order moment estimation of the gradient.

In this learning rate optimization algorithm, we try to use Nadam as an optimizer, which can be regarded as the combination of Nesterov and Adam [28,29]. In Nadam, the main parameter update formula is shown below:

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\hat{V}_t + \varepsilon}} \left(\frac{\beta_1 m_t}{1 - \beta_1^{t+1}} + \frac{(1 - \beta_1) g_t}{1 - \beta_1^t} \right) \quad (20)$$

where the definition of each parameter is the same as that of Adam. The momentum m_{t-1} at time $t - 1$ is replaced by the momentum at time t , thus taking into account the “future factor” and achieving the effect of Nesterov.

To verify the proposed optimization algorithm of the learning rate, changes in the learning rate are observed in Figure 4. The fixed learning rate is set as 0.001, and the total epochs is set as 100. We define the first 10 epochs as the gradual warmup stage, the next 10 epochs as the keeping constant stage, and the rest as the cosine annealing stage.

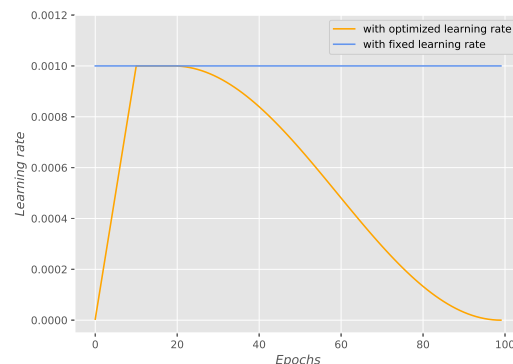


Figure 4. Changes of learning rate using optimization algorithm.

The comparative losses of the model during the training process and validation are shown in Figure 5. We can observe that, in the first two epochs, the convergence rate of the proposed optimization algorithm is slower, which is shown in Figure 5a. Because the first 10 epochs of the optimization algorithm fall into the stage of gradual warmup and the learning rate is relatively low at the beginning, in the training set, the loss convergence performance with the optimization algorithm of learning rate is better than that without the learning rate optimization algorithm.

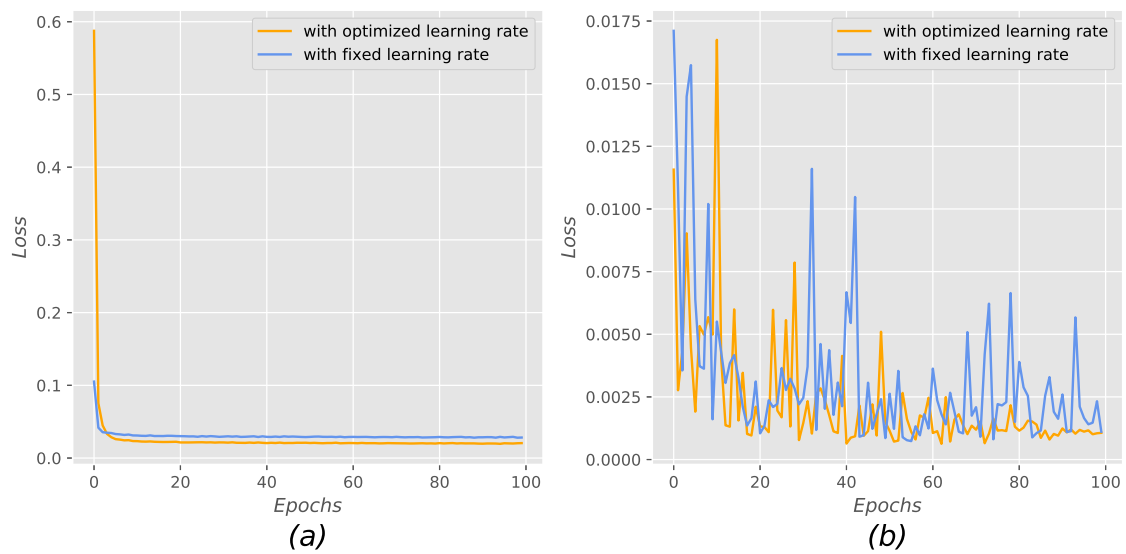


Figure 5. The comparative losses of the model. (a) the convergence of the loss function on the training set; (b) the convergence of the loss function on the validation set.

Figure 5b shows the validation loss curve of the proposed algorithm, which fluctuates significantly at the beginning of the epochs. In the first 10 epochs, it is in a gradual warmup stage and the learning rate increases slowly. Therefore, it fluctuates more significantly than the loss curve of the fixed learning rate. In the latter part of the epochs, it performs more stably than the loss curve with fixed learning rate. This is due to the involvement of the cosine annealing algorithm in the later stage. After the keeping constant stage, the learning rate enters the cosine annealing stage, and its annealing according to the cosine function $[0,1]$ interval. In the early annealing stage, the learning rate decreases slowly still remains at a large value, which helps the Nadam optimizer accumulate momentum out of the local optimal point and look for a better convergence point. Then, the learning rate gradually decreases, the model can converge quickly to the best point. Finally, the learning rate changes slowly with a small value and slowly approaches the optimal point to avoid missing it. After the above learning rate annealing, the model finally converges to a better state. Therefore, the proposed algorithm of the learning rate is helpful to accelerate the convergence and improve the accuracy of the model.

5. Performance Assessment

There are two different NLSTMs in PSNLSTM. The grid search method is used to match the lengths of time steps, then two sizes of time steps for PSNLSTM are set as 7 and 4 to obtain the different temperature features of the PMSMs. Accordingly, we set NLSTM as one of the comparative models. NLSTM is an advanced variant of LSTM, and LSTM is widely used in the prediction of temperature series. Therefore, it is necessary to compare the temperature prediction results of LSTM and PSNLSTM of PMSMs. As mentioned in Section 2.1, NLSTM is equivalent to two LSTMs which are composed of nested structures. To prove the superiority of NLSTM structure in temperature prediction of PMSM, stacked LSTM is also set as a comparative model. The stacked LSTM in this paper has two layers of LSTM network in the loop layer, called LSTM-2. In this study, LSTM, LSTM-2, NLSTM, and PSNLSTM, respectively, predict the temperature of the core components ϑ_{SY} , ϑ_{ST} and ϑ_{SW} , and all these models utilizing the temperature features at the moment t to predict the temperature at the next moment $t + 1$. The time steps for the comparative models are set as 7. The other hyper-parameters are listed in Table 2. In addition, the experiment platform is as follows: Win10 (64 bits), i7-6700HQ Intel(R)Core(TM), 16 GB, GTX960M-2G, and the versions of the deep learning framework were Keras 2.2.4 and Tensorflow 1.12.0, version of PyCharm 2020.3.2 (JetBrains, Prague, Czech Republic).

Table 2. The hyper-parameters of each network in the experiment.

Hyper-Parameter	LSTM	LSTM-2	NLSTM	PSNLSTM
Hidden layer	3	4	4	4
Units	64	(64, 64)	64	64
Time steps	7	7	7	7&4
Weight	normal	normal	normal	normal
Optimizer	Nadam	Nadam	Nadam	Nadam
Learning rate	0.001	0.001	0.001	0.001
Warm-up epochs	10	10	10	10
Epochs	100	100	100	100
Gaussian noise	1×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}
Drop out	0.2	0.2	0.2	0.2

To evaluate the performance of PSNLSTM and learning rate optimization algorithm, we compare it with LSTM, LSTM-2, and NLSTM for temperature prediction of ϑ_{SY} , ϑ_{ST} and ϑ_{SW} . Tables 3–5 show the performance of each model from the four technical indicators of MSE, MAE, RMSE, R^2 , and STDPE.

Table 3. The temperature prediction performance of ϑ_{SY} .

Model	MSE (%)	MAE (%)	RMSE (%)	R^2 (%)	STDPE
LSTM	0.0927	2.3625	3.0448	99.7374	0.0304
LSTM-2	0.0897	2.2004	2.9947	99.7460	0.3000
NLSTM	0.0627	1.7879	2.5044	99.8223	0.0230
PSNLSTM	0.0508	1.6860	2.2537	99.8561	0.0222

Table 4. The temperature prediction performance of ϑ_{ST} .

Model	MSE (%)	MAE (%)	RMSE (%)	R^2 (%)	STDPE
LSTM	0.1321	2.4961	3.6339	99.8435	0.0661
LSTM-2	0.1683	2.9113	4.1018	99.8006	0.0615
NLSTM	0.0934	2.1675	3.0567	99.8892	0.0510
PSNLSTM	0.0998	2.455845	3.1598	99.8816	0.0467

Table 5. The temperature prediction performance of ϑ_{SW} .

Model	MSE (%)	MAE (%)	RMSE (%)	R^2 (%)	STDPE
LSTM	0.4380	4.0302	6.6183	99.6873	0.0357
LSTM-2	0.3902	4.4855	6.2463	99.7215	0.0409
NLSTM	0.2609	3.3056	5.1074	99.8138	0.0306
PSNLSTM	0.2198	3.4557	4.6888	99.8430	0.0301

The values of R^2 in Tables 3–5 for all of the models are close to 1, which means that all of these models achieve an excellent fitting effect. NLSTM performs better than LSTM in the view of R^2 because the NLSTM captures more abstract features contained in the temperature data than LSTM. The design of PSNLSTM with different time steps even creates the possibility to get more comprehensive information from the data. The NLSTM with a longer time step contains the temperature information from a longer time ago, which is conducive to learning the temperature change trend, and theoretically can reduce the prediction error of the temperature change point to a certain extent. The other NLSTM is better at learning temperature changes in a short period of time, can grasp the details of temperature changes, and theoretically can improve the accuracy of non-abrupt temperature changes to a certain extent. The values of MSE, MAE, and RMSE should be close to 0,

which reveals the good accuracy of the models. The values of MSE, MAE, and RMSE for the NLSTM are smaller compared with the LSTM and LSTM-2 in Tables 3–5. Meanwhile, the performance of PSNLSTM is better compared with the NLSTM in the temperature prediction of ϑ_{SW} and ϑ_{SY} . In particular, the MSE of PSNLSTM in Table 5 is 49.82% lower than LSTM and 43.67% lower than LSTM-2. The performance of PSNLSTM and NLSTM are generally close for the prediction of ϑ_{ST} . The first four evaluation indicators of PSNLSTM are slightly worse than NLSTM, but its STDPE is 8.43% lower than that of NLSTM, that is, the prediction of PSNLSTM is more stable and its error volatility is lower. In addition, in the temperature prediction of ϑ_{SY} , ϑ_{ST} and ϑ_{SW} , the error volatility of the PSNLSTM prediction result is the lowest. In order to compare the performance difference between PSNLSTM and competitive models, the difference of the values between two networks for MSE, for MAE, for RMSE and for STDPE, respectively, and then the average of the three values is calculated. In this study, the temperatures of ϑ_{SY} , ϑ_{ST} and ϑ_{SW} were predicted, so there are three previously mentioned averages between each competitive model and PSNLSTM. Finally, the three averages are averaged, and the averages of evaluation indicators are obtained, which can describe the overall difference between each competitive model and PSNLSTM in PMSM temperature prediction. The overall error of PSNLSTM is 25.34% lower than LSTM, 28.23% lower than LSTM-2, and 3.68% lower than NLSTM. The results indicate a very slight superiority of PSNLSTM over NLSTM.

The measured and predicted temperatures of the ϑ_{SY} and ϑ_{ST} implemented, respectively, by LSTM, LSTM-2, NLSTM and PSNLSTM are shown in Figures 6 and 7. Moreover, the prediction error curves of the models are also provided, which is calculated by Equation (16). The black error curves on the right correspond to the predictions of each model on the left.

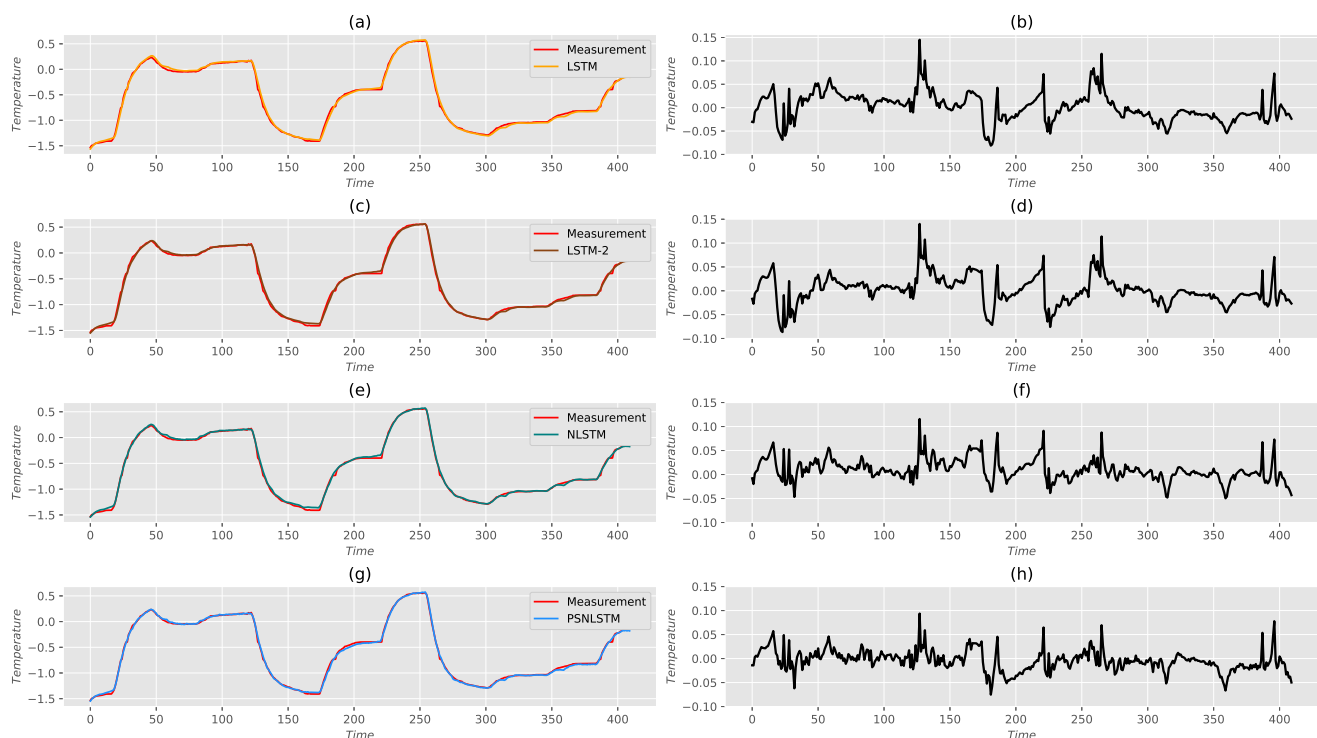


Figure 6. Temperature prediction of ϑ_{SY} for each model. (a,c,e,g) the temperature fitting curves of ϑ_{SY} for each model; (b,d,f,h) the temperature prediction error curves of the ϑ_{SY} for each model, which are obtained by subtracting the measured values from the predicted values. In addition, the error curves correspond to the left fitting curves respectively.

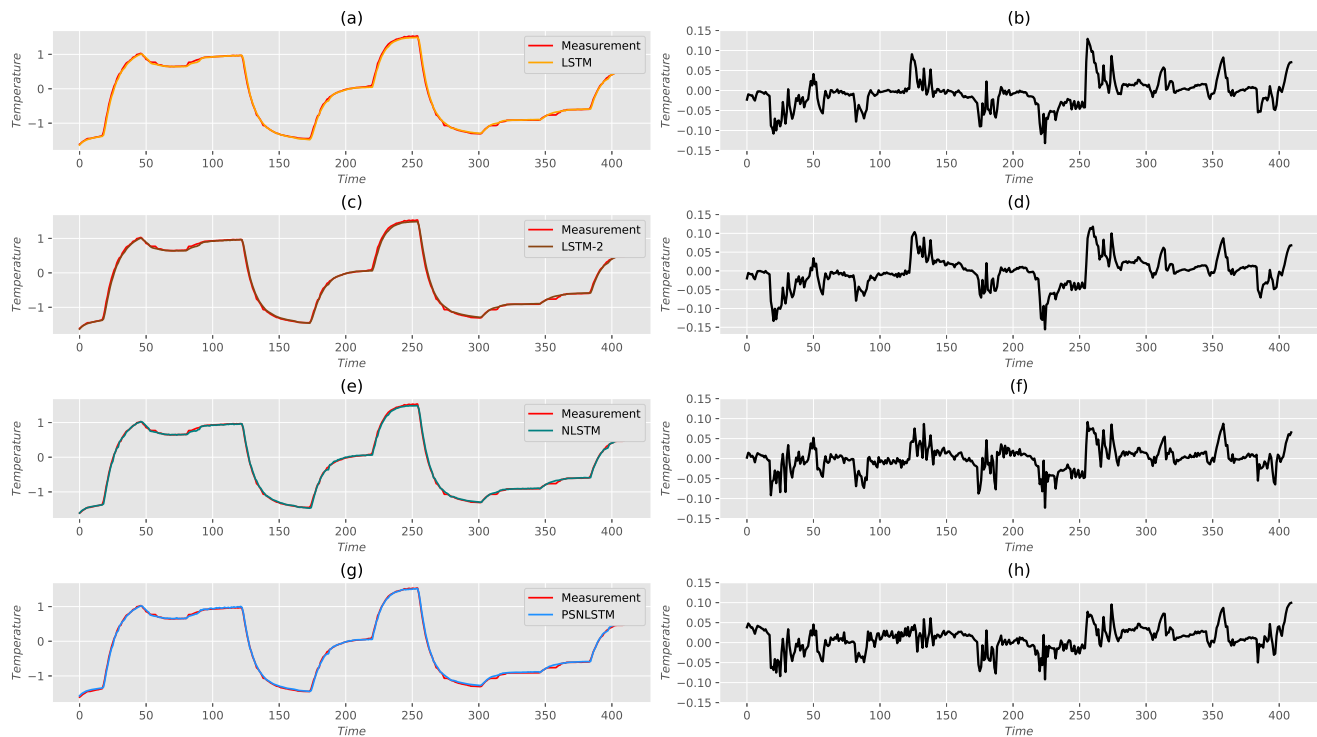


Figure 7. Temperature prediction of ϑ_{ST} for each model. (a,c,e,g) the temperature fitting curves of ϑ_{ST} for each model; (b,d,f,h) the temperature prediction error curves of the ϑ_{ST} for each model, which are obtained by subtracting the measured values from the predicted values. In addition, the error curves correspond to the left fitting curves, respectively.

It can be observed that the prediction temperature curves and measurement curves fit well for these four deep learning models. The fluctuation of the error curves of PSNLSTM and NLSTM, as attested visually, is less than the fluctuation of the respective curves for LSTM and LSTM-2. A less abrupt nature of the error curves is more obvious at the temperature transition time instants for both PSNLSTM and NLSTM and even a slight advantage can be attested at those time instants for PSNLSTM over NLSTM.

The measured and predicted temperatures of the ϑ_{SW} implemented, respectively, by LSTM, LSTM-2, NLSTM, and PSNLSTM are shown in Figure 8. It can be observed that the range of error curves in Figure 8 implemented by the four models, respectively, are more extensive than in Figures 6 and 7, so the temperature prediction of ϑ_{SW} reveals difficulty. The error ranges of LSTM and LSTM-2 in Figure 8 are about 0.5, while the error range of PSNLSTM is 0.33. Compared with them, the advantage of PSNLSTM is obvious. Moreover, PSNLSTM has a slight advantage over NLSTM, which is mainly reflected in the sudden change points of temperature.

The performances shown in Figures 6–8 are consistent with the results revealed in Tables 3–5. The weak advantage of PSNLSTM over NLSTM is mainly reflected in the mutation points of temperature, which is relatively obvious in the most difficult ϑ_{SW} temperature prediction. To a certain extent, it supports the structural feature of PSNLSTM, which consists of two NLSTMs with different time steps. However, this structure also improves the accuracy of temperature prediction while increasing a certain amount of parameters. In this study, the temperature prediction effect for the ϑ_{PM} component of the permanent magnet synchronous motor is not shown because the four deep learning models mentioned in the article are not ideal for the component. Both PSNLSTM and NLSTM have certain advantages over LSTM and LSTM-2 in temperature prediction. Compared with NLSTM, although PSNLSTM is slightly worse than NLSTM in ϑ_{ST} temperature prediction, there are many slight indications that PSNLSTM was slightly better than NLSTM overall.

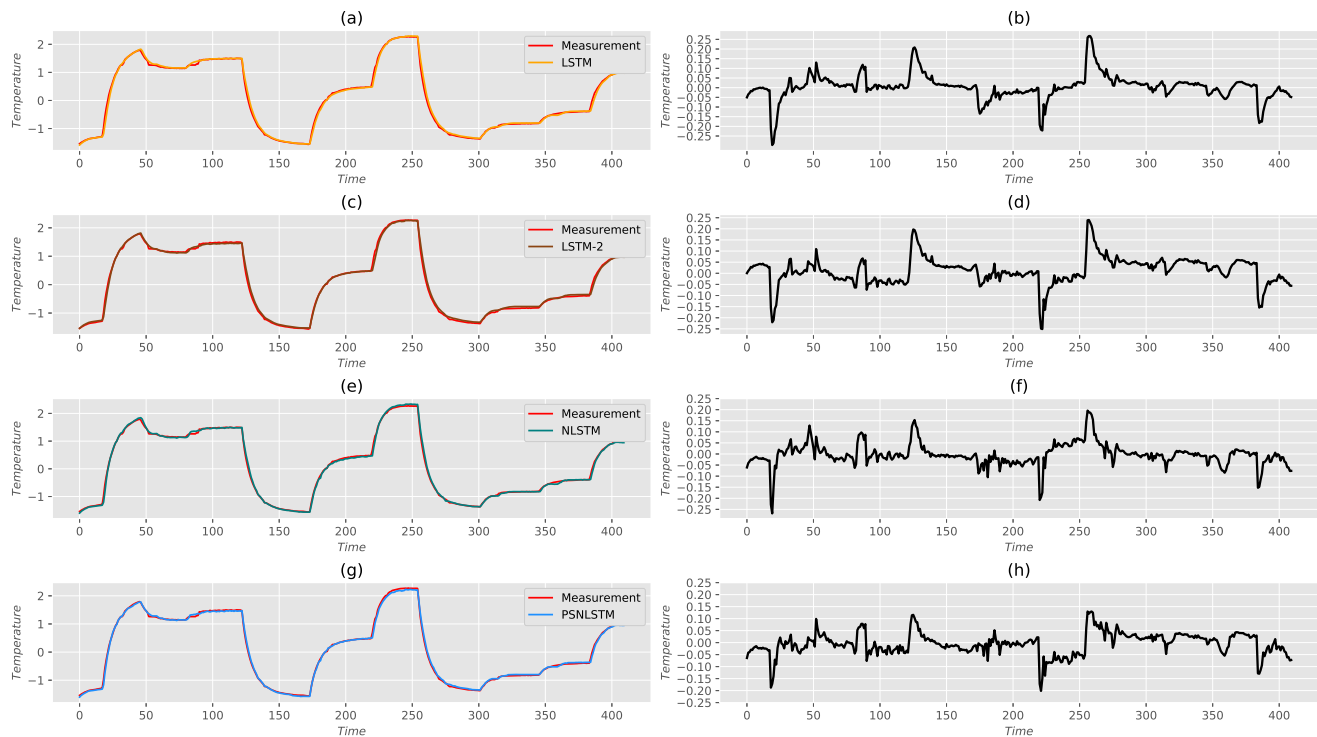


Figure 8. Temperature prediction of θ_{SW} for each model. (a,c,e,g) the temperature fitting curves of θ_{SW} for each model; (b,d,f,h) the temperature prediction error curves of the θ_{SW} for each model, which are obtained by subtracting the measured values from the predicted values. In addition, the error curves correspond to the left fitting curves, respectively.

6. Conclusions

This paper systematically expounds on the current research on the temperature prediction of permanent magnet synchronous motors. The NLSTM, a novel deep learning model, is chosen to predict the temperature of the PMSMs. Then, the PSNLSTM, which combines the NLSTM and Pseudo-Siamese networks, is proposed. The learning rate optimization algorithm of gradual warmup and cosine annealing with the adaptive optimizer Nadam is also proposed to accelerate the convergence of the model and to improve the prediction performance. Both the proposed model and optimization algorithm of learning rate are verified by the experiments. A slight improvement is detected concerning the comparison of PSNLSTM with NLSTM, while a clear advantage of both is indicated concerning LSTM and LSTM-2. More details will be confirmed in future studies, which will include more extended datasets.

Author Contributions: Conceptualization, Y.C. (Yuefeng Cen) and G.C.; methodology, Y.C. (Yongping Cai) and Y.C. (Yuefeng Cen); formal analysis, Y.C. (Yongping Cai) and C.Z.; writing—original draft preparation, Y.C. (Yuefeng Cen) and Y.C. (Yongping Cai); writing—review and editing, X.Y., C.Z., and Y.Z.; visualization, G.C. and X.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Foundation of China (Grant Nos. Nsf61902349 and Nsf61803337).

Acknowledgments: We are grateful to the Kaggle data science competition platform and the University of Paderborn in Germany for their dataset: <https://www.kaggle.com/wkirsngn/electric-motor-temperature> (accessed on 1 April 2021).

Conflicts of Interest: The authors declare that there is no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Fan, T.; Li, Q.; Wen, X. Development of a High Power Density Motor Made of Amorphous Alloy Cores. *IEEE Trans. Ind. Electron.* **2013**, *61*, 4510–4518. [[CrossRef](#)]
2. Wu, P.S.; Hsieh, M.F.; Cai, W.L.; Liu, J.H.; Huang, Y.T.; Caceres, J.F.; Chang, S.W. Heat Transfer and Thermal Management of Interior Permanent Magnet Synchronous Electric Motor. *Inventions* **2019**, *4*, 69. [[CrossRef](#)]
3. Guo, H.; Ding, Q.; Song, Y.; Tang, H.; Wang, L. Predicting Temperature of Permanent Magnet Synchronous Motor Based on Deep Neural Network. *Energies* **2020**, *13*, 4782. [[CrossRef](#)]
4. Zhu, Y.; Xiao, M.; Lu, K.; Wu, Z.; Tao, B. A simplified thermal model and online temperature estimation method of permanent magnet synchronous motors. *Appl. Sci.* **2019**, *9*, 3158. [[CrossRef](#)]
5. Habibinia, D.; Rostami, N.; Feyzi, M.R.; Soltanipour, H.; Pyrhönen, J. New finite element based method for thermal analysis of axial flux interior rotor permanent magnet synchronous machine. *IET Electr. Power. Appl.* **2019**, *14*, 464–470. [[CrossRef](#)]
6. Feng, G.; Lai, C.; Iyer, K.L.V.; Kar, N.C. Improved high-frequency voltage injection based permanent magnet temperature estimation for PMSM condition monitoring for EV applications. *IEEE Trans. Appl. Supercon.* **2020**, *30*, 1–5. [[CrossRef](#)]
7. Boglietti, A.; Cavagnino, A.; Staton, D.; Shanel, M.; Mueller, M.; Mejuto, C. Evolution and modern approaches for thermal analysis of electrical machines. *IEEE Trans. Ind. Electron.* **2009**, *56*, 871–882. [[CrossRef](#)]
8. Kral, C.; Haumer, A.; Lee, S.B. A Practical Thermal Model for the Estimation of Permanent Magnet and Stator Winding Temperatures. *IEEE Trans. Veh. Technol.* **2017**, *67*, 216–225. [[CrossRef](#)]
9. Qiao, G.; Wang, M.; Liu, F.; Liu, Y.; Zheng, P.; Sui, Y. Analysis of Magnetic Properties of AlNiCo and Magnetization State Estimation in Variable-Flux PMSMs. *IEEE Trans. Magn.* **2019**, *55*, 1–6. [[CrossRef](#)]
10. Wallscheid, O.; Huber, T.; Peters, W.; Böcker, J. A critical review of techniques to determine the magnet temperature of permanent magnet synchronous motors under real-time conditions. *EPE J.* **2016**, *26*, 11–20. [[CrossRef](#)]
11. Balamurali, A.; Kundu, A.; Clandfield, W.; Kar, N.C. Non-invasive parameter and loss determination in PMSM considering the effects of saturation, cross-saturation, time harmonics and temperature variations. *IEEE Trans. Magn.* **2020**, *57*, 8202206. [[CrossRef](#)]
12. Giangrande, P.; Madonna, V.; Nuzzo, S.; Spagnolo, C.; Gerada, C.; Galea, M. Reduced Order Lumped Parameter Thermal Network for Dual Three-Phase Permanent Magnet Machines. In Proceedings of the 2019 IEEE Workshop on Electrical Machines Design, Control and Diagnosis (WEMDCD), Athens, Greece, 22–23 April 2019; pp. 71–76.
13. Rostami, N.; Feyzi, M.R.; Pyrhonen, J.; Parviainen, A.; Niemela, M. Lumped-parameter thermal model for axial flux permanent magnet machines. *IEEE Trans. Electr. Power. Appl.* **2017**, *49*, 1178–1184. [[CrossRef](#)]
14. Yu, X.; Shi, S.; Xu, L.; Liu, Y.; Miao, Q.; Sun, M. A Novel Method for Sea Surface Temperature Prediction Based on Deep Learning. *Math. Probl. Eng.* **2020**, *2020*, 1–9. [[CrossRef](#)]
15. Hochreiter, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int. J. Uncertain. Fuzz.* **1998**, *6*, 107–116. [[CrossRef](#)]
16. Wallscheid, O.; Kirchgässner, W.; Böcker, 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.
17. Kirchgässner, W.; Wallscheid, O.; Böcker, J. Deep residual convolutional and recurrent neural networks for temperature estimation in permanent magnet synchronous motors. In Proceedings of the 2019 IEEE International Electric Machines and Drives Conference (IEMDC), San Diego, CA, USA, 11–15 May 2019; pp. 1439–1446.
18. Moniz, J.R.A.; Krueger, D. Nested lstms. In Proceedings of the Ninth Asian Conference on Machine Learning (ACML2017), Seoul, Korea, 15–17 November 2017; pp. 15–17.
19. Li, Y.; Yu, Z.; Chen, Y.; Yang, C.; Li, Y.; Allen, L.X.; Li, B. Automatic Seizure Detection using Fully Convolutional Nested LSTM. *Int. J. Neural Syst.* **2020**, *30*, 2050019. [[CrossRef](#)]
20. Ma, X.; Zhong, H.; Li, Y.; Ma, J.; Cui, Z.; Wang, Y. Forecasting transportation network speed using deep capsule networks with nested lstm models. *IEEE Trans. Intell. Transp.* **2020**, *99*, 1–12. [[CrossRef](#)]
21. Roy, S.K.; Harandi, M.; Nock, R.; Hartley, R. Siamese networks: The tale of two manifolds. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV2019), Seoul, Korea, 27 October–2 November 2019; pp. 3046–3055.
22. Hughes, L.H.; Schmitt, M.; Mou, L.; Wang, Y.; Zhu, X.X. Identifying corresponding patches in SAR and optical images with a pseudo-siamese CNN. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 784–788. [[CrossRef](#)]
23. Pontes, E.L.; Huet, S.; Linhares, A.C.; Torres-Moreno, J.M. Predicting the semantic textual similarity with siamese CNN and LSTM. *arXiv* **2018**, arXiv:1810.10641.
24. Wallscheid, O.; Böcker, J. Global identification of a low-order lumped-parameter thermal network for permanent magnet synchronous motors. *IEEE Trans. Energy Convers.* **2015**, *31*, 354–365. [[CrossRef](#)]
25. Goyal, P.; Dollár, P.; Girshick, R.; Noordhuis, P.; Wesolowski, L.; Kyrola, A.; Tulloch, A.; Jia, Y.; He, K.J. Accurate, large minibatch sgd: Training imagenet in 1 h. *arXiv* **2017**, arXiv:1706.02677.
26. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.
27. Kingma, D.P.; Adam, B.J. A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.

28. Reddi, S.J.; Kale, S.; Kumar, S. On the convergence of adam and beyond. *arXiv* **2019**, arXiv:1904.09237.
29. Dozat, T. Incorporating nesterov momentum into adam. In Proceedings of the Workshop track at International Conference on Learning Representations (ICLR2016), San Juan, Puerto Rico, 2–4 May 2016.