

Article

A Long-Term Prediction Method of Computer Parameter Degradation Based on Curriculum Learning and Transfer Learning

Yuanhong Mao , Zhong Ma, Xi Liu, Pengchao He and Bo Chai

Xi'an Microelectronics Technology Institute, Xi'an 710065, China

* Correspondence: yuanhongmao@hotmail.com; Tel.: +86-158-2910-1891

Abstract: The long-term prediction of the degradation of key computer parameters improves maintenance performance. Traditional prediction methods may suffer from cumulative errors in iterative prediction, which affect the model's long-term prediction accuracy. Our network adopts curriculum learning and transfer learning methods, which can effectively solve this problem. The training network uses a dual-branch Siamese network. One branch intermixes the predicted and annotated data as input and uses curriculum learning to train. The other branch uses the original annotated data for training. To further align the hidden distributions of the two branches, the transfer learning method calculates the covariance matrices of the time series of the two branches by correlation alignment loss. A single branch is used in the test for prediction without increasing the inference computation. Compared with the current mainstream networks, our method can effectively improve the accuracy of long-term prediction with the improvements above.

Keywords: long-term prediction; time series; curriculum learning; transfer learning

MSC: 62P30



Citation: Mao, Y.; Ma, Z.; Liu, X.; He, P.; Chai, B. A Long-Term Prediction Method of Computer Parameter Degradation Based on Curriculum Learning and Transfer Learning. *Mathematics* **2023**, *11*, 3098. <https://doi.org/10.3390/math11143098>

Academic Editor: Paweł Kliber

Received: 19 June 2023

Revised: 9 July 2023

Accepted: 11 July 2023

Published: 13 July 2023



Copyright: © 2023 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

In recent decades, embedded computers in aviation and aerospace systems have become more integrated and intelligent. Because of their complex structures, embedded computer systems' development, production and maintenance costs are increasing. The functions of computers in working and storage conditions are subject to gradual degradation under the prolonged effects of stress factors such as heat, humidity, and vibration, which may eventually cause functional failure. During the maintenance of embedded computers, predicting the degradation trend of these critical functional parameters is desirable. In this way, maintenance personnel can monitor the health status of the products in real time and perform timely maintenance before a failure occurs, thereby reducing downtime and maintenance costs. Hence, the degradation analysis of electronic systems based on critical parameters has emerged as a research hotspot for scholars.

The reliability of computers and circuits deployed on aerospace equipment should be the most important factor in maintenance work. It is necessary to predict long-term computer degradation. Algorithms should predict computer failure due to occur within a few weeks or longer to facilitate early component replacement. Therefore, the long-term monitoring and prediction of key parameters of aerospace computers and their circuits can improve product reliability more effectively. For example, the flight control computers equipped in unmanned aerial vehicles (UAV) are subject to accelerated degradation in humid and hot climates and eventually fail to function. The power consumption and resistance of the entire computer rise significantly before its failure, which shows a noticeable degradation process. The timely repair and replacement of electronic components can improve the UAVs' operational capability.

The theoretical research and application of computer degradation are in the exploratory stage. In the research process, scholars identify the key parameters that influence the computer’s function by analyzing the weak part of the computer. They perform failure mode and mechanism analyses based on these parameters. So, the reliability of the entire electronic system is simplified by studying the degradation of several key characterization parameters. Many studies have conducted various explorations in this direction following this idea. Mao et al. [1] collected data on the key parameters of embedded computers during temperature-accelerated aging and evaluated real-time input data at normal working temperatures. They used the real input data under storage to update the acceleration factor and reasonably estimate its current degradation trend.

The current mainstream research methods mainly include physical-failure-model-based, mathematical–statistical-model-based, and machine learning-based approaches. A brief description of the methods and their advantages and disadvantages is presented in Table 1. With the development of deep learning networks, data-driven methods have become a hot research topic. These methods can handle complex prediction problems that are difficult to describe with physical or statistical models.

Table 1. Mainstream electronics product degradation prediction methods.

Method	Characteristics	Advantages	Disadvantages
Approaches based on a physical model [2–7]	Applicable to equipment with a clear degradation mechanism and weak generalization ability.	The accuracy is higher because of the precise description of degradation mechanisms and impact factors.	<ul style="list-style-type: none"> • Requires significant expertise and experience. • Poor generalization for specific devices. • Difficult to obtain models for some overly complex scenarios.
Statistics approaches [8–13]	Build a statistical model based on empirical knowledge and present the degradation predictions as a probability density function based on the statistical results.	Not dependent on specific failure physics, a variety of more mature models are available	<ul style="list-style-type: none"> • Real data with a large number of degraded samples are required. • Scenarios that are too complex remain challenging to process.
Machine learning approaches [14–21]	A data-driven approach to learning mechanical degradation patterns from existing observational data rather than building physical or statistical models.	Strong model expression, with excellent prediction for complex models.	<ul style="list-style-type: none"> • Need to collect a large number of samples. • Poor interpretability of prediction results.

Essentially, the long-term degradation data of key computer parameters are time series data. Time series data analysis has extensive applications in finance, meteorology, agriculture, industry, and medicine [22]. Particularly in recent years, with the advancements in sensor and network technology, maintenance personnel can more easily collect key computer parameters automatically at regular intervals. It means that a significant amount of time series data of key parameters are available now. Computer parameter degradation should be a long-term time series prediction problem. Researchers can adopt time series methods to deal with it. These time series methods can extract patterns from past degradation trend data and forecast future development trends. A critical computer parameter degradation method based on time series analysis will offer guidance for maintenance work and has high academic significance.

Predicting computer parameter degradation is challenging, and differs significantly from traditional time series forecasting methods. Standard time series prediction focuses on the correspondence between the annotated inputs and the expected outputs. It optimizes the model by minimizing the difference between the model’s output values and the annotated values. However, long-term computer degradation prediction needs existing data to forecast future trends iteratively. The prediction model uses the predicted value y at time t as input for the next time $t + 1$ during the test. In this way, the iterative inputs

result in the prediction of more steps in the future. This iterative approach theoretically transforms a single-step model into a multi-step prediction model. However, there is inevitably some error in the predicted values at each step. The prediction model uses these previous predictions as input for the next step, which can lead to an accumulation of errors over long iterations. Therefore, excessively long prediction steps can lead to a poor prediction accuracy of the model, which should be appropriately addressed.

The network employs a Siamese architecture network, where one branch is trained with annotated values and the other with a curriculum learning approach. This curriculum learning method combines ground-truth and predicted values as next-step inputs to train the model, gradually increasing the proportion of predicted values as inputs to reduce error accumulation. The distance between the covariance matrices of the output prediction sequences from the two branches is compared during training. This transfer learning approach can reduce errors between iterative and ground-truth training.

This paper proposes the curriculum and transfer learning methods to address the challenges of long-term computer parameter degradation prediction. Our contributions are as follows:

- The network employs a Siamese architecture network, where one branch is trained with annotated data and the other with a curriculum learning approach. The curriculum learning branch combines annotated data and predicted output as the next-step inputs to train the model. The proportion of predicted values in the inputs gradually increases during the training process to avoid accumulative prediction error.
- During the training phase, the network output trained with the ground-truth data is regarded as the source domain, while the network is trained iteratively with the predicted value as the target domain. Correlation alignment (CORAL) loss can facilitate time series features to align the covariance between domains. Our network incorporates CORAL loss within deep networks to learn a nonlinear transformation to align source and target domain distributions within feature space.

These improvements enable higher accuracy in the long-term iterative prediction of key computer parameters, providing more time for subsequent maintenance work. Our paper is organized as follows: The related works are discussed in Section 2. We propose our method of long-term computer parameter degradation estimation in Section 3. Section 4 presents the experiments with our datasets. We conclude with a summary in Section 5.

2. Related Works

This section discusses the existing electronic product degradation prediction methods, time series forecasting, and analysis methods.

2.1. Methods in Electronic Product Degradation Prediction

Physical-failure-based models describe the degradation mechanism of electronic systems. The degradation parameters of the physical model are related to the material properties and stress levels. They are identified by comparative experiments or finite element analysis.

The University of Maryland CALCE Center proposed the life consumption monitoring (LCM) methodology for electronic devices [3]. The LCM method combines the monitoring of environmental and operational stresses with the physical model of failure in electronic devices to calculate the cumulative damage and predict the product prediction. Renwick [4] obtained the condition degradation pattern of capacitor devices by monitoring the electrical stress and performed failure prediction analysis. Rana [5] developed physical failure models for different electronic components and completed a life prediction study. Liu [6] proposed a reliability assessment based on integrating highly accelerated life testing and accelerated degradation testing. Rockwell [7] used early warning circuits embedded in the product for early fault diagnosis for welded and corroded parts with low cycle fatigue characteristics.

Statistical-model-based methods, also known as empirical-model-based methods, present degradation predictions as probability density functions by building statistical models based on empirical knowledge and presenting the probability density functions based on statistical results. Many kinds of statistical models are commonly used, such as the auto-regressive model, the Wiener process model, the Weibull process model, the Gaussian process model, the Markov model, etc. Kim [8] proposed a state-of-health prediction for lithium-ion batteries with the Seasonal ARIMA (SARIMA) and auto-regressive integrated moving average with exogenous variables (ARIMAX) models. Li [9] addressed the degradation prediction of electronic products based on the Wiener degradation process model and Bayesian posterior estimation to realize real-time updates of parameters. Wei [10] applied the Weibull distribution to model the distribution of each key component in a complex electronic system. It dynamically depicted the reliability changes in the entire system under the stress impacts. Wan [11] proposed a stochastic model of thermal reliability analysis and prediction for a whole electronic system based on the Markov process to estimate the thermal reliability of an electronic system. Wang [12] proposed a generalized Gaussian process to construct a one-stage maximum-likelihood method for parameter estimation for degradation procedures. Shi [13] proposed a method for forecasting the remaining life of a multi-component computer based on Copula theory. Each component's degradation distribution function was deduced from kernel density estimation and generated the remaining life prediction model with their correlation.

Data-driven approaches use artificial intelligence techniques to learn patterns of electronic degradation from existing data rather than building physical or statistical models. They can handle complex prediction problems that are difficult to describe with physical or statistical models. With the continuous development of deep learning, data-driven approaches are receiving more and more attention in electronic device degradation prediction. Fan [14] utilized the existing constant-stress accelerated test data in storage conditions to assess the computer degradation trend using the support vector machine method. Jiang [15] proposed a reliable cycling aging prediction based on a data-driven model to address the urgent issue of the adaptive and early prediction of lithium-ion batteries' remaining useful life. Li [16] introduced a deep learning-based battery health prognostics approach to predict the future degradation trajectory in one shot without iteration or feature extraction. Zhao [17] constructed a probabilistic degradation prediction framework to estimate the probability density of target outputs based on parametric and non-parametric approaches. The method can naturally provide a confidence interval for the target prediction. Deng [18] adopts a new multi-scale dilation convolution fusion unit with different dilated factors for remaining useful life (RUL) prediction. Liu [19] proposed a novel fault diagnostic application of the Gaussian–Bernoulli deep belief network (GB-DBN) for electronics-rich analog systems, which can more effectively capture high-order semantic features from the raw output signals.

2.2. The Forecasting and Analysis Methods of Time Series

This paper employs a time series approach to address the degradation of crucial computer parameters and focuses on predictive analysis methods for time series. Traditional methods determine the time series parameter model and solve model parameters to complete the prediction. Typical methods include ARIMA (Auto-Regressive Integrated Moving Averages) [23] and the Holt–Winters method. While traditional time series models can solve simple prediction problems, they may only suffice if the time series contains fewer variables and dimensions or if the change patterns are overly complex.

Time series decomposition is a valuable method in time series analysis. This approach assumes that a time series is the result of the superposition or coupling of long-term trends (Secular trend, T), seasonal variations (Seasonal Variation, S), cyclical fluctuations (Cyclical Variation, C), and irregular fluctuations (Irregular Variation, I). A typical method in this approach is Prophet [24].

Time series data prediction is closely related to regression analysis in machine learning. Support Vector Regression (SVR) applies SVM to time series function regression by mapping x to a high-dimensional feature space using a nonlinear transformation and estimating the time series. Gradient Boosting Regression Tree (GBRT) introduces gradient descent [25] to solve regression problems by calculating the negative gradient of the loss function to minimize it and ultimately obtain the optimal model.

With the tremendous achievements of deep learning methods in computer vision and natural language processing, deep learning approaches have gradually been introduced into time series prediction applications. By constructing various network architectures, deep neural networks can better represent high-dimensional data, avoiding manual feature engineering and model design. Deep learning facilitates end-to-end training by optimizing a predefined loss function.

Convolutional neural networks (CNNs) extract local features in the time dimension and gradually aggregate them through multiple layers to obtain hidden information from past time sequences. Borovykh [26] constructed a stacked structure of dilated convolutions in the network to aggregate more historical records for future time series prediction. Chen [27] proposes a temporal convolutional network (TCN) that treats the sequence as a one-dimensional input and captures long-term relationships through iterative multi-layer convolution. The TCN uses causal dilated convolutions [28] and residual convolution skip connections to provide an extensive temporal receptive field for modeling.

Recurrent neural networks (RNNs) learn the hidden states within all time series before prediction, serving as feature representations of past information. Combined with current inputs, they provide the next step prediction. Long Short-Term Memory (LSTM) [29] and its variant Gate Recurrent Unit (GRU) [30] are essential for RNN-based time series prediction. The DeepAR network [31] employs an LSTM model to solve time series prediction problems. Its previous expected output is used as input to replace the input of the next moment in the prediction phase. Rangapuram [32] propose an RNN-based deep state space model that considers the current state to be related only to the last moment. Wen [33] used an encoder–decoder structure to forecast the multi-horizon time steps in the future simultaneously.

Self-attention networks, originating in natural language processing, can be quickly applied to time series prediction due to the similarity of these tasks. RNNs must sequentially aggregate all the hidden information from $t-n$ to t when processing and analyzing time series. The connection between each time step usually needs to be improved. RNNs are less efficient because of their sequential processing procedure. The model based on the attention mechanism can realize the association between any units in the input time series. Association weighting provides representations in the lower layer features to the upper layer. The self-attention mechanism can better realize the contextual information interaction in time series. Wu [34] tried the time series prediction task using the Transformer architecture with good results. The Transformer model has great potential to improve prediction performance. However, Transformer also has limitations such as high computational effort, high memory footprint, and an encoder–decoder architecture, which prevent it from being applied to more prolonged time series prediction problems. The Informer [35] selects $O(\log L)$ -dominant queries based on query–key similarity, achieving a similar improvement in computational complexity to LogTrans. Researchers have explored frequency-domain self-attention mechanisms in time series modeling. Autoformer [36] designs a short-term trend decomposition architecture with an autocorrelation mechanism as an attention module. It measures the delay similarity between input signals and aggregates the top k similar subsequences to produce an output with $O(N \log N)$ complexity. FEDformer [37] applied the attention mechanism in the frequency domain using the Fourier transform and wavelet transform. It achieved linear complexity by randomly selecting a subset of fixed-size frequencies. Li [38] proposed a combination of CNN and Transformer to enhance contextual information extraction. Lim [39] used LSTM to model the input sequence as a preprocessing step and fed it into the upper transformer to compensate for the sequence information using attention.

For all neural networks, the self-attention method currently performs the best in time series prediction because it can pay more attention to the historical context of time series on a larger scale. Although the algorithms above have improved prediction accuracy for public time series datasets, reducing iterative error is more critical for long-term prediction in real applications. From another perspective, the application of time-series-based time networks is also limited by the limitations of large-scale data. For scenarios with limited sample data, this can also affect performance.

3. Our Approach

3.1. Overview of Our Network Architecture

Our network employs a dual-branch structure during training, as shown in Figure 1. y_i is the real output at i step in the collected dataset, which is annotated in the training process. \hat{y}_i is the prediction output at i step, which may be regarded as an iterative input at $i + 1$ step. One branch uses the curriculum learning training method and combines annotated and predicted values as inputs to train the model. The proportion of predicted value inputs gradually increases during the training process. The other branch always uses annotated data for training to preserve the real data distribution. To ensure better fits of the actual data distribution in the curriculum learning training, a transfer learning approach compares the distance between the covariance matrices of the output sequences from the two branches. Deep transfer learning confuses the distribution between the hidden layers of the two branches to make the data distribution using prediction from real values as similar as possible to that when using a prediction from the predicted value.

During the end-to-end optimization process, the Mean Squared Error (MSE) loss function compares the output time series from each branch to their respective annotated data. In addition, the correlation alignment (CORAL) loss between the two branches is evaluated based on the covariance distance between their output sequences.

We adopted the dual-branch network architecture just during the training. During testing, the data are fed into a single trained network branch. The output from the previous step should be regarded as input for the subsequent step.

During the loss function statistics, the time series of the outputs from the two branches are optimized with the ground truth as the loss functions, respectively. In addition, the correlation alignment (CORAL) loss between the two branches is evaluated in terms of the covariance distance according to the corresponding two output time sequences.

It should be noted that this network architecture is only used for the training process. During the testing process, only one network branch works appropriately. One step's output should be used as input for the next step in the testing process.

The LSTNET [40] structure is the backbone of our Siamese network architecture due to its effectiveness in accounting for the accumulation of linear trends and periodic fluctuations in time series data. Its hypothesis is consistent with the changing trends required in the current degradation process of key computer parameters. We chose the LSTNET structure for the network architecture because it can better account for the accumulation of linear and periodic fluctuations in the degradation trend.

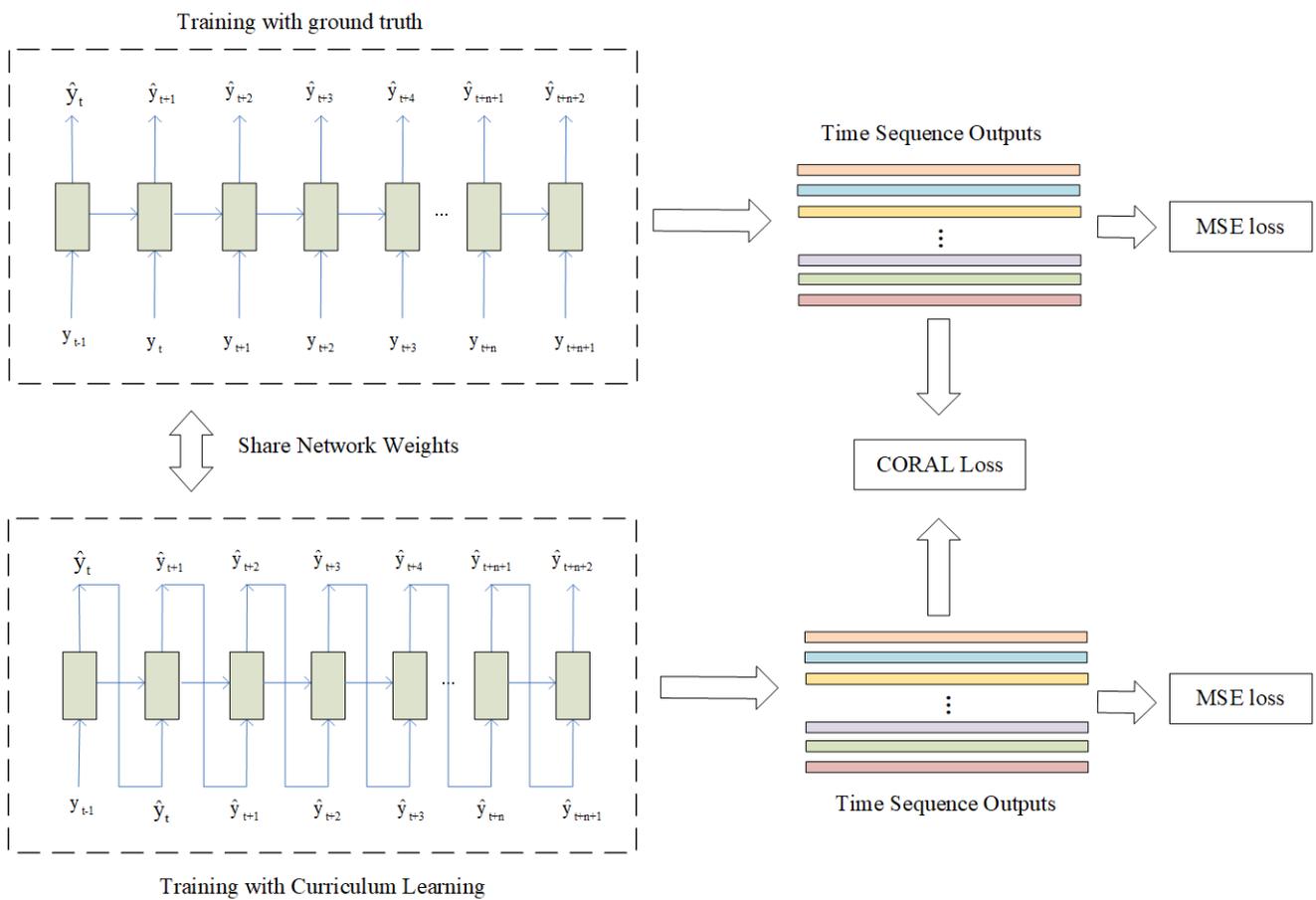


Figure 1. Overall network architecture.

3.2. Curriculum Learning Iterative Training in Degradation Prediction

The iterative training methodology employed in the curriculum learning branch introduces an exposure bias into time series forecasting. During training, the time series forecasting network can predict future steps based on currently available data. The predicted outputs from the network are compared with their respective annotated data. The network parameters are updated accordingly based on the difference between them. However, during subsequent training iterations, the predicted outputs from previous steps are utilized as input data in the following steps. This iterative forecasting approach can result in the accumulation of errors.

The traditional deep learning-based time series methodologies exhibited limited predictive capabilities during the initial stages of training. Generating consistently accurate forecasts has yet to be achieved. When a significant deviation occurs at a particular step, the biased output value as input data will exacerbate subsequent forecast inaccuracies. It can result in insufficient network convergence and poor forecast accuracy during training.

Curriculum learning adopts a structured learning methodology, just like the human learning process. Initially, an instructor imparts knowledge onto the student and waits until the student has acquired a foundational understanding before allowing them to engage in independent learning. The curriculum learning branch’s input data combines the previously predicted output and annotated data. As illustrated in Figure 2, during the training phase, the network ceases to rely solely on predicted outputs as network inputs for subsequent steps. Instead, it selects its initial steps output with probability p and the annotated input with probability $1 - p$ as its input for the next steps. Throughout this scheduled sampling process, the value of p fluctuates between 0 and 1. When the network needs still more development at the beginning of training, a smaller value of p is optimal, resulting in a higher proportion of annotated data being utilized in input samples

to train the network. As training progresses and the model matures, p should incrementally increase to maximize the use of the model’s own predicted output as the iterative input during training. This gradual learning approach enables the model to adapt to training with expected outputs as cyclic inputs while maintaining prediction accuracy.

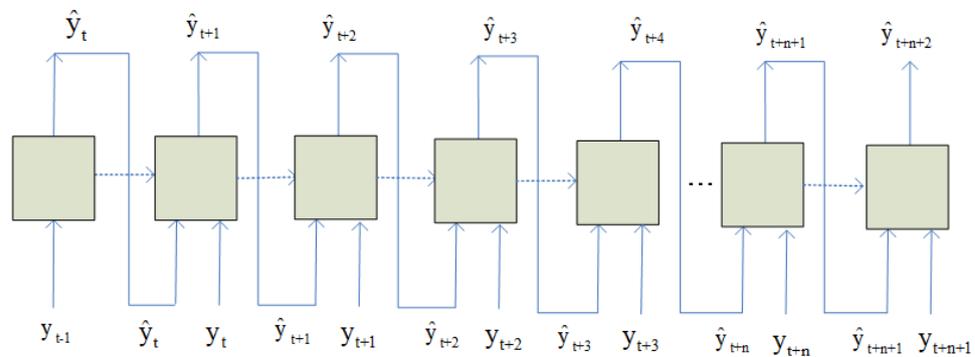


Figure 2. Curriculum learning iteration training.

During the training phase, the predicted output is selected with probability p while the annotated data are selected with probability $(1 - p)$. The value of p ranges from 0 to 1. As the number of training iterations increases, p incrementally increases from 0 to 1. p should be set to 1 at the end of training.

3.3. Domain Adaptation between Two Branches

The iterative errors will inevitably be cumulated when training the network with predicted values. During iterative training, there is a discrepancy between the distributions of predicted data and the distribution of annotated data. More alignment of their feature spaces is essential to diminish this discrepancy. Domain transfer learning can accomplish this goal.

During the training phase, the network output trained with the ground-truth data is regarded as the source domain, while the network is iteratively trained with the predicted value as the target domain. Throughout the learning process, synchronizing the output of both sub-networks to the corresponding time series keeps their sequence alterations consistent in the time dimension. The similarity between both branches is determined by measuring the distance between their output sequence covariances to evaluate two time series.

Covariance quantifies how each dimension deviates from its mean value. The covariance matrix can effectively measure the sequence correlation over time dimensions for time series data. The covariance matrix assesses the correlation of changes across the periods within the time series and serves as a representation of internal correlation. Comparing the correlation of two time series can evaluate their relationship to learn their joint distribution.

Our network adopts the CORAL loss function to measure the distance between two distributions. CORAL loss can facilitate feature transformation to align the covariance between different domains. Our network incorporates CORAL as a loss function within deep networks to learn a nonlinear transformation to align source and target domain distributions within feature space. The covariance matrices for the source and target domains are calculated as shown in Formulas (1)–(3). The covariances for the source and target domains are represented as C_s and C_t . Here, D_s and D_t are the feature presentation of source and target data, respectively. The covariance of the source domain and target domain are indicated as C_s and C_t , which are calculated with the equations below. N_s and N_t represent the sequence length in the source and target domains, respectively.

$$C_s = \frac{1}{N_s - 1} (D_s^T D_s - \frac{1}{N_s} (1^T D_s)^T (1^T D_s)) \tag{1}$$

connected by bus. The AD input and DA output accuracy significantly impact the flight control embedded computers. During the flight, the analog signal of the rotation angle of the servo is input to the computer through A/D conversion. The computer calculates and transfers the following rotation action to the servo through D/A conversion. If this part of the component degrades, it will significantly impact flight accuracy. Therefore, this function belongs to the key of the flight control computer and needs more attention.

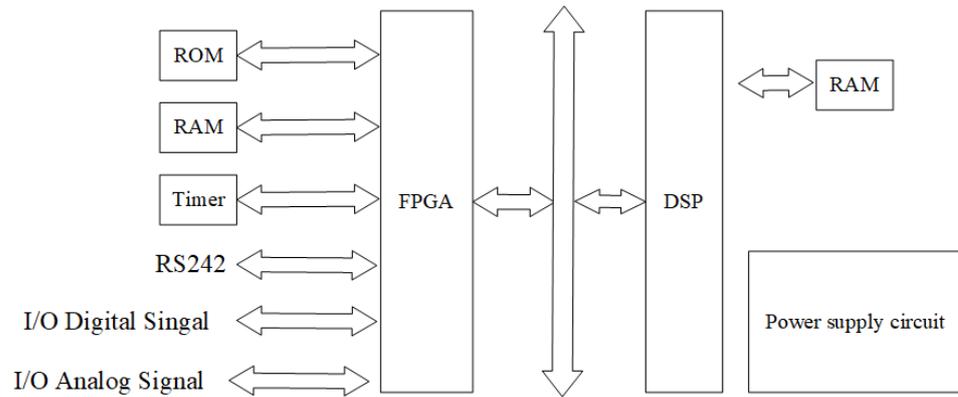


Figure 4. The simplified embedded computer architecture diagram.

From the data analysis of the natural storage environment, accelerated experiment, and the long-term power-on test of the computer, it is found that the parameters of the computer’s A/D- and D/A-related circuits degrade more obviously with increased serving time. Therefore, their degradation mode greatly influences the degradation of computer function. The changes in the critical parameters of the embedded computer were statistically analyzed by day. We collected the data in this way and constructed the time series datasets.

The aircraft’s flight state is controlled with the airfoil servos adjustment. There is a constant data exchange between the flight control computer and the vehicle’s servos. The flight control computer is constantly monitoring the changes in the analog model of the control Totoro and simultaneously converting the adjusted digital quantities into analog outputs for the Totoro. Therefore the degradation of A/D and D/A in the computer affects the accuracy of the vehicle.

4.2. Implementation Details

The hardware platform for the experiment was Nvidia Tesla V100 GPU. The software platform uses PyTorch 3.7 for training and testing. In the experiment, the LSTNET network was used as the backbone network. In the process of training, the collected data above were used. The network adopts the Siamese structure. The two branches share the network weight. The ground truth was used in training to input and calculate the loss function. The other branch used the curriculum learning method to learn. Excellent performance can be achieved by gradually increasing the proportion of predicted output as the iterative input. The batch size was 256 in training. The gradient descent algorithm adopts the Adam algorithm. The learning rate of the network decreases from 10^{-3} to 10^{-5} according to the number of iterations. The number of iterations was 5000.

4.3. Experiment Metric

For time series forecasting, the difference between the predicted and annotated values is mainly measured by MSE (Mean Squared Error) and MAE (Mean Absolute Error). The equations are shown below, where N is the steps of the prediction, y_i is the ground truth, and \hat{y}_i is the predicted value.

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

$$MAE = \frac{\sum_{i=1}^N |\hat{y}_i - y_i|}{N} \tag{5}$$

4.4. Experiment Result

We compared the results with the current mainstream networks developed in recent years. As the number of prediction steps grows, our algorithm performs better than other methods. Table 2 shows univariate results with different prediction lengths, including 96, 192, 336, 720 for key computer parameter datasets. A lower MSE or MAE indicates a better prediction.

Table 2. Experiment Results.

Models	Our Model		Autoformer [36]		Informer [35]		Reformer [41]		Transformer [42]		LSTNET [40]		LSTM [43]	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
96	0.052	0.159	0.086	0.219	0.088	0.223	0.114	0.259	0.104	0.251	0.121	0.268	0.243	0.357
192	0.091	0.214	0.117	0.245	0.158	0.301	0.195	0.349	0.206	0.365	0.240	0.381	0.449	0.476
336	0.136	0.259	0.227	0.369	0.251	0.398	0.273	0.412	0.282	0.439	0.348	0.462	0.817	0.753
720	0.262	0.335	0.332	0.465	0.420	0.519	0.438	0.457	0.462	0.527	0.507	0.541	1.017	1.184

As shown in Table 2, our method exhibits better accuracy because our model adopts the transfer learning method. During the training process, this enables the iterative prediction sequence to achieve distribution alignment with the real sequence, thereby more effectively eliminating prediction error accumulation compared to previous methods.

Our method’s advantages are further demonstrated in Figure 5. The long-term prediction process of our method can effectively use the iterative output for the next prediction, which is more accurate than other methods. In Figure 5, the Autoformer network represents the Transformer methods, and the LSTM network represents the RNN methods. The blue line represents the ground truth. The red line represents the output prediction.

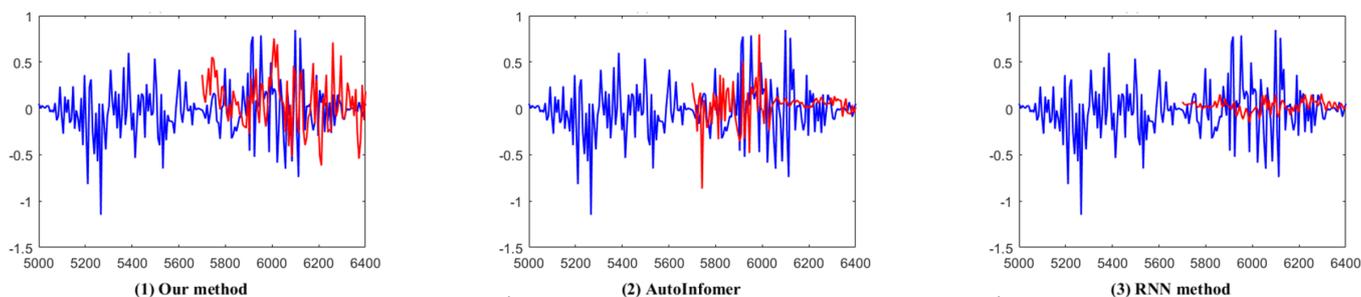


Figure 5. The prediction result comparison.

4.5. Ablation Studies

To validate the impact of curriculum learning and transfer learning methods on our model, we performed ablation experiments to compare their effects on long-term prediction. In Table 3, curriculum learning and transfer learning can separately improve the accuracy of the network with 720 prediction steps. A lower MSE or MAE indicates a better prediction. The combination of the two approaches can produce better results.

Table 3. Ablation experiment.

Method	MSE	MAE
Baseline	0.507	0.541
Curriculum learning	0.351	0.407
Our method (curriculum learning + transfer learning)	0.262	0.335

4.6. Different Input Substitution Schemes

Our experiments demonstrated the influence of different p substitution schemes during curriculum learning. We compared the linear, convex, and concave substitution functions, as shown in Figure 6. In the convex function, a percentage of p changed faster in the early training stages, whereas p changed faster in the later stages in the concave function. The change in p is constant with linear function during the training.

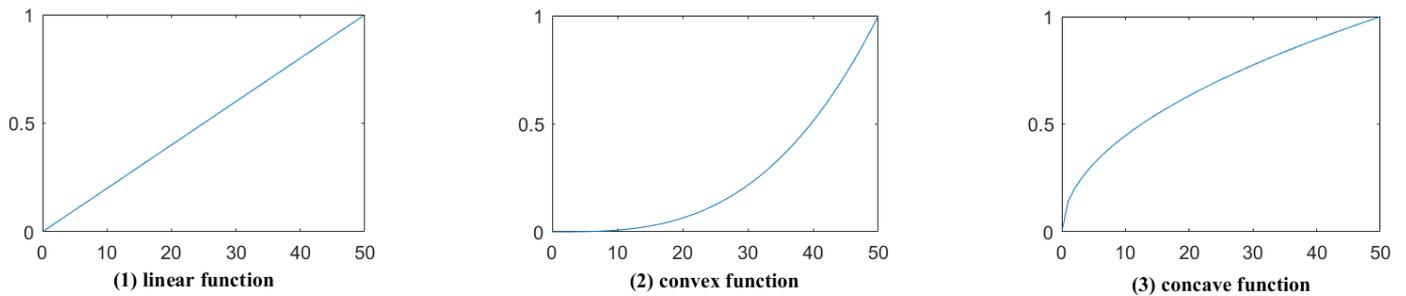


Figure 6. The p proportion different change function schemes.

It is evident from the tests that too high a rate of change in p in the early iterations may lead to lower performance, as shown in Table 4. A lower MSE or MAE indicates a better prediction. Too high a prediction input proportion in the early training stages may lead to unstable convergence in the model.

Table 4. Experiments with p proportion schemes.

p Change Scheme	MSE	MAE
linear function	0.262	0.335
convex function	0.264	0.349
concave function	0.355	0.391

4.7. Influence of Different Queue Lengths

The experiments evaluated the effect of different-length covariance matrices on the CORAL loss function. The two first-in-first-out branch queues were compared according to different queue length settings. Table 5 shows that the 512 lengths can achieve better results. A lower MSE or MAE indicates a better prediction. Too long queues mean more computation and no significant increase in performance.

Table 5. Experiments with different queue length.

Queue Length	MSE	MAE
128	0.371	0.435
256	0.305	0.398
512	0.262	0.335
1024	0.268	0.347

5. Conclusions

With the development of networks and sensor technology, it is possible to collect datasets more efficiently. It is possible to employ data-driven computer degradation analysis. Our paper uses a combination of curriculum learning and transfer learning to effectively reduce cumulative errors in the prediction process for long-term prediction scenarios of key computer features. We propose a Siamese network architecture oriented towards correlation alignment. During training, one branch uses annotated data, while the other uses curriculum learning in iterative prediction. Moreover, the correlation of the time series generated by the two branches is measured by optimizing the CORAL loss function, which realizes the alignment of the prediction time series distribution. Compared to the

time series prediction methods developed in recent years, our approach can effectively address long-term prediction in embedded computers. The maintenance and reliability of electronic devices can be improved.

Current time series networks based on deep learning rely on complete datasets, but many applications are challenging to achieve. We need to preprocess to impute the missing value for incomplete data. This can be performed for simple cases by interpolating before and after values. For complex cases, the missing parts can be generated by using self-encoders and adversarial networks.

Author Contributions: Y.M.: Conceptualization, Methodology, Software, Formal Analysis, and Writing. Z.M.: Data Curation and Resources. X.L.: Supervision. P.H.: Project Administration. B.C.: Conceptualization. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Innovation Fund of Xi'an Microelectronics Technology Institute under Grant No.YL-220009.

Data Availability Statement: Data are not available due to commercial restrictions.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following notations are used in this manuscript:

p	The probability of the iterative prediction input
y_i	The annotated output at i step
\hat{y}_i	The prediction output at i step
D_s	The feature presentation of source domain
D_t	The feature presentation of target domain
C_s	The covariance matrices of source domain
C_t	The covariance matrices of target domain
N_s	The length of the sequence in the source domain
N_t	The length of the sequence in the target domain
d	The size of the covariance matrices is $(d \times d)$

References

- Mao, Y.; Ma, Z.; Gao, S.; Li, L.; Yuan, B.; Chai, B.; He, P.; Liu, X. A Method of Embedded Computer Degradation Trend Prediction. In Proceedings of the 2022 5th International Conference on Pattern Recognition and Artificial Intelligence (PRAI), Chengdu, China, 19–21 August 2022; pp. 1338–1343. [\[CrossRef\]](#)
- Liu, L.; Lin, Z.; Jiang, S. Progress in Research and Application of Failure Physical Model for Electronic Products. *Equip. Environ. Eng.* **2015**, *20*, 1–10.
- Ramakrishnan, A.; Pecht, M.G. A life consumption monitoring methodology for electronic systems. *IEEE Trans. Components Packag. Technol.* **2003**, *26*, 625–634. [\[CrossRef\]](#)
- Renwick, J.; Kulkarni, C.S.; Celaya, J.R. Analysis of Electrolytic Capacitor Degradation under Electrical Overstress for Prognostic Studies. In Proceedings of the Annual Conference of the PHM Society, Beijing, China, 21–23 October 2015.
- Rana, Y.S.; Banerjee, S.; Singh, T.; Varde, P.V. Experimental program for physics-of-failure modeling of electrolytic capacitors towards prognostics and health management. *Life Cycle Reliab. Saf. Eng.* **2017**, *6*, 179–185. [\[CrossRef\]](#)
- Liu, K.; Lv, C.; Dang, W.; Li, L.; Zou, T.; Li, P. Research on reliability assessment of space electronic products based on integration of highly accelerated life test and accelerated degradation test. In Proceedings of the 2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Bali, Indonesia, 4–7 December 2016; pp. 1651–1654. [\[CrossRef\]](#)
- Liu, Z.; Jia, Z.; Vong, C.M.; Han, J.; Yan, C.; Pecht, M. A Patent Analysis of Prognostics and Health Management (PHM) Innovations for Electrical Systems. *IEEE Access* **2018**, *6*, 18088–18107. [\[CrossRef\]](#)
- Kim, S.; Lee, P.Y.; Lee, M.; Kim, J.; Na, W. Improved State-of-health prediction based on auto-regressive integrated moving average with exogenous variables model in overcoming battery degradation-dependent internal parameter variation. *J. Energy Storage* **2022**, *46*, 103888. [\[CrossRef\]](#)
- Lin, W.; Chai, Y.; Liu, Q. Remaining Useful Life Prediction of Electronic Products Based on Wiener Degradation Process. *IFAC-PapersOnLine* **2019**, *52*, 24–28. [\[CrossRef\]](#)
- Wei, L.; Wu, W.; Xu, J. Reliability Modeling of Complex Electronic System Based on Weibull Distribution. In Proceedings of the 2018 3rd International Conference on Modelling, Simulation and Applied Mathematics (MSAM 2018), Shanghai, China, 22–23 July 2018. [\[CrossRef\]](#)

11. Wan, Y.; Huang, H.; Das, D.; Pecht, M. Thermal reliability prediction and analysis for high-density electronic systems based on the Markov process. *Microelectron. Reliab.* **2016**, *56*, 182–188. [[CrossRef](#)]
12. Wang, Z.; Wu, Q.; Zhang, X.; Wen, X.; Zhang, Y.; Liu, C.; Fu, H. A generalized degradation model based on Gaussian process. *Microelectron. Reliab.* **2018**, *85*, 207–214. [[CrossRef](#)]
13. Shi, H.; Kang, H.; Lizhi, Z. Kernel density prediction method for residual life of multi component systems based on Copula theory. *Comput. Integr. Manuf. Syst.* **2023**, *29*, 212–223. [[CrossRef](#)]
14. Fan, Y.; Ma, X.; Chen, J.; Zhigang, Y. Evaluation method for constant stress accelerated storage test of electronic complete machine based on support vector machine. *Missile Space Deliv. Technol.* **2018**, *21*, 365.
15. Jiang, B.; Dai, H.; Wei, X.; Jiang, Z. Multi-Kernel Relevance Vector Machine With Parameter Optimization for Cycling Aging Prediction of Lithium-Ion Batteries. *IEEE J. Emerg. Sel. Top. Power Electron.* **2023**, *11*, 175–186. [[CrossRef](#)]
16. Li, W.; Sengupta, N.; Dechent, P.; Howey, D.; Annaswamy, A.; Sauer, D.U. One-shot battery degradation trajectory prediction with deep learning. *J. Power Sources* **2021**, *506*, 230024. [[CrossRef](#)]
17. Zhao, Z.; Wu, J.; Wong, D.; Sun, C.; Yan, R. Probabilistic Remaining Useful Life Prediction Based on Deep Convolutional Neural Network. *SSRN Electron. J.* **2020**. [[CrossRef](#)]
18. Deng, F.; Bi, Y.; Liu, Y.; Yang, S. Deep-Learning-Based Remaining Useful Life Prediction Based on a Multi-Scale Dilated Convolution Network. *Mathematics* **2021**, *9*, 3035. [[CrossRef](#)]
19. Liu, Z.; Jia, Z.; Vong, C.M.; Bu, S.; Han, J.; Tang, X. Capturing High-Discriminative Fault Features for Electronics-Rich Analog System via Deep Learning. *IEEE Trans. Ind. Inform.* **2017**, *13*, 1213–1226. [[CrossRef](#)]
20. Khemani, V.; Azarian, M.; Pecht, M. Prognostics and Secure Health Management (PSHM) of Electronic Systems in Zero-Trust Environment. *Annu. Conf. Phm Soc.* **2021**, *13*. [[CrossRef](#)]
21. Yin, Z.; Li, J.; Su, D. A Novelty Method for Bayesian Reliability Assessment of Electronic Equipment. *Microelectron. Comput.* **2014**, *31*, 107–110.
22. Mao, Y.; Sun, C.; Xu, L.; Liu, X.; Chai, B.; He, P. A survey of time series forecasting methods based on deep learning. *Microelectron. Comput.* **2023**, *40*, 8–17. [[CrossRef](#)]
23. Zhang, G. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing* **2003**, *50*, 159–175. [[CrossRef](#)]
24. Taylor, S.J.; Letham, B. Forecasting at Scale. *Am. Stat.* **2018**, *72*, 37–45. [[CrossRef](#)]
25. Elsayed, S.; Thyssens, D.; Rashed, A.; Jomaa, H.S.; Schmidt-Thieme, L. Do we really need deep learning models for time series forecasting? *arXiv* **2021**, arXiv:2101.02118.
26. Borovykh, A.; Bohte, S.; Oosterlee, C.W. Conditional Time Series Forecasting with Convolutional Neural Networks. *arXiv* **2017**, arXiv:1703.04691.
27. Chen, Y.; Kang, Y.; Chen, Y.; Wang, Z. Probabilistic forecasting with temporal convolutional neural network. *Neurocomputing* **2020**, *399*, 491–501. [[CrossRef](#)]
28. Borovykh, A.; Bohte, S.; Oosterlee, C.W. Dilated convolutional neural networks for time series forecasting. *J. Comput. Financ.* **2019**, *22*, 73–101. [[CrossRef](#)]
29. Liu, C.; Jin, Z.; Gu, J.; Qiu, C. Short-Term Load Forecasting using A Long Short-Term Memory Network. In Proceedings of the 2017, IEEE PES Innovative Smart Grid Technologies Conference Europe, (ISGT-Europe), Torino, Italy, 26–29 September 2017.
30. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.
31. Salinas, D.; Flunkert, V.; Gasthaus, J.; Januschowski, T. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *Int. J. Forecast.* **2020**, *36*, 1181–1191. [[CrossRef](#)]
32. Rangapuram, S.S.; Seeger, M.W.; Gasthaus, J.; Stella, L.; Wang, Y.; Januschowski, T. Deep state space models for time series forecasting. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 7796–7805.
33. Wen, R.; Torkkola, K.; Narayanaswamy, B.; Madeka, D. A multi-horizon quantile recurrent forecaster. *arXiv* **2014**, arXiv:1711.11053.
34. Wu, N.; Green, B.; Ben, X.; O'Banion, S. Deep transformer models for time series forecasting: The influenza prevalence case. *arXiv* **2020**, arXiv:2001.08317.
35. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In Proceedings of the 35th AAAI Conference on Artificial Intelligence/33rd Conference on Innovative Applications of Artificial Intelligence/11th Symposium on Educational Advances in Artificial Intelligence, ELECTR NETWORK, Vancouver, Canada, 2–9 February 2021; pp. 11106–11115.
36. Wu, H.; Xu, J.; Wang, J.; Long, M. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In Proceedings of the Advances in Neural Information Processing Systems 34 (NEURIPS 2021), 35th Conference on Neural Information Processing Systems (NeurIPS), ELECTR NETWORK, Online Conference, 6–14 December 2021; Volume 34.
37. Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; Jin, R. FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting. In Proceedings of the International Conference on Machine Learning, ICML 2022, 17–23 July 2022, Baltimore, MD, USA; Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., Sabato, S., Eds.; PMLR: New York City, NY, USA, 2022; Volume 162, pp. 27268–27286.
38. Li, S.; Jin, X.; Xuan, Y.; Zhou, X.; Chen, W.; Wang, Y.X.; Yan, X. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Av. Neural Inf. Process. Syst.* **2019**, *32*, 5244–5254.

39. Lim, B.; Arık, S.Ö.; Loeff, N.; Pfister, T. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *Int. J. Forecast.* **2021**, *37*, 1748–1764. [[CrossRef](#)]
40. Lai, G.; Chang, W.C.; Yang, Y.; Liu, H. Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks. In Proceedings of the 41st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), University Michigan, Ann Arbor, MI, USA, 8–12 July 2018; pp. 95–104. [[CrossRef](#)]
41. Kitaev, N.; Kaiser, U.; Levskaya, A. Reformer: The Efficient Transformer. *arXiv* **2020**, arXiv:2001.04451.
42. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the Advances in Neural Information Processing Systems 30, Long Beach, CA, USA, 4–9 December 2017; Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R., Eds.; NIPS: Long Beach, CA, USA, 2017; pp. 5998–6008.
43. Zhong, G.; Lin, X.; Chen, K.; Li, Q.; Huang, K. Long Short-Term Attention. In Proceedings of the 10th International Conference on Brain Inspired Cognitive Systems (BICS), Guangzhou, China, 13–14 July 2019; Ren, J., Hussain, A., Zhao, H., Huang, K., Zheng, J., Cai, J., Chen, R., Xiao, Y., Eds.; BICS: Guangzhou, China, 2020; Volume 11691, pp. 45–54. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.