Environmental Science and Pollution Research
https://doi.org/10.1007/s11356-019-06156-0

RESEARCH ARTICLE

Transfer learning for neural network model in chlorophyll-a dynamics prediction

Wenchong Tian [1,2,3] · Zhenliang Liao [1,2,3,4] · Xuan Wang [1,2,3]

Received: 4 April 2019 / Accepted: 31 July 2019

Abstract

Neural network models have been used to predict chlorophyll-a concentration dynamics. However, as model generalization ability decreases, (i) the performance of the models gradually decreases over time; (ii) the accuracy and performance of the models need to be improved. In this study, Transfer learning (TL) is employed to optimize neural network models (including feedforward neural networks (FNN), recurrent neural networks (RNN) and long short-term memory (LTSM)) and overcome these problems. Models using TL are able to reduce the influence of mutable data distribution and enhance generalization ability. Thus, it can improve the accuracy of prediction and maintain high performance in long-term applications. Also, TL is compared with parameter norm penalties (PNP) and dropout—two other methods used to improve model generalization ability. In general, TL has a better prediction effect than PNP and dropout. All the models, including FNN with different architectures, RNN and LSTM, as well as models optimized by PNP, dropout, and TL, are applied to an estuary reservoir in eastern China to predict chlorophyll-a dynamics at 5-min intervals. According to the results of this study, (i) models with TL produce the best prediction results; (ii) the original models and the models with PNP and dropout lose their ability to predict within 3 months, while TL models retain a high prediction accuracy.

Keywords Transfer learning · Chlorophyll-a dynamics · Feedforward neural networks · Recurrent neural network · Long short-term memory

Introduction

Lakes and reservoirs are commonly susceptible to algal blooms, resulting in adverse effects on drinking water security (Zhang et al. 2014). An early-warning, proactive approach to algal bloom prediction models can be effective in preventing the occurrence of algal blooms and minimizing the adverse effect of algal blooms on water bodies (Oh et al. 2007; Tian et al. 2017). Mechanisms of algal and chlorophyll-a dynamics are studied for this purpose (Vahtera et al. 2007; Nausch et al. 2012). Both deductive and inductive modeling methods have been developed over the years (Recknagel et al. 1997; Zhang et al. 2014) to predict the dynamics of chlorophyll-a and algae. Inductive models such as feedforward neural network (FNN) and recurrent neural network (RNN) have been applied in algae and chlorophyll-a dynamics prediction. For example, an artificial RNN was applied to predict the algae population dynamic in the Nakdong River-Reservoir System (Jeong et al. 2006). An artificial neural network was set up for predicting *Spirulina platensis* (Pappu et al. 2013) and the growth of the microalga *Karlodinium veneficum* (García-Camacho et al. 2016). Artificial neural network was also used for the management of estuarine algal blooms by Wei et al. (2001).

However, the models' application is not perfect. For instance, in some applications, the prediction accuracy of neural network needs improvement (Recknagel 1997; Coad et al.

Responsible editor: Marcus Schulz

✉ Zhenliang Liao
zl_liao@tongji.edu.cn

[1] UNEP-Tongji Institute of Environment for Sustainable Development, College of Environmental Science and Engineering, Tongji University, Shanghai 200092, People's Republic of China

[2] Shanghai Institute of Pollution Control and Ecological Security, Shanghai 200092, People's Republic of China

[3] Key Laboratory of Yangtze River Water Environment (Ministry of Education),, Tongji University, 200092 Shanghai, People's Republic of China

[4] College of Civil Engineering and Architecture, Xinjiang University, Urumqi 830046, People's Republic of China

Published online: 13 August 2019

2014; Zhang et al. 2015; Tian et al. 2017). The performance of the models gradually decreases over time contributing to poor model performance, especially in a long-term application (Tian et al. 2017).

The main reason of these problems is that the models' generalization ability is not enough and decreases over time. A model's generalization ability is a measurement of the model's performance on the dataset outside of the training sample (Lecun et al. 2015). Therefore, a model with low generalization ability demonstrates poor performance in different situations, such as different periods and cases. In an outdoor culture, the nutrient condition and algae (or chlorophyll-a) dynamics in the water body are mutable and unstable, which leads to nonlinear dynamics. Therefore, the identical-distribution assumption, which is a major assumption in many machine-learning and data-mining algorithms (Pan and Yang 2010), does not hold steady for algae (or chlorophyll-a) dynamics data over different cases and periods. A model, which has a fixed architecture and is trained by the data from a single period, is hard to generalize to the dynamics of multiple cases and periods, which leads to poor generalization ability (Jeong 2000; Jeong et al. 2003; Tian et al. 2017).

Improving the generalization ability of chlorophyll-a prediction models is needed to help models avoid these problems. To achieve this, one solution is to use as much data as possible to cover almost every situation (Lilover and Stips 2008). Another solution is to design different models with different architectures based on multiple environmental conditions, which is called the adaptive agent technique (Recknagel 2001). However, both of these approaches are computationally expensive, and it is difficult to define all possible environmental conditions (Janssen et al. 2004). Cross-validation (Lecun et al. 2015), PNP (Lecun et al. 2015) and dropout (Srivastava et al. 2014; Lecun et al. 2015), recently developed in a deep learning field, are helpful in increasing generalization ability and applied in many cases. Also, some research found that extra training with data from nearby periods can help the neural network model maintain accuracy (Tian et al. 2017).

The transfer learning (TL) is employed in this study to overcome the problem of generalization decreasing. TL refers to situations in which something learned in one setting is used to improve generalization in another setting (Lecun et al. 2015). It differs from traditional machine-learning techniques, which try to learn each task from scratch. Instead, the TL method attempts to transfer knowledge from previous tasks to a target task when the latter has fewer training data (Pan and Yang 2010). In general, if one machine-learning model has a good performance in one task, it can be generalized to similar tasks and achieve good performance without too much data and training.

Specifically, for the prediction of algae and chlorophyll-a dynamics, neural network models with a generalization ability supplement can achieve accurate predictions, even under different architectures and time periods. Thus, TL can be employed to a neural network model for generalization ability improvement. To implement TL, the model runs supplementary training using data from the nearest period, every time before model application. Thus, the model supplements its generalization ability by repeated training itself through repeated applications.

In this paper, the FNN, RNN, and long short-term memory, also PNP, dropout, and TL optimized FNN, RNN, and LSTM were established for chlorophyll-a dynamics prediction in an estuary reservoir in eastern China for a long-term application, under a small-time interval condition.

Methodology

Chlorophyll-a dynamics prediction based on FNN, RNN, and LSTM

In terms of chlorophyll-a dynamics, it is widely recognized that water quality, hydrology, biology, climate condition, and even some random events influence chlorophyll-a dynamics (Grover 1991; Seitzinger 1991) and usually exhibit strong non-linear characteristics (Recknagel 1997; Coad et al. 2014). However, the types of features related to chlorophyll-a dynamics and how they influence the dynamics are hard to predict in different cases. To predict the dynamics, a deep understanding of these two conditions is needed to establish a deductive model. Thus, the dynamics are hard to confirm based on deductive models alone (Recknagel 1997). To avoid the complex modeling process and achieve an accurate prediction, inductive models (e.g., machine-learning models) are employed. With suitable architecture, data, and training, the neural network is able to fit any non-linear continuous function (Lecun et al. 2015), including chlorophyll-a dynamics. All the parameters of the model are confirmed automatically by the training algorithm. It is not necessary to understand the water quality, hydrology, and biology completely before the prediction.

There are many machine-learning models for chlorophyll-a time series prediction. Normally, these kinds of models use the features of the environment and the historical chlorophyll-a as input and use the prediction of chlorophyll-a as output. Thus, the prediction model can be described as Eq. 1, where y_t is the chlorophyll-a concentration at time point t, x_t is the features which influence the dynamic at time point t, f is the prediction model which is a machine-learning model and can be confirmed by the data of y_t and x_t, θ is the parameter group of f. In this paper, the Feedforward neural network (FNN), Recurrent neural network (RNN), long short-term memory (LSTM) are employed as the model f to establish the prediction model.

$$y_{t+1} = f(x_t, y_t; \theta) \tag{1}$$

FNN is a kind of machine-learning method that has been applied many times to predict algae and chlorophyll-a dynamics. The basic architecture of the FNN is shown in Fig. 1 where X is the input layer, Y is the output layer and A are hidden layers. W and b are the weights and bias between two nearby layers. They can be updated automatically by training. With different architecture, the number of neural nodes and layers can be changed based on different applications.

RNN is a kind of deep learning model. With the help of parameter sharing and state transformation (Lecun et al. 2015), RNN has the advantage of analysis of sequence data. The architecture of RNN is shown in Fig. 2. X(t) and Y(t) represent the input and output at time point t; L(t) is an FNN structure at time point t. The node groups o(t) and h(t) together are called an RNN cell. Similar to the FNN, RNN updates its parameters by training.

LSTM, which is more complex in architecture, has a stable and robust training process compared with RNN (Graves 2008; Graves and Jaitly 2014). The main difference between RNN and LSTM is that LSTM uses the LSTM cell (Fig. 3), which includes a self-recycle and gate system to replace the RNN cell in the architecture. Here g, f, q, p, s, h are the nodes. Usually, g is called the update gate, f is called the forget gate, q is called the quite gate (Lecun et al. 2015).

For all three of these machine-learning models, the tangent function (Eq. 2) is used to activate all of the nodes, except the gate system of LSTM (the g, f, q in Fig. 3), which uses the sigmoid function (Eq. 3). The normalization of the input (Eq. 4, where $X\_in^{(i)}$ is the final input of the machine-learning model, $X^{(i)}$ is the original input of the ith sample and X (in all samples) is used to improve the model generalization ability.

$$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \tag{2}$$

$$\text{sigmoid}(x) = \frac{1}{1 + \exp(-x)} \tag{3}$$

$$X\_\text{in}^{(i)} = \frac{X^{(i)} - \max(X)}{\max(X) - \min(X)} \tag{4}$$

The architecture of FNN (the number of layers and neural nodes), RNN (the number of RNN cells), and LSTM (the number of LSTM cells) were confirmed by trial and error, which means the architecture with the lowest MSE on the training set was chosen. The detail of the architecture selection is given in the results section. All the models here were implemented in Python 3.6 with the help of TensorFlow.

In this paper, the algorithm used for training of all the models is processed by the Adam optimization algorithm (Kingma and Ba 2014; Lecun et al. 2015), which is an advance technique compared with BP, SGD, and momentum-based algorithm and is used in many deep learning fields. To make sure the training process is valid, we used the train-dev

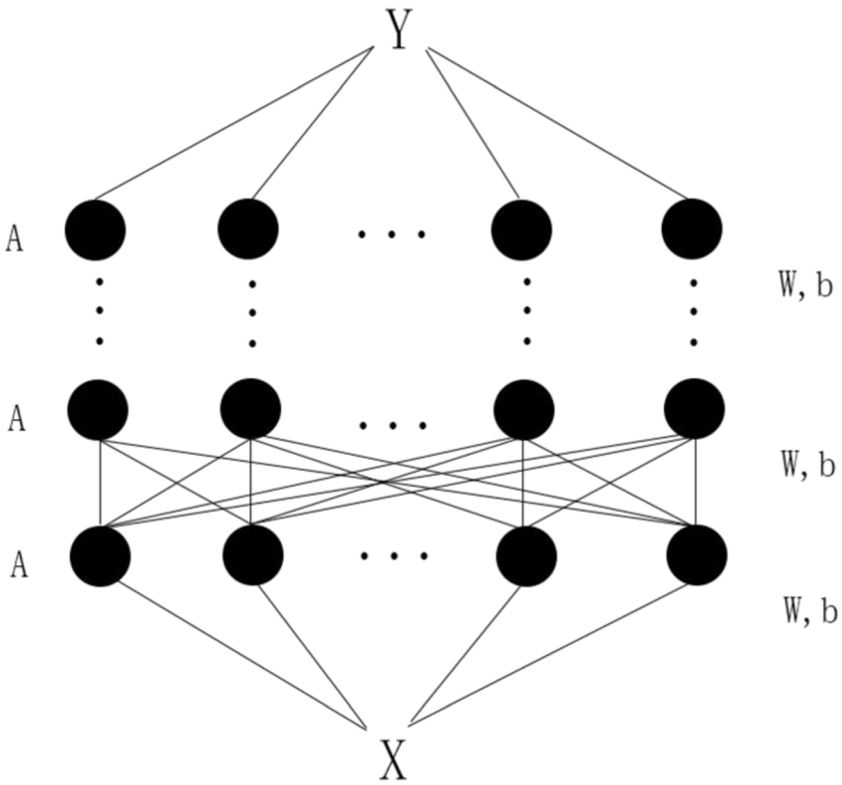

Fig. 1 The architecture of FNN

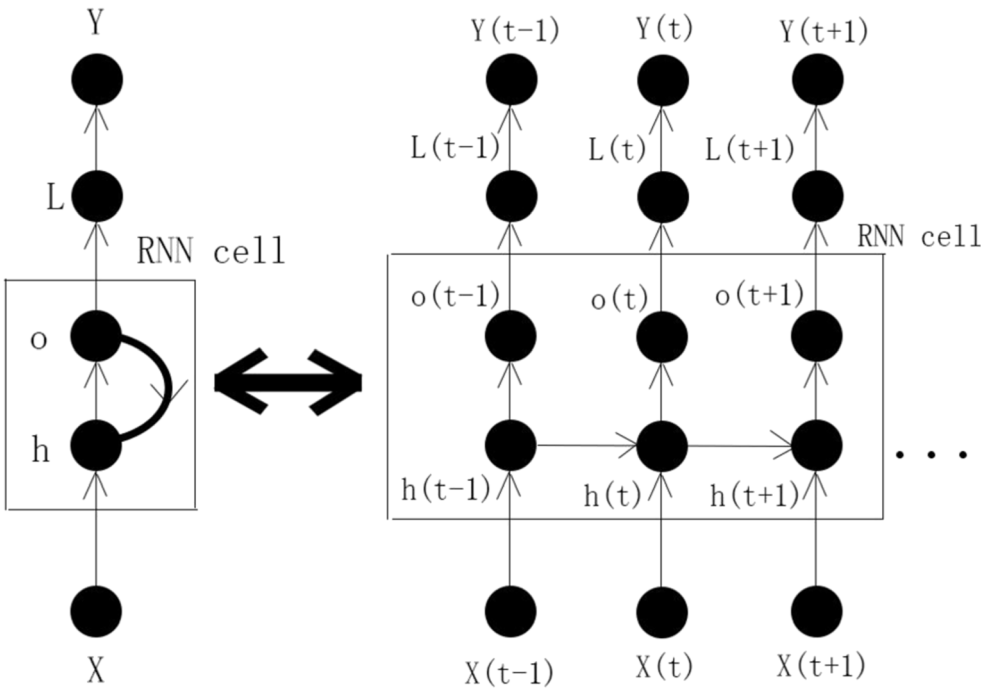

Fig. 2 The architecture of RNN

technique (Lecun et al. 2015). In this technique, the training data is divided into the training set and the validation set or test set. The training set is used to run the Adam algorithm for updating the model parameters, and the validation set is used to check model performance. If a model has a mean square error (MSE) close to 0 for both the training set and the validation set, it means that the training process is valid and has improved the model's generalization ability.

However, for algae and chlorophyll-a prediction, these methods have the following problems: (i) the prediction accuracy needs improvement and (ii) the prediction accuracy decreases over time. The main reason of these two problems is that the generalization ability decreases over time. Thus, improving the generalization ability is a key step in optimization of machine-learning models.

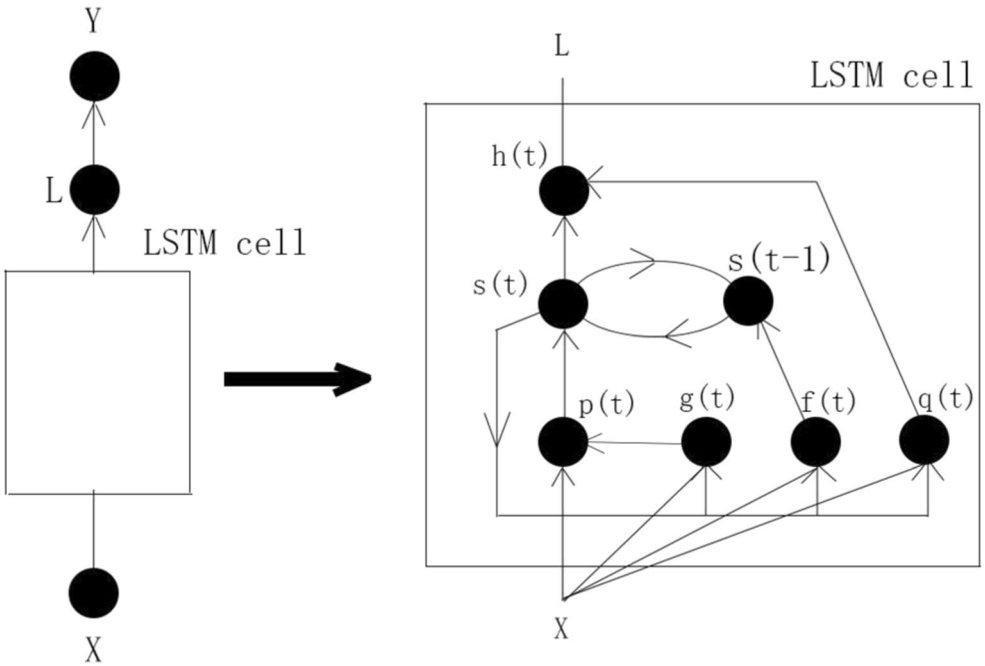

Fig. 3 The architecture of LSTM cell

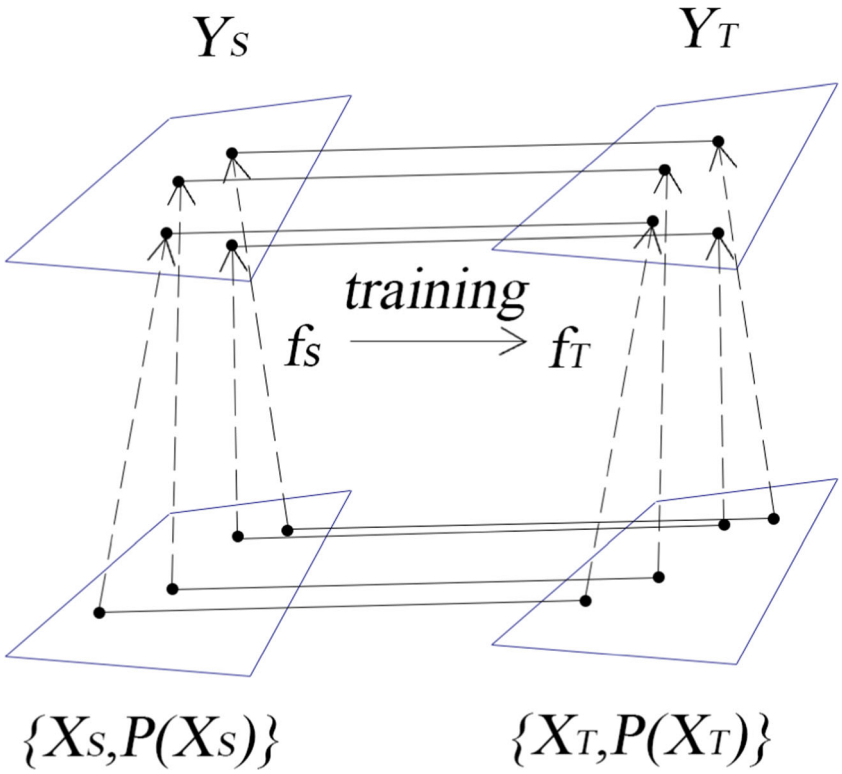

Fig. 4 The transfer learning

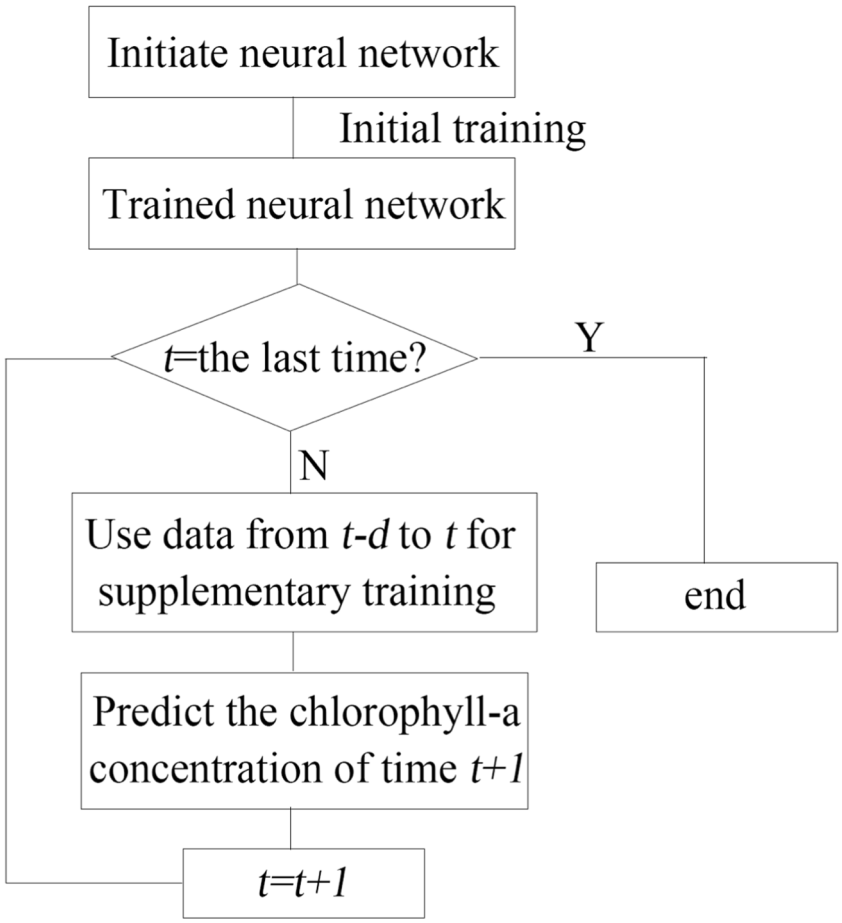

Fig. 5 The routine of the TL, *d* is the number of the data for supplementary training.

Table 1 Architecture of the eight models, where N is the number of neural nodes in hidden layers, M is the number of RNN cell or LSTM cell

	Architecture
FNN1	7-N-1
FNN2	7-N-N-1
FNN3	7-N-N-N-1
FNN4	7-N-N-N-N-1
FNN5	7-N-N-N-N-N-1
FNN6	7-N-N-N-N-N-N-1
RNN	7-M RNN cell-1
LSTM	7-M LSTM cell-1

Table 3 MSE of initial training of the models on training set and validation set

	Training set	Validation set
FNN1	1.651	0.759
FNN2	2.040	0.668
FNN3	2.846	0.723
FNN4	1.374	0.645
FNN5	1.277	0.679
FNN6	1.718	0.822
RNN	1.570	0.901
LSTM	1.418	0.836

Brief review of parameter norm penalties (PNP) and dropout

To improve the generalization ability of machine-learning models, one effective way is to avoid overfitting (Lecun et al. 2015). Therefore, many methods have been developed, including PNP and dropout, which are effective methods for generalization ability improvement in deep learning. These two methods reduce the complexity of neural network models to prevent overfit and improve generalization ability at some level. In this section, these methods are briefly reviewed.

The PNP is an effective method which can avoid overfitting and improve generalization ability. It modifies the loss function of the neural network model to help it focus on both model accuracy and architecture complexity. It can be described as shown in Eq. 5, where $\overline{J(w)}$ is the new loss function, $J(w)$ is the original loss function, $\Omega(w)$ is the penalty function, and λ is a parameter which represents the importance of the penalty function. By minimizing this loss function, the error of the model in the training set will be small, and the model will not be too complex. Using the penalty function serves to control model's complexity and avoid overfitting.

$$\overline{J(w)} = J(w) + \frac{\lambda}{2}\Omega(w) \tag{5}$$

According to different penalty functions, there are two kinds of PNP, L1 and L2. L1 PNP can find a sparse neural network making it useful for feature selection (Lecun et al. 2015); thus, the L1 function is applied in this paper.

Dropout (Srivastava et al. 2014) is a method proposed in recent years to control the complexity of the model. This method resets the weight to zero according to a certain probability during the training process, thereby reducing the complexity of the model and solving the problem of overfitting at some level.

The transfer learning in chlorophyll-a dynamics prediction

A major assumption in machine learning is that the training and future data must be in the same distribution. However, in chlorophyll-a dynamic prediction, this assumption may not hold. The main reason is that water quality, hydrology, biology, and climate condition are unstable and mutable, which leads to different data distributions in each case and period. Also, how these features influence chlorophyll-a dynamic changes with time (Wei et al. 2014; Zhang et al. 2014). Therefore, if a machine-learning model is only trained with the data from one period, it is most likely to lose its prediction ability in another period. TL is very useful in dealing with this situation.

Table 2 MSE on training set of models with different N and M

N (for FNN) or M (for RNN and LSTM)	1	2	3	4	5	6	7	8	9
FNN1	1.810	1.896	2.229	2.274	1.718	1.857	1.945	1.651	1.976
FNN2	2.088	2.045	2.267	2.086	2.048	2.045	2.121	2.040	2.056
FNN3	2.980	3.808	2.865	3.006	2.923	3.600	3.001	2.846	3.081
FNN4	1.832	1.983	2.921	3.246	2.099	2.141	2.192	1.374	2.253
FNN5	1.866	1.929	2.024	1.929	1.960	2.131	1.952	1.277	2.490
FNN6	1.882	1.860	1.948	1.955	2.236	2.282	2.176	1.718	2.284
RNN	1.656	1.626	1.570	1.997	1.982	3.091	3.424	2.857	4.036
LSTM	2.107	1.786	1.418	4.910	3.160	3.203	1.976	11.321	16.551

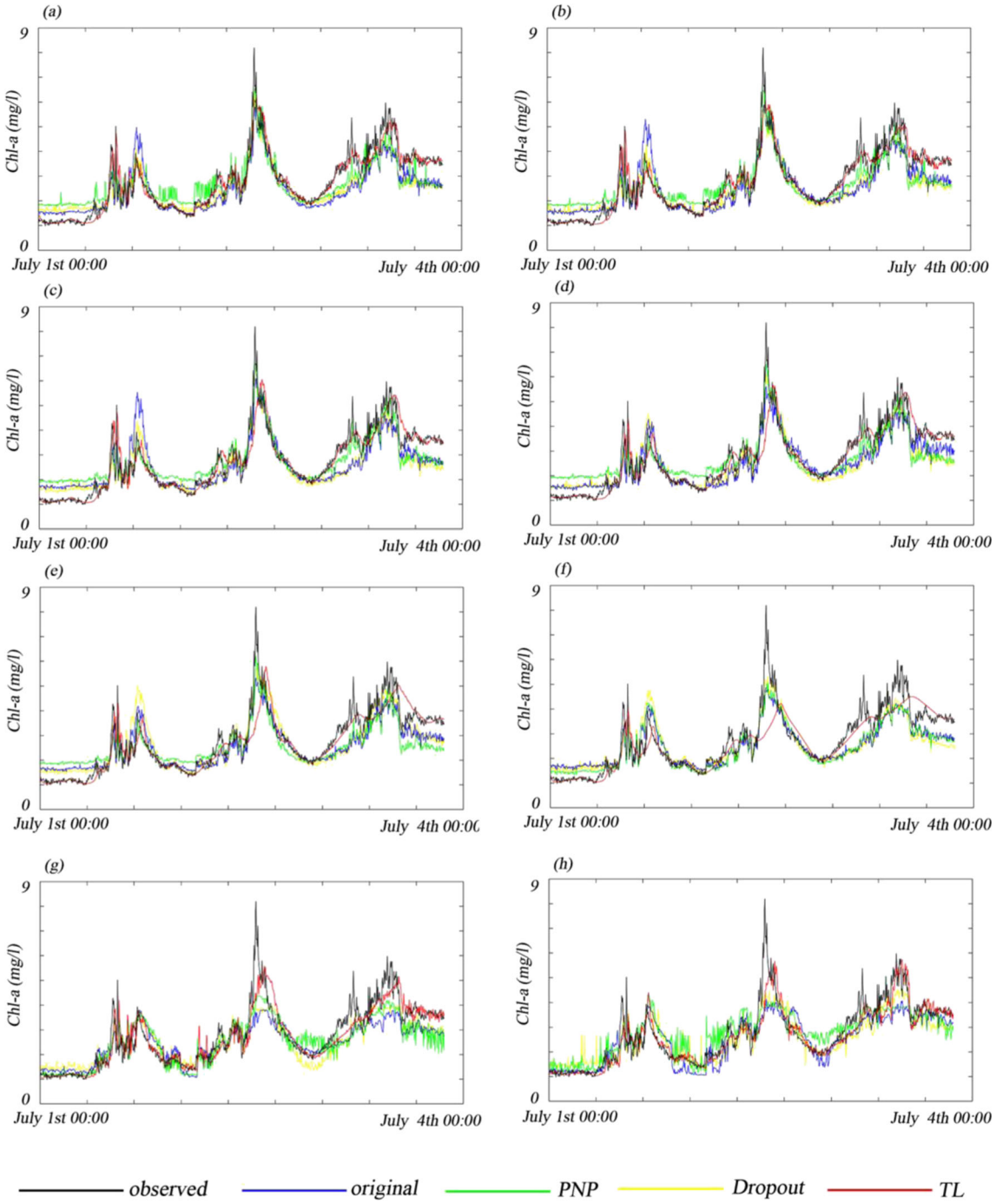

Fig. 6 Results of 1-month lag prediction of FNN1 (**a**), FNN2 (**b**), FNN3 (**c**), FNN4 (**d**), FNN5 (**e**), FNN6 (**f**), RNN (**g**), LSTM (**h**)

TL is designed to achieve generalization ability on different distributions (Pan and Yang 2010). The feature space (X) and its marginal probability distribution ($P(X)$) comprise the domain ($D = \{X, P(X)\}$). The output space (or label space Y) and the objective predictive model function (f) comprise the task ($T = \{Y, y = f(x)\}$). Traditional machine learning tries to get a good task, which means to find a good f based on the training data $\{x_i, y_i | x_i \in X, y_i \in Y\}$. When it comes to TL (Fig. 4), it tries to achieve a target task ($T_T = \{Y_T, y = f_T(x)\}$) on a target domain ($D_T = \{X_T, P(X_T)\}$) given the source task ($T_S = \{Y_S, y = f_S(x)\}$) on the source domain ($D_S = \{X_S, P(X_S)\}$) (Pan and Yang 2010). Which means using f_S and D_S to find f_T, this process can be achieved by model training. Usually, if the source and the target (including domains and tasks) are similar, it is easy for TL to achieve the target well and find f with good performance.

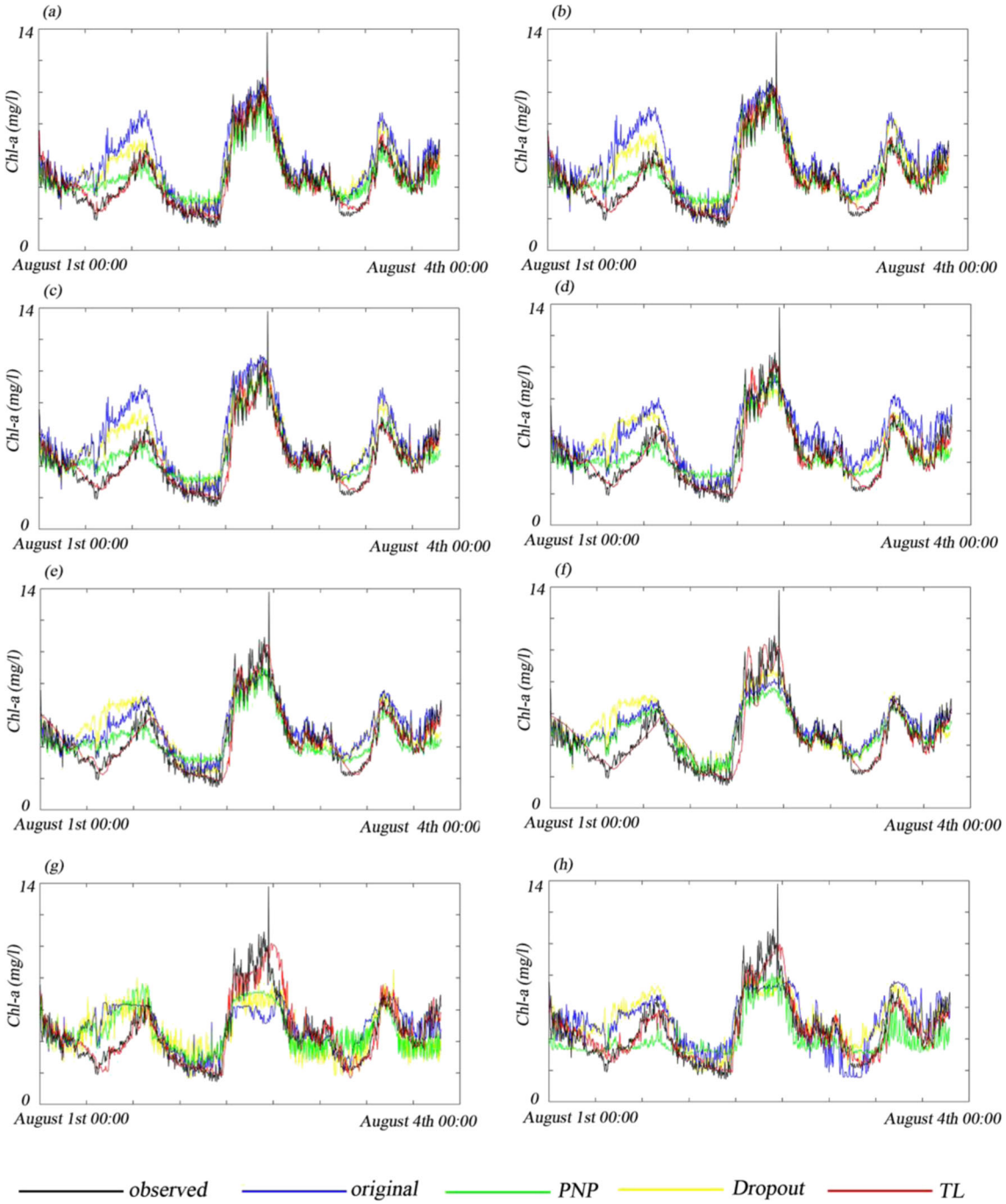

Fig. 7 Results of 2-month lag prediction of FNN1 (**a**), FNN2 (**b**), FNN3 (**c**), FNN4 (**d**), FNN5 (**e**), FNN6 (**f**), RNN (**g**), LSTM (**h**)

When applied to the chlorophyll-a prediction model, the domain includes the feature space X—the space of input features that influence the dynamic, and its data distribution $P(X, t)$ (as the chlorophyll-a dynamics change with cases and time, the distribution is related to the time variable t). The task includes the output space Y—the space of output chlorophyll-a concentration, and the model function $y_{t+1} = f_t(x_t, y_t)$ (x_t is the input feature and y_t and y_{t+1} is the chlorophyll-a concentration).

Traditionally, the machine-learning method tries to achieve the task based on one domain. However, when it comes to the next time period, the domain and the task change as $\{X, P(X, t+1)\}$ and $\{Y, y_{t+2} = f_{t+1}(y_{t+1}, x_{t+1})\}$. Thus, the previous model function may not make a good prediction for this period.

TL can be employed in this situation to achieve the target task. The source has the domain X, the $P(X, t)$ and the task with Y and $y_{t+1} = f_t(x_t, y_t)$. The target has a similar domain with X and $P(X, t+1)$, and a similar task with Y and $y_{t+2} = f_{t+}$

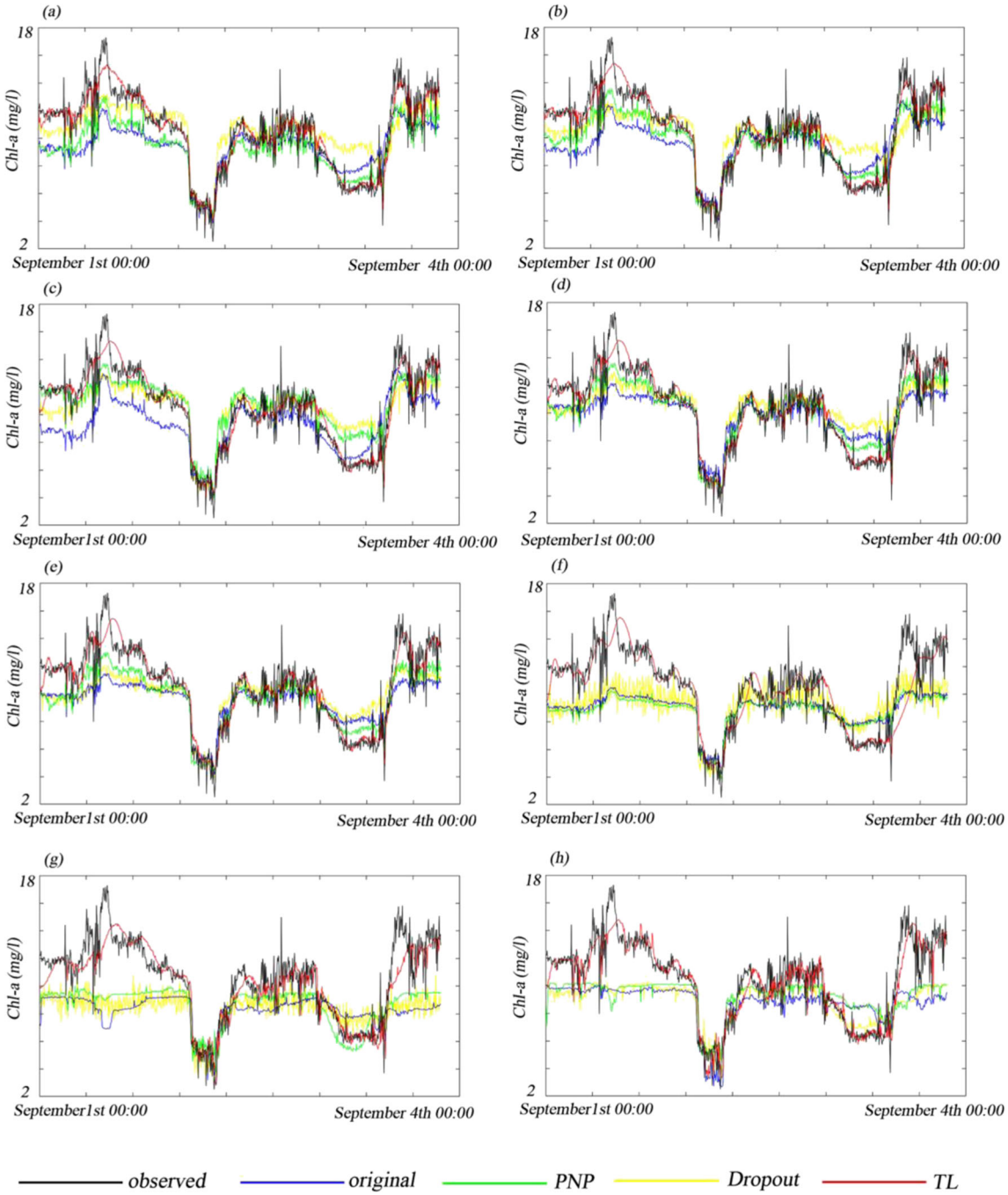

Fig. 8 Results of 3-month lag prediction of FNN1 (**a**), FNN2 (**b**), FNN3 (**c**), FNN4 (**d**), FNN5 (**e**), FNN6 (**f**), RNN (**g**), LSTM (**h**)

$_1(x_{t+1}, y_{t+1})$. Then, it is possible to achieve the target and get the $y_{t+2} = f_{t+1}(x_{t+1}, y_{t+1})$ by supplementary training of the source objective model $y_{t+1} = f_t(x_t, y_t)$ with training data from the target. The $y_{t+1} = f_t(x_t, y_t)$ with supplementary training improves its generalization ability and generates to task. Therefore, models can achieve a better prediction accuracy and performance in a long-term application.

When it comes to a continuous prediction, the model runs the supplementary training every time before prediction, thus achieving TL for the model. The routine is shown in Fig. 5.

Indices of the model performance

Six indices are applied for the evaluation of model performance for both model training and model prediction. They are the *MSE* (Eq. 6), the Bayesian information criterion (BIC) (Eq. 7), the statistical result of the Kolmogorov-Smirnov (K-S) test (Eqs. 8 and 9), The Conditional Kolmogorov Complexity (CKC), the bias feature (*BF*, Eq. 11), and the accuracy feature (*AF*, Eq. 12).

Table 4 The MSE of models

		MSE			
		Original	PNP	Dropout	TL
FNN1	1-month lag	0.422	0.371	0.368	0.133
	2-month lag	1.884	0.658	1.101	0.422
	3-month lag	3.990	2.787	2.485	1.002
FNN2	1-month lag	0.379	0.298	0.370	0.155
	2-month lag	2.609	0.580	1.372	0.496
	3-month lag	3.602	1.517	2.903	1.013
FNN3	1-month lag	0.424	0.254	0.393	0.214
	2-month lag	2.526	0.587	1.381	0.451
	3-month lag	3.906	1.539	2.613	1.113
FNN4	1-month lag	0.289	0.297	0.360	0.258
	2-month lag	1.811	0.606	1.605	0.551
	3-month lag	2.658	1.131	2.656	1.118
FNN5	1-month lag	0.315	0.398	0.313	0.311
	2-month lag	1.072	0.824	1.696	0.506
	3-month lag	3.651	1.904	3.417	1.152
FNN6	1-month lag	0.390	0.366	0.445	0.338
	2-month lag	1.131	1.022	1.710	0.728
	3-month lag	6.062	6.712	5.714	2.156
RNN	1-month lag	0.554	0.492	0.457	0.294
	2-month lag	2.509	1.923	2.392	0.776
	3-month lag	10.919	7.125	9.547	1.787
LSTM	1-month lag	0.497	0.491	0.397	0.281
	2-month lag	1.952	1.561	1.714	0.540
	3-month lag	7.157	6.355	6.060	1.427

Table 5 *P* value K-S test

		P value (K-S test)			
		Original	PNP	Dropout	TL
FNN1	1-month lag	1.463e-12	1.013e-21	1.860e-16	0.887
	2-month lag	2.486e-26	3.961e-18	3.854e-22	0.073
	3-month lag	7.90e-64	3.649e-43	7.441e-14	0.504
FNN2	1-month lag	1.116e-8	1.013e-21	5.073e-14	0.466
	2-month lag	2.769e-38	1.938e-33	2.669e-24	0.165
	3-month lag	5.725e-49	2.670e-24	3.854e-22	0.705
FNN3	1-month lag	7.046e-13	6.319e-29	1.071e-14	0.543
	2-month lag	1.179e-29	6.732e-19	8.889e-23	0.148
	3-month lag	3.575e-46	8.719e-12	3.442e-17	0.395
FNN4	1-month lag	6.442e-8	6.704e-30	1.751e-11	0.543
	2-month lag	5.672e-40	1.737e-20	2.244e-17	0.466
	3-month lag	2.929e-45	4.873e-13	6.733e-19	0.332
FNN5	1-month lag	2.473e-11	4.220e-26	1.503e-08	0.303
	2-month lag	1.207e-25	1.102e-19	1.052e-18	0.466
	3-month lag	4.470e-81	2.926e-27	2.785e-42	0.363
FNN6	1-month lag	1.342e-10	4.498e-09	9.612e-11	0.003
	2-month lag	5.013e-27	3.255e-15	2.244e-17	0.105
	3-month lag	4.81e-134	5.273e-142	2.568e-74	0.019
RNN	1-month lag	1.454e-15	1.286e-06	1.978e-07	0.623
	2-months lag	8.719e-12	3.597e-10	1.205e-05	0.466
	3-months lag	1.73e-186	2.513e-160	8.487e-146	0.010
LSTM	1-month lag	8.558e-8	1.589e-13	3.560e-06	0.118
	2-month lag	7.355e-24	3.529e-37	6.849e-31	0.093
	3-month lag	7.35e-152	2.414e-146	1.525e-142	0.332

The MSE is used as a measurement of model accuracy. And is given as Eq. 6, where n represents the number of time points, $y_p(t_k)$ and $y(t_k)$ are the predicted and observed chlorophyll-a values at time point t_k. The closer it is to 0, the better the approximation.

The BIC is a measure of both model performance and model complexity, a small BIC means a better model. It given as Eq. 7, where the L is the likelihood of model function and M is the number of model parameters.

Bias feature (*BF*, Eq. 8) and accuracy feature (*AF*, Eq. 9) are also used to measure prediction accuracy. In addition, *BF* is a measure of systematic overestimation or underestimation of the target values; and *AF* characterizes the average difference between the observed and the predicted values. When both of them are in the range from 0.95 to 1.11, this indicates an acceptable model performance (García-Camacho et al. 2016).

The K-S test attempts to determine if two data sets differ significantly. The K-S test has the advantage of making no assumption about the distribution of data. Its statistic for a given cumulative distribution function $F_{1,\ n}(x)$, $F_{2,\ m}(x)$ is shown in Eq. 10, where n and m are the sizes of the first and second sample, respectively. The threshold value of the rejected null hypothesis at level α is shown in Eqs. 11 and 12. If $D_{n,\ m} > K$, then the null hypothesis should be rejected at level α, which means the two data distributions are the same.

The *CKC* is employed here to measure the similarity between the source and the target (Pan and Yang 2010; Vitányi 2013). For source and target, or two data sets, a smaller *CKC* indicates their data distribution are similar to each other.

$$\mathrm{MSE} = \frac{1}{n}\sum_{k=1}^{n}\left(y(t_k) - y_p(t_k)\right)^2 \tag{6}$$

$$BIC = -2\log(L) + M\log(n) \tag{7}$$

$$\log(\mathrm{BF}) = \frac{\sum_{k=1}^{n}\log\frac{y_p(t_k)}{y(t_k)}}{n} \tag{8}$$

$$\log(\mathrm{AF}) = \frac{\sum_{k=1}^{n}\left|\log\frac{y_p(t_k)}{y(t_k)}\right|}{n} \tag{9}$$

$$D_{n,m} = \sup_x |F_{1,n}(x) - F_{2,m}(x)| \tag{10}$$

Table 6 The *BF* of models

		BF			
		Original	PNP	Dropout	TL
FNN1	1-month lag	1.216	1.243	1.210	1.091
	2-month lag	1.177	1.147	1.143	1.079
	3-month lag	1.287	1.207	1.244	1.121
FNN2	1-month lag	1.202	1.218	1.209	1.095
	2-month lag	1.168	1.104	1.152	1.079
	3-month lag	1.337	1.196	1.262	1.130
FNN3	1-month lag	1.225	1.212	1.217	1.101
	2-month lag	1.175	1.115	1.150	1.083
	3-month lag	1.321	1.197	1.264	1.120
FNN4	1-month lag	1.183	1.226	1.207	1.110
	2-month lag	1.142	1.088	1.149	1.082
	3-month lag	1.312	1.200	1.266	1.131
FNN5	1-month lag	1.191	1.245	1.182	1.121
	2-month lag	1.166	1.117	1.167	1.083
	3-month lag	1.242	1.229	1.275	1.131
FNN6	1-month lag	1.209	1.186	1.214	1.145
	2-month lag	1.228	1.241	1.220	1.108
	3-month lag	1.240	1.213	1.276	1.156
RNN	1-month lag	1.210	1.206	1.212	1.138
	2-month lag	1.329	1.249	1.304	1.106
	3-month lag	1.319	1.325	1.360	1.151
LSTM	1-month lag	1.213	1.215	1.194	1.128
	2-month lag	1.266	1.228	1.215	1.098
	3-month lag	1.341	1.289	1.283	1.143

Table 7 The *AF* of models

		AF			
		Original	PNP	Dropout	TL
FNN1	1-month lag	1.104	1.003	1.075	1.014
	2-month lag	1.105	1.114	0.978	1.001
	3-month lag	0.810	0.951	0.854	0.997
FNN2	1-month lag	1.068	0.996	1.079	1.021
	2-month lag	1.086	1.059	1.007	0.996
	3-month lag	0.784	0.931	0.844	1.008
FNN3	1-month lag	1.045	0.962	1.086	1.023
	2-month lag	1.114	0.962	0.989	0.999
	3-month lag	0.797	0.945	0.846	1.008
FNN4	1-month lag	1.059	0.982	1.072	1.033
	2-month lag	1.040	1.026	0.997	1.001
	3-month lag	0.796	0.967	0.882	1.019
FNN5	1-month lag	1.047	1.039	1.030	1.035
	2-month lag	1.085	1.070	1.046	0.997
	3-month lag	0.848	0.987	0.871	1.008
FNN6	1-month lag	1.055	1.085	1.057	1.054
	2-month lag	1.156	1.173	1.131	1.022
	3-month lag	0.868	1.000	0.920	0.877
RNN	1-month lag	1.074	1.081	1.067	1.036
	2-month lag	1.262	1.208	1.245	1.030
	3-month lag	0.935	0.971	1.008	0.995
LSTM	1-month lag	1.094	0.969	1.037	1.043
	2-month lag	1.191	1.123	1.157	1.017
	3-month lag	0.912	1.114	0.871	0.994

$$c(\alpha) = \sqrt{-\frac{1}{2} ln(\alpha)} \tag{11}$$

$$K = c(\alpha)\sqrt{\frac{m+n}{mn}} \tag{12}$$

Case study

FNN, RNN, and LSTM are optimized by TL, and all of these models (including original models, PNP optimized models, dropout optimized models, and TL optimized models) were applied to an estuary reservoir in eastern China to predict the chlorophyll-a dynamics.

The surface of the estuary reservoir is about 60 km^2, and the effective storage capacity is about 4.35×10^8 million m^3. The water quality data was obtained every 5 min by online electronic monitors placed at the outlet of the reservoir, with the depth of 2 to 3 m underwater since June 6, 2015. The data includes water temperature, pH, electronic conductivity, oxidation-reduction potential (ORP), turbidity, dissolved oxygen (DO), and chlorophyll-a. The training data is from June 5–9, 2015, with a time interval of 5 min for a total of 1440 samples. One thousand sample data were randomly chosen from the original data as the training set; the remaining 440 data was the validation set. Test data were collected over three time periods: July 1–3, August 1–3, and September 1–3, 2015, all with time intervals of 5 min.

Results

Model configuration and initial training

Six FNNs, one RNN, and one LSTM were employed to establish prediction models. Water temperature, pH, electronic conductivity, ORP, turbidity, DO, and current chlorophyll-a were the inputs, and the chlorophyll-a of the next point in time was the output. Normalization was applied to every input to improve the model's generalization ability.

The architectures of FNN, RNN, and LSTM are listed in Table 1. The number of nodes in hidden layers of FNN (called N) and the number of RNN cells or LSTM cells (called M)

Table 8 BIC of models

		BIC			
		Original	PNP	Dropout	TL
FNN1	1-month lag	495.109	495.048	495.044	494.780
	2-month lag	497.307	495.406	496.032	495.109
	3-month lag	501.273	498.949	498.386	495.886
FNN2	1-month lag	927.500	927.406	927.490	927.247
	2-month lag	931.059	927.749	928.896	927.642
	3-month lag	932.957	929.130	931.613	928.345
FNN3	1-month lag	1359.998	1359.799	1359.961	1359.755
	2-month lag	1363.349	1360.201	1361.353	1360.031
	3-month lag	1365.995	1361.610	1363.510	1360.937
FNN4	1-month lag	1792.283	1792.292	1792.365	1792.247
	2-month lag	1794.511	1792.669	1794.163	1792.599
	3-month lag	1796.038	1793.378	1796.034	1793.388
FNN5	1-month lag	2224.756	2224.854	2224.754	2224.775
	2-month lag	2225.763	2225.406	2226.759	2224.986
	3-month lag	2230.384	2227.116	2228.398	2225.884
FNN6	1-month lag	2657.289	2657.260	2657.354	2657.346
	2-month lag	2658.296	2658.133	2659.226	2657.718
	3-month lag	2667.603	2668.901	2666.909	2660.006
RNN	1-month lag	2927.768	2927.689	2927.646	2927.454
	2-month lag	2930.926	2929.870	2930.710	2928.061
	3-month lag	2947.590	2940.003	2944.845	2929.635
LSTM	1-month lag	4218.270	4218.262	4218.148	4218.013
	2-month lag	4220.495	4219.828	4220.085	4218.324
	3-month lag	4230.641	4229.039	4228.450	4219.610

were confirmed based on trial and error, which means the N and M with the lowest MSE on training set were chosen. The details of these results are given in Table 2. $N = 8$ and $M = 3$ was the best choice in this case. Accordingly, the chosen architecture of all the models are given as follow: (i) FNN have one input layer (7 nodes), one to six hidden layers (8 nodes in each layer), and one output layer (1 node). (ii) RNN includes one input layer (7 nodes), one RNN cell with three hidden layers (8 nodes), and one output layer (1 node). (iii) LSTM is the same as RNN, except that it replaces the three RNN cells with three LSTM cells.

To test the effect of TL, the architecture of the original models and the optimized models (including PNP models, dropout models, and TL models) were the same. The original models only ran the initial training by the training data. The optimized models (PNP, dropout, and TL) ran both initial training and supplementary training during application. The Adam algorithm was employed to train all the models. The MSE of different models on both the training set and the validation set are shown in Table 3. These models achieved low MSE on both the training and validation set, which indicates an effective training.

Table 9 *P* value of K-S test among 4 data set (data on June 1st to 3rd (a), July 1st to 3rd (b), August 1st to 3rd (c), and September 1st to 3rd (d))

	a–b	a–c	a–d	b–c	b–d	c–d
P value	6.9e-316	7.5e-187	3.0e-57	6.8e-80	0.00	2.4e-210

Model prediction

All the trained models (including the original models, PNP models, dropout models, and TL models) were applied to predict chlorophyll-a dynamics with intervals of 5 min. The first test prediction was from July 1 to 3, the second test prediction was from August 1 to 3, and the third test prediction was September 1–3, 2015. TL was applied according to the methodology, and the number of supplementary training data (d in Fig. 5) was 30. All results are given in Figs. 6, 7, and 8, and the analysis of the results is given in Tables 4, 5, 6, and 7.

According to the MSE and the *P* values of the K-S test, all of the optimized models were more likely performed better than the original models. Models with TL had the lowest prediction MSE and the best K-S test (P value larger than 0.05) among all the other models. According to AF and BF, TL models achieved best result among all four models, as all its AF and BF were closer to the range from 0.95 to 1.11.

According to Figs. 6, 7, and 8, in the 1-month lag prediction, all the models achieved reasonable values. When it comes to the 2-month lag prediction, the performance of the original model obviously decreased, leading to poor accuracy. The PNP and dropout models had a better performance than the original models, but their accuracy still decreased. In the 3-month lag prediction, all the original models, PNP models, and dropout models lost their validation and became useless in prediction. However, the models with TL maintained a higher prediction accuracy over time which indicate that TL was able to reduce the decrease in performance of the neural network model.

BIC of models

The BIC of all the models are given in Table 8. The BIC of TL models are most likely better than other models, which mean TL can help the models achieve better performance on prediction without increasing too many model parameters.

K-S test and CKC of dataset

To further discuss the data distribution of the data set, the K-S test and the distribution of chlorophyll-a data were given (Table 9 and Fig. 9). According to Table 9, it shows that all the *P* values were less than 0.05, which proves that the data distributions are different in each period.

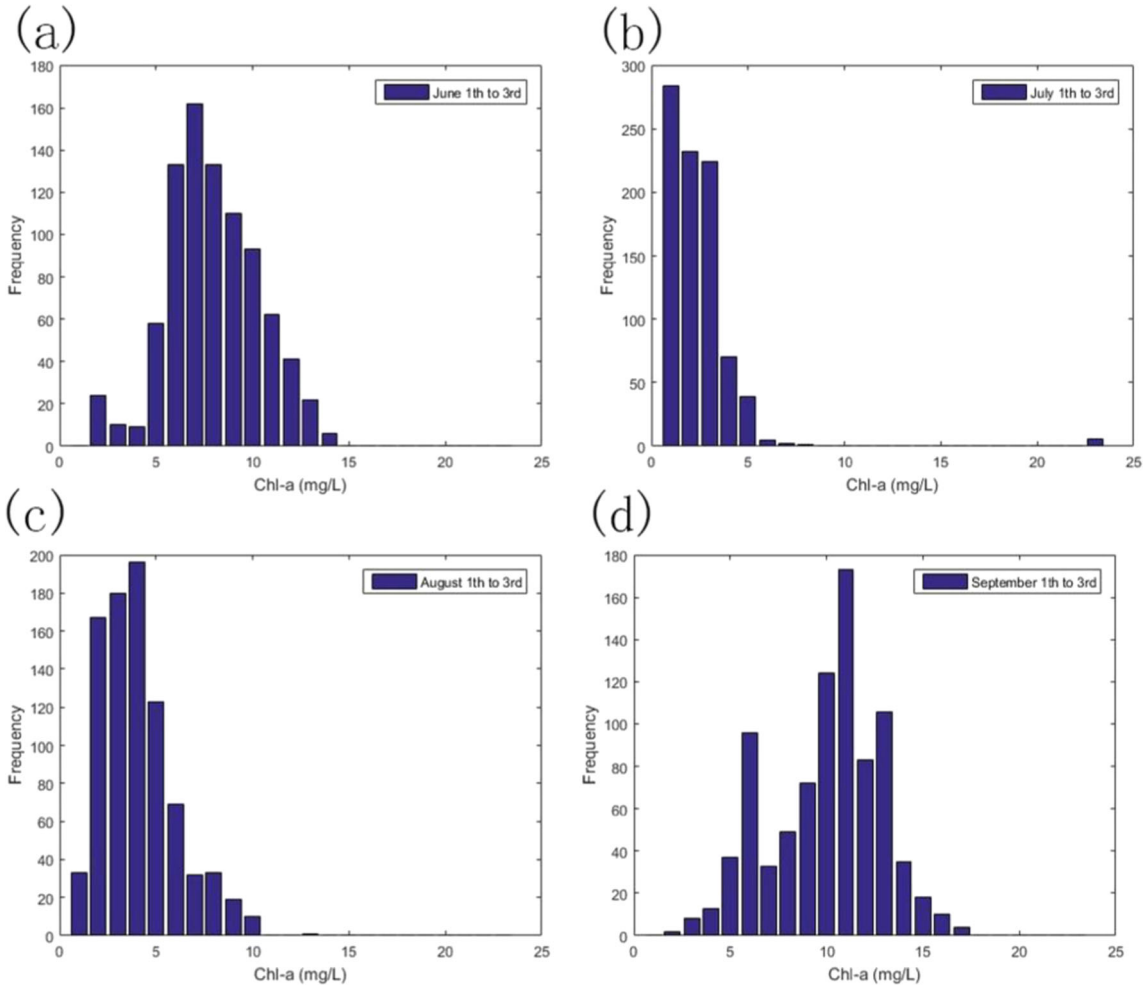

Fig. 9 The distribution of chlorophyll-a data on June 1st to 3rd (**a**), July 1st to 3rd (**b**), August 1st to 3rd (**c**), and September 1st to 3rd (**d**)

To measure the similarity between each data set, the CKC values were given in Table 10. The CKC between the 1-month lag data set and the 2-month lag data set is the smallest of all the other CKC values. It shows that as the time lag increases, the similarity between data sets decreases.

Discussion

The supplement of generalization ability

According to the results of MSE, AF, and BF in Tables 4, 5, 6, and 7, the performance of original models (including FNNs, RNN, and LSTM) keep decreasing over time. This indicates a bad generalization of original models.

Table 10 The *CKC* of each pair of the data set

	1-month lag and 2-month lag	1-month lag and 3-month lag	2-month lag and 3-month lag
CKC	2.669	30.469	19.414

The main reason causing the lack of the generalization ability is that the nutrient condition and chlorophyll-a (or algae) dynamics in the water body are mutable and unstable, which leads to nonlinear dynamics. Thus, the distribution of both training data and test data are not identical distribution. In the case above, according to the *P* values in Table 9 and Fig. 9, the data distribution was different in each time period. Thus, it is hard for just one model, which is trained by data from only one period, to fit the chlorophyll-a dynamics of different periods and achieve enough generalization ability.

However, with the help of TL, these original models keep learning the data distribution during a long-term application; thus, the influence of mutable and unstable data distribution can be reduced. Therefore, the generalization ability of these original models can be supplemented to avoid the decreasing of performance. In the results of MSE, AF, and BF in Tables 4, 5, 6, and 7, the performance of models with TL was better than the original models. Therefore, the TL, which applies a source model and some extra training to get the target model, is very suitable in this situation to avoid performance decreasing.

The performance of TL

The performance of the TL is strongly dependent on the similarity of data distribution between the source and the target. It is easy for the TL to receive a good performance if the source and the target are similar to each other.

According to the results in Table 10, the CKC value between the 1-month lag data set and the 3-month lag data set is larger than it is between the 1-month lag data set and the 2-month lag data set. It means that the data distribution of the 1-month lag data set is more similar to that of the 2-month lag data set than the 3-month lag data set. Thus, the TL should achieve better performance over a 2-month lag test than the 3-month lag test.

This analysis based on the CKC values is consistent with the results in Tables 4, 5, 6, and 7, which show that the TL achieves better performance in the 2-month lag test than the 3-month lag test.

The efficiency of generalization ability improvement

If we compare the MSE (Table 4) and the BIC (Table 8) of all the models, it is easy to find that (1) improving the complexity of the model cannot improve model generalization ability, and (2) the PNP and the dropout are not as efficient as TL in generalization ability improvement.

The reason for the first phenomenon is that increasing the model complexity helps nothing with reducing the influence of mutable data distribution. It only focuses on the fitting ability of the model rather than considering the generalization improvement so that the performance of the model is not improved with the model complexity. Both PNP and dropout improve generalization ability by avoiding overfitting. However, they can only help with the generalization ability improvement under the identical-distribution assumption, so that they might lose their function when the data distribution changed too much.

Therefore, the TL, which directly aimed to learn the mutable data distribution by supplementary training, is more efficient in generalization improvement than PNP and dropout.

Finally, considering the number of model parameters and the model performance, the FNN1 with TL is more suitable for chlorophyll-a dynamic prediction in this case, as it achieves both good prediction and the lowest BIC.

Conclusion

Chlorophyll-a (or algae) dynamics prediction models based on neural networks usually suffer from lack of generalization ability, leading to many problems such as low accuracy and decreasing performance. The main reason is that the data distributions of chlorophyll-a (or algae) in different time periods are not identical. To avoid this, TL technique is employed to optimize neural network models. Based on the results, transfer learning (TL) can reduce the influence of unstable data distribution and improve the generalization ability. Accordingly, (1) TL models have a better prediction accuracy than original models, PNP models, and dropout models, and (2) TL models maintain a higher performance during long-term application.

Also, the effect of TL is limited by the similarity of the source and target. The TL is able to achieve good performance if the source and the target are similar to each other. Finally, compared with PNP and dropout, TL is an efficient way to enhance model performance. Also, considering the model complexity and the performance, the FNN1 model with TL is the most suitable method for chlorophyll-a dynamic prediction in this case.

Funding information This study was financially supported by The National Key Research and Development Program of China (Grant No.2016YFE0123300 and 2017YFC0405406) and the National Natural Science Foundation of China (Grant No. 51578396).

References

Coad P, Cathers B, Ball JE, Kadluczka R (2014) Proactive management of estuarine algal blooms using an automated monitoring buoy coupled with an artificial neural network. Environ Model Softw 61(C):393–409

García-Camacho F, López-Rosales L, Sánchez-Mirón A, Belarbi EH, Chisti Y, Molina-Grima E (2016) Artificial neural network modeling for predicting the growth of the microalga karlodinium veneficum. Algal Res 14:58–64

Graves A. 2008 Offline handwriting recognition with multidimensional recurrent neural networks [C]// International Conference on Neural Information Processing Systems. Curran Associates Inc 2008:545-552

Graves A, Jaitly N 2014 Towards end-to-end speech recognition with recurrent neural networks [C]// International Conference on Machine Learning. 2014:1764-1772

Grover JP (1991) Resource competition in a variable environment: phytoplankton growing according to the variable-internal-stores model. Am. Nat. 1991:811–835

Janssen F, Neumann T, Schmidt M (2004) Inter-annual variability in cyanobacteria blooms in the Baltic Sea controlled by wintertime hydrographic conditions [J]. Mar Ecol Prog 275(1):59–68

Jeong KS 2000 Statistical evaluation and application of artificial neural networks on water quality of the lower Nakdong River. MSc thesis, Pusan National University, Busan, p. 74

Jeong KS, Kim DK, Whigham P et al (2003) Modelling microcystis aeruginosa, bloom dynamics in the Nakdong River by means of evolutionary computation and statistical approach [J]. Ecological Modelling 161(1–2):67–78

Jeong KS, Recknagel F, Joo GJ 2006 Prediction and elucidation of population dynamics of the blue-green algae microcystis aeruginosa, and the diatom Stephanodiscus hantzschii, in the Nakdong river-reservoir system (South Korea) by a recurrent artificial neural network [M]// Ecological Informatics. Springer Berlin Heidelberg, 2006:196-213

Kingma D, Ba J 2014 Adam: a method for stochastic optimization [J]. Computer Science

Lecun Y, Bengio Y, Hinton G (2015) Deep learning [J]. Nature 521(7553):436–444

Lilover MJ, Stips A (2008) The variability of parameters controlling the cyanobacteria bloom biomass in the Baltic Sea [J]. J Mar Syst 74(20):S108–S115

Nausch M, Nausch G, Mohrholz V et al (2012) Is growth of filamentous cyanobacteria supported by phosphate uptake below the thermocline [J]. Estuar Coast Shelf Sci 99(99):50–60

Oh HM, Ahn CY, Lee JW, Ahn CY, Lee JW, Chon TS, Choi KH, Park YS (2007) Community patterning and identification of predominant factors in algal bloom in Daechung Reservoir (Korea) using artificial neural networks [J]. Ecol Model 203(1-2):109–118

Pan SJ, Yang Q (2010) A Survey on Transfer Learning [J]. IEEE Trans Knowl Data Eng 22(10):1345–1359

Pappu JSM, Vijayakumar GK, Ramamurthy V (2013) Artificial neural network model for predicting production of Spirulina platensis, in outdoor culture [J]. Bioresour Technol 130(2):224–230

Recknagel F (1997) ANNA – artificial neural network model for predicting species abundance and succession of blue-green algae. Hydrobiologia. 349(1):47–57

Recknagel F (2001) Applications of machine learning to ecological modelling [J]. Ecol Model 146(1):303–310

Recknagel F, French M, Harkonen P, Yabunaka KI (1997) Artificial neural network approach for modelling and prediction of algal blooms. Ecol Model 96(s 1–3):11–28

Seitzinger SP (1991) 1991. The effect of pH on the release of phosphorus from Potomac estuary sediments: implications for blue-green algal blooms. J. Estuarine Coastal Shelf Sci. 33(4):409–418

Srivastava N, Hinton G, Krizhevsky A et al (2014) Dropout: a simple way to prevent neural networks from overfitting [J]. J Mach Learn Res 15(1):1929–1958

Tian W, Liao Z, Zhang J (2017) An optimization of artificial neural network model for predicting chlorophyll-a dynamics [J]. Ecol Model 364:42–52

Vahtera E, Laamanen M, Rintala J-M (2007) Use of different phosphorus sources by the bloom-forming cyanobacteria Aphanizomenon flos-aquae and Nodularia spumigena. Aquat Microb Ecol 46:225e237

Vitányi PMB (2013) Conditional Kolmogorov complexity and universal probability [J]. Theor Comput Sci 501:93–100

Wei B, Sugiura N, Maekawa T (2001) Use of artificial neural network in the prediction of algal blooms [J]. Water Res 35(8):2022

Wei L, Qin B, Zhu G (2014) Forecasting short-term cyanobacterial blooms in Lake Taihu, China, using a coupled hydrodynamic-algal biomass model. Ecohydrology 7(2):794–802 (9)

Zhang X, Recknagel F, Chen Q, Cao H, Li R (2014) Spatially-explicit modelling and forecasting of cyanobacteria growth in Lake Taihu by evolutionary computation. Ecol Model 306:216–225

Zhang Y, Huang JJ, Chen L, Lan Q (2015) Eutrophication forecasting and management by artificial neural network: a case study at Yuqiao Reservoir in North China. J Hydroinf 17:4