IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 7, 2021, accepted March 25, 2021, date of publication March 31, 2021, date of current version April 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3069990

An Effective Machine Learning Scheme to Analyze and Predict the Concentration of Persistent Pollutants in the Great Lakes

CHUNXUE WU[1], (Member, IEEE), BIN LI[1], AND NAIXUE XIONG[2], (Senior Member, IEEE)
[1]School of Optical Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China
[2]Department of Mathematics and Computer Science, Northeastern State University, Tahlequah, OK 74464, USA

Corresponding author: Chunxue Wu (wcx@usst.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFC0810204, and in part by the Shanghai Key Laboratory of Modern Optical System.

ABSTRACT Persistent organic pollutants (POPs) are highly toxic and difficult to degrade in the natural ecology, which has a severe negative impact on the ecological environment. Quantifying changes in the concentrations of persistent organic pollutants in the Great Lakes is challenging work. Machine learning (ML) methods are potent predictors that have recently achieved impressive performance on time series tasks. ARIMA model, Linear Regression methods, XGBoost algorithm, and Long Short-Term Memory (LSTM) are commonly used for estimating time-series changes. Traditionally Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) have been standard criteria to measure the error between the actual value and predicted value; however, Euclidean distance (ED) cannot effectively calculate the similarity between two-time series. We proposed an alternative criterion called Penalty Dynamic Time Wrapping (Penalty-DTW) based on Dynamic Time Wrapping (DTW). It can accurately measure the difference between the actual value and the predicted value. We study the benefits of Penalty-DTW vs. ED under the above ML algorithms. Further, considering the machine learning algorithm's uncertainty, we proposed combining LSTM and deep ensemble methods to quantify algorithms uncertainty and make a confident prediction. We find improved LSTM model outperformed other predictive power models by comparing pollutant concentration prediction. The prediction results show that the concentration of pollutants has a stable downward trend in recent years. Simultaneously, we found that pollutants' concentration correlates with seasons, which positively guides environmental pollution control in the Great Lakes.

INDEX TERMS Pollutants, time series, penalty-DTW, XGBoost, LSTM, deep ensemble, uncertainty.

I. INTRODUCTION

The Great Lakes of North America are a series of large interconnected freshwater lakes and are generally on or near the Canada-United States border. They are lakes Superior, Michigan, Huron, Erie, and Ontario. Over recent decades, with the economic development around the Great Lakes, the increase of human activities and other factors have directly led to the environmental pollution of the Great Lakes [1]. Eriksen *et al.* [2], and Mason *et al.* [3] confirmed the existence of microplastics in Great Lakes' surface water. Baldwin *et al.* [4] found the presence of plastics in the Great Lakes' water. These microplastics contain many chemicals, and persistent organic pollutants (POPs) are one of them. POPs are difficult to degrade in the environment. They have the characteristics of bio-accumulation, persistence, long-distance transport, and high toxicity. POPs are persistent in the environment, having long half-lives in soil, sediment, air, or biota [5].

A variety of methods have been developed for pollutants concentration prediction. Time series models have been used to predict future changes of pollutants for the concentration of pollutants that occur in cyclic or repeating patterns. Statistical and mathematical analysis method such as Auto-Regressive Integrated Moving Average (ARIMA) model [6] is fitted to time series data either to understand the data better or to predict future points in the series. Shamshad Ahmad *et al.* analyzed water quality based on the multiplicative ARIMA model, which has both nonseasonal and seasonal components [7]. LY Siew *et al.* have presented

The associate editor coordinating the review of this manuscript and approving it for publication was Pengcheng Liu.

the ARIMA model and improved the ARIMA model called ARFIMA for forecasting pollution [8]. Taking advantage of its strictly statistical approach, the ARIMA method only requires a time series's preliminary data to generalize the forecast. However, the ARIMA model has apparent disadvantages: first, It requires that the time series data be stable or stable after differentiation. Second, the ARIMA model always assumes linear relationships between independent and dependent variables. It can only capture linear relations but not nonlinear relations.

Machine learning (ML) has progressed dramatically over the past two decades, from laboratory curiosity to a practical technology in widespread commercial use [9]. ML algorithms also have a wide range of time series analysis applications, especially in pollutant concentration prediction. Marta Venier and Ronald A Hites presented multiple linear regression (LR) methods to address what rate of atmospheric concentrations of a specific chemical decreasing around the Great Lakes region [10]. Marla C. Maniquiz *et al.* used the same multiple LR methods to develop estimation models of pollutant loads [11]. Compared with the ARIMA model, the LR model is more straightforward and faster, leading to widespread deployment in practice. EXtreme Gradient Boosting (XGBoost) algorithm is proposed by Chen and Guestrin [12], have achieved state-of-art performance on a wide variety of machine learning tasks, including regression, time series. To control air pollution effectively, JunMa *et al.* propose a methodology framework combining XGBoost and Grid Importance Rank (GIR) [13]. Compared with SVR, RF, and GBDT, authors find that XGBoost has the best performance. In [14], Random Forest (RF), XGBoost, and Deep Neural Network (DNN) machine learning methods are used to investigate PM2.5 concentration prediction. A grid search on hyperparameters with 10-fold cross-validation was carried out to find the best model. Authors find the XGBoost technique demonstrated the highest performance and an acceptable time of training. In addition to the above conventional ML algorithms, W. Qiao *et al.* designed a hybrid forecast model for hourly gas consumption based on an improved whale optimization algorithm and relevance vector machine (improved support machine learning). The results show that the convergence accuracy and convergence speed of the new algorithm is higher than other algorithms [15].

Due to the improvement of computer computing power, deep learning algorithms have achieved success in many fields, including speech [16], natural language [17] and vision [18]. Deep learning algorithms achieve accuracy that is far beyond that of classical ML methods and has high adaptability. These positive aspects make AI methods an applicable technique in different engineering issues [19]. Deep learning methods can identify the structure and pattern of data such as non-linearity and complexity in time series forecasting [20]. In particular, Long Short-Term Memory (LSTM) has achieved great success in time series forecasting. Yuanyuan Wang *et al.* proposed a new water quality prediction method based on Long Short-Term Memory (LSTM) for water quality in Taihu Lake [21]. The authors trained the best model through a series of simulations and parameter selection. By comparing with BP neural network, it is concluded that LSTM has the best performance in predicting water quality. In [22], authors chose Mean square error (MSE) as LSTM loss function, adopt Adaptive moment estimation (Adam) as LSTM optimization algorithm, and set the number of iterations to 100. The results indicate that the predicted values of the LSTM model and the actual values were in good agreement. Weibiao Qiao and Zhe Yang proposed a novel hybrid model called WT-SAE-LSTM, which is a combination of wavelet transform (WT), stacked autoencoder (SAE), and LSTM to forecast electricity price time series [24]. The results showed that the designed model outperforms other AI methods. Despite the fact that the LSTM method has recently achieved impressive performance on time series forecasting, LSTM is poor at quantifying predictive uncertainty and tends to produce overconfident predictions. Overconfident incorrect predictions can be harmful or offensive [25]. Hence uncertainty quantification is crucial in time series forecasting.

The above research has been widely used in time series forecasting and has achieved good performance. There are key problems in comparing the predicted value and the actual value of the time series. When comparing the accuracy of algorithms applied in time series data, The RMSE has been used as a standard statistical metric to measure model performance. However, there is a piece of growing evidence that RMSE is poor distance measurement in time series domain [30], [31]. What we need is a dynamic and elastic method to evaluate time series similarity. Dynamic time warping (DTW) algorithm can make up for the shortcomings of Euclidean distance and calculate the time series's similarity dynamically. This method allows non-linear alignments between two-time series to accommodate similar sequences, but locally out of phase [32]. DTW is widely used in various domains, including word image matching [33], voice recognition [34], human action recognition [35].

This study aimed to find and develop a model that would analyze and predict the concentration of POPs in the Great Lakes. The main challenges are twofold: First, this study's model should be flexible and robust to handle the time series; second, we must develop better criteria to choose which model is the best. To solve these challenges, we used machine learning algorithms, mainly including Linear Regression (LR) models, XGBoost model, and ensemble LSTM model, to analyze the series of pollutant concentration and make predictions. Then we proposed a new type of similarity comparison measure called Penalty-DTW combined with Root Mean Square Error (RMSE) to ensure the validity of models.

The main contribution of the work are;

1. **We combine the LSTM algorithm and deep ensemble methods to quantify algorithms' uncertainty and make a confident prediction. The difference between the**

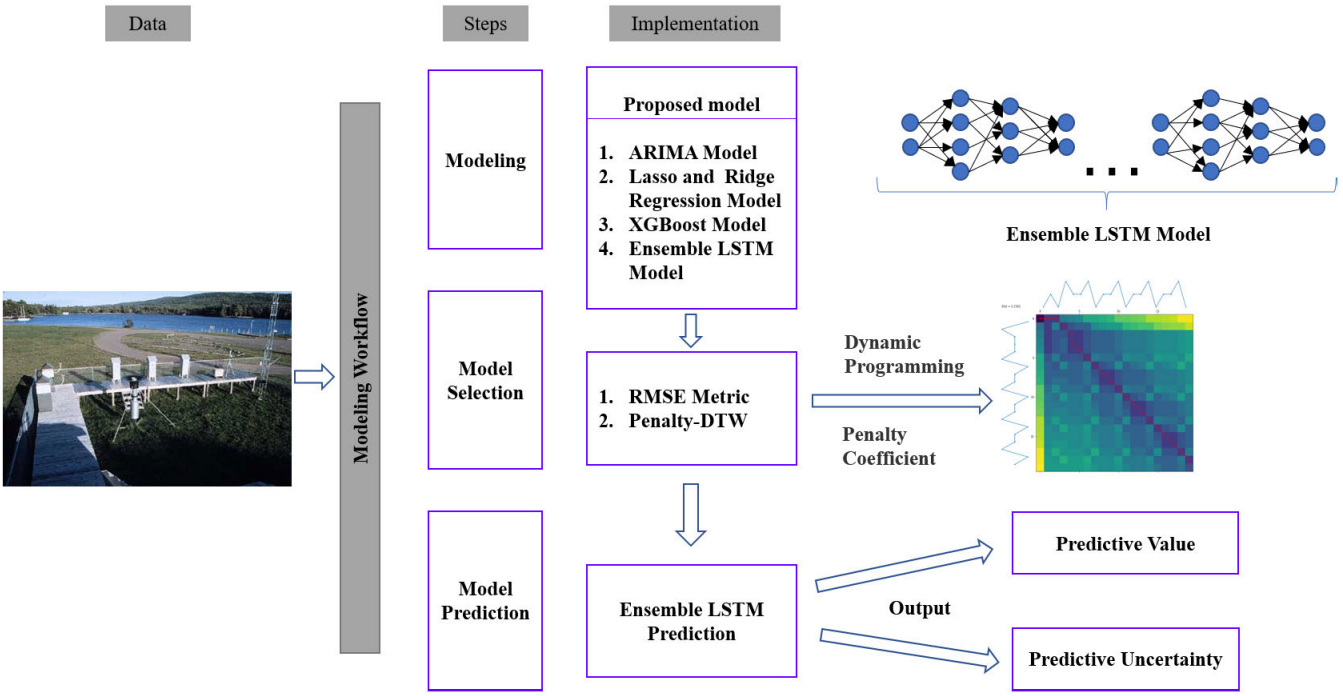


FIGURE 1. This paper presents the design schemes for the concentration prediction of persistent organic pollutants in the Great Lakes. Here, a complete workflow was performed using machine learning algorithms and the proposed ensemble LSTM model combined with the Penalty-DTW metric. This paper combines the LSTM algorithm and deep ensemble methods to quantify algorithms' uncertainty and make a confident prediction. The difference between the proposed model and the existing LSTM model improves prediction accuracy and measures prediction uncertainty. As a supplement to RMSE, we propose a new similarity comparison measure, Penalty-DTW, to compare LR, ARIMA, and LSTM algorithms' performance. Compared with the RMSE metric, Penalty-DTW is more suitable as the metric of time series data.

proposed model and the existing LSTM model is that it improves prediction accuracy and measure prediction uncertainty.

2. **As a supplement to RMSE, we propose a new similarity comparison measure, Penalty-DTW, to compare LR, ARIMA, and LSTM algorithms' performance. Compared with the RMSE metric, Penalty-DTW is more suitable as the metric of time series data**.

Fig.1 shows the complete workflow of this work.

The rest of this paper is organized as follows: In the second section, this article will introduce the previous research's efforts and shortcomings on model uncertainty and the criteria and problems that this article will try to solve. In the third section, we elaborate on the model methods and definitions in detail, including the study area, data preparation, and method. The fourth section describes in detail the experimental process of this study. The fifth section compares the methods used in this study with different criteria. The sixth chapter summarizes some of this study's characteristics and the work to be carried out in the future.

II. RELATED WORK

A. UNCERTAINTY OF DEEP LEARNING

The uncertainty measurement of deep learning algorithms has been ignored for a long time. People only care about the deviation between the predicted value and the deep learning algorithm's actual value. Until the past 20 years, due to the rapid development of deep learning algorithms, more and more deep learning algorithms have been deployed to the actual application scenarios. It is necessary to research model uncertainty to make the deep learning algorithm have good performance in the data distribution and the data outside the data distribution,

In the initial classification task, a softmax function is added to the last layer of the deep network to obtain the probability, but the probability cannot represent the model's uncertainty. The Bayesian neural network can obtain a confidence output first proposed by Mackay *et al.* [26]. It puts the prior distribution on the weights of the neural network and then derives the distribution of a parameter function group. Bayesian network is easy to understand; it is difficult to reason in practical application. Therefore, many approximate reasoning methods have been proposed. In the early process of the Bayesian neural network, the variational Inference is an approximate reasoning method, which takes the marginalization needed in the Bayesian reasoning process as the optimization problem [27]. Variational Inference simplifies the Bayesian neural network calculation and improves the reliability of the Bayesian neural network. The focus of the Bayesian neural network is to find a good approximation of posterior distribution. The prediction value and interval are calculated as the expected value of the posterior distribution. The accurate prediction depends on the accurate approximation of the difficult posterior probability. Monte Carlo hidden Markov (MCMC) method shows a random walk behavior. The Markov chain's basic idea generates the posterior distribution samples, and Monte Carlo

integration is carried out based on the samples (effective samples) when the Markov chain reaches stationary distribution. Due to the complexity and high dimension of BNNs posterior function, this random walk behavior makes these methods not suitable for reasoning in any reasonable time. To solve the problem of MCMC in bnns, Yarin gal *et al.* Proposed a new Monte Carlo method and MC dropout [28]. As a Bayesian approximation method, the MC dropout method is interpreted as a Bayesian approximation of the Gaussian process, which solves the high computational cost of Bayesian approximation. MC dropout method does not need to modify the existing network model, and it only needs the dropout layer in the neural network model. In the model training, MC dropout and dropout layers are the same, but in the test process, when the neural network propagates forward, the neural network's dropout layer cannot be closed.

In recent years, non-Bayesian methods are also gradually developed. The core idea of the non-Bayesian method is to train multiple probabilistic neural networks and construct the best network model through self-sampling and ensemble learning. Lakshminarayanan *et al.* Evaluated the uncertainty of deep learning on the Imagenet dataset by using ensemble learning method [29]. Compared with the standard Bayesian method, his implementation method is simple and suitable for distributed computing and large-scale deep network.

To more accurately predict the concentration of sustainable organic pollutants in the Great Lakes, this paper combines the LSTM algorithm and ensemble learning method of the non-Bayesian method to estimate the model uncertainty in the depth network. Improved LSTM can estimate the uncertainty of the predicted value and get the predicted value more accurately.

III. SYSTEM MODELS AND DEFINITIONS

A. STUDY AREA AND DATA SETS

The study area is Eagle Harbor(47.46306°N, 88.14972°W), located on the north shore of the Keweenaw Peninsula. The harbor is irregularly shaped, about 4900 feet long and 1100 feet in width. The sampling site is located 1km east of Eagle Harbor, about 100m from the lake.

B. DATA PREPARATION AND SPLIT

Integrated Atmospheric Deposition Network (IADN) is a long-term monitoring program run by the U.S. Environmental Protection Agency's Great Lakes National Program Office. It has measured the concentrations of persistent toxic chemicals in Great Lakes air and precipitation since 1990s [39]. The sampling time-frequency of pollutant concentration is once every two weeks. Since that time, over a million measurements of the concentrations of PCBs, pesticides, PAHs, flame retardants, and trace metals have been made. This study mainly analyzes Polychlorinated biphenyl (PCBs) ' total concentration in Eagle Harbor, a persistent organic pollutant. Features we analyze mainly include *Phase*, *Site*, *Date*, *Unit*, and *SuitePCBs*.

The standard k-fold cross-validation method is widely used in ML algorithms. It can evaluate the stability of ML models and optimizes model parameters to make more accurate predictions. However, it does not work well in time series data because it ignores the inherent temporal components. In this study, *TimeSeriesSplit* in scikit-learn package is used. It provides train/test indices to split time series data samples observed at fixed time intervals in train/test sets. In each split, test indices must be higher than before, and thus shuffling in cross-validator is inappropriate. This cross-validation object is a variation of KFold. In the kth split, it returns first k folds as train set and the $k+1$th fold as test set [40]. In this experiment, we set the number of time series split to 5.

C. METHODS

1) PENALTY-DTW

DTW is a method to measure the similarity of time series. Unlike the traditional Euclidean distance measurement methods, the algorithm solves the template matching problem of different time series lengths. It is based on dynamic programming; to some extent, this is similar to solving the longest common substring (LCS) problem.

We now assume that there are two time series $P(0\ldots M)$ and $Q(0\ldots N)$. We construct a two-dimensional array of $dp[i][j]$, which represents the square of the similar distance between the series $P[0\ldots i]$ and $Q[0\ldots j]$. $dp[i][j]$ satisfies (1):

$$dp[i][j] = \begin{cases} (P[0]-Q[0])^2 & i=0,\ j=0 \\ (P[0]-Q[j])^2 + dp[0][j-1] & i=0 \\ (P[i]-Q[0])^2 + dp[i-1][0] & j=0 \\ (P[i]-Q[j])^2 & \\ +min(dp[i-1][j], dp[j-1][i], & \\ dp[i-1][j-1]) & i,j>0 \end{cases} \tag{1}$$

Array dp is used to solve the problem that the two sequences' lengths must be equal. Refer to (1), we concluded that when we get the value of $dp[m-1][n-1]$, we get the similarity of time series $P(0\ldots M)$ and $Q(0\ldots N)$.

Although DTW has solved the problem of time sequence length cannot equal, there is a problem we have to think about: when a time series is come by another time series translation, from the actual, two time series are equal, but when we calculate according to the similarity of the DTW, found that two time series is not equal, not only that, two time series similarity can vary widely.

To solve this problem, we propose some improved strategies for the DTW algorithm. We first need to calculate a penalty coefficient α, the multiplication of the penalty coefficient, and the DTW calculation result is the final similarity we need.

Based on dynamic programming, we first work out the longest common substring of two time series. Since time series P and Q both are numerical types, we set the tolerance

of the maximum standard deviation. Within this tolerance range, the data of the two values are regarded as equal and will be added to the common substring. When we get the longest common sequence *com_len*, the penalty coefficient α is calculated as follows:

$$\alpha = e^{-\sum_{i=0}^{n} \frac{com\_len_i \times com\_len_i}{lenP \times lenQ}} \tag{2}$$

lenP is the length of time series *P* and *lenQ* is the length of *Q*. It can be seen from (2) that the longer the common substring of two time series is, the smaller the penalty coefficient α is, the smaller the similarity will be.

The algorithm logic is shown in Algorithm 2.

Algorithm 1 Penalty Coefficient

Require: Time series *P* and *Q*;
Ensure: The longest common sequence of *P* and *Q*;

```
lenP ← P.length()
lenQ ← Q.length()
comLen ← 0
std ← (std(P) >= std(Q) ? std(P) : std(Q))
paths[LenP + 1][lenQ + 1] ← inf
subMatrix[LenP][lenQ] ← 0
for i = 0 to lenP do
    for j = 0 to lenQ do
        dist ← P[i] − Q[j]
        cost ← dist ∗ ∗2
        paths[i + 1][j + 1] ← cost + min(paths[i][j + 1], paths[i + 1][j], paths[i][j])
        if math.abs(dist) < std then
            if i == 0 or j == 0 then
                subMatrix[i][j] = 1
            else
                subMatrix[i][j] = subMatrix[i][j] + 1
                comLen = max(comLen, subMatrix[i][j])
            end if
        end if
    end for
end for
paths = sqrt(paths)
return paths, comLen
```

2) ARIMA

In statistics and econometrics, particularly in time series analysis, the ARIMA model summarizes the Autoregressive Moving Average (ARMA). Both models are fitted to time series data. ARIMA models are applied in some cases where data show evidence of nonstationarity, where an initial differencing step (corresponding to the "integrated" part of the model) can be applied one or more times to eliminate the nonstationarity [6]. The core idea of the ARIMA model is to combine the AR model and the MA model. ARIMA model is denoted ARIMA(p, d, q) where p, d and q are nonnegative integers. p is the lags of the time series data used in the prediction model, also known as the Auto-Regressive (AR) term. d is the degree of differencing (the number of times the data have had past values subtracted). q is the lags of the prediction errors used in the prediction model, known as Moving Average (MA).

3) LINEAR REGRESSION

LR is a linear approach to modeling the relationship between a scalar response and one or more explanatory variables. It was the first type of regression analysis to be studied rigorously and used extensively in practical applications [41]. In LR, the relationships are modeled using linear predictor functions whose unknown model parameters are estimated from the data. Such models are called linear models [42]. Linear regression models are often fitted using the least-squares approach. However, they may also be fitted in other ways, such as by minimizing the "lack of fit" in some other norm (as with least absolute deviations regression), or by minimizing a penalized version of the least-squares cost function as in Ridge Regression (*l1_norm* penalty) and Lasso Regression (*l2_norm* penalty).

4) XGBoost

EXtreme Gradient Boosting (XGBoost) algorithm is a scalable tree boosting system widely used by data scientists and provides state-of-the-art results on many problems, proposed by Chen and Guestrin [12]. The core idea of XGBoost is to add trees constantly and constantly split the characteristics to grow a tree. Every time a tree is added, it is actually to learn a new function and use the new function to fit the residual. XGBoost can accurately predict the classification and regression problems, and it also has excellent performance for the prediction of time series.

5) LONG SHORT-TERM MEMORY

The techniques of ANNs are well indeed established as one of the most robust mathematical-based solutions promising reliable connections between the inputs and output(s) [43]. LSTM(Long Short-Term Memory) was proposed by Hochreiter and Scheduler 1997 and recently improved and promoted by Alex graves [44]. It is well-suited to classify, processing, and to make predictions based on time series data. LSTM is a variant of recurrent neural networks. It adds input gate, output gate, and forget gate based on recurrent neural network [45], which helps deal with vanishing or exploding gradients. Vanilla LSTM is a variation of origin LSTM, has become the state-of-the-art model for ML problems in recent years [46]. By utilizing vanilla LSTM, model make full use of the long short-term memory power to obtain precise accuracy. Vanilla LSTM have achieved great success in predictive accuracy in natural language processing, speech recognition domains.

IV. PROPOSED MODELS

A. ARIMA MODEL

When building an ARIMA model for time series analysis, we need first to detect the stationarity of time series. If the series data is not stationary, we need to carry out a differential

process. Generally speaking, the time series's first difference can achieve the series's stationarity, but sometimes the second difference check is needed. Then we identify the model, that is, to determine the categories of AR(p), MA(q) or ARIMA(p, d, q) models. Next step is estimating the parameters p and q. When parameters the estimation is completed, we must check whether the time series is white noise. Only when the time series is white noise can we build the model.

In testing stationarity of time series, we use Dicky-Fuller [48] test method to detect whether the time series is stationary. In parameter estimation, we mainly use the BIC criterion to select models among a finite set of models, and the lower BIC is preferred. In testing white noise of time series, we use the Lagrange Multiplier (LM) test for residual correlation LM Test. Suppose the corresponding $p - value$ of F statistic is greater than the significance level α. In that case, the original hypothesis is acceptable, and the residual sequence is considered a white noise sequence, and the model has passed the test.

B. LINEAR REGRESSION AND XGBoost MODEL

LR and XGBoost models need multi-dimension features to train the model, but now we only have one-dimension features. We have to extract features from the one-dimension time series. We mainly use the two methods to extract features. The first method is to shift the series n steps back, then we get a feature column, in which the current value of the time series is aligned with its value at time $t - n$. The second method is to obtain the time characteristics of the current time, mainly including the *month*, *week*, *quarter* of the current time, whether the current time is summer (the concentration of the pollutant in summer may be higher than that in other months).

For LR, we used Lasso Regression and Ridge Regression models. They both add some more constraints to the loss function to resolve the problem of overfitting. Ridge regression responds to ordinary least squares' problems by imposing a penalty on the size of the coefficients. Lasso regression is a linear model that estimates sparse coefficients.

For XGBoost, the most notable is parameter tuning. We mainly tune three types of parameters: general parameters, booster parameters, and learning task parameters. General parameters are set automatically by XGBoost generally. Booster parameters define characteristics of building XGBoost tree, mainly includes *max_depth*, *min_child_weight*, *sub_sample* and other features. Learning task parameters are used to define the optimization objective the metric to be calculated at each step, mainly includes *objective*, $eval_metric$. *objective* and *eval_metric*, we set $objective = linear$ and $eval\_metric = rmse$. The general approach includes:

Step 1. Choose a relatively high learning rate to determine the optimum number of trees for this learning rate. We initialize $learning\_rate = 1$.
Step 2. Tune tree-specific parameters.
Step 3. Tune regularization parameters (lambda, alpha) for XGBoost can help reduce model complexity and enhance performance.
Step 4. Lower the learning rate and decide the optimal parameters.

In step2, we combine GridSearch with 5-fold cross-validation. The whole dataset is divided into five parts. Each time GridSearch is used, one part is taken as the test set, and the other four parts are used as the training set to train the model. Then the performance of the model on the test set is calculated. Early Stopping is also used to ensure that the loss does not decline before it stops; we set $early\_stopping\_rounds = 10$.

C. ENSEMBLE LSTM MODEL

Before the ensemble LSTM model training, we carried out data transformation, divided into three steps. First of all, we converted our single column of data into a two columns dataset: the first column containing this day's (t) pollutant concentration and the second column containing the next day's ($t+1$) pollutant concentration, which is to be predicted. Because the stationary time series is more comfortable to model and can lead to more accurate prediction, we then transformed the time series into a stationary one, which is the same as the ARIMA model. Finally, we scale the time series values because the activation function adopted by the LSTM model is tangent function:

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}.$$

Its value range is -1 to 1, so we have to transform the dataset to the range -1 to 1. The specific conversion method is as follows:

$$X_{scaled} = \frac{X - X.min\,(axis = 0)\,(max - min)}{X.max\,(axis = 0) - X.min\,(axis = 0)} + min.$$

Completing the data transformation, we developed ensemble LSTM model based on Vanilla LSTM model. Different from the standard Vanilla LSTM model, our model has two output gates, one is used to output the predictive value, and the other is used to evaluate the predictive uncertainty. We use the variance of ensemble LSTM predictions as approximate measurement of uncertainty. The ensemble model algorithm as follows.

Algorithm 2 Training Ensemble LSTM for Uncertainty Prediction

Require: The models ensemble size, M.
Ensure: Ensemble LSTM model.
1: Let each neural network parametrize a distribution over the outputs. Use RMSE and Penalty-DTW scoring rules as the training criterion.
2: Initialize θ_1, θ_2,..., θ_M randomly
3: Minimize scoring rule;

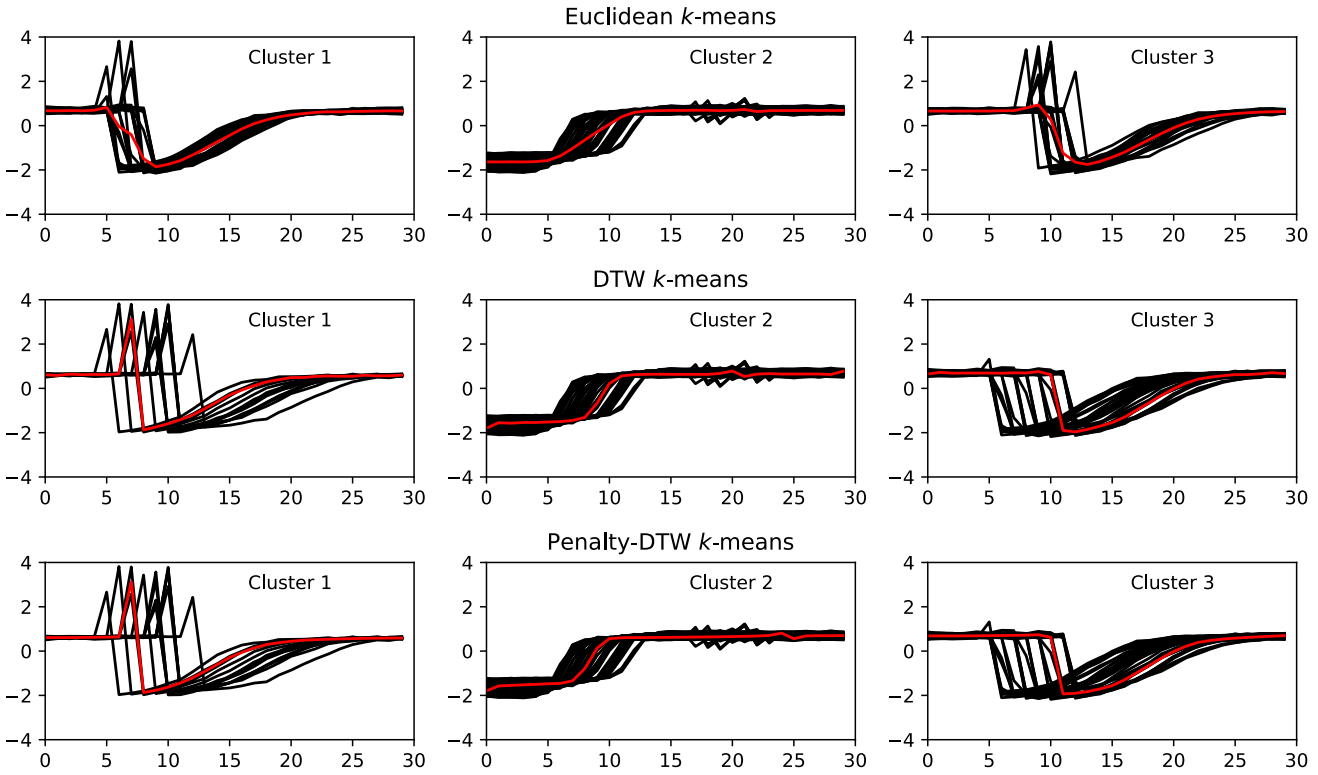


FIGURE 2. Results on the cylinder-bell-funnel dataset. k-means clustering with 3 metrics: Euclidean distance, Dynamic time Wraping, and Penalty-DTW. Each subfigure represents series from a given cluster and their centroid (in red).

V. EXPERIMENTAL RESULTS ANALYSIS AND DISCUSSION

A. PENALTY-DTW METRIC

1) COMPARISON OF EUCLIDEAN DISTANCE AND DTW

Time series cluster is typical verification to measure the similarity of different time series. We performed a simple k-means cluster experiment on the cylinder-bell-funnel dataset. The dataset is a deceptively simple-looking three-class problem. All classes are of length 128.

The algorithm's variants are available: standard Euclidean k-means, k-means based on DTW, and k-means based on Penalty-DTW. In Fig.2, each row corresponds to the result of a different clustering. In a row, each subfigure corresponds to a cluster. It represents the set of time series from the training set assigned to the considered cluster (in black) and the barycenter of the cluster (in red). The first row is the result that k-means clustering based on standard Euclidean distance. One problem is that Euclidean Distance assumes the ith point in one sequence is aligned with the ith point in the other. The second row shows the result of a k-means clustering that uses DTW as a primary metric, which allows a more intuitive distance measure to be calculated. In this example, we used preprocessing method to process input time series data. This preprocessing way ensures that each output time series have zero mean and unit variance.

2) COMPARISON OF DTW AND PENALTY-DTW ON TOY DATA

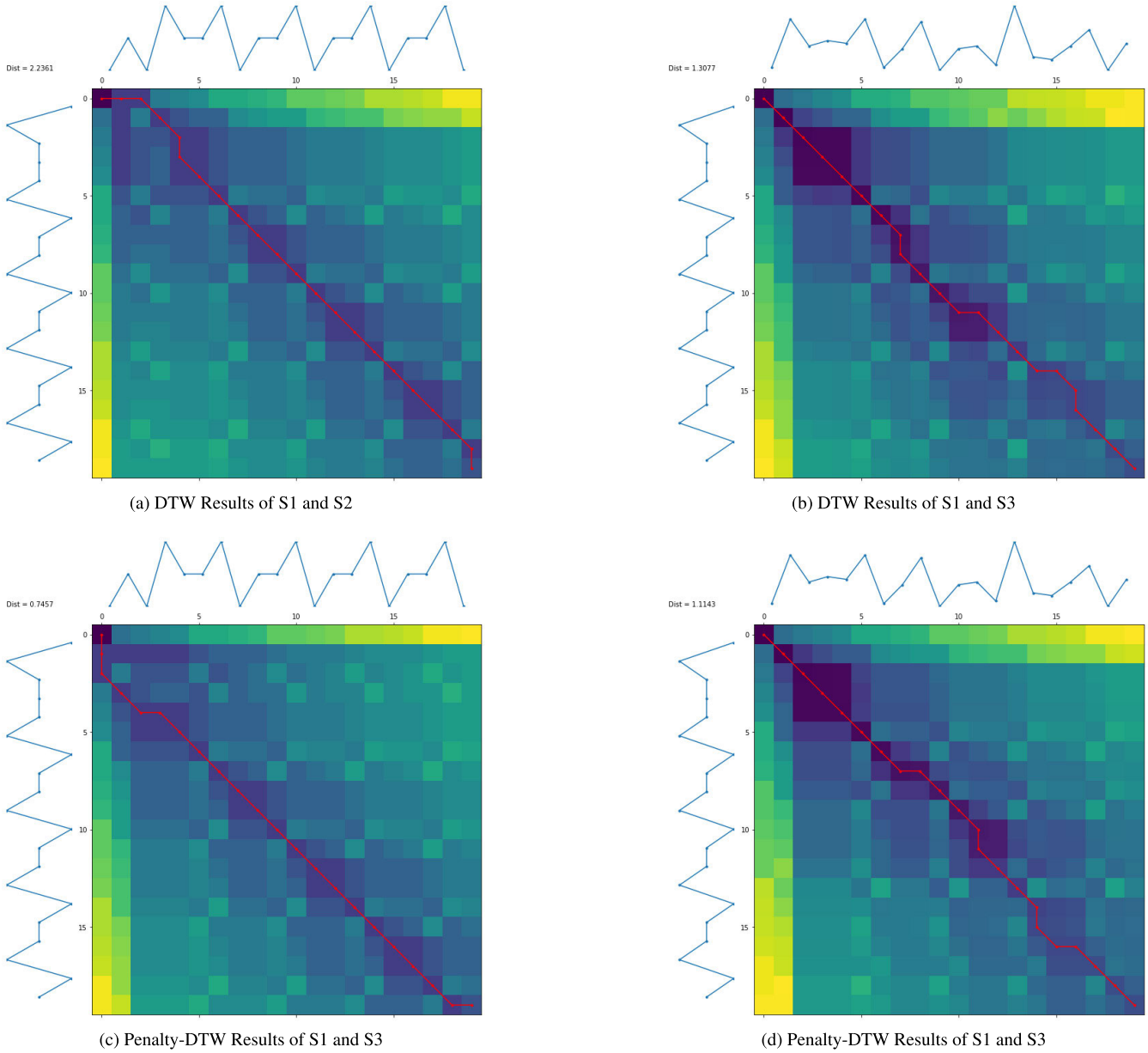
To prove obviously the effectiveness of the Penalty-DTW algorithm, we chose $S1$, $S2$ and $S3$ as time series with a length of 20, in which $S2$ is translated from time series $S1$ and $S3$ is another randomly generated sequence, $S1$, $S2$ and $S3$ are as follows:

$$\begin{aligned} s1 &= [0, 2, 1, 1, 1, 2, 0, 1, 1, 2, 0, 1, 1, 2, 0, 1, 1, 2, 0, 1], \\ s2 &= [0, 1, 0, 2, 1, 1, 2, 0, 1, 1, 2, 0, 1, 1, 2, 0, 1, 1, 2, 0], \\ s3 &= [0.1, 1.9, 0.9, 1.1, 1, 1.9, 0.1, 0.8, 1.8, 0, 0.8, 0.9, \\ &\quad 0.2, 2.4, 0.5, 0.4, 0.9, 1.5, 0, 1]. \end{aligned}$$

Fig.3 illustrate DTW and Penalty-DTW computation between time series and plots the optimal alignment path. The images represent cost matrix, that is the squared Euclidean distance for each time point between both time series, which are represented at the left and at the top of the cost matrix. The optimal path, that is the path that minimizes the total cost to go from the first time point to the last one, is represented in red on the images. The upper left corner of the image shows the results of DTW and Penalty-DTW.

Fig.3(a), (b) show the results of the unimproved DTW algorithm. We found that the similarity between time series $S1$ and $S2$ is 2.236, whereas $S1$ and $S3$ is 1.3077, the distance of $S1$ and $S2$ time series is larger than that of $S1$ and $S3$, which is not consistent with the actual situation.

Fig.3(c), (d) show the results of the improved DTW algorithm. We observe that the similarity distance of $S1$ and $S2$ time series is smaller than that of $S1$ and $S3$; that is to say, the improved Penalty-DTW algorithm works better.



(a) DTW Results of S1 and S2

(b) DTW Results of S1 and S3

(c) Penalty-DTW Results of S1 and S3

(d) Penalty-DTW Results of S1 and S3

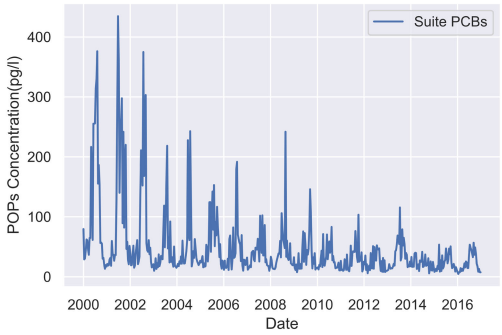
FIGURE 3. Evaluating Penalty-DTW Results, (a) showed the DTW results of series S1 and S2, (b) showed the DTW results of series S1 and S3. The images represent cost matrix. The curves on the left and top of the image denote the squared Euclidean distance for each time point between both time series. The red line in the graph indicated the optimal path.

B. ARIMA RESULTS

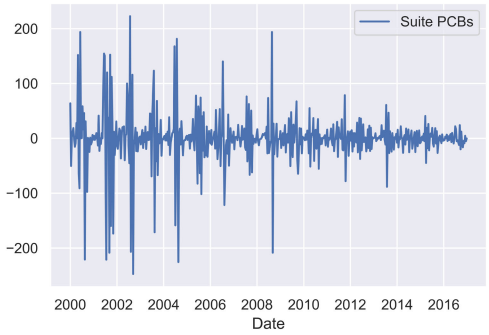
Fig.4 shows the time series of PCBs pollutant concentration since 2000. It can be seen from the figure that the overall trend of PCBs concentration is decreasing year by year, and there may be a seasonal trend. In some months of the year, the concentration of PCBs is on the high side, whereas in some months, PCBs' concentration is on the low side. To ensure the stationarity of the data, the first difference was performed for the nonstationary data. The results of the difference are shown in Fig.4. The raw data is significantly smoother than before after differential operation.

We used Dicky-Fuller test methods to check whether the data are stationarity. Dicky-Fuller test results are shown in Table.1. From the results of the Dickey-Fuller test, we can see the value of the test statistic is $-1.144889e+01$, which is far less than the critical value (1%) of $-3.439327e+00$; that is to say, it is perfect to reject the original hypothesis, with 99% reliability to ensure the stability of the time series.

The critical point of ARIMA model construction lies in selecting p, q, and d parameters. To select these parameters more accurately, we adopted the BIC order criterion. We took the grid search method to select p and q according to the BIC criterion to select the appropriate parameters and observed that BIC is the lowest when p is 2, and q is 6. That is to say, the order of the autoregression term is 6, the order of difference item is 1, and the order of Moving Average is 6.



(a) Time series trend of PCBs concentration. x-axis denotes the year, and the y-axis denotes the pollutant concentration.



(b) First Difference Results for Time series data. x-axis denotes the year, and the y-axis denotes differential results.

FIGURE 4. Results about time series trend of PCBs and first difference results for PCBs.

TABLE 1. Dickey Fuller test results.

Evaluation criteria	Value
Test Statistic	$-1.144889e+01$
p-value	$5.924966e-21$
Critical Value (1%)	$-3.439327e+00$
Critical Value (5%)	$-2.865502e+00$
Critical Value (10%)	$-2.568880e+00$

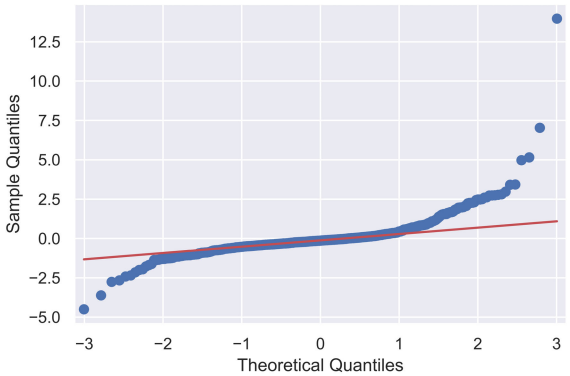


FIGURE 5. Theoretical Quantiles.

Before model fitting, we also need to test the model by residual. Fig.5 is the QQ chart to check whether it satisfies the positive and negative distribution. We see that the residual satisfies the distribution of the positive and the negative. The D-W method was then performed to test the residual's autocorrelation, and the D-W test result is 1.986. According to the D-W test characteristics, when the D-W test value is close to 2, there is no autocorrelation, which indicates that the model works well. Fig.5 shows the ARIMA model fitting results. The RMSE value is 60.84, ARIMA model has not good prediction results.

C. LINEAR REGRESSION RESULTS

We shifted the series n steps back. Then we get a feature column, in which the current value of the time series is aligned with its value at time $t - n$. In this experiment, we chose the value of n in the range of $[2, 26]$. Then we added *weekday*, *quarter*, *month*, *year*, *dayofyear*, *weekofyear*, *is_weekend*, *is_summer*, *is_winter* and other features to the dataset. Datasets is split into 5 train-test folds.

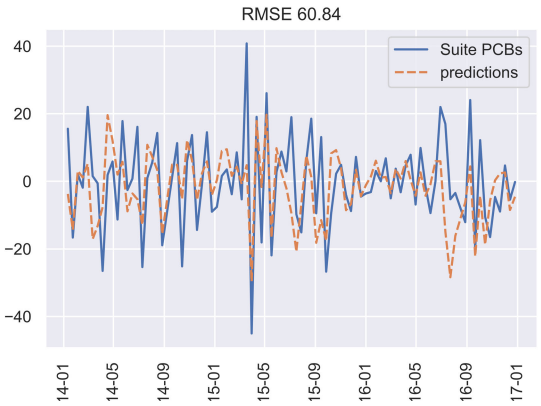


FIGURE 6. ARIMA prediction of PCBs concentration.

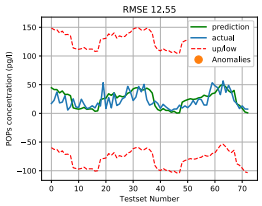
In order to make our model have better generalization ability, we applied regularization to the model. Fig.7(a) and Fig.7(b) respectively show the performance of Lasso Regression and Ridge Regression on the testset. The value of RMSE on Ridge Regression is 13.78, and the value of RMSE on Lasso Regression is 12.55. Both models are better than the regression model without regularization.

D. XGBoost RESULTS

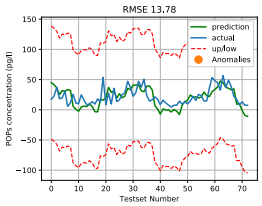
Fig.7(c) shows the performance of the XGBoost model in the test set. The effect is not apparent when predicting a sharp increase or decrease in PCBs concentration. However, compared with the ARIMA model, XGBoost is outstanding in predicting some stable data. The RMSE value of the XGBoost model is reduced from 60.84 to 16.49, which significantly improves the performance.

E. ENSEMBLE LSTM RESULTS

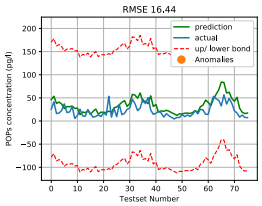
Our goal is to achieve state-of-art prediction performance of the LSTM model and give the predictive uncertainty.



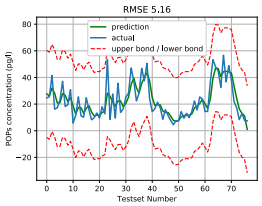
(a) Lasso Prediction of PCBs Concentration Results.



(b) Ridge Prediction of PCBs Concentration Results.



(c) XGBoost Prediction of PCBs Concentration Results.



(d) LSTM Prediction of PCBs Concentration Results.

FIGURE 7. **Prediction of PCBs concentration results. x-axis denotes time series, and y-axis denotes POPs concentration. Blue line represents true values, and the green line represents prediction values.**

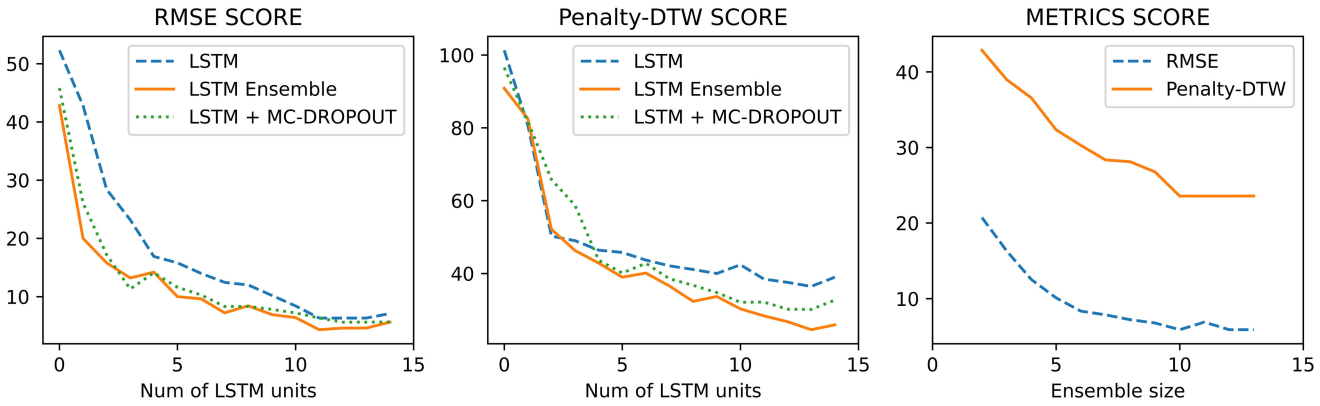


FIGURE 8. **Measuring predictive ability under RMSE and Penalty-DTW evaluation metrics. In left plot, x-axis denotes number of LSTM units, y-axis denotes RMSE scores. Second plot shows the effect of training 3 models using Penalty-DTW. In third plot, x-axis denotes ensemble size (number of networks in the ensemble), y-axis denotes both RMSE scores and Penalty scores. Ensemble LSTM significantly outperform MC-dropout and LSTM performance with the corresponding M in terms of all 2 metrics.**

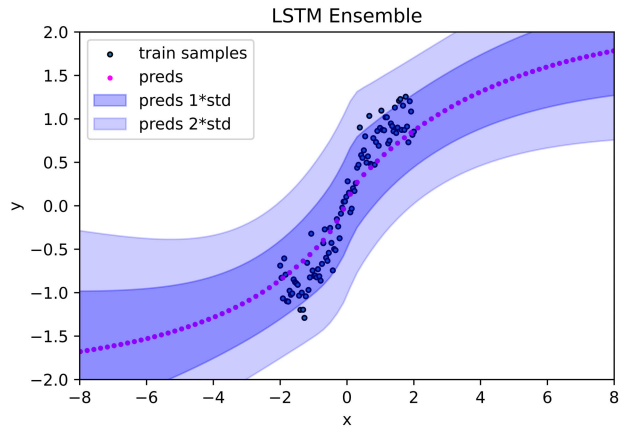


FIGURE 9. **Results on part of test datasets using ensemble LSTM. x-axis denotes x. On the y-axis, Red dots represent prediction results, the blue shadow represents the Standard deviation of prediction results, that is, the uncertainty.**

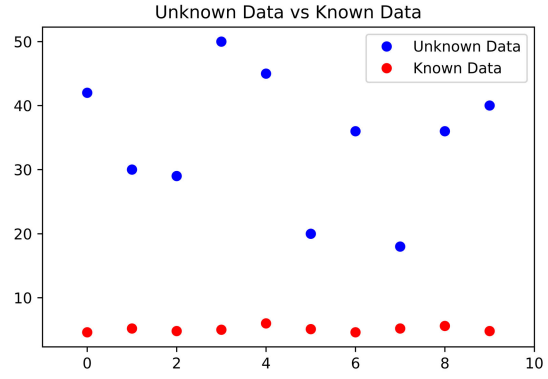
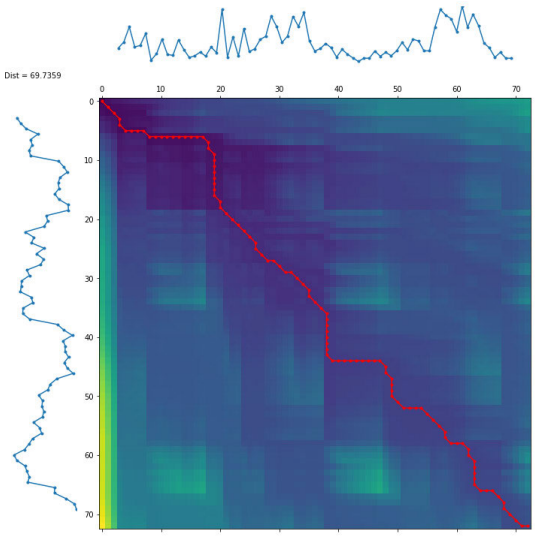


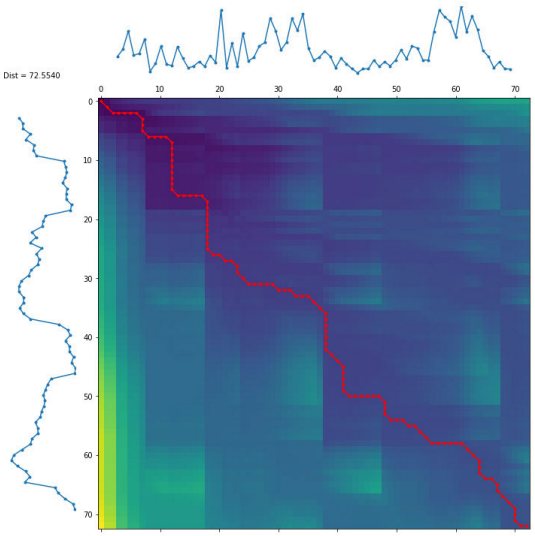
FIGURE 10. **Uncertainty results of unknown data and known data. x-axis denotes x, y-axis denotes uncertainty values. Blue dots represent unknown inputs, and red dots represent known inputs.**

Datasets are split into 5 train-test folds. For LSTM, We used a vanilla LSTM with 2-hidden layers and tanh non-linearities with batch normalization. One layer is the LSTM layer with different units; the other layer is the Dense layer. We set the number of LSTM layer units as a variable to seek the optimal network architecture. We trained with Adam optimizer with a constant learning rate of 0.001 and weight decay 10^{-5}. For MC-dropout, we added dropout after each nonlinearity with 0.1 as the dropout rate. For ensemble LSTM, we set default ensemble size values are M = 5. We trained for 100 epochs. When measuring the predictive uncertainty, the experiment's commonly used way is to train an ensemble model and obtain predictions. Then use the empirical variance according to the predictions as approximate uncertainty.

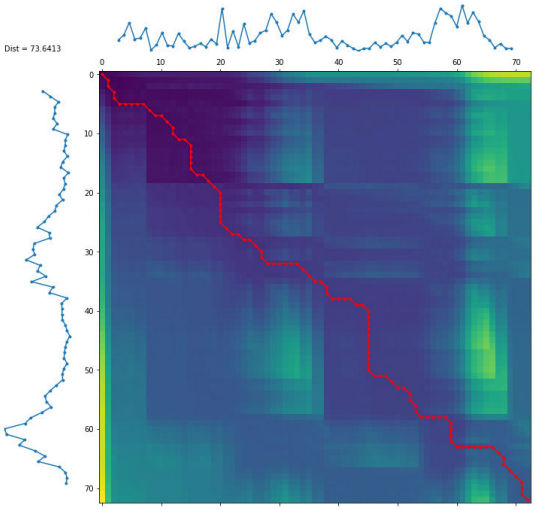
Results are shown in Fig.8. We observe that increasing the number of LSTM units significantly improves the performance in terms of RMSE score and Penalty-DTW score.



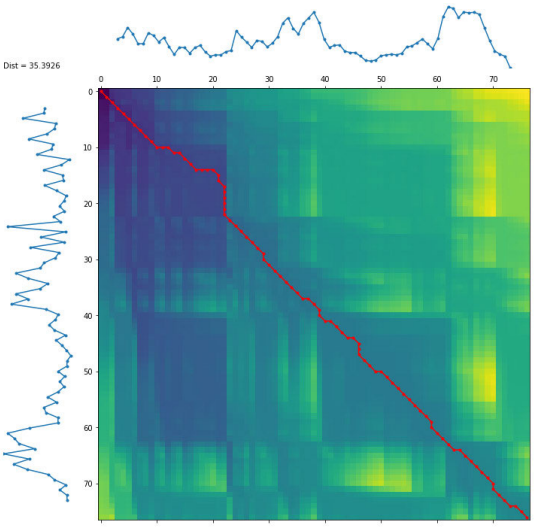
(a) DTW Results of S1 and S2



(b) DTW Results of S1 and S3



(c) Penalty-DTW Results of S1 and S3



(d) Penalty-DTW Results of S1 and S3

FIGURE 11. Evaluating Penalty-DTW Results, (a) showed the DTW results of series S1 and S2, (b) showed the DTW results of series S1 and S3. The images represent cost matrix. The curves on the left and top of the image denote the squared Euclidean distance for each time point between both time series. The red line in the graph indicated the optimal path.

Meanwhile, under RMSE and Penalty-DTW score, ensemble LSTM leads to better performance than LSTM. LSTM also performs much better than MC-dropout in terms of both metrics.

Fig.7(d) shows the performance of the LSTM model on the test set. We observe that LSTM can predict the static data well and gets better performance for some peak data. The RMSE of the test set of the LSTM model is 5.16, which dramatically improves the performance of XGBoost.

To measure the uncertainty of prediction results, we experimented with an ensemble LSTM model trained before. The performance of the ensemble LSTM model on testsets is shown in Fig.9. The larger the area of the blue shadow, the higher the uncertainty of the prediction results. We observe that the shadow area of the prediction result is in an acceptable range, which indicates that the uncertainty of the prediction result is small and the reliability of the prediction is high.

To assess the model's robustness to known data and unknown data, we used the trained model and selected two groups of input data to evaluate predictive uncertainty; one is from our training data, the other is generated randomly. From Fig.10, we observe that the standard deviation of training data is much smaller than the standard deviation of random data. This result shows that ensemble LSTM the model can express higher uncertainty to unknown input data.

F. MODEL SELECTION

Fig.11 shows the similarity between prediction and actual in four models. In the training step, the LSTM

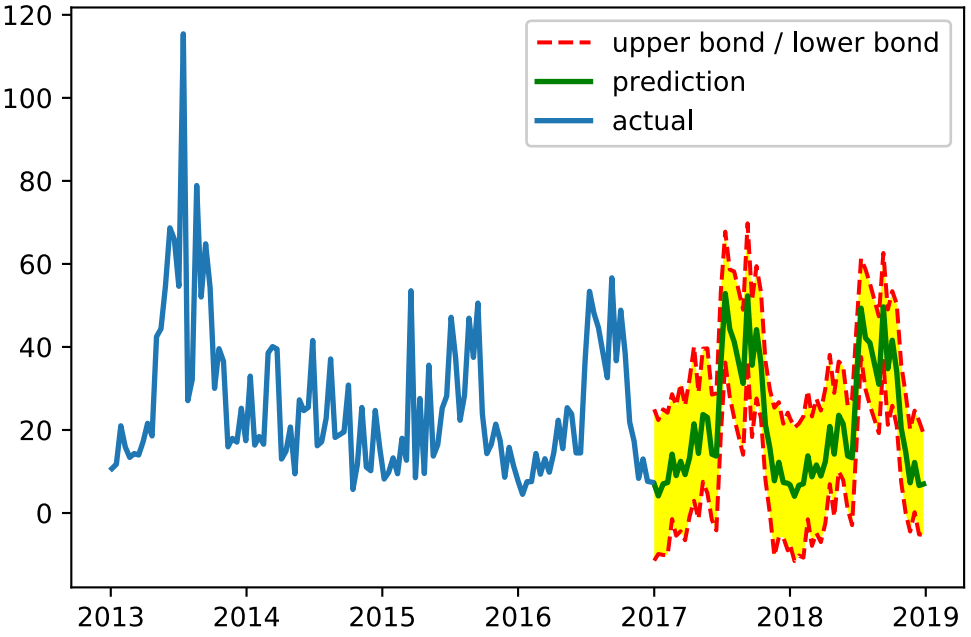


FIGURE 12. Prediction of PCBs.

model produces a better prediction of POPs concentration (*RMSE* = 3.25, *Penalty-DTW* = 30.14) than did Ridge Regression(*RMSE* = 10.78, *Penalty-DTW* = 61.74), Lasso Regression(*RMSE* = 10.55, *Penalty-DTW* = 62.55) and XGBoost(*RMSE* = 14.22, *Penalty-DTW* = 68.64).

The goodness-of-fit of the model illustrates how well the models fit the training dataset. The prediction and generalization abilities of the model cannot be evaluated using the goodness-of-fit of the model because it is measured by the data that were used to calibrate the model [47]. The test step accurately takes into account the prediction ability of the model. Results reflect the performance of the LSTM(*RMSE* = 5.16, *Penalty-DTW* = 35.39) was better than Ridge Regression(*RMSE* = 13.78, *Penalty-DTW* = 69.74), Lasso Regression(*RMSE* = 12.55, *Penalty-DTW* = 72.55) and XGBoost model(*RMSE* = 16.44, *Penalty-DTW* = 73.64).

For the Root Mean Square Error criteria, we conclude that the LSTM model's performance and the LR model on the dataset are better than other models. To more accurately verify the similarity between the predicted time series and the actual time series, we adopt an improved method based on DTW. The results show that the ensemble LSTM model has the best prediction ability.

G. MODEL PREDICTION

The predicted results of the model are shown in the Fig.12. It can be seen from the figure that the overall concentration of PCBs will gradually decrease in the next two years. In a year, the concentration from July to October will be higher than that in other months, and the concentration from December to March will be lower than that in other months.

VI. CONCLUSION AND FUTURE WORK

This study proposed a verification method based on the dynamic time warping (Penalty-DTW) algorithm. We proved that the algorithm is superior to the traditional Root Mean Square difference in the similarity comparison of time series. Meanwhile, we combined the LSTM model and ensemble methods to measure uncertainty and made an accurate and confident prediction. We found that the ensemble LSTM model and LR model have good performance in predicting PCBs concentration in Eagle Harbor through the similarity analysis. It can be reached from the results that the concentration of PCBs has a stable trend in recent years and is decreasing year by year. However, there is a considerable correlation between PCBs concentration and season. The concentration of PCBs will increase from July to October and reach the peak value around July. The concentration of PCBs will decrease from December to March, an excellent reference for environmental pollution control. It is required for us to study further and explore this time series model's application in other regions and pollutants.

Future research in this field can be carried out from the following two aspects. Firstly, we will continue to study the uncertainty in the prediction of pollutant concentration, and use different evaluation criteria, put forward different methods to measure the uncertainty of the prediction results In the following research. We can try to combine the MC-dropout method and deep ensemble method to consider the uncertainty comprehensively. Secondly, the study of pollutants in the Great Lakes is only limited to certain pollutants in one region. We can expand the pollutants to other regions in the following study.

ACKNOWLEDGMENT

The authors would like to thanks for all anonymous reviewers for their very insightful comments and constructive suggestions to polish this paper in high quality.

REFERENCES

[1] C. C. Lee, B. B. Barnes, S. C. Sheridan, E. T. Smith, C. Hu, D. E. Pirhalla, V. Ransibrahmanakul, and R. Adams, "Using machine learning to model and predict water clarity in the great lakes," *J. Great Lakes Res.*, vol. 46, no. 6, pp. 1501–1510, Dec. 2020.

[2] M. Eriksen, S. Mason, S. Wilson, C. Box, A. Zellers, W. Edwards, H. Farley, and S. Amato, "Microplastic pollution in the surface waters of the Laurentian Great Lakes," *Mar. Pollut. Bull.*, vol. 77, nos. 1–2, pp. 177–182, Dec. 2013.

[3] S. A. Mason, L. Kammin, M. Eriksen, G. Aleid, S. Wilson, C. Box, N. Williamson, and A. Riley, "Pelagic plastic pollution within the surface waters of Lake Michigan, USA," *J. Great Lakes Res.*, vol. 42, no. 4, pp. 753–759, Aug. 2016.

[4] A. K. Baldwin, S. R. Corsi, and S. A. Mason, "Plastic debris in 29 Great Lakes tributaries: Relations to watershed attributes and hydrology," *Environ. Sci. Technol.*, vol. 50, no. 19, pp. 10377–10385, Oct. 2016.

[5] K. C. Jones and P. de Voogt, "Persistent organic pollutants (POPs): State of the science," *Environ. Pollut.*, vol. 100, nos. 1–3, pp. 209–221, 1999.

[6] D. Asteriou and S. G. Hall, "ARMA models and the Box–Jenkins methodology," in *Applied Econometrics*. London, U.K.: Macmillan Education, 2016, pp. 275–296.

[7] S. Ahmad, I. H. Khan, and B. P. Parida, "Performance of stochastic approaches for forecasting river water quality," *Water Res.*, vol. 35, no. 18, pp. 4261–4266, Dec. 2001.

[8] L. Y. Siew, L. Y. Chin, and P. M. J. Wee, "ARIMA and integrated ARFIMA models for forecasting air pollution index in Shah Alam, Selangor," *Malaysian J. Anal. Sci.*, vol. 12, no. 1, pp. 257–263, 2008.

[9] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, Jul. 2015.

[10] Venier, M, and R. A. Hites, "Time trend analysis of atmospheric POPs concentrations in the Great Lakes region since 1990," *Environ. Sci. Technol.*, vol. 44, p. 8050–8055, Nov. 2010.

[11] M. C. Maniquiz, S. Lee, and L.-H. Kim, "Multiple linear regression models of urban runoff pollutant load and event mean concentration considering rainfall variables," *J. Environ. Sci.*, vol. 22, no. 6, pp. 946–952, Jun. 2010.

[12] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd Int. Conf. Knowl. Discovery Data Mining*, San Francisco, CA, USA, Aug. 2016, pp. 785–794.

[13] J. Ma, J. C. P. Cheng, Z. Xu, K. Chen, C. Lin, and F. Jiang, "Identification of the most influential areas for air pollution control using XGBoost and grid importance rank," *J. Cleaner Prod.*, vol. 274, Nov. 2020, Art. no. 122835.

[14] M. Zamani Joharestani, C. Cao, X. Ni, B. Bashir, and S. Talebiesfandarani, "PM2.5 prediction based on random forest, XGBoost, and deep learning using multisource remote sensing data," *Atmosphere*, vol. 10, no. 7, p. 373, Jul. 2019.

[15] W. Qiao, K. Huang, M. Azimi, and S. Han, "A novel hybrid prediction model for hourly gas consumption in supply side based on improved whale optimization algorithm and relevance vector machine," *IEEE Access*, vol. 7, pp. 88218–88230, 2019.

[16] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[17] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, Aug. 2018.

[18] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[19] W. Qiao, M. Khishe, and S. Ravakhah, "Underwater targets classification using local wavelet acoustic pattern and multi-layer perceptron neural network optimized by modified whale optimization algorithm," *Ocean Eng.*, vol. 219, Jan. 2021, Art. no. 108415.

[20] S. Siami-Namini, N. Tavakoli, and A. Siami Namin, "A comparison of ARIMA and LSTM in forecasting time series," in *Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Orlando, FL, USA, Dec. 2018, pp. 1394–1401, doi: 10.1109/ICMLA.2018.00227.

[21] Y. Wang, J. Zhou, K. Chen, Y. Wang, and L. Liu, "Water quality prediction method based on LSTM neural network," in *Proc. 12th Int. Conf. Intell. Syst. Knowl. Eng. (ISKE)*, Nanjing, China, Nov. 2017, pp. 1–5.

[22] P. Liu, J. Wang, A. Sangaiah, Y. Xie, and X. Yin, "Analysis and prediction of water quality using LSTM deep neural networks in IoT environment," *Sustainability*, vol. 11, no. 7, p. 2058, Apr. 2019.

[23] J. Zhao, F. Deng, Y. Cai, and J. Chen, "Long short-term memory—Fully connected (LSTM-FC) neural network for $PM_{2.5}$ concentration prediction," *Chemosphere*, vol. 220, pp. 486–492, Apr. 2019.

[24] W. Qiao and Z. Yang, "Forecast the electricity price of U.S. using a wavelet transform-based hybrid model," *Energy*, vol. 193, Feb. 2020, Art. no. 116704.

[25] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in AI safety," 2016, *arXiv:1606.06565*. [Online]. Available: http://arxiv.org/abs/1606.06565

[26] D. J. C. MacKay, "A practical Bayesian framework for backpropagation networks," *Neural Comput.*, vol. 4, no. 3, pp. 448–472, May 1992.

[27] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *J. Amer. Stat. Assoc.*, vol. 112, no. 518, pp. 859–877, Apr. 2017.

[28] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2015, pp. 1050–1059.

[29] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," 2016, *arXiv:1612.01474*. [Online]. Available: http://arxiv.org/abs/1612.01474

[30] T. Chai and R. R. Draxle, "Root mean square error (RMSE) or mean absolute error (MAE)?—Arguments against avoiding RMSE in the literature," *Geoscientific Model. Develop.*, vol. 7, pp. 1247–1250. Jun. 2014.

[31] B.-K. Yi, H. V. Jagadish, and C. Faloutsos, "Efficient retrieval of similar time sequences under time warping," in *Proc. 14th Int. Conf. Data Eng.*, Orlando, FL, USA, Feb. 1998, pp. 201–208, doi: 10.1109/ICDE.1998.655778.

[32] C. A. Ratanamahatana and E. Keogh, "Making time-series classification more accurate using learned constraints," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2004, pp. 11–22.

[33] T. M. Rath and R. Manmatha, "Word image matching using dynamic time warping," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Los Alamitos, CA, USA, Dec. 2003, p. 521.

[34] L. Muda, M. Begam, and I. Elamvazuthi, "Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques," *Comput. Sci.*, vol. 2, no. 3, pp. 138–143, Mar. 2010.

[35] S. Sempena, N. Ulfa Maulidevi, and P. Ruswono Aryan, "Human action recognition using dynamic time warping," in *Proc. Int. Conf. Electr. Eng. Informat.*, Jul. 2011, pp. 1–5.

[36] T. E. Keskin, M. Düğenci, and F. Kaçaroğlu, "Prediction of water pollution sources using artificial neural networks in the study areas of Sivas, Karabük and Bartin (Turkey)," *Environ. Earth Sci.*, vol. 73, no. 9, pp. 5333–5347, May 2015.

[37] W. Deng, G. Wang, X. Zhang, Y. Guo, and G. Li, "Water quality prediction based on a novel hybrid model of ARIMA and RBF neural network," in *Proc. IEEE 3rd Int. Conf. Cloud Comput. Intell. Syst.*, Shenzhen, China, Nov. 2014, pp. 33–40.

[38] M. Liu and J. Lu, "Support vector machine—An alternative to artificial neuron network for water quality forecasting in an agricultural nonpoint source polluted river?" *Environ. Sci. Pollut. Res.*, vol. 21, no. 18, pp. 11036–11053, Sep. 2014.

[39] *Integrated Atmospheric Deposition Network (IADN) Data Visualization Tool IADN Data Viz*, Indiana Univ., Bloomington, IN, USA, 2020.

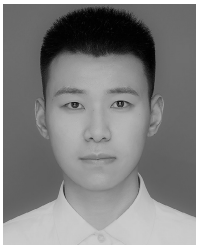
[40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, and O. Grisel, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[41] X. Yan and X. G. Su, "Regression analysis," in *Linear Regression Analysis: Theory and Computing*. Singapore: World Scientific, 2009.

[42] H. L. Seal, "Studies in the history of probability and statistics. XV: The historical development of the gauss linear model," *Biometrika*, vol. 54, pp. 1–24, Jun. 1967.

[43] W. Qiao, H. Moayedi, and L. K. Foong, "Nature-inspired hybrid techniques of IWO, DA, ES, GA, and ICA, validated through a K-fold validation process predicting monthly natural gas consumption," *Energy Buildings*, vol. 217, Jun. 2020, Art. no. 110023.
[44] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[45] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Phys. D, Nonlinear Phenomena*, vol. 404, Mar. 2020, Art. no. 132306.
[46] Y. Wu, M. Yuan, S. Dong, L. Lin, and Y. Liu, "Remaining useful life estimation of engineered systems using vanilla LSTM neural networks," *Neurocomputing*, vol. 275, pp. 167–179, Jan. 2018.
[47] J. Henseler and S. Marko, "Goodness-of-fit indices for partial least squares path modeling," *Comput. Statist.*, vol. 28, pp. 565–580, Apr. 2013.
[48] D. A. Dickey and W. A. Fuller, "Distribution of the estimators for autoregressive time series with a unit root," *J. Amer. Stat. Assoc.*, vol. 74, no. 366, pp. 427–431, Jun. 1979.

CHUNXUE WU (Member, IEEE) received the Ph.D. degree in control theory and control engineering from the China University of Mining and Technology, Beijing, China, in 2006. He is currently a Professor with the Computer Science and Engineering and Software Engineering Division, School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, China. His research interests include wireless sensor networks, distributed and embedded systems, wireless and mobile systems, and networked control systems.



BIN LI is currently pursuing the master's degree in computer science and technology with the University of Shanghai for Science and Technology. He is also engaged in big data and artificial intelligence research under the guidance of Prof. C. Wu.

NAIXUE XIONG (Senior Member, IEEE) received the Ph.D. degree in sensor system engineering from Wuhan University, in 2007, and the Ph.D. degree in dependable communication networks from the Japan Advanced Institute of Science and Technology, in 2008. He is currently an Associate Professor (fifth year) with the Department of Mathematics and Computer Science, Northeastern State University, OK, USA. Before he attended Northeastern State University, he worked with Georgia State University, Wentworth Technology Institution, and Colorado Technical University (Full Professor about five years) about ten years. He has published over 200 international journal articles and over 100 international conference papers. Some of his works were published in the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE or ACM TRANSACTIONS, ACM Secom Workshop, IEEE INFOCOM, ICDCS, and IPDPS. His research interests include cloud computing, security and dependability, parallel and distributed computing, networks, and optimization theory.

Dr. Xiong has received the Best Paper Award in the 10th IEEE International Conference on High Performance Computing and Communications (HPCC-08) and the Best Student Paper Award in the 28th North American Fuzzy Information Processing Society Annual Conference (NAFIPS2009). He has been the General Chair, the Program Chair, the Publicity Chair, a Program Committee Member, and an Organizing Committee Member of over 100 international conferences, and a Reviewer of about 100 international journals, including IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS (Part: A/B/C), IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON MOBILE COMPUTING, and IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS. He is serving as the Editor-in-Chief and an Associate Editor or an Editor Member for over ten international journals, including an Associate Editor for IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS: SYSTEMS, *Information Science*, the Editor-in-Chief for *Journal of Internet Technology* (JIT) and *Journal of Parallel and Cloud Computing* (PCC), and a Guest Editor for over ten international journals, including *Sensors* journal, *WINET*, and *MONET*.