Air Quality Prediction based on LSTM-Kalman Model

Xijuan Song[1,2], Jijiang Huang[1], Dawei Song[1,2]

1. Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences

2. University of Chinese Academy of Sciences

Xi'an, Beijing, China

songxijuan18@mails.ucas.ac.cn, huangjijiang@opt.ac.cn, songdawei18@mails.ucas.ac.cn

***Abstract*—In this paper, a time prediction model based on LSTM and Kalman filtering, LSTM-Kalman model, is proposed for the prediction of time series data with long-term and short-term characteristics. The LSTM-Kalman model uses the unique memory feature of LSTM to "store" the information contained in the pre-order data. Then it is used to obtain the underlying time series of the predicted problem in subsequent processing. Next Kalman filtering model dynamically adjusts the basic time data sequence obtained by LSTM processing. In the end, we will get the adjusted predicted value. Here we select the observation indicators in the air quality data set and builds training and test set samples to train the LSTM-Kalman model and test the performance of the sample model. As a comparison we also trained the performance of the LSTM model.In this work, we compare the RMSE(Root Mean Square Error) and R-Squared (determination coefficient) of the LSTM model with the LSTM-Kalman model predictions. The results show that the LSTM-Kalman model is better than the LSTM model, and the LSTM-Kalman model has a better fit to the predicted values of the model data.**

Keywords—Air quality prediction; LSTM model; Kalman filtering; LSTM-Kalman model; Sequence learning

I. INTRODUCTION

In a certain area, the composition of pollutants in the air is usually unchanged, but the data indicators of a certain pollutant are changing at different times. Therefore, indicator data of the pollutant are a dynamically changing time series data. In the long term, each time series has a certain trend of change. In the short term, different time series has their own changing trends. The Kalman filter is an auto-regressive filter using a recursive algorithm. It is able to estimate the state of a dynamic system through a series of incomplete and noisy-containing measurements. The state space method is used to design the filter in the time domain, which is suitable for the estimation of multidimensional stochastic processes. Kalman filtering iterates and recurses with known models, and it can be used not only as a state estimator but also as a state predictor. The core task of state estimation is to design the filter. The Kalman filter is widely used as a linear optimal estimator. The system equations and observation equations in reality are mostly nonlinear. There are many algorithms for nonlinear state estimation, such as sliding time domain estimation, particle filtering, integrated Kalman filter, undirected Kalman filter and extended Kalman filter. Kalman filtering also has several obvious drawbacks. The most significant of these is its dependence on the dynamic model. Therefore, we consider using LSTM to improve the Kalman filter to predict the value of a certain pollutant [1~3]. On the one hand, it can get rid of its dependence on the dynamic model, so that the model can learn from the data. On the other hand, it enhances the long-term prediction ability of the Kalman filter. The LSTM-Kalman is superior to the independent Kalman filter and the independent LSTM in time regularization.

There are several variants of LSTM [4~6]. Unfortunately, the results show that no single variant can significantly improve the standard LSTM architecture. Therefore, we chose to use the standard LSTM architecture combined with Kalman filter to analyze the air quality observations.

II. BASIC THEORY

A. Kalman Filtering [7~14]

Kalman filtering introduces state space theory into the mathematical modeling process of physical systems. It uses a recursive method to solve the linear filtering problem of discrete data and not all the previous data are needed. Instead, Kalman filtering uses the previous estimate and the most recent observation to estimate the current value of the signal. Simply speaking, the Kalman filtering process mainly consists of two steps: estimation of state variables and correction of state variables. The specific mathematical modeling process is as follows:

1)Estimation of state variables

First, the current time state quantity is pre-estimated from the estimated value of the previous time and the controllable input to the system.

$$X_{(k|k-1)}=AX_{(k-1|k-1)}+BU_{(K)} \tag{1}$$

978-1-5386-8178-7/19/$31.00 ©2019 IEEE

Where $X_{(k-1|k-1)}$ represents the estimated value of the previous moment, and U(k) represents the control input of the system, $X_{(k|k-1)}$ represents the pre-estimated value of the state quantity estimated from the previous moment, A represents a state transition matrix from k-1 to k, and B represents a conversion factor between the control input and the state quantity, both of which are determined by the nature of the system.

Then, the mean square error matrix of the current time is pre-estimated by the mean square error matrix of the previous moment.

$$P_{(k|k-1)}=AP_{(k-1|k-1)}A'+Q \quad (2)$$

Where $P_{(k-1|k-1)}$ is the mean squared error estimate of the previous moment, A' represents the transpose of matrix A, and Q represents the mean square error matrix of process noise.

2)Correction of state variables

First, the estimated value of the current time is corrected by the difference between the measured value at the current time and the estimated measured value.

$$X_{(k|k)}= X_{(k|k-1)}+Kg_{(k)}(Z_{(k)}-HX_{(k|k-1)}) \quad (3)$$

Where $Z_{(k)}$ is the measured value, $HX_{(k|k-1)}$ is the product of the estimated state quantity and the measurement system parameter, and the product result is the system measurement value without considering the noise. $Kg_{(k)}$ is called the gain factor of the system.

Second, the mean squared error value of the current time is updated from the estimated value of the state quantity at the current time of the update, and an expression of the gain factor under the minimum mean square error criterion is obtained.

$$P_{(k|k)}=E\{[Xs_{(k|k)}-X_{(k|k)}][Xs_{(k|k)}-X_{(k|k)}]\quad (4)$$

For (4) to find the first derivative of $Kg_{(k)}$, and make it 0. You can get the value of Kg(k) corresponding to the minimum $P_{(k|k)}$:

$$Kg_{(k)}=P_{(k|k-1)}H'/(HP_{(k|k-1)}H'+R) \quad (5)$$

Substituting $Kg_{(k)}$ into the above equation and finding the relation between $P_{(k|k)}$ and $Kg_{(k)}$, the estimated mean square error of the current time is obtained:

$$P_{(k|k)}=(I-Kg_{(k)}H)P_{(k|k-1)} \quad (6)$$

When the system enters the k+1 state, $P_{(k|k)}$ is the $P_{(k-1|k-1)}$ appearing in the above equations. In this way, the algorithm can proceed to the auto-regressive operation.

B. LSTM Theory [15~19]

As a special kind of recurrent neural network, LSTM also has a recursive structure similar to RNN. However, unlike the naive RNN network, LSTM uses four processing modules that interact in a special way to implement long-term dependencies. The LSTM network module is shown in Fig.1:

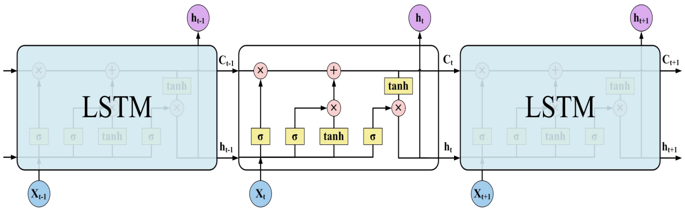

Fig. 1. LSTM network expansion structure.

1)Forward propagation process

Unlike RNN, the hidden layer of LSTM adds a Ct called cell state in addition to state ht.In addition, the gate structure is also designed in LSTM to increase or delete the information of the cell state. Specifically, each gate is a fully connected layer, and then the activation function controls the passage of information according to the activation value.

The forward propagation steps of LSTM are as follows:

First, the forgotten gate layer determines what information is discarded from the LSTM cell state. The output of the forgotten gate can be expressed as:

$$f_t=\sigma(W_f\cdot[h_{t-1},x_t]+b_f) \quad (7)$$

$\sigma(\cdot)$ in the equation is the sigmoid function, and $[h_{t-1},x_t]$ represents the operation of concatenating two vectors into one longer vector.

Next, determine what information is stored in the LSTM unit, which contains two parts. First, the sigmoid layer of the input gate results as information to be updated, and secondly, a new candidate value vector C_t is created by the tanh layer and added to the cell state. The new state of the cell will be multiplied by the old state C_{t-1} and f_t to obtain the information after forgetting, plus it · Ct to achieve the update. The expression of the above steps is as follows:

$$i_t=\sigma(W_i\cdot[h_{t-1}, x_t]+b_i) \quad (8)$$

$$C'_t=\tanh(W_C\cdot[h_{t-1}, x_t]+b_C) \quad (9)$$

$$C_t = ft \circ C_{t-1}+i_t \circ C'_t \quad (10)$$

The ◦ symbol in the above expression represents the element-by-element multiplication of two vectors.

Finally, the status of the LSTM cell unit is updated by the output gate. First, using a sigmoid layer to determine which parts of the current cell state require output; then, the cell state is processed by tanh to obtain a value between -1 and 1, and it is multiplied by the sigmoid output to obtain the output value. The arithmetic expression of the above processing is as follows:

$$o_t=\sigma(Wo\cdot[h_{t-1},x_t]+bo) \quad (11)$$

$$h_t=o_t \circ \tanh(C_t) \quad (12)$$

2)Model weight update

Like other neural network models, LSTM also uses a back propagation algorithm during training, which is not repeated here.

III. Model and Algorithm

In this paper, the long-short-term memory recurrent neural network LSTM is used as a static prediction model to predict the value of the air pollution indicator in time series as the basic time series. The Kalman filter is used as a dynamic adjustment model to dynamically adjust the base time.

A schematic diagram of the LSTM static prediction model expanded by time is shown in Fig.2, where gn represents the observed value at time n. From the vertical perspective, each node represents a layer of the neural network, namely the input layer, the hidden layer and the output layer. From the horizontal direction, each node represents its calculation process at different times. First, we input the meta-data of the first moment into the LSTM neural network and output it via the hidden layer. Secondly, the output of the hidden layer and the meta-data of the second moment are used as the input of the LSTM neural network calculation at the next moment. By analogy, the meta-data at the nth time are calculated and output to the output layer via the hidden layer. Finally, the output layer outputs the predicted value.

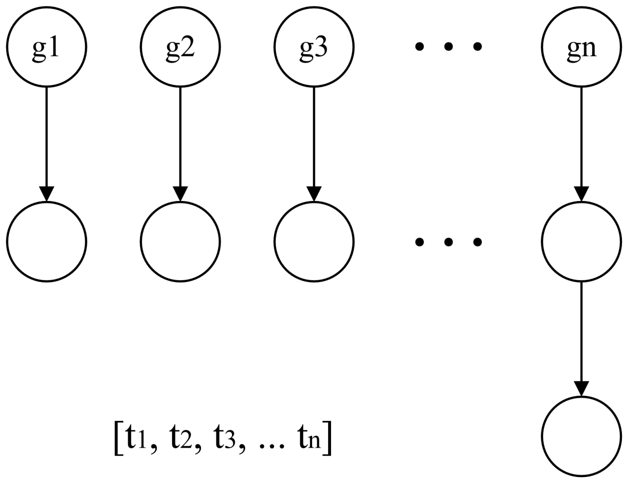

Fig. 2. LSTM static prediction.

The Kalman filter dynamic adjustment model is shown in Fig.3. When predicting a certain pollutant indicator data, the Kalman filter calculates the optimal estimate of the nth time based on the observed value gn of the contaminant at n and the previous n-1 predicted values.Further, an adjustment value at the nth time is obtained, and the subsequent prediction value is updated according to the adjustment value.

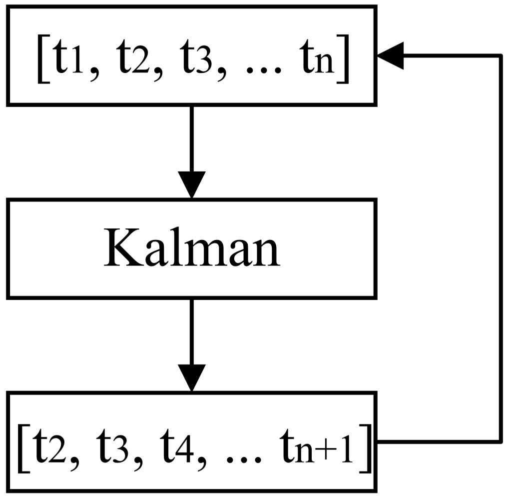

Fig. 3. Kalman filter dynamic adjustment.

The algorithm steps are shown in Table I.

TABLE I. ALGORITHM STEPS

```
dataset = DataSetCollection().load_dataset(DataSetName.AirQuality)
scaler = MinMaxScaler(feature_range=(0, 1))
dataset = scaler.fit_transform(dataset)
train_size = int(len(dataset) * 0.67)
test_size = len(dataset) - train_size
train, test = dataset[0:train_size, :], dataset[train_size:len(dataset), :]
look_back = 3
trainX, trainY = create_dataset(train, look_back)
testX, testY = create_dataset(test, look_back)
model = Sequential()
model.add(LSTM(4, input_shape=(None, 1)))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')
model.fit(trainX, trainY, epochs=100, batch_size=1, verbose=2)
trainPredict = model.predict(trainX)
testPredict = model.predict(testX)
kf = KalmanFilter(initial_state_mean= testY[0][0])
kf = kf.em(observations, n_iter=5, em_vars='all')
measurements_predicted = (kf.smooth(testPredict[:, 0])[0])[:, 0]
```

First load and normalize the dataset, then split the dataset into 67% train data sets and 33% test data sets. We divide the data set into two columns, the first column is the pollution amount at time t, and the second column is the pollution amount at time t+1. Next, create and fit the LSTM network. Timesteps means the recursive relationship between each input data of the LSTM and the previous data input. In LSTM, there are one input in the input layer and four neurons in the hidden layer. The activation function is sigmoid, iterates 100 times, and batch size is 1. The output layer is the predicted value. The network layer is organized by sequential model of Keras, which is stacked by model.add() to form a model. After building the model, we need to use model. compile () method to compile the model. The loss function and optimizer are specified when compiling the model. Then we train the network according to batch iterations on the training data and make LSTM predictions. In the end, create and fit the Kalman Filter by using KalmanFilter.em() to learn the parameters, and use KalmanFilter.smooth() to predict the hidden state sequence.

IV. ALGORITHM ANALYSIS AND EVALUATION PRINCIPLES

In a certain area, the composition of pollutants in the air is usually unchanged, but the data indicators of a certain pollutant are changing at different times.Therefore, the indicator data of the pollutant are a dynamically changing time series data. In the long term, each time series has a certain trend of change. In the short term, different time series has their own changing trends.

In this paper, the long-short-term memory recurrent neural network LSTM is used as a static prediction model to predict the value of the air pollution indicator in time series as the basic time series. The Kalman filter is used as a dynamic adjustment model to dynamically adjust the base time.

The prediction method is divided into two parts: static prediction and dynamic adjustment. Firstly, the static prediction model is used to predict the predicted value of a certain pollutant at the current time, and then the dynamic adjustment model is used to dynamically adjust the static prediction indicator according to the observed value of the pollutant.

In order to analyze the performance of the LSTM-kalman model prediction, we select the RMSE (Root Mean Square Error) of the LSTM-kalman model and the LSTM model prediction value and the R-Squared (determination coefficient) of the model prediction value as the evaluation index.

Among them, RMSE is the mean of the square root of the error between the predicted value and the true value, which is used to measure the deviation between the observed value and the true value. The smaller the RMSE, the better the fit. R-Squared interprets the ability of the model to predict new observations. The data are used to characterize the quality of a fitting process. The closer the R-square is to 1, the better the fit.

V. RESULTS AND ANALYSIS

We write programs in Python and use the Keras framework to build an LSTM network. We also use machine learning library of scikit-learn for data processing and regression analysis. The pykalman library implements kalman filtering. The predict method in the kalman class gets the predicted value matrix of the state and is used to get the predicted value.

We use the Air Quality Data Set as the experimental data set and then select an observation. Two-thirds of the observation data set is used as a training set to train the estimated sample model, and 1/3 is used as a test set to test the performance of the training sample model.

Observations of indicators in samples, the LSTM model predictions, and the LSTM-Kalman model predictions as a function of time series are shown in Fig.4, Fig.5, and Fig.6.

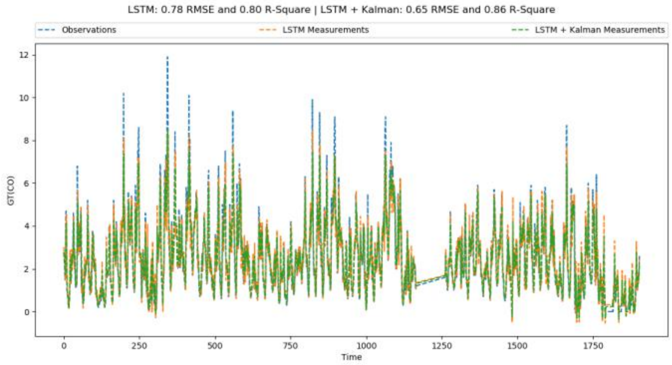

Fig. 4. Observations of CO, LSTM model predictions of CO and the changes of the LSTM-Kalman model predictive value of CO over time series.

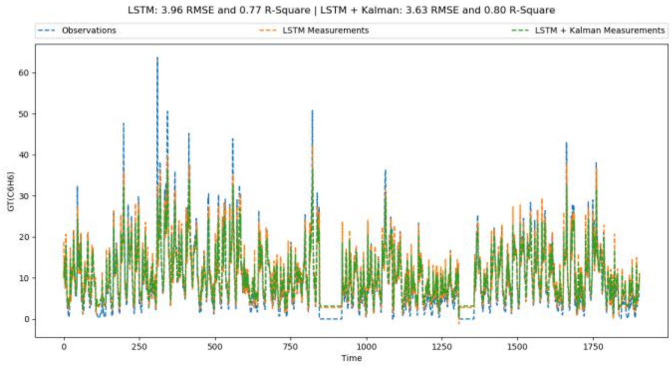

Fig. 5. Observations of C_6H_6, LSTM model predictions of C_6H_6 and the changes of the LSTM-Kalman model predictive value of C_6H_6 over time series.

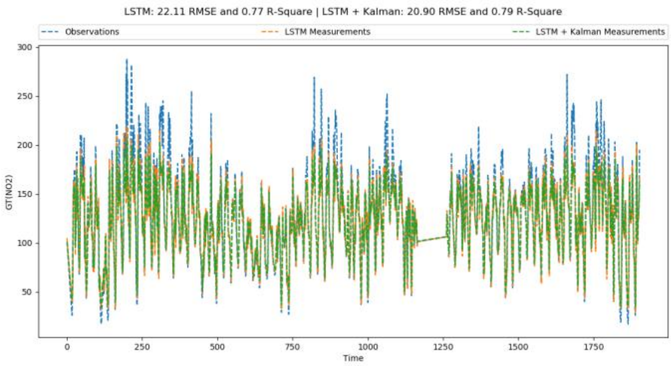

Fig. 6. Observations of NO_2, LSTM model predictions of NO_2 and the changes of the LSTM-Kalman model predictive value of NO_2 over time series.

The RMSE of the model predictions is shown in Table II. The RMSE of CO, C_6H_6, and NO_2 for the predicted value of the static prediction model LSTM is higher than the RMSE of the LSTM-Kalman model. The static prediction model has a significant decrease in RMSE after Kalman dynamic adjustment. This shows that the trend of LSTM-Kalman model prediction is more stable than that predicted by the LSTM model. So the LSTM-Kalman model fits even better.

TABLE II. RMSE OF EACH MODEL

Model	CO	C_6H_6	NO_2
LSTM	0.78	3.96	22.11
LSTM-Kalman	0.65	3.63	20.90

The R-Squared of the model predictions is shown in Table III. The static prediction model LSTM predicts that the R-Squared of CO, C_6H_6, and NO_2 is lower than the R-Squared of the LSTM-Kalman model. The static prediction model has a significant rise in R-Squared after Kalman dynamic adjustment. This shows that LSTM-Kalman model has better explaining ability than LSTM model in prediction. The LSTM-Kalman model fits the data better.

TABLE III. R-SQUARED OF EACH MODEL

Model	CO	C_6H_6	NO_2
LSTM	0.80	0.77	0.77
LSTM-Kalman	0.86	0.80	0.79

VI. CONCLUSION

In this paper, we discuss the prediction model of air quality pollutant indicator and related technologies, and propose a prediction model of the indicator values of LSTM and Kalman filtering. The long-short-term memory recurrent neural network is used as the static prediction model. It summarizes the variation of values from historical data. The Kalman filtering model is used as a dynamic model, and the observation data is dynamically adjusted based on the predicted values of the long-short-term memory recurrent neural network.

We compare LSTM-Kalman with LSTM model predictions. The results show that the RMSE of the predicted value of the static prediction model LSTM is higher than the RMSE of the LSTM-Kalman model while the R-Squared is lower than the R-Squared of the LSTM-Kalman model. The static prediction model has a significant decrease in RMSE after Kalman dynamic adjustment and a significant increase in R-Squared. Therefore, the trend predicted by the LSTM-Kalman model is smoother and more explanatory than the trend predicted by the LSTM model. The LSTM-Kalman model fits even better.

REFERENCE

[1] Coskun H, Achilles F, DiPietro R, Navab N and Tombari F. Long Short-Term Memory Kalman Filters: Recurrent Neural Estimators for Pose Regularization[J]. 2017.

[2] Pérezortiz J A, Gers F A, Eck D and SchmidhuberJ. Kalman filters improve LSTM network performance in problems unsolvable by traditional recurrent nets[J]. Neural Networks, 2003, 16(2):241-250.

[3] Juan A. Pérez-Ortiz, Jürgen Schmidhuber, Gers F A and Eck D. Improving Long-Term Online Prediction with Decoupled Extended Kalman Filters[C]. International Conference on Artificial Neural Networks. Springer Berlin Heidelberg, 2002.

[4] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink and J. Schmidhuber. LSTM: A search space odyssey. IEEE Transactions on Neural Networks and Learning Systems, 2016.

[5] Chalvatzaki G, Koutras P, Hadfield J, et al. On-line Human Gait Stability Prediction using LSTMs for the fusion of Deep-based Pose Estimation and LRF-based Augmented Gait State Estimation in an Intelligent Robotic Rollator[J]. 2018.

[6] Alahi A, Goel K, Ramanathan V, et al. Social LSTM: Human Trajectory Prediction in Crowded Spaces[C].Computer Vision & Pattern Recognition. 2016.

[7] Yajin X, Zhihai X U, Huajun F, Qi L I, and Yueting C. Real-time Video Stabilization Based on Minimal Spanning Tree and Modified Kalman Filter[J]. Guangzi Xuebao/acta Photonica Sinica, 2018, 47(1): 110002.

[8] Da L, Xiaonan M, Xunjiang Z, et al. Iterative Estimation Algorithm of Star Tracker's Star Imaging Model[J]. Guangzi Xuebao/acta Photonica Sinica, 2019, 48(1).

[9] R. G. Krishnan, U. Shalit, and D. Sontag. Deep kalman filters. In NIPS Workshop on Advances in Approximate Bayesian Inference and Black Box Inference, 2015.

[10] Chen R. Mixture kalman filters[J]. Journal of the Royal Statistical Society B, 2010, 62(3):493-508.

[11] Wan E A, Rudolph V D M. The Unscented Kalman Filter[M].Kalman Filtering and Neural Networks. 2002.

[12] G. Welch and G. Bishop. An Introduction to the Kalman Filter. Technical Report 1, University of North Carolina at Chapel Hill, 2006.

[13] Kalman R E. A New Approach To Linear Filtering and Prediction Problems[J]. Journal of Fluids Engineering, 1960.

[14] Harvey A C. Forecasting, Structural Time Series Models and the Kalman Filter[M]. Forecasting, structural time series models and the Kalman filter. Cambridge University Press, 1990.

[15] Inc G. Convolutional, long short-term memory, fully connected deep neural networks[C]. IEEE International Conference on Acoustics. IEEE, 2015.

[16] Kalchbrenner N, Danihelka I , Graves A. Grid Long Short-Term Memory[J]. Computer Science, 2015.

[17] Gers F A, Schmidhuber, Jürgen, Cummins F. Learning to Forget: Continual Prediction with LSTM[J]. Neural Computation, 2000, 12(10): 2451-2471.

[18] Hochreiter S, Schmidhuber, Jürgen. Long Short-Term Memory[J]. Neural Computation, 1997, 9(8):1735-1780.

[19] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2):157-166, 1994.