The original paper number: STOTEN-D-20-23854

Title: A Novel Encoder-Decoder Model based on Read-first LSTM for Air Pollutant Prediction

Dear Editor and Reviewers:

Enclosed is the revised version of the paper entitled “A Novel Encoder-Decoder Model based on Read-first LSTM for Air Pollutant Prediction”. We are truly grateful to yours and reviewers’ critical comments and thoughtful suggestions. Based on these comments and suggestions, we have made careful modifications on the original manuscript. According to the comments of the reviewers, we have completed the adjustment and modification of the paper structure, content, and experiment. In addition, in this response, we adjusted the positions of the figures and tables based on the reviewer's comments. Therefore, it should be noted that in this response, the positions of the figures and tables in the response may be different from the positions of the figures and tables in the questions. These changes made to the text are in red so that they will be easily identified.

We appreciated the thorough reviews and responses of all reviewers. Below is our response to their comments resulting in a number of clarifications.

Best regards

Dr. Hui Wang

***We thank the Editor for organizing the review for our submission. We have carefully followed the reviewers’ comments and prepared a major revision in this submission. The detailed comments are addressed point-to-point as follows. We hope that the Editor and the anonymous reviewers find this revision satisfactory.***

• Editors and Reviewers Comments:

Reviewer #1: The authors proposed an improved LSTM as a more powerful temporal feature extractor and applied it to build a an encoder-decoder model for pollutant prediction. The reviewer does not have primary concerns over the paper. Following are some of the suggestions and comments for the authors to consider.

1. The sentence for highlights are too long.

Response: Thanks to the reviewer's suggestion, we have simplified the highlights part. The review highlights model prediction performance improvement and solving long time series prediction problem; RLSTM effectively solving the problem of insufficient extraction of pollutants and meteorological data features; EDSModel yields higher-accuracy predictions by fully extracting data features, and overcomes long-term dependency; EDSModel has been applied as one of the practical auxiliary models in the national urban pollution prediction tasks.

2. LSTM has been extensively studied with many extension and improvement. The authors are suggested to further highlight which innovation they made with LSTM and justify their innovation is substantial.

Response: According to the reviewer's suggestion, we have highlighted and summarized the innovations of RLSTM, and verified the advanced nature of the model. In the 3.2.1 Feature Extraction Process section of the paper, we describe in detail the functions, characteristics and implementation process of each gate of RLSTM. In the 3.2.2 Advantages of RLSTM section of the paper, we analyzed the structure and advantages of RLSTM compared to LSTM. According to the characteristics of the model, in the experimental part, we respectively compared the performance of the LSTM and RLSTM models to verify the effectiveness of RLSTM. Thanks again for the reviewer's comments, we have appropriately added the contents of the reply to the Sections 3.2.1, 3.2.2, 4 and 5 of the paper.

3. The paper could be even better structured if the authors could start with the air pollution prediction problem to inspire the use of deep-learning method. I think it could fit better the scope of this journal.

Response: Thanks to the reviewer's suggestion. According to the reviewer's suggestion, we adjusted the overall structure of this paper. In the Introduction section, first, we pointed out the hazards of environmental pollutants and the importance of pollutant concentration prediction; secondly, we pointed out the characteristics of pollutants and the problems existing in the existing pollutant concentration prediction methods; Finally, we point out the characteristics of deep learning technology and its advantages in the task of pollutant concentration prediction. This paper focuses on the shortcomings of existing methods in the task of pollutant concentration prediction. In order to solve the problems encountered in the existing long-term sequence prediction tasks, the RLSTM network model is proposed. Then, we verify the correctness of the RLSTM model proposed in this paper through a large number of comparative experiments and analysis. Finally, it proves the advantages of RLSTM in long-term pollutant concentration sequence prediction tasks. Thanks again for the reviewer's comments.

4. There lacks an in-depth discussion. The experimental results are mainly focused on evaluating the performance of deep learning methods. Given the scope of the journal, insights on environments could be discussed.

Response: Thanks to the reviewer's suggestion. According to the reviewer's suggestion, we have revised this paper and added a large number of references related to environmental pollution [39-60]. In addition, we added new experiments in Section 4.4.1 of the paper. We combine the prediction of pollutant concentration with AQI to more accurately analyze the distribution characteristics of pollutant data.We compare with existing methods in terms of data and model. For example, from the aspect of data, we consider the characteristics of the changes in the values of pollutants, the distribution characteristics of the data, including the mutation points of the pollutants and the magnitude of the values, etc.; from the model, we consider the characteristics of the internal structure of the model and make improvements. In Sections 4 and 5, we analyzed in detail the distribution characteristics of pollutants and the performance of the model. Because this article is a new prediction model for pollutant concentration prediction, we pay more attention to the performance of the model in the task of pollutant concentration prediction. Thanks again for the reviewer's comments, we have appropriately added the contents of the reply to the Section 4 and 5 of the paper.

Reviewer #2: The authors have proposed an alternative to the LSTM encoder-decoder network for predicting air pollution. The alternate network introduces a "read first LSTM" that aims to do some feature extraction before carrying out the other LSTM steps. The method, according to the authors performs better than the LSTM encoder-decoder network.

Here are a few comments for the authors:

1. The authors mention in pg 10 that "The traditional LSTM has two key problems. (1) The gating units within LSTM are independent, so LSTM does not efficiently extract the long-term sequence characteristics of the input data". Evidence to support this statement should be provided.

Response: Thanks to the reviewer's suggestion. According to the reviewer’s suggestion, we have added the corresponding literature to the this paper, for example, [53] Zhang, P., Xue, J., Lan, C., Zeng, W., Gao, Z., & Zheng, N. (2019). Eleatt-rnn: Adding attentiveness to neurons in recurrent neural networks. *IEEE Transactions on Image Processing*, *29*, 1061-1073. [59] Yang, B., Sun, S., Li, J., Lin, X., & Tian, Y. (2019). Traffic flow prediction using LSTM with feature enhancement. *Neurocomputing*, *332*, 320-327.[60]Yao K, Cohn T, Vylomova K, Duh K, Dyer C. Depth-gated recurrent neural networks. arXiv preprint arXiv:1508.03790. 2015 Aug;9. Thanks again for the reviewer's comments, we have appropriately added the references of the reply to the Section 3.3 of the paper.

2. In the paragraph below fig 1, the authors referred to "the update information it". Fig. 1 should be labelled properly to show this variable and others. Please also label the different gates.

Response: Thanks to the reviewer's suggestion. The "Update information" in this paper refers to the input information used to update the memory storage unit . We have modified Figure 1 and Figure 2 according to the reviewer's suggestion. We have labeled the main variables, and different gates are represented by unreasonable symbols, such as: means unit state, represents forget gate, represents input gate, etc. Thanks again for the reviewer's comments.

3. The authors mention that the forget and input gates in the LSTM architecture are independent, but also retain an independent forget and input gate. The main understanding of the new architecture is that it removes unnecessary features, but this does not combine the hyperparameter of either gate making them dependent as the authors have insinuated. Please reword your criticism of the LSTM architecture.

Response: Thanks to the reviewer's suggestion. Due to our description problem, I am very sorry to cause some confusion to the reviewer. According to the reviewer's suggestion, we have modified the description of each problem of LSTM and added specific variables. In addition, in RLSTM, we have modified each door. For example, the forge gate is based on the introduction of read gate , and the forge gate is strongly associated with the introduction of read gate ; similarly, the write gate is also based on the read gate , and the write gate is strongly related to the read gate . Therefore, during the time series feature extraction phase, the LSTM will encounter the following problems: (1) When forget gate selectively forgetting the cell memory information , the update information is not referred to, and the effect of the update information at time on the forget of cell memory information is ignored. (2) At time , the update of the memory information of cell is mainly completed through the cooperation of the forget gate and the input gate . However, when the input gate selects the information for updating the cell state , it does not refer to the information forgotten by the forget gate . Therefore, the forget gate and the input gate are two independent processes in LSTM. (3) Because the forget gate and input gate are independent in the feature extraction process, the hidden feature information output by the output gate may have problems such as feature information redundancy or insufficient feature extraction [19]. For the description of RLSTM, we have also modified it in Section 3.2 and described it in more detail. Thanks again for the reviewer's comments, we have appropriately added the contents of the reply to the Sections 3.1 and 3.2 of the paper.

4. Please explain what the difference between feature extraction of the new gate and the job of the forget gate for readers to understand. At first glance, it seems like they might do the same thing.

Response: Thanks to the reviewer's suggestion. Yes, the Sigmoid function used by the RLSTM forget gate is the same as the LSTM. However, we added a read gate to the independent variable, that is, associate the forget gate with the read gate . Thus, the forget gate realizes the forgetting of the state information of the historical unit according to the read gate . In this way, the forget gate can selectively forget the unit state information based on the read feature information . We have explained in the paper that “Unlike LSTM, RLSTM forgets the feature information with a small contribution in the memory unit according to the output of the read gate and the current input ”.

5. From the look of things in fig.2, it looks like the output gate has been removed and only the squashing function is left. Please address why this is the case. It looks like everything from the cell state (memory) is fed to the computation in the next time step as the previous hidden state.

Response: Thanks to the reviewer's suggestion. What I need to explain here is that because traditional LSTM is generally used for feature extraction and prediction tasks at each moment, it is generally based on the output hidden state . That is, each time LSTM not only extracts the input features at the current time, but also uses the current hidden feature output for related prediction tasks. Since the prediction result of LSTM at each time is distinguishable, according to the current unit state , the hidden feature information output at each time is required to be different. Therefore, the traditional LSTM will weight the unit state through the output gate , and then output the hidden features of the current unit . However, the RLSTM model we proposed is mainly used to extract the time series features of the input data, that is, our focus is on the extraction process and results of the whole time series feature. For the hidden features output at each time, we use a to perform a nonlinear transformation on the unit state , which is also a normalization process, that is, the range of hidden feature values is between [-1.0,1.0]. In the whole sequence feature extraction process, we have no related downstream tasks, and only need to output the unit state or hidden feature at the last moment. Therefore, for the prediction tasks of [72-1h], [72-24h] and [72-48h] in this paper, we only need to input the unit state at the last moment into the fully connected layer or Decoder. However, our model is also suitable for prediction tasks at each moment, because the unit state is different at each moment, so we can completely output the whole unit state information through the function. The above content is some similarities and differences between LSTM and RLSTM in the output of hidden feature. These points have been expressed through the mathematical formulas in the paper. Thanks again for the reviewer's comments.

6. Continuing from the previous comment, why is the previous cell state a function of the read state rt if ht-1 already has the information. Isn't this redundant?

Response: Thanks to the reviewer's suggestion. Because of the input information, whether it is the hidden feature at the last time, the unit state at the last time, or the current input , due to the limited storage capacity of the memory unit , we must filter out some redundant information. Moreover, if the memory unit stores too much non-important or redundant information, this will also have an impact on our target tasks. Therefore, our first step is to select the input feature information. The selection method is similar to the attention mechanism. That is, we use the sigmoid function to weight the input features, a small weight indicates that the importance of the feature information is less, and a large weight indicates that the feature is more important. The weighting process is completed through continuous training of the neural network, and the weight range is between [0,1]. The content introduced above is the function and implementation mechanism of our read gate . The realization principle and process of read gate, we have given in the text in the form of text and formula, such as formula (1). Thanks again for the reviewer's comments.

7. Please provide a reference for an example for this statement "Recently, the mainstream pollutant concentration prediction model is mainly based on LSTM Encoder-Decoder, using one LSTM as the Encoder…" in section 3.3.

Response: Thanks to the reviewer's suggestion. We have added the latest LSTM-based Encoder-Decoder prediction model literature to the corresponding position. [54] Kristiani, E., Yang, C. T., Huang, C. Y., Lin, J. R., & Nguyen, K. L. P. (2020). PM2. 5 Forecasting Using LSTM Sequence to Sequence Model in Taichung City. In *Information Science and Applications* (pp. 497-507). Springer, Singapore. [55] Lyu, P., Chen, N., Mao, S., & Li, M. (2020). LSTM based encoder-decoder for short-term predictions of gas concentration using multi-sensor fusion. *Process Safety and Environmental Protection*, *137*, 93-105. [56] Du, S., Li, T., & Horng, S. J. (2018, December). Time series forecasting using sequence-to-sequence deep learning framework. In *2018 9th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP)* (pp. 171-176). IEEE. Thanks again for the reviewer's comments, we have appropriately added the references of the reply to the Section 3.3 of the paper.

8. Mark the temporal direction and depth in Fig. 4.

Response: Thanks to the reviewer's suggestion. According to the reviewer's suggestion, we have marked the temporal direction and depth in Figure 4.

9. Were the RNN, GRU and LSTM encoder-decoder results presented in Table 3 trained on the same data? If not, please specify this to the readers.

Response: Thanks to the reviewer's suggestion. According to the reviewer's suggestion, we added Section 4.2 to this article. Data-related issues are introduced in 4.2.1 Datasets, 4.2.3 Training and 4.2.4 Evaluation respectively. That is, RNN, GRU and LSTM encoder-decoder results presented in Table 4 and Table 5 trained on the same data. Thanks again for the reviewer's comments, we have appropriately added the contents of the reply to the Section 4.3, 4.4.1, 4.4.2 of the paper.

10. Provide a column with the references to the results in Table 3 if they're from literature.

Response: Thanks to the reviewer's suggestion. According to the reviewer's suggestion, we added a references to the existing comparison models in Table 4 and Table 5. In addition, we have added some comparative model results of the latest research, such as XGBoost approach, multiple linear regression (MLR), etc.

11. Finally, specify the number of layers used for the encoder and decoder as well as other hyperparameters that will be useful for replications such as batch size, epoch and the equipment used to train.

Response: We really admire the suggestions given by the reviewer. According to the reviewer's suggestion, we add all the hyperparameters to Table 3. In addition, we have given the relevant parameter setting mechanism, which has been added to the 4.2.3 Training, 4.2.4 Evaluation and 4.3 Parameter Setting sections. 4.2.3 *Training*：The hyperparameters in our EDSModel are determined during the training process, that is, the best performance model is selected on the validation set through the RMSE. We manually specify the hyperparameter ranges: learning rate {0.01, 0.005, 0.003, 0.001}, dropout rate {0.0, 0.1, 0.2, 0.3, 0.4, 0.5}, regularization parameter {0.1, 0.01, 0.001, 0.0001} and decay rate{0.99, 0.95, 0.90, 0.85}. For different datasets, we have found that the following setting work well: set the dropout to 0.2, decay rate to 0.99, regularization parameter to 0.0001 and learning rate to 0.001 for EDSModel. When using the comparison models, these settings still work well. 4.2.4 *Evaluation*: The setting of hyperparameters in this study is based on the results of many experiments, leading to the final selection of the optimal set of hyperparameters. The validation set used in this study is closely related to the training stage, and after each epoch, the RMSE and MAE of the prediction model on the validation set are calculated. Therefore, the optimal model is selected based on the model error calculated on the validation set. The specific process is as follows: for each experiment, the number of epochs selected was 100. After training an epoch, we tested the trained model on the validation set. If the RMSE and MAE of the prediction model on the validation set became smaller, we updated and saved the model parameters. After many parameter adjustments and experiments, when the prediction effect of the prediction model on the validation set was optimal, the training ended. Finally, get the prediction result by iterating all the samples in the test set. 4.3 Parameter Setting: In the experiment, dropout was used as a general trick to avoid model overfitting. According to historical experience and research results in the field of deep learning, the effect is obvious when the value of the training stage is 0.5. Therefore, in the training stage of different prediction models, the value of dropout is 0.5 for the hidden layer of the recurrent network, and the fully connected layer. In the verification and testing stage, for each model, the value of dropout is 1.0. After the experiments, the layer selection of the EDSModel is shown in Table 3, and the parameters used for model testing are shown in Table 3. Thanks again for the reviewer's comments, we have appropriately added the content of the reply to the Sections 4.2.3, 4.2.4 and 4.3 of the paper.

Reviewer #3: This study proposed a RLSTM for pollutant prediction, and the effectiveness and superiority of RLSTM and the prediction model have been demonstrated. The hybridization of the model, which has EDS model, stacked RLSTM as encoder, LSTM as decoder, is interesting and novel. Overall, this is an interesting research and can make contribution to environmental modeling fields, while there are a number of flaws and fuzziness, which need be clarified. Further comments are shown below:

1.The Graphical Abstract is not attractive and informative.

Response: Thanks to the reviewer's suggestions. Based on the reviewer's suggestion, we have revised the graph in Graphical Abstract and marked the extraction process of pollutants and meteorological data features.

2.The maximum characters per bullet point should be less than 85 characters in the highlights.

Response: Thanks to the reviewer's suggestions, we have simplified the highlights part. The review highlights model prediction performance improvement and solving long time series prediction problem; RLSTM effectively solving the problem of insufficient extraction of pollutants and meteorological data features; EDSModel yields higher-accuracy predictions by fully extracting data features, and overcomes long-term dependency; EDSModel has been applied as one of the practical auxiliary models in the national urban pollution prediction tasks.

3. The references can be much enhanced and updated. As stated. "To date, deep learning models have proved to be the state of the art in spatiotemporal prediction, [3],[14-18]". There are quite many of publications in STOTEN, which used deep learning methods (or machine learning techniques) for air pollutant predictions (ex. PM2.5) over last two years.

Response: Thanks to the reviewer's suggestions. According to the reviewer's suggestion, we have updated the references and added a large number of references to the latest research results. For example, we have added the latest research results [39-56], and completed the update of the references such as [1], [27], etc. Thanks again to the reviewers for their comments, we have appropriately added the references to the corresponding paragraphs of this paper.

4. It is not suitable to claim "All the previous studies were based on shallow data analysis and failed to fully extract the historical data distribution characteristics,...", which could cause many debated!

Response: Thank you very much for the reviewer’s suggestion, and also provided a reference for my future writing. We have revised this part of the content, and elaborated and summarized the existing research methods in more detail. We have modified this part of the content to: “The above researches are generally based on the shallow feature extraction of pollutant and meteorological data, including two aspects: spatial dimension and temporal dimension. From the spatial dimension, for multi-site pollutant concentration prediction tasks, some researchers have used CNN to extract spatial features of pollutants and meteorological data [38],[40,41]. However, for single-site pollutant concentration prediction, from the time dimension, they cannot fully extract the time series distribution features of historical data, especially the complex internal interactions between long-term series data, including deterministic and statistical approaches.” Thanks again for the reviewer's comments, we have appropriately added the content of the reply to the Section 2 Related work of the paper.

5. Why does Decoder use the original LSTM instead of RLSTM?

Response: First of all, we are very sorry. Because, we did not explain in detail why Decoder uses LSTM and is not an RLSTM network. Therefore, we need to explain and describe this. First of all, the reason why we choose RLSTM as the encoder is to prove that compared with LSTM, RLSTM is more suitable for the extraction of long-term series of pollutants and meteorological data features. Secondly, in order to verify the above argument, we choose the controlled variable method to control the consistency of some variables. That is, when the decoder is the same, the LSTM is used as the decoder of the EDSModel, and the prediction model based on the RLSTM as the encoder is better than the prediction model with the LSTM as the encoder. Finally, according to the experimental results, it is verified that the performance of the prediction model EDSModel based on RLSTM as the encoder is better than the comparison model. Thanks again for the reviewer's comments, we have appropriately added the content of the reply to the Section 3.4 of the paper.

6. It seems that RLSTM adds one more input to the other two gates (forget and write gate) without reducing the input information, because there are still xt, ht-1, ct-1 information. It is mentioned in your description 3.2.2 "Read gate filtering out the redundant information in the input data". Can you briefly explain how is the model filtering the redundant information?

Response: Thanks to the reviewer's suggestions. According to the reviewer's question, because the filtering of redundant information is related to the read gate and the forget gate. Therefore, we divide this question into two answers, from the working principle of the read gate and forget gate. For the read gate: Because of the input information, whether it is the hidden feature at the last time, the unit state at the last time, or the current input , due to the limited storage capacity of the memory unit , we must filter out some redundant information. Moreover, if the memory unit stores too much non-important or redundant information, this will also have an impact on our target tasks. Therefore, our first step is to select the input feature information. The selection method is similar to the attention mechanism. That is, we use the sigmoid function to weight the input features, a small weight indicates that the importance of the feature information is less, and a large weight indicates that the feature is more important. The weighting process is completed through continuous training of the neural network, and the weight range is between [0,1]. The content introduced above is the function and implementation mechanism of our read gate . The realization principle and process of read gate, we have given in the text in the form of text and formula, such as formula (1). Thanks again for the reviewer's comments. For the forget gate, the Sigmoid function used by the RLSTM forget gate is the same as the LSTM. However, we added a read gate to the independent variable, that is, associate the forget gate with the read gate . Thus, the forget gate realizes the forgetting of the state information of the historical unit according to the read gate . In this way, the forget gate can selectively forget the unit state information based on the read feature information . The above is the whole process of RLSTM filtering redundant information. Thanks again for the reviewer's comments.

7. Will it be clearer to express the error in percentage? Because RMSE and MAE cannot show the percentage of error.

Response: Thanks to the reviewer's suggestions. First of all, I need to apologize that because we are doing regression tasks, it is difficult to use percentages to measure the error of the prediction results. In classification tasks similar to NLP and image recognition, percentages are often used to represent the performance of the model. Secondly, our measurement methods, including RMSE and MAE, are common methods in current research tasks of pollutant concentration prediction. Finally, if the reviewer has a percentage evaluation method suitable for measuring error, we will use it according to the actual situation.

8. Recently, GRU has become more and more popular. Will there be similar effects if GRU is improved? Why choose modified LSTM instead of GRU?

Response: Thanks to the reviewer's suggestions. 1. According to relevant research, the performance of LSTM is better than that of GRU network (Hládek, D., Staš, J., & Ondáš, S. (2019, October). Comparison of Recurrent Neural Networks for Slovak Punctuation Restoration. In *2019 10th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)* (pp. 95-100). IEEE.). The experimental results in this paper also verify this point. The accuracy of LSTM is higher than that of GRU. But for some complex problems, such as machine translation, the computational complexity of GRU is lower than LSTM, but its performance is lower than LSTM. 2. Because, for the problem of pollutant concentration prediction, the input data dimension is relatively low, and the calculation complexity is small. Therefore, considering the time complexity and prediction performance, we prefer LSTM as the improvement object, and use the improved LSTM as the feature extractor. Thanks again for the reviewer's comments, we have appropriately added the reference of the reply to the Section 3.1 of the paper.

9. Are the time periods corresponding to Figure 6 a-d in different seasons? Please briefly explain.

Response: Thanks to the reviewer's suggestions. Thank you very much for the questions raised by the reviewer. Here, we explain the doubts raised by the reviewer. First, the time periods in Figure 8 (a)-(d) do not overlap. Secondly, Our method of selecting samples for the four time periods is random selection, that is, randomly selecting four non-overlapping samples from the whole Shanghai test set. Therefore, the samples of the four time periods are not necessarily from the four seasons. Finally, we use the four samples as an auxiliary means to visualize the prediction performance of each model. Of course, the specific performance of each model refers to the experimental results in Table 5. Since there are a large number of test results like Figure 8 for time series prediction in different time periods, we cannot display all the results like the visualization method Figure 8. Therefore, for the series prediction of the time period 1-24h, we randomly selected 4 test samples as an aid to visually show the prediction performance of the EDSModel. Thanks again for the reviewer's comments, we have appropriately added the content of the reply to the Section 4.4.2 of the paper.

10. Similar question as 5, what is the period in Figure 7?

Response: Thanks to the reviewer's suggestions. Similar to question 9, here, we will provide further answers to the reviewer’s doubts. First, the time periods in Figure 9 do not overlap. Secondly, Our method of selecting samples for the two time periods is random selection, that is, randomly selecting four non-overlapping samples from the whole Shanghai test set. Finally, we use the two samples as an auxiliary means to visualize the prediction performance of each model. In order to ensure the consistency of the length of the input time series in the paper, the length of the sequence used in the whole paper is 72 hours. In order to further demonstrate the difficulty of long-term sequence prediction, we will predict the pollutant concentration in the next 48 hours as the target task. In order to ensure that the length of the time series of the model input data in the paper is consistent, the length of the series used in the whole paper is 72 hours. In order to further demonstrate the difficulty of long-term series forecasting, we have extended the time length of pollutant concentration prediction, that is, predicting the pollutant concentration in the next 48 hours as the forecast target. we have appropriately added the content of the reply to the Section 4.6 of the paper.

11. Why is your study use data from the past 72 hours to predict the next 48 hours? Is there any reference? (e.g. calculating the correlation of data, Trial and error method? etc.)

Response: Thanks to the reviewer's suggestions. 1. Because according to relevant research, long-term sequence feature extraction and time-series prediction are a difficult task. Therefore, we use the input sequence of the last time to be 72 hours, and the target predicted pollutant concentration time periods are 1, 24 and 48 hours. We use the past 72 hours to predict the pollutant concentration in the next 48 hours. The purpose of this is to show the advantages of EDSModel in extracting long-term sequence features and pollutant concentration prediction tasks. 2. Thank you very much for the reviewer’s suggestion. We have added the corresponding literature to the corresponding position of the paper, such as [57] Jin, X., Yang, N., Wang, X., Bai, Y., Su, T., & Kong, J. (2019). Integrated predictor based on decomposition mechanism for PM2. 5 long-term prediction. *Applied Sciences*, *9*(21), 4533. Thanks again for the reviewer's comments, we have appropriately added the content and references of the reply to the Section 4.3, 4.4, and4.6 of the paper.

12. Could you list detailed parameters of the model with a table? For example, the number of neurons, weight initializing method, regularizor etc.

Response: Thanks to the reviewer's suggestions. According to the reviewer's suggestion, we add all the hyperparameters to Table 3. In addition, we have given the relevant parameter setting mechanism, including training method, which has been added to the Sections 4.2.3 Training, 4.2.4 Evaluation and 4.3 Parameter Setting. The implementation code of the EDSModel has been published on the personal homepage, and readers can access it through the github address. *4.2.3 Training*：The hyperparameters in our EDSModel are determined during the training process, that is, the best performance model is selected on the validation set through the RMSE. We manually specify the hyperparameter ranges: learning rate {0.01, 0.005, 0.003, 0.001}, dropout rate {0.0, 0.1, 0.2, 0.3, 0.4, 0.5}, regularization parameter {0.1, 0.01, 0.001, 0.0001} and decay rate{0.99, 0.95, 0.90, 0.85}. For different datasets, we have found that the following setting work well: set the dropout to 0.2, decay rate to 0.99, regularization parameter to 0.0001 and learning rate to 0.001 for EDSModel. When using the comparison models, these settings still work well. We train all models using the SGD optimizer with batch size 128. *4.2.4 Evaluation*: The setting of hyperparameters in this study is based on the results of many experiments, leading to the final selection of the optimal set of hyperparameters. The validation set used in this study is closely related to the training stage, and after each epoch, the RMSE and MAE of the prediction model on the validation set are calculated. Therefore, the optimal model is selected based on the model error calculated on the validation set. The specific process is as follows: for each experiment, the number of epochs selected was 100. After training an epoch, we tested the trained model on the validation set. If the RMSE and MAE of the prediction model on the validation set became smaller, we updated and saved the model parameters. After many parameter adjustments and experiments, when the prediction effect of the prediction model on the validation set was optimal, the training ended. Finally, get the prediction result by iterating all the samples in the test set. *4.3 Parameter Setting*: In the experiment, dropout was used as a general trick to avoid model overfitting. According to historical experience and research results in the field of deep learning, the effect is obvious when the value of the training stage is 0.5. Therefore, in the training stage of different prediction models, the value of dropout is 0.5 for the hidden layer of the recurrent network, and the fully connected layer. In the verification and testing stage, for each model, the value of dropout is 1.0. After the experiments, the layer selection of the EDSModel is shown in Table 3, and the parameters used for model testing are shown in Table 3. Thanks again for the reviewer's comments, we have appropriately added the content of the reply to the Sections 4.2.3, 4.2.4 and 4.3 of the paper.