

PAPER • OPEN ACCESS

Computer dynamic model and time series prediction of air by LSTM recurrent neural network

To cite this article: Haoran Zhao *et al* 2021 *J. Phys.: Conf. Ser.* **2033** 012085

View the [article online](#) for updates and enhancements.

You may also like

- [Data-driven Detection of Multimessenger Transients](#)
Iftach Sadeh
- [Differential resistance based self-sensing recurrent neural network model for position estimation and control of antagonistic Shape Memory Alloy actuator](#)
Hari N. Bhargaw, Samarth Singh, B A Botre et al.
- [DAVI: Deep learning-based tool for alignment and single nucleotide variant identification](#)
G Gupta and S Saini



The Electrochemical Society
Advancing solid state & electrochemical science & technology

242nd ECS Meeting

Oct 9 – 13, 2022 • Atlanta, GA, US

Extended abstract submission deadline: April 22, 2022

Connect. Engage. Champion. Empower. Accelerate.

MOVE SCIENCE FORWARD



Submit your abstract



Computer dynamic model and time series prediction of air by LSTM recurrent neural network

Haoran Zhao*, Ziyuan Li, Shuyang Xu

School of Information Science & Engineering Lanzhou University, Lanzhou, China

*Corresponding author: haoranzhao@lzu.edu.cn

Abstract. Air quality is closely related to people's daily life. In order to predict air quality with high accuracy, the air pollution monitoring data of Lanzhou City from May 13, 2015 to April 18, 2020 is used as the basis, and the LSTM model based on the deep learning library TensorFlow is used to predict the air quality of Lanzhou City and compared with the RNN model. The experimental results show that the mean square error of the model is 39.579212, which is more accurate than the RNN model but takes a longer time, and provides a new prediction method with the scientific and reasonable theoretical basis for air pollution prevention and control work.

Keywords: Air Quality, TensorFlow, LSTM, Prediction.

1. Introduction

With the continuous improvement of people's production and living standards in today's society, environmental issues have gradually come into people's view [1]. Lanzhou City, Gansu Province, is an ancient cultural city with a history of 2200 years. For a long time, as an essential industrial base and comprehensive transportation hub in the northwest, one of the critical central cities in the western region, and a vital node city of the Silk Road Economic Belt, Lanzhou has played an indispensable role in the political, economic and cultural development of the northwest [2]. At the same time, as a city standing in the northwest, it also inevitably receives a certain amount of environmental impact [3]. Nowadays, environmental problems are becoming more serious, especially in spring, when Lanzhou receives dust storms and other climatic causes [4] [5]. In order to further reveal and manage the pollution situation of air quality in Lanzhou, it is necessary to understand the trend of air change, get timely, accurate and comprehensive air quality information, and make a precise prediction of air quality.

There are many air pollution prediction methods, including numerical prediction model methods, statistical prediction model methods, and machine learning-based model prediction methods. Numerical prediction model methods, such as the Nested Air Quality Prediction Modeling System [6] and Model-3/CMAQ model [7], are based on atmospheric dynamics to build a numerical model of atmospheric pollutant transport in the air for prediction. This method has been widely used, but the application range is single, and it is not easy to achieve real-time online prediction of air quality [8]. Statistical forecasting models such as statistical regression models, clustering models and backpropagation models are based on historical data to determine the intrinsic development pattern to complete the prediction [9], which has become one of the mainstream forecasting methods. Traditional machine learning algorithms such as Boost [10], MFO [11], and SVM [12] can also achieve air quality forecasting in a short period but have



the disadvantages of harrowing feature extraction and high uncertainty in the long-term forecasting [13]. In recent years, with the rapid development of deep learning technology, we have seen its great potential in the field of air quality prediction [14]. In this paper, a tensor flow-based LSTM (recurrent neural network) time series model is proposed to predict air pollution in Lanzhou City, and its performance is compared with the RNN model.

2. Theory Introduction

2.1. RNN model theory

RNN, namely recurrent neural network, is applied to the scene where the input data is a sequence and the front and back data are dependent. Compared with the fully connected neural network, RNN has a cyclic hidden layer, which makes the value of this layer depend on both the current input and depend on the value of the previously hidden layer. Its structure is shown in Figure 1 [15].

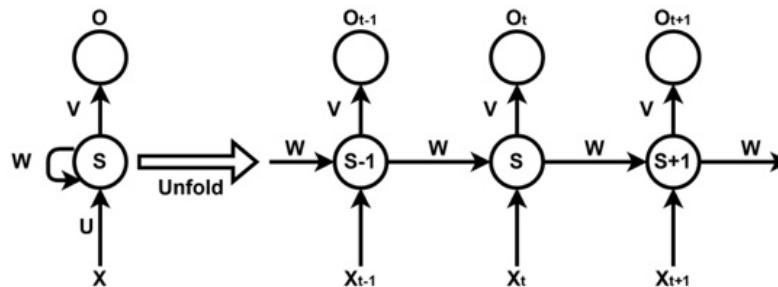


Figure 1. RNN structure

In Figure 1, each circle is a unit, and each unit's work is the same. Therefore, the extended network can be folded as the structure of the left half of the figure, using a single unit to calculate repeatedly. According to the expansion structure of RNN and its critical sequence data, we make the following definition based on Figure 1[16]:

- X_t : Represents the input at time t
- O_t : Represents the output at time t
- S_t : Represents the memory at time t

As a sort of neural network, RNN is also like other neural networks. It adds all kinds of parameters to the input value and obtains the parameters by continuous learning based on the existing data. The basic formula is as follows (1):

$$S_t = f(U * X_t + W * S_{t-1}) \quad (1)$$

Among them, f is the activation function. Standard activation functions include \tanh , $Sigmoid$, $Relu$. U is the weight matrix of input X , and W is the memory value of the hidden layer at the last moment (S_{t-1}), as the weight matrix of this input.

According to the basic formula of RNN, it can be seen that in the process of learning, RNN not only needs to receive time input data, but receive the last moment (i.e., The data stored in the hidden layer.). After the calculation of the basic formula, the result is satisfactory. It will be input into the output formula (2), that is:

$$O_t = g(VS_t) \quad (2)$$

Wherein g is the activation function, and V is the weight matrix of the output layer. After continuous calculation, the output value O_t of the cyclic network is affected by the memory value S with the front

number of $t - 1$, which solves the problem that the fully connected neural network cannot process the sequence input with the data related to each other.

However, in the RNN model, sufficient sequence length can easily lead to extreme nonlinear behaviors, including gradient vanishing or gradient explosion. The recursive calculation of RNN is similar to continuous matrix multiplication. Because the weight of RNN is fixed, the exact weight is used in all time steps. Therefore, with constant learning, the weight coefficient will grow or decay exponentially, making the gradient change considerably [17].

2.2. LSTM model theory

LSTM is an improved model based on RNN to solve the gradient vanishing problem. Its cycle unit modules have different structures, and four network layers influence each other in a particular way. As shown in Figure 2, LSTM uses a "gate" to control discarding or adding information to realize the function of forgetting or remembering to avoid the long-term dependence of RNN [18].

Gate is a structure that can control the information selection, composed of a *sigmoid* activation function and a dot operation [19]. As shown in Figure 1, an LSTM unit has three such gate structures: forgetting gate, input gate, and output gate. The data update process of LSTM is as follows:

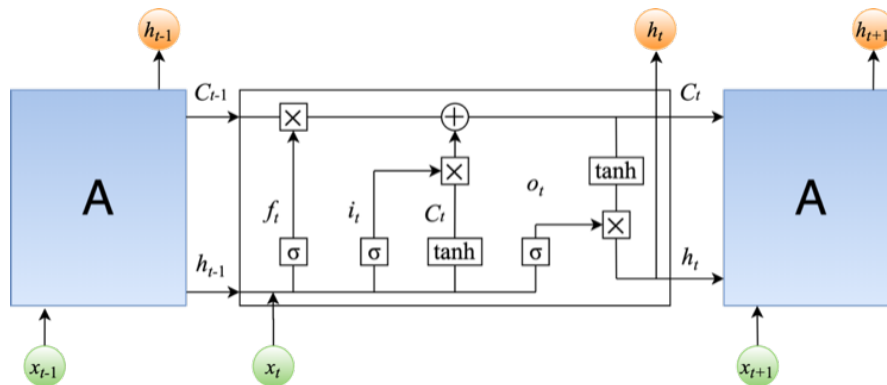


Figure 2. LSTM structure

At time t , the input gate calculates the input i_t of the input gate according to the output h_{t-1} at the last moment and the current input x_t and decides whether to update the current state to the cell state of LSTM. The calculation of i_t is as follows (3):

$$i_t = \text{sigmoid}(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad (3)$$

W_{xi} is the weight matrix of the current input x_t of the forgetting gate, W_{hi} is the weight matrix of the output h_{t-1} at the last moment of the forgetting entrance, b_i is the offset, and *sigmoid* is used as the activation function.

The forgetting gate outputs an output value between [0,1] through the activation function *sigmoid* according to the output result h_{t-1} of the hidden layer at the previous time and the current input x_t . When the output value is 0, all the information at the earlier time is discarded, and when the output value is 1, it means that all the information is retained. The calculation of output value f_t is as follows (4):

$$f_t = \text{sigmoid}(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \quad (4)$$

W_{xf} is the weight matrix of the current input x_t of the forgetting gate, W_{hf} is the weight matrix of the output h_{t-1} at the last moment of the forgetting entrance, b_f is the offset, and *sigmoid* function is used as the activation function.

Current moment candidate memory value \tilde{C}_t and a tan function to control which new information will be added, the candidate memory value \tilde{C}_t is calculated as follows (5):

$$\tilde{C}_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (5)$$

W_{xc} is the weight matrix of the current input x_t of the candidate memory value. W_{hc} is the weight matrix of the output h_{t-1} of the previous time of the candidate memory value, b_c is the offset, and the tanh function is used as the activation function.

Cell state value C_t at the current moment is composed of input gate and forgetting gate respectively to mediate candidate unit \tilde{C}_t and its own state C_{t-1} at the last moment. The calculation formula is shown in Formula (6):

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C} \quad (6)$$

f_t is the forgetting gate value at the current moment, and i_t is the input gate value at the current moment.

The output gate o_t controls the output of the cell state value, which is multiplied by the activation function *sigmoid* to determine which information will be output. The calculation is shown in equation (7):

$$o_t = \text{sigmoid}(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \quad (7)$$

Where W_{xo} is the weight matrix of the current input x_t of the output gate, W_{ho} is the weight matrix of the output h_{t-1} at the last moment of the output gate, b_o is the offset, and *sigmoid* function is used as the activation function.

The output of the final LSTM unit is h_t , and the calculation is shown in equation (8):

$$h_t = o_t \cdot \tanh(C_t) \quad (8)$$

o_t is the value of the output gate, and C_t is the cell status value at the current moment.

3. Establishment of Prediction Model

3.1. Data preparation

Through the national urban air quality real-time release platform of China's environmental monitoring station, we obtained the air quality data used in the experiment. The accurate air quality data of Lanzhou from May 13, 2014, to April 18, 2020, are selected as the experimental data set, composed of the hourly data of these days. The 24-hour PM2.5 content, PM10 content, PM100 content, AQI index, and other weather indicators are recorded, respectively, with a total of 16 hands and 50043 data points (as shown in Figure 3)

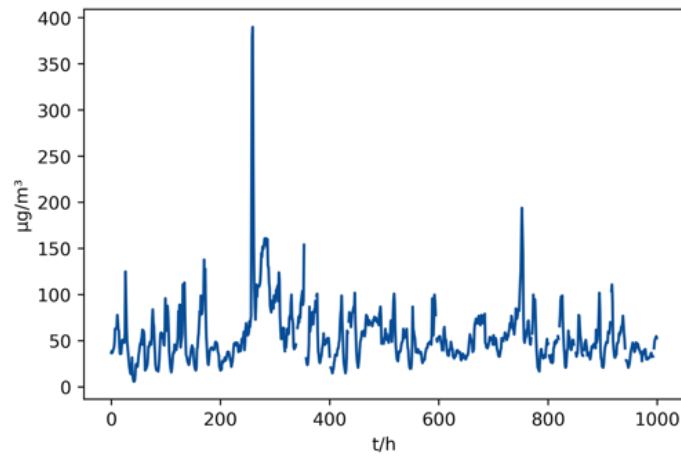


Figure 3. Original PM2.5 data points (Top 1000)

3.2. Data preprocessing

In the process of data collection, data may be missing due to human operation or equipment failure. Incomplete data will affect the prediction results of the model, so it is necessary to complete the missing data. The air quality data has a specific periodic change law. In the experiment, the average load data at the same time of the first day and the second day as the missing value are used to complete the missing data.

At the same time, because the LSTM neural network model is sensitive to the input data scale, in order to reduce the impact of data scale on the model training effect, we normalize the data, as shown in equation (9):

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (9)$$

3.3. Model structure design

As the most popular machine learning framework, TensorFlow is widely used in machine learning and deep learning. The RNN neural network model (see Figure 4) and LSTM neural network model (see Figure 4) for air quality data prediction are designed based on the TensorFlow framework.

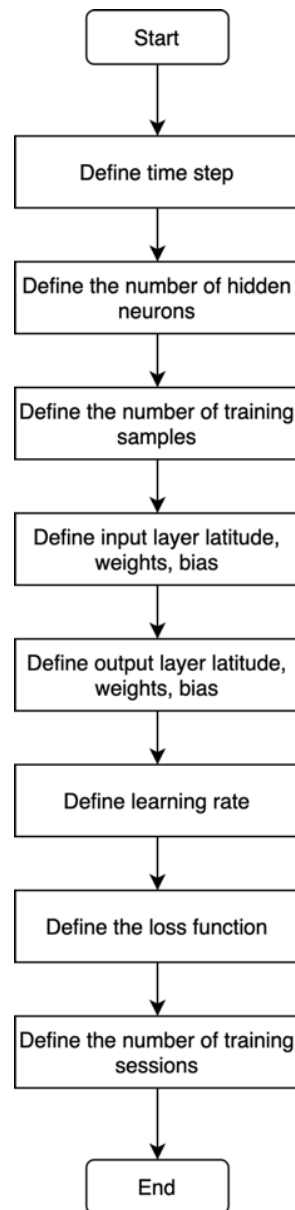


Figure 4. LSTM neural network model implementation

The experimental data is divided into two parts. The hourly data of the first 1900 days is used as the model training set, and the hourly data of the last 185 days is used as the test set. The model's prediction performance is tested by using the test set to compare with the prediction results of the model. Through constant parameter tuning and optimization of the model, the RNN model parameters are finally determined as follows: 128 Hidden layer neurons, the batch size is 16, the window size is 2850, the Adam Optimizer is 0.001, the model uses the first 24 data points to predict the next one. Also, the parameters of the LSTM model are the same as the RNN model [20].

4. Result Analysis

The trained LSTM and RNN were used to predict the next 185 days' hourly data according to the way that use the first 24 data points to indicate the next one, with a total of 4440 data points (185×24). After the end of prediction, the predicted data and the accurate data are inversely normalized, and the mean square error of the model in the prediction is calculated to obtain the fitting curve between the predicted

value and the actual value. The blue represents the actual data and the orange represents the predicted data, as shown in Fig. 5 and Fig. 6.

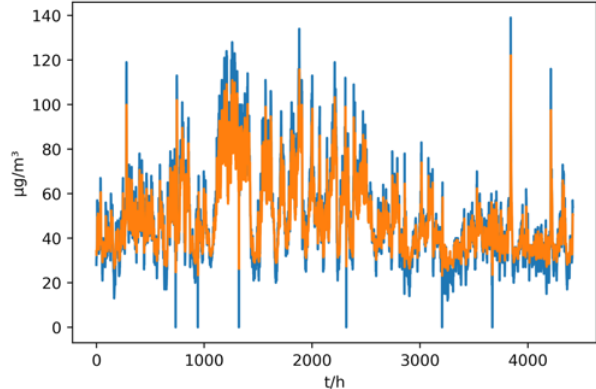
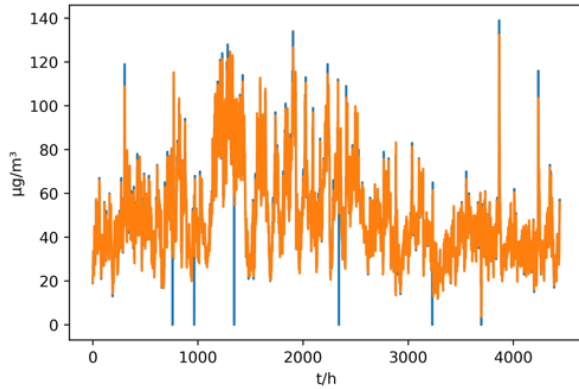


Figure 5. Original PM2.5 data points (Top 1000) **Figure 6.** Original PM2.5 data points (Top 1000)

By comparing the original data, LSTM prediction results, and RNN prediction results (as Figure 7). In the picture, blue represents accurate data, orange represents LSTM prediction data, and green represents RNN prediction data. We can see that LSTM is better than RNN in predicting normal data fluctuation. Although both of them can reasonably forecast the trend of changes in data, LSTM performs better than RNN in order of magnitude prediction. At the same time, we can find that many extreme mutations are not predicted in LSTM, while RNN reasonably predicts the occurrence of mutations.

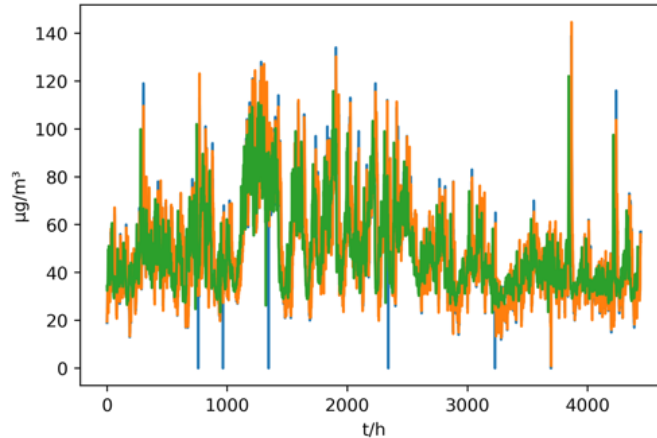


Figure 7. Comparison of predicted results

Table 1. Comparison of model prediction results

Model	Model Parameters			MSE	Time
LSTM	batch_size=16	epoch = 50	Optimizer = Adam	39.5792	13.3
RNN	batch_size=16	epoch = 50	Optimizer = Adam	55.3997	1.67

By comparing Table 1, we can find that LSTM has lower MSE and higher accuracy in predicting the air quality of Lanzhou. However, at the same time, the training time of the LSTM model is higher than that of the RNN model.

5. Conclusion

In this paper, considering the influence of historical air quality data, time information, and weather information on short-term air quality prediction, a short-term air quality prediction model based on long-term and short-term memory networks is proposed. The outliers of air quality data are corrected, and missing values are filled. The PM_{2.5} concentrations, temperature, and other weather data are normalized. The sample data are reconstructed. Then the model is trained with this data, and the prediction results are compared with the RNN model. The experimental results show that:

- Air quality prediction model based on LSTM recurrent neural network is established on Lanzhou's actual air quality data set from May 13, 2014, to April 18, 2020. The result verifies that the LSTM recurrent neural network has an excellent fitting effect on air quality prediction and the mean square error of the training set and test set of the model is small, which means this model can be used to guide the future air quality prediction.

- The stability of the LSTM recurrent neural network is verified. The training is repeated ten times under the same parameters, and the mean square error of each model on the test set is obtained. The standard deviation of RMSD is slight in many experiments, which indicates that the LSTM recurrent neural network model has good stability and can obtain a more stable fitting effect in many repeated experiments. So, the prediction result has high reliability.

- LSTM recurrent neural network model has better accuracy than RNN in time series prediction. However, since LSTM recurrent neural network has a more complex network structure, the solving process is more complex and time-consuming than other models, and thus it is not suitable for real-time prediction. Air quality prediction is a decision-making process in advance, so the real-time requirement is not high.

In future research, we should focus on the time-consuming optimization of the LSTM recurrent neural network model so that the air prediction model can better adapt to the demand of real-time prediction.

References

- [1] Li M, Zhang L. Haze in China: Current and Future Challenges [J]. *Environmental Pollution*, 2014, 189 (Complete): 85 - 86.
- [2] Wang L. A Preliminary Study on the Evolution of Urban Functions in Lanzhou City [J]. *Lanzhou Academic Journal*, 1995 (01): 50 - 52.
- [3] Chu, P. C., Chen, Y., Lu, S., Li, Z., & Lu, Y. (2008). Particulate air pollution in Lanzhou China. *Environment International*, 34 (5), 698 - 713.
- [4] Chen, L. H., Yu, Y., Chen, J. B., Li, W. Y., & Li, J. L. (2010). Characteristics of Main Air Pollution in Lanzhou During 2001--2007. *Plateau Meteorology*, 29 (6), 1627 - 1633.
- [5] Yu, Y., Xia, D. S., Chen, L. H., Liu, N., Chen, J. B., & Gao, Y. H. (2010). Analysis of particulate pollution characteristics and its causes in Lanzhou, Northwest China. *Huan jing ke xue= Huanjing kexue*, 31 (1), 22 - 28.
- [6] Wang, Z. F., Xie, F. Y., Wang, X. Q., An, J., & Zhu, J. (2006). Development and application of nested air quality prediction modeling system. *CHINESE JOURNAL OF ATMOSPHERIC SCIENCES-CHINESE EDITION-*, 30 (5), 778.
- [7] Ching, J., & Byun, D. (1999). Introduction to the Models-3 framework and the Community Multiscale Air Quality model (CMAQ). *Science Algorithms of the EPA Models-3 Community Multiscale Air Quality (CMAQ) Modeling System*.
- [8] Ma, J. Y., Yi, H. H., Tang, X. L., & Zhang, Y. (2014). Development and Research Progress of the Air Quality Modelling [J]. *Shanghai Environmental Sciences*, 2014 (1): 17 - 23.
- [9] Baker, K. R., & Foley, K. M. (2011). A nonlinear regression model estimating single source concentrations of primary and secondarily formed PM_{2.5}. *Atmospheric Environment*, 45 (22), 3758 - 3767.
- [10] Pan, B. (2018, February). Application of XGBoost algorithm in hourly PM_{2.5} concentration prediction. In *IOP conference series: earth and environmental science* (Vol. 113, No. 1, p. 012127). IOP publishing.

- [11] Liu, X., Zhang, X., & Zhao, X. (2020, October). The Prediction of PM_{2.5} Concentration with an Intelligent Hybrid Model. In *Journal of Physics: Conference Series* (Vol. 1624, No. 2, p. 022015). IOP Publishing.
- [12] Leong, W. C., Kelani, R. O., & Ahmad, Z. (2020). Prediction of air pollution index (API) using support vector machine (SVM). *Journal of Environmental Chemical Engineering*, 8(3), 103208.
- [13] Hable-Khandekar, V., & Srinath, P. (2017, August). Machine Learning Techniques for Air Quality Forecasting and Study on Real-Time Air Quality Monitoring. In *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)* (pp. 1-6). IEEE.
- [14] Li, X., Peng, L., Hu, Y., Shao, J., & Chi, T. (2016). Deep learning architecture for air quality predictions. *Environmental Science and Pollution Research*, 23(22), 22408 - 22417.
- [15] Kawakami, K. (2008). Supervised sequence labelling with recurrent neural networks. *Ph. D. thesis*.
- [16] Zhang, J., & Man, K. F. (1998, October). Time series prediction using RNN in multi-dimension embedding phase space. In *SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 98CH36218)* (Vol. 2, pp. 1868-1873). IEEE.
- [17] Manaswi, N. K. (2018). Rnn and lstm. In *Deep Learning with Applications Using Python* (pp. 115-126). Apress, Berkeley, CA.
- [18] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.
- [19] Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2018, December). A comparison of ARIMA and LSTM in forecasting time series. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 1394-1401). IEEE.
- [20] Reimers, N., & Gurevych, I. (2017). Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. *arXiv preprint arXiv:1707.06799*.