Multivariate Time Series Forecasting with LSTM for Madrid, Spain pollution

Shaheen Alhirmizy
College of Science
Kirkuk University
Kirkuk, Iraq
shaheengeolo@gmail.com

Banaz Qader
Computer Science Department
Kirkuk University
Kirkuk, Iraq
bnazanwr@gmail.com

Abstract- Time series forecasting is something of a dark horse in the field of data science and it is most critical factor that decides whether a business, temperatures or any environmental factors effect will rise or fall, A single time-dependent variable means A univariate time series while A Multivariate time series like environmental data has more than one time-dependent variable. Each variable depends on its past values and also on other variables past values. this paper used Neural networks like Long Short-Term Memory (LSTM) for forecasting Spain capital Madrid Air Quality using a dataset that reports on the weather and the level of pollution each hour for two years from 2015to 2016 the data includes the date-time, the pollution concentration of the following: SO2, NO_2, NO, CO. Almost the best problems modelling for multiple input variables are recurrent neural networks and they are the great solution for multiple input time series forecasting problems, where classical linear methods can't . this paper used LSTM model for multivariate time series forecasting in the Keras and Tensor Flow deep learning library in a Python SciPy environment with Machine Learning scikit-learn, Pandas, NumPy and Matplotlib libraries.

Keywords— LSTM, Keras, Tensor Flow, Python SciPy, Deep Learning

I. Introduction

The most important applied techniques of data science are the time series forecasting. It is used in various life aspects such as business, in planning of production and inventory, supply chain management, and widely in finance. It relies on statistical analysis and it has a well-founded theoretic background dependable by dynamic systems. So far, in comparison with the latest and more common machine learning themes like image recognition, natural language processing and pattern recognition, time series forecasting holds something of outsider status, and it obtains a few or no processing at all in preliminary courses to machine learning (ML) and data science[20]. In ML, the model is trained, tested, and it is kept at exigency until obtained persuasive results, and then evaluated on a holdout dataset. Finally, the model is deployed to production, after reaching to satisfied model's performance. Whereas in environmental area, new data is registered as it has gotten from several resources such as earth stations and satellite sensors. But after a several months the used model may need to be updated due to increasing coming huge training data. Model training is an activity that is performed one-time or often at periodic intervals to retention the efficiency of the model regarding new information. Therefore, forecasting Complex environmental systems require perception and discernment their dynamic. The complex systems formation is done by interacting several elements in a non-linear form. These complex systems demonstrate relevant nascent phenomena that cannot be interpreted by its elements analysis on an individual basis, but needs more comprehensive method to reveal concealed structure in its dynamics so as to discover transitions, or to predict undesirable immoderate events. Nature represents many physical systems where the interaction among a specific behavior, randomness and time delay leads to a wide diversity of complex dynamics [1][2][3]. This kind of behavior can be existed in neuronal dynamics [4][5], heartbeat behavior [6][7][8], earthquake activity [9][10][11][21], optical systems [12], stock markets [8][13], social networks [14], and others [15][16][17]. In this paper, Long Short-Term Memory networks (LSTMs) are taken on because they are a special type of Recurrent Neural Network (RNN). RNNs are one of neural network kinds utilized for analyzing sequences functions. Sequences are consisting of various types ofobjects like videos, images, or different factors of multidimensional environmental. It is notable that these sequences' objects depend on each other by utilizing RNNs. Where, the environmental variations cannot be distinctly perceptible in the attempt to study the non-sequent factors separately. Whereas studying all subsequent factors collectively lead to instant observation of environmental phenomenon's, environmental pollutions and climate variations.

II. Understanding LSTM Networks

Recurrent Neural Networks are networks consist of loops Fig. 1 which allow to information to be kept on running. RNN doesn't begin from essential point every time, but they learn from preceding phenomenon. While conventional neural networks can't do this.

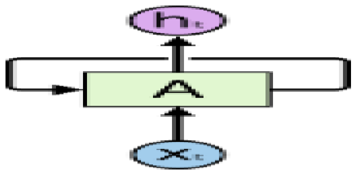

Fig. 1. Recurrent Neural Networks have loops

The graph shown in Fig. 1 illustrates an RNN, where box (A) represents the neural network body that gets some input (xt) and outputs a value (ht). The information is conveyed by a loop from one step of the network to the next. RNNs look ambiguous due to these loops' existence. Nevertheless, it becomes clear that RNNs aren't all as more different as traditional neural networks. A recurrent neural network can be considered as numerous duplicates of the same network, each network passes a message to the next one. Fig. 2

demonstrates what happens in case of unrolling the loop of RNN.

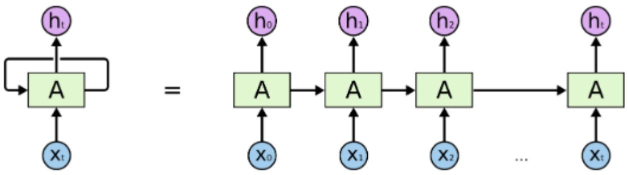

Fig. 2. An Unrolled Recurrent Neural Network

As shown in Fig. 2, the recurrent neural networks are tightly linked to sequences and lists like a chain-nature. They're considered the neural network's naturalistic design for utilizing these data. Recently, RNNs have been attained unbelievable success in applying to a different issue such as language modeling, image captioning, speech recognition, translation. The key of these successes is due to the use of LSTMs which are a very distinctive type of recurrent neural networks. LSTMs which is much better than the standard version are used for many purposes. Nearly, all excitative results are accomplished depending on recurrent neural networks. Hence these LSTMs are the essential that will lead to explore this paper [18].

III. THE PROBLEM OF LONG-TERM DEPENDENCIES

The idea of RNNs is the ability of them to relate previous information to the present tasks, but there are also some situations that demand more context. Theoretically, RNNs have unlimited ability to handle such "long-term dependencies. Though, some of RNNs don't seem to be able to learn them. Memorable that the big problem for RNNs is the information for long periods of time. In practice, their presumptive manner, not long-time learning. While a special type of RNN which is named Long Short Term Memory networks (LSTMs) was introduced by (Hochreiter & Schmidhuber, 1997) and it is capable of learning long-term dependencies [17]. All recurrent neural networks are made up of recursive neural network modules appearing in a form of chain. This frequent module in standard RNNs will be represented in a very simple function like a single tan(h) layer as illustrated in Fig. 3.

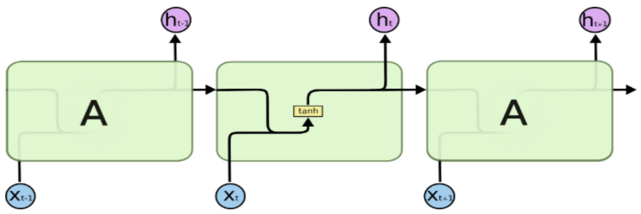

Fig. 3. The Repeating Module in a Standard RNN Contains a Single Layer

LSTMs also reveal in the chain-like structure as shown in Fig. 4, but the frequent module has a various structure. Where it has four layers related in a special way instead of having a single neural network layer.

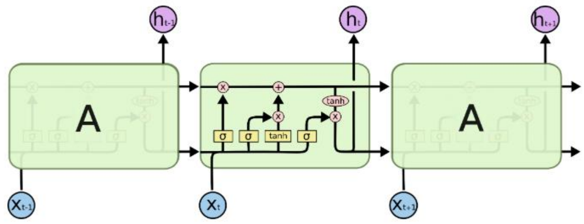

Fig. 4. The Repeating Module in an LSTM Contains Four Interacting Layers

IV. METHODOLOGY

The methodology of this paper comprised of three parts:

A. *Forecasting of Air Pollution*

The Raw pollution dataset of this paper consist of many columns' files of CSV for many environmental factors concentrations merged to one CSV file with python merge script and minimized to pollution concentrations of each hour for the following: SO2, NO_2, NO, CO with the date-time. later the dataset minimized for two years 2015 and 2016 for best fitting the expected model in order to get best forecasting for Spain capital Madrid Air Quality. This paper framed dataset for forecast the pollution concentrations at the next hour depending on the pollution concentrations of prior hours

B. *Preparation of Basic Data*

For using dataset as an index in python Pandas converted to a single date-time so that we can use it. The rows of NA values for every variable removed and few NA values founded within pollution dataset changed to 0 values. Data set parsed as as the Pandas DataFrame index using python. Clearer names are specified for each column after The NAN columns were dropped. After preparation of data the five rows of data looked like as TABLE 1 below:

TABLE 1. FIRST FIVE ROWS OF DATA

date	CO	NO	NO_2	SO_2
31/12/2016 23:00	0.7	101.0	59.0	9.0
31/12/2016 22:00	0.7	113.0	68.0	8.0
31/12/2016 21:00	0.9	170.0	83.0	11.0
31/12/2016 20:00	1.0	164.0	86.0	13.0
31/12/2016 19:00	0.7	82.0	73.0	9.0

A quick plot of each series for each of four columns in data after loading the new created pollution "clean1.csv" file and plotted each series as a separate subplot Using a python matplotlib a Fig. 5 a plot with 4subplots showing the 2 years of data for each variable. Finally, the data preparation completes in an easy-to-use form.

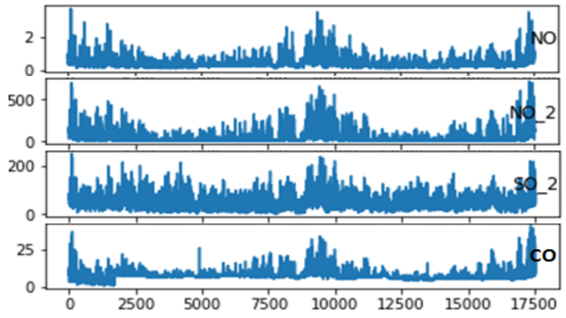

Fig. 5. Every Hour of years from 2015 to 2016

C. *Preparation of LSTM Data for Multivariate Forecast Model*

Before LSTM Data Preparation in Anaconda Python a separate environment created with installation of a Keras with TensorFlow, scikit-learn, Pandas, NumPy and Matplotlib libraries. The preparation of pollution dataset for the LSTM must be framed as a supervised learning problem and the input variables must normalized [19]. In order to predict the pollution at the current hour (t) the problem framed as supervised learning problem and more than one hour of pollution measurements at the prior time steps for 4 input variables used. The dataset splitted into two parts train and test sets, then the train and test also splitted into input and output variables. while the input variables are reshaped into the 3D dimension expected by LSTMs, namely [samples, timesteps, features]. The first configuration was to define the first hidden layer of LSTM with 50 neurons and the output layer with 1 neuron for forecasting pollution. The input shape will be 3-time step with 4 features. The efficient Adam version of stochastic gradient descent used with the Mean Absolute Error (MAE) loss function. In order for best model fitting this paper used 50 training epochs with a batch size of 2000 of four iterations. The internal state of the LSTM in Keras is reset at the end of each batch to helpful internal state which is a function of a number of days Finally, by setting the validation data argument in the fit () function to kept track of both the training and test loss during experiment of model. The training and test loss of run plotted finally fig (6). The forecasting of the test dataset done After fitting the model. and inverted the scaling on the test dataset with the expected pollution numbers. With predictions and real values in their original scale, to get error in the same units as the variable itself. then the model error scores Root Mean Squared Error (RMSE) calculated. Finally, the final RMSE of the model on the test dataset calculated and printed its value 0.141 at the end in order to store large arrays of numbers on disk the model has been saved as HDF5 file format.

V. Result and Discussions

This LSTM Model evaluated by running it and observing the created plot for the train and test loss during training. the compilation of the model done with by calling the python fit() function, to call history which contains a trace of the loss and other metrics specified Interestingly. stochastic of LSTM which makes a different diagnostic plot for each experiment Fig. 6. In order to diagnose the Overfitting and Underfitting of this model by reviewing its performance over time. This paper recorded the scores at the end of each epoch. the log loss (binary_crossentropy) and measured accuracy of each epoch was optimized during the model compiling, with recorded and calculated accuracy in the history trace for each training epoch Fig. 7. After calling fit() Each score is accessed by a key in the history object by default. the first running of this model was underfit cause the performance on the training set was better than the validation Fig. 6 (b,c) the main reason of this underfit is data size was big about 39000 rows and performance has leveled off. After minimizing the data size to 17000 with changing the configuration of model, the number of hidden layers changed from 10 to 50 and the capacity of the model increased, this led the performance improved and also there was another reason for this performance change related about increasing the number of training epochs. In this case the performance of the model behaved good on both the train and validation sets Fig. 6 (a,d). The train and validation loss decreased and stabilized around the very near points to each other. Ideally, this paper liked to see good fit model performance, with a lot of data there is no possible. Repeated the diagnostic run multiple times. For more robust idea of the behavior of the model. over time the train and validation traces plotted for each run. At the end of each training epoch The Train and test loss are printed. Then achieved a respectable RMSE like 0.141, which is lower than an RMSE of 2 others found with a persistence model and the result prediction values for one hour later prove the fitting of the model TABLE 2.

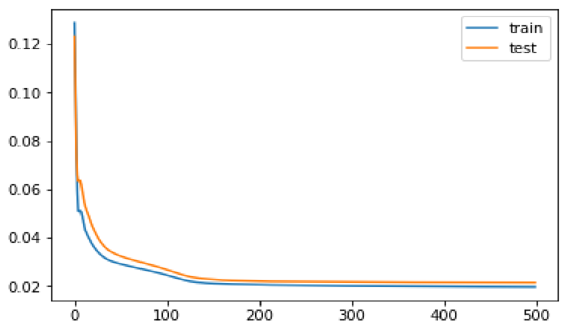

Fig. 6.a training and testing loss plot

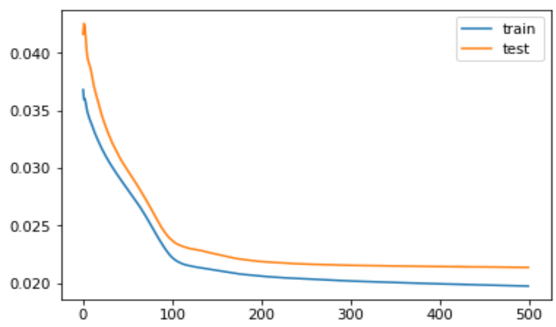

Fig. 6.b training and testing loss plot

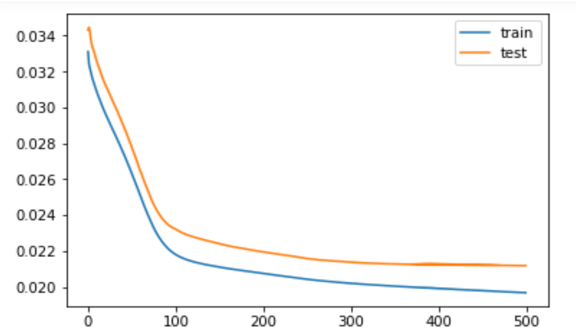

Fig. 6.c training and testing loss plot

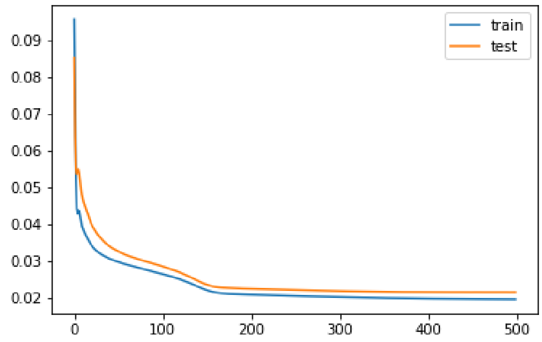

Fig. 6.d training and testing loss plot

```
Train on 8760 samples, validate on 8490 samples
Epoch 1/500
8760/8760 [==============================] - 1s 144us/step - loss: 0.0453 - val_loss: 0.038
Epoch 2/500
8760/8760 [==============================] - 0s 30us/step - loss: 0.0381 - val_loss: 0.0429
Epoch 3/500
8760/8760 [==============================] - 0s 30us/step - loss: 0.0375 - val_loss: 0.0383
Epoch 4/500
8760/8760 [==============================] - 0s 29us/step - loss: 0.0333 - val_loss: 0.0355
Epoch 5/500
8760/8760 [==============================] - 0s 30us/step - loss: 0.0329 - val_loss: 0.0348
Epoch 6/500
8760/8760 [==============================] - 0s 29us/step - loss: 0.0322 - val_loss: 0.0340
Epoch 7/500
8760/8760 [==============================] - 0s 27us/step - loss: 0.0314 - val_loss: 0.0342
```

Fig. 7. history trace for each training epoch

TABLE 2. PREDICTION VALUES FOR CURRENT TIME AND ONE HOUR LATER

time	Co	No	No_2	So_2
31/12/2016 23:00	0.7	101	59	9
31/12/2016 22:00	0.7	113	68	8
31/12/2016 21:00	0.9	170	83	11
31/12/2016 20:00	1	164	86	13
31/12/2016 19:00	0.7	82	73	9
time+1	**Co**	**No**	**No_2**	**So_2**
31/12/2016 24:00:00 AM	0.5	103	58	7
31/12/2016 23:00	0.8	115	67	6
31/12/2016 22:00	0.7	168	85	10
31/12/2016 21:00	0.9	162	87	14
31/12/2016 20:00	0.8	81	80	10

VI. Conclusion

The difficulty of predicting multivariate time series and the development of intelligent systems that can design predictive complex environmental dataset automatically Requires a good Machine Learning model. To tackle such problems, this paper proposes an approach based on Neural networks like Long Short-Term Memory (LSTM) in keras and TensorFlow environment. The proposed approach can process multivariate environmental time series data and is robust to the number of samples, numeric ranges of data etc. Running the model many times with increasing the number of memory cells in a hidden layer validate the effectiveness of the approach in accomplishing multivariate time series forecasting. And also, this paper recommends to best dataset preparation and using data size less than 2000 rows for better fit model. In every try for validation the model could be updated in each time step. tests are needed in order to determine if it should be better to refit the model from scratch or update the weights with a few more training epochs including the new sample. After many tests for model reasonable predictions made cause The LSTM network may be able to learn the trend in the data. this model can be used to predict new environmental observations and also recommend Varying the time step index of the sequence output and training epochs to see if there is a relationship between the index and how hard the problem is to learn.

Acknowledgment

We thank Dr. Jason Brownlee professional developer, and machine learning practitioner from Australia [Machine Learning Mastery] who provided us many mini courses about how to get started with machine learning algorithms, insight and expertise that greatly assisted the research.

References

[1] Mitchell, M. Complexity: a guided tour. Oxford: University press (2009), pp.368.

[2] Crutchfield, J. P. "Between order and chaos." Nature Phys. 8(1), pp.17–24, (Jan. 2012).

[3] Charbonneau, P. "Natural complexity: Glass and Jamming Transitions: From Exact Results to Finite-Dimensional Descriptions". Annual Review of Condensed Matter Physics: Princeton Univ. press. 8, pp. 265-288, (2017).

[4] Sancristóbal, B., Rebollo, B., Boada, P., Sanchez-Vives, M. V., & Garcia-Ojalvo, J. "Collective stochastic coherence in recurrent neuronal networks." Nature Physics, 12(9), pp.881-888, (2016).

[5] Palchykov, V., Mitrovic, M., Jo, H.-H., Saramäki, J. & Pan, R. K. "Inferring human mobility using communication patterns." Scientific Reports, 4 (6174), (2014).

[6] Parlitz, U., Berg, S., Luther, S., Schirdewan, A., Kurths, J., & Wessel, N. "Classifying cardiac biosignals using ordinal pattern statistics and symbolic dynamics." Computers in biology and medicine, 42(3), pp.319-327, (2012).

[7] Zanin, M., Zunino, L., Rosso, O. A., & Papo, D. "Permutation entropy and its main biomedical and econophysics applications: a review." Entropy, 14(8), pp.1553-1577, (2012).

[8] Soriano, M. C., García-Ojalvo, J., Mirasso, C. R., & Fischer, I. "Complex photonics: Dynamics and applications of delay-coupled

semiconductors lasers." Reviews of Modern Physics, 85(1), pp.421, (2013).

[9] Corral, A. "Long-term clustering, scaling, and universality in the temporal occurrence of earthquakes." Physical Review Letters, 92(10), pp.108501, (2004).

[10] Kagan, Y. Y. "Worldwide earthquake forecasts." Stochastic Environmental Research and Risk Assessment, 31(6), pp.1273-1290, (2017).

[11] Neiman, A. B. & Russell, D. F. "Models of stochastic biperiodic oscillations and extended serial correlations in electroreceptors of paddlefish." Physical Review E, 71(6), pp. 061915, (2005).

[12] Zunino, L., Zanin, M., Tabake, B. M., Pérez, D. G. & Rosso, O. A. "Forbidden patterns, permutation entropy and stock market inefficiency." Physica A: Statistical Mechanics and its Applications, 388(14), pp.2854–2864, (2009).

[13] Lindner, J. F., Kohar, V., Kia, B., Hippke, M., Learned, J. G., & Ditto, W. L. "Strange nonchaotic stars." Physical review letters, 114(5), pp.054101, (2015).

[14] Peng, C. K., Mietus, J., Hausdorff, J. M., Havlin, S., Stanley, H. E., & Goldberger, A. L. "Long-range anticorrelations and non-Gaussian behavior of the heartbeat." Physical review letters, 70(9), pp.1343. (1993).

[15] Ginzburg, N. S., Rozental, R. M., Sergeev, A. S., Fedotov, A. E., Zotova, I. V., & Tarakanov, V. P. "Generation of rogue waves in gyrotrons operating in the regime of developed turbulence." Physical review letters, 119(3), pp. 034801. (2017).

[16] Akhmediev, N., Kibler, B., Baronio, F., Belić, M., Zhong, W. P., Zhang, Y., ... & Lecaplain, C. "Roadmap on optical rogue waves and extreme events." Journal of Optics, 18(6), pp.063001, (2016).

[17] Hochreiter, S., & Schmidhuber, J. "Long short-term memory." Neural computation, 9(8), pp.1735-1780, (1997).

[18] Christopher olah, "Understanding LSTM Networks" http://colah.github.io/Christopher olah, colah's blog/, Posted on August 27, 2015.

[19] Jason Brownlee, "How to Diagnose Overfitting and Underfitting of LSTM Models." https://machinelearningmastery.com/diagnose-overfitting-underfitting-lstm-models/, September 1, 2017

[20] Jason Brownlee, "Long Short-Term Memory Networks with Python, Develop Sequence Prediction Models with Deep Learning." Edition: v1.5, 2018, online book, https://machinelearningmastery.com/lstms-with-python/.

[21] Robert, J., Geller, Yan, Y. K., David, D. J & Mulargia, F. "Earthquakes cannot be predicted". Science 275(5306), pp.1616, (1997).