STAT 527 Final Project

Multivariate Time Series Forecasting Using LSTM and CNN Experiments on Climate Daily Data

Emily (Wendan) Yan, Wenqian Shao, Carl (Kaiyuan) Duan,
Shujie Chen, Hanyi Wang

1. Introduction

1.1 Method Introduction

Deep learning is an important part of machine learning methods based on artificial neural networks, and it is about learning multiple levels of representation and abstraction that help to make sense of data such as images, sound, and text. Common applications include image and speech recognition. The famous deep learning models are Classic Neural Networks (Multilayer Perceptrons), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs).

1.2 Task Introduction

Time series forecasting is a crucial task in machine learning areas and it has a wide range of applications in economics and finance, traffic and environment, and supply chain management. There are many approaches for modeling and forecasting time series, such as the classic ARIMA model and machine learning methods, such as linear regression, ridge regression, and deep learning models, such as Convolution Neural Networks (CNN), Recurrent Neural Networks (RNN) and Graph Neural Networks (GNN). These models have been applied to the forecasting tasks. However, in many cases, time series forecasting involves multivariates. This multivariate time series data structure is more interesting and has more prevalence in real-life or business situations. Regarding the multivariate time series forecasting problem, there are also methods to achieve this goal, such as Vector Auto Regressive Moving Average models (VARMA), i.e. a vector form of autoregressive integrated moving average (ARIMA), which can be used to examine the relationships among several variables in multivariate time series analysis and also machine learning methods such as deep learning methods RNN, CNN, etc. Our project is aimed at applying Convolutional neural networks (CNN) and Long short-term memory (LSTM)(a variant of the RNN model) to build multivariate time series forecasting models and compare the two methods' performance on a specific multivariate time series forecasting task.

2. Data

2.1 Data Description

[Daily climate time series dataset](#) is a multivariate time series dataset that provides measurements of temperature, humidity, wind speed, and pressure from January 1st, 2013 to April 24th, 2017 in the city of Delhi, India. It was collected from Weather Underground API. This dataset has 5 variables, including:

- date: date of format YYYY-MM-DD;
- meantemp: mean temperature averaged out from multiple 3 hour intervals in a day;
- humidity: humidity value for the day (units are grams of water vapor per cubic meter volume of air);
- wind_speed: wind speed measured in kmph;
- meanpressure: pressure reading of weather (measure in atm).

2.2 Exploratory Data Analysis

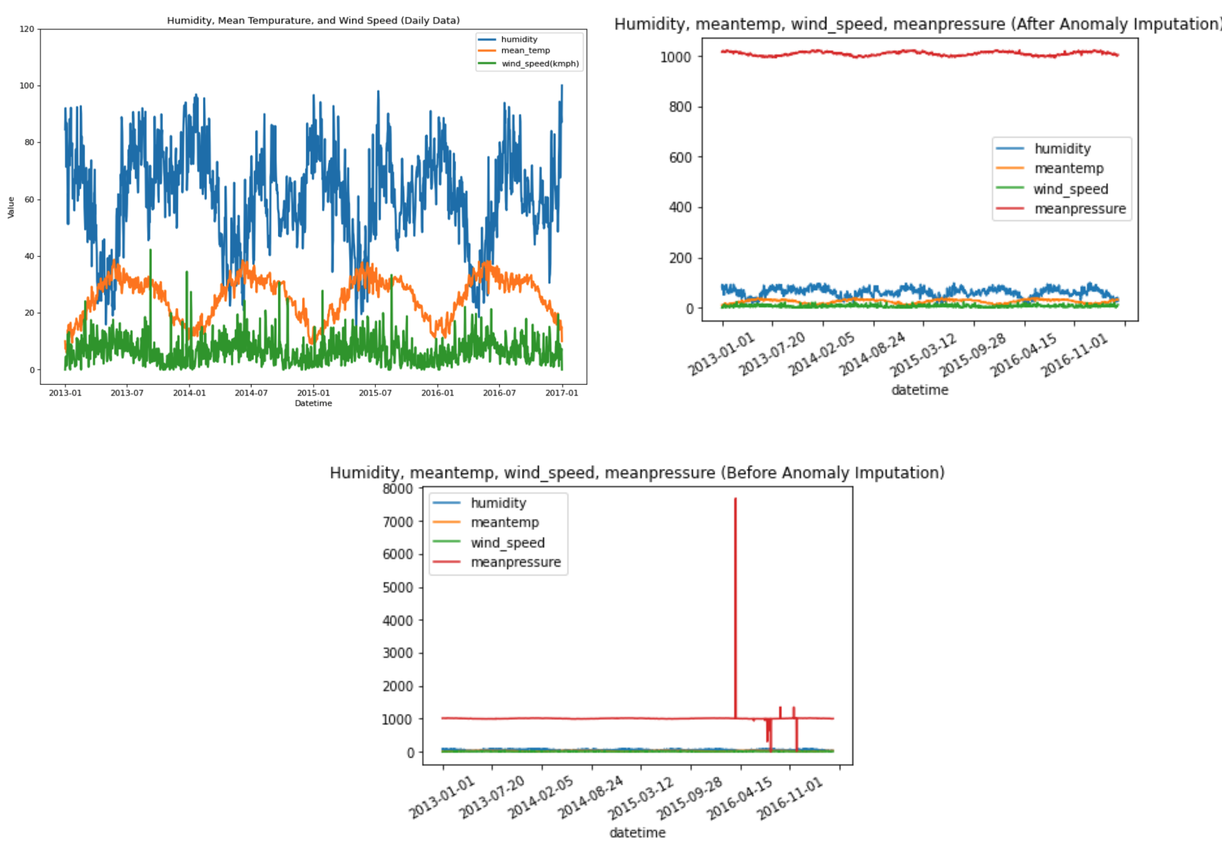

Figure 1: Humidity, Mean Temperature, Mean Pressure, and Wind Speed (Daily Data)

From the figure 1, we can see that the fluctuation of the humidity variable is bigger than the other two variables. Therefore, we are considering setting the humidity variable as the response variable so we can build models that have more significant variations over time and give us a clear and intuitive sense of the performance of these models. Based on this consideration, our main purpose for this dataset is to predict the humidity using the date, temperature, wind speed, and pressure.

Another observation is the mean pressure seems to have anomalies on some of the dates since they are 8 times higher than the average of the rest of the data points. It is considered outliers in the CNN model and we did anomaly imputation based on STL(Seasonal and Trend decomposition using Loess) decomposition and variable span smoother. We used the library supersmoother for the implementation of variable span smoother.

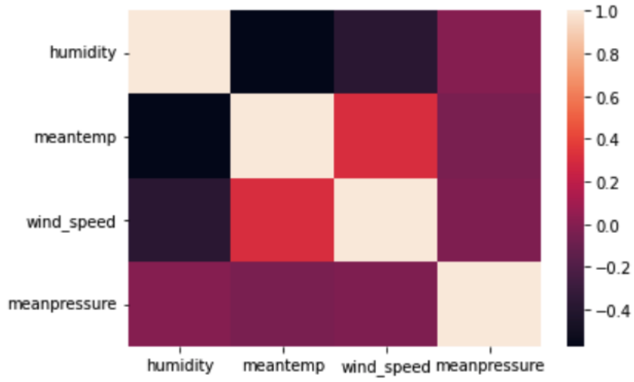

Figure 2: Correlation Plot of all Features

	humidity	**meantemp**	**wind_speed**	**meanpressure**
humidity	1.000	-0.572	-0.374	0.001
meantemp	-0.572	1.000	0.306	-0.039
wind_speed	-0.374	0.306	1.000	-0.021
meanpressure	0.001	-0.039	-0.021	1.000

Table 1: Correlation Coefficients of all Features

In figure 2 and table 1, we can see that there are no highly correlated variables, all of the absolute values of correlation coefficients are less than 0.5. Hence do not need to worry about the problem caused by the high correlation.

After Exploratory Data Analysis, we have a basic understanding of the data in this dataset. Then we start our neural network modelings for this dataset.

3. Models

This project's purpose is to build neural networking models to fulfill multi-step time series forecasting. More specifically, we want to use the data in the [Daily climate time series dataset] to build neural networking models to achieve the goal of predicting the humidity of the upcoming seven days.

3.1 Convolutional neural network (CNN)

For CNN, we have tried Multi-step Time Series Forecasting With a Univariate CNN, Multi-step Time Series Forecasting With a Multichannel CNN, and Multi-step Time Series Forecasting With a Multihead CNN.

Before we build our models, we need to prepare our dataset and define our evaluation metric:

1. Load and Prepare Dataset: After loading the data, we write a function to deal with missing data problems whose logic is to fill missing values with a value at the same time one day ago.
2. Evaluation Metric: In our project, we will use Root Mean Squared Error (RMSE) as our metric, and the performance metric for this problem will be the RMSE from day 1 to day 7.

3.1.1 Multi-step Time Series Forecasting With a Univariate CNN

First, we start with building a multi-step time series forecasting model for univariate data, which in this setting, is the daily humidity, in order to get some intuition about the performance of the CNN method and also as a comparative result with the following multivariate time series models.

Convolutional neural networks are capable of automatically learning features from sequence data, support multiple-variate data, and can directly output a vector for multi-step forecasting. The method logic of Multi-step Time Series Forecasting With a Univariate CNN is using this property of CNN to fulfill the Multi-step Time Series Forecasting task.

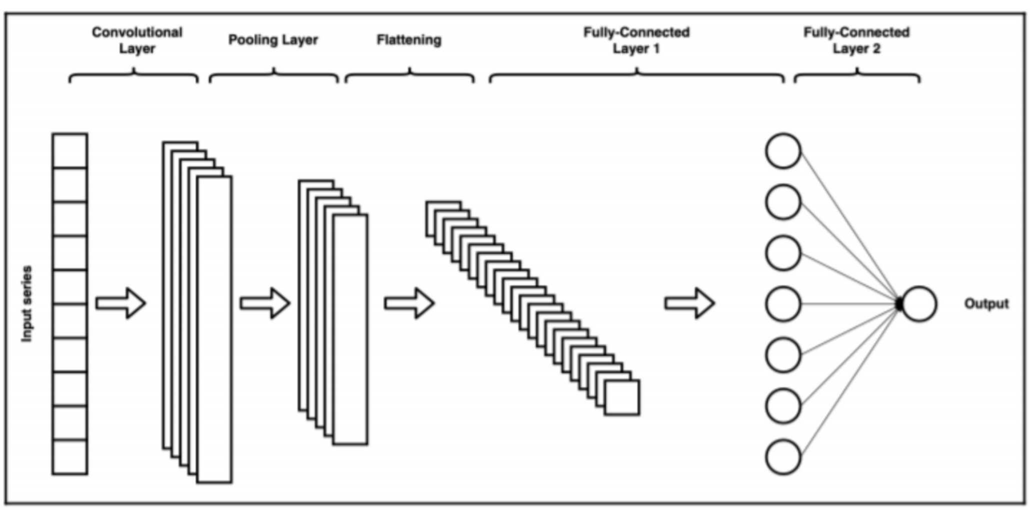

Figure 3: Architecture for Multi-step Time Series Forecasting With a Univariate CNN

In practice, for this scenario, the numbers of prior days we use as input are 7 and 14, which are achieved by changing the n_input variable and the model we build is a CNN model with one convolution layer with 16 filters and a kernel size of 3, and a pooling layer will reduce these feature maps by 1/4 their size, a fully connected layer, the output layer which predicts the next seven days in the sequence.

3.1.2 Multi-step Time Series Forecasting With a Multichannel CNN

The method logic of Multi-step Time Series Forecasting With a Multichannel CNN is as follows: We separate multivariate time series into univariate ones and perform feature learning on each univariate series individually, and then a traditional MLP is concatenated at the end of feature learning that is used to do the forecasting.

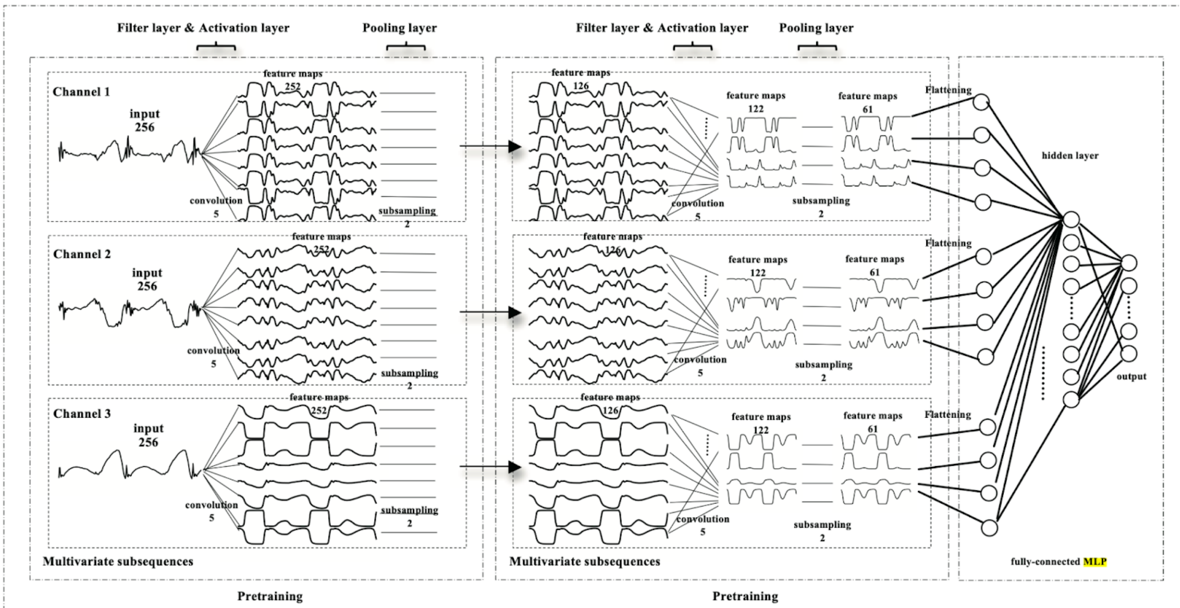

Figure 4: Architecture for Multi-step Time Series Forecasting With a Multichannel CNN

In practice, with a little trial and error, we fulfill one model that performs well by using two convolutional layers with 32 filter maps followed by a pooling layer, then another convolutional layer with 16 filter maps and a pooling layer, then the fully connected layer with 100 nodes, the output layer which predicts the seven days in the sequence. And we use 14 days of prior observations across eight of the input variables as we run in the final section of the prior section that results in slightly better performance.

3.1.3 Multi-step Time Series Forecasting With a Multihead CNN

Multihead is a variant of CNN in which for each time, the time series is processed on an utterly independent convolution. In Multihead CNN, each head uses a one-dimensional CNN to extract the features of each input channel and combine them together in a convolutional head. They are combined together and are called convolutional heads. Multihead CNN combines multidimensional input data well to represent the relationship between multiple variables and can be used for multivariate prediction.

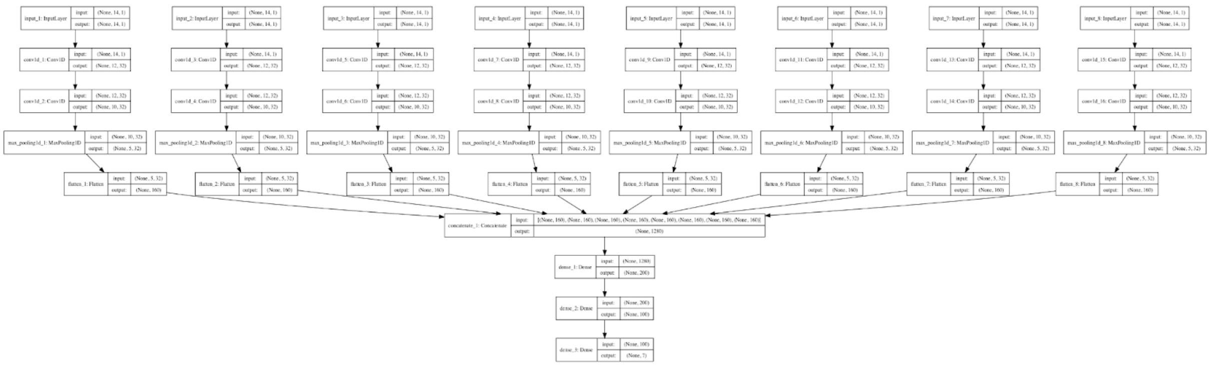

Figure 5: Architecture for Multi-step Time Series Forecasting With a Multihead CNN

In practice, for this scenario, we further extend the CNN model to have a separate sun-CNN model or head for each input variable. When the model is used, it will require eight arrays as input: one for each of the submodels. To do this, we create a list of 3D arrays where each 3D array contains one feature.

3.2 Long short-term memory (LSTM)

Unlike CNN, Long Short Term Memory Network(LSTM) is an advanced RNN, another powerful and robust type of neural network besides CNN.

While convolutional neural networks employ filters within convolutional layers to transform data, recurrent neural networks reuse activation functions from other data points in the sequence to generate the next output in a series. When processed, the RNN passes the previously hidden state to the next step in the sequence. The hidden state acts as neural network memory, which contains information about data the network has seen before. However, due to short-term memory, the typical RNNs are only able to use information in the last parts of the sequence. In this project, typical RNNs would predict based on the weather conditions in the past few days and may miss important information. Given this, this project uses the LSTM method to make the prediction, which allows the model to predict the weather 10 hours after the data based on the weather conditions for the past 24 days.

The method logic of LSTM is as follows：

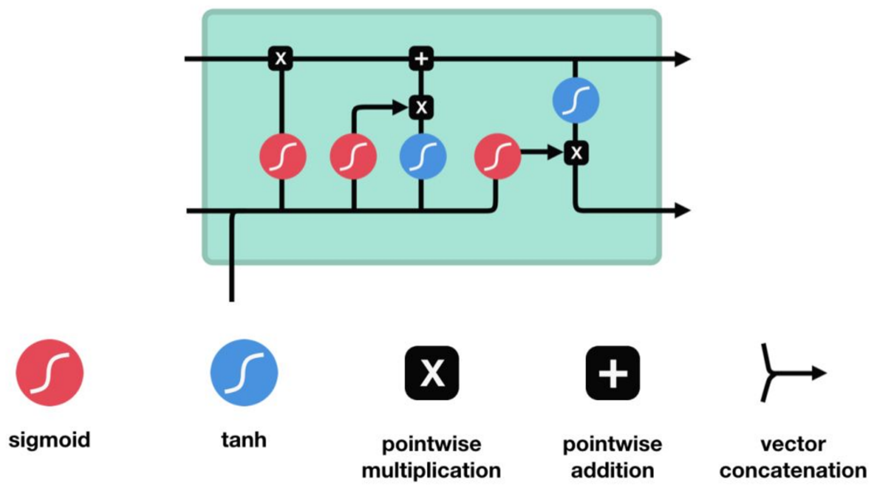

Figure 6: Architecture for the LSTM model

The core concepts of LSTM lie in the cell state and the gate structure. The cellular state acts as a pathway for information to travel through a sequence. Therefore, even information of earlier time steps can be carried into cells of later time steps, which overcomes the effect of short-term memory. Information is added and removed through a "gate" structure, which learns what information to save or forget during training. The activation function Tanh is used to help adjust the values flowing through the network by limiting the number to a range of -1 to 1. Similarly, the Sigmoid activation function "compresses" the value to the range 0 to 1. This allows the model to update or forget information (use a [0,1] scale to represent the importance of the information).

4. Summary

In this project, we use several models to fit the daily climate time series data and predict the humidity. We tried multi-step time series forecasting with a univariate CNN, multi-step time series forecasting with a multichannel CNN, multi-step time series forecasting with a multihead CNN and LSTM models.

For the detailed setup of the CNN model, the objective is to use an input of 14 days of daily humidity to predict the subsequent 7 days of daily humidity. The model was trained on 178 weeks of data and tested on 46 weeks of data later. After serialization, there are 1226 rows of training data, there are 46 weeks of target data to predict in the testing set.

	MSE	**RMSE**
Univariate multi-step CNN	104.332	10.214
Multichannel multi-step CNN	133.958	11.574
Multi-headed	120.434	10.974

multi-step CNN		
LSTM	236.449	15.377

Table 2: MSE and RMSE of the models

Table 2 provides the experimental results of the forecasting models of the dataset. We observe that for this dataset, the performance of the CNN models varied not too much. And CNN has better performance than LSTM.

5. Individual Contribution

There are five members collaborating on this project. Our individual contributions are as follows:
Emily (Wendan Yan): data cleaning, method research, notebook for running the CNN models
Wenqian Shao: selection of the dataset, method research, paper writing (Data and models part)
Carl (Kaiyuan Duan): selection of the dataset, method research, notebook for running the LSTM models
Shujie Chen: method research, paper writing (Introduction and models part)
Hanyi Wang: method research, paper writing (models part)

Appendix

Our code can be found at the following link:
https://github.com/LilyHanyiWang/time-series-forecasting-with-ML.git

References

[1] J. Yin *et al.*, "Experimental Study of Multivariate Time Series Forecasting Models," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, New York, NY, USA, 2019, pp. 2833–2839. doi: 10.1145/3357384.3357826.

[2] J. Alammar, "The Illustrated Transformer." https://jalammar.github.io/illustrated-transformer/ (accessed Jun. 11, 2022).

[3] A. Vaswani *et al.*, "Attention Is All You Need," arXiv, arXiv:1706.03762, Dec. 2017. doi: 10.48550/arXiv.1706.03762.

[4] "How To Do Multivariate Time Series Forecasting Using LSTM," *Analytics India Magazine*, Jul. 11, 2021. https://analyticsindiamag.com/how-to-do-multivariate-time-series-forecasting-using-lstm/ (accessed Jun. 11, 2022).

[5] J. Brownlee, “Convolutional Neural Networks for Multi-Step Time Series Forecasting,” *Machine Learning Mastery*, Oct. 07, 2018. https://machinelearningmastery.com/how-to-develop-convolutional-neural-networks-for-multi-step-time-series-forecasting/ (accessed Jun. 11, 2022).

[6] “Daily Climate time series data.” https://www.kaggle.com/sumanthvrao/daily-climate-time-series-data (accessed Jun. 11, 2022).

[7] S. Verma, “Understanding Input and Output shapes in LSTM | Keras,” *Medium*, Oct. 05, 2021. https://shiva-verma.medium.com/understanding-input-and-output-shape-in-lstm-keras-c501ee95c65e (accessed Jun. 11, 2022).

[8] J. H. Friedman, “A VARIABLE SPAN SMOOTHER,” Oct. 1984.