BATCH DATA PIPELINE FOR VISUALIZING AND FORECASTING AIR POLLUTION*

1st Nguyen Hoai Bao
University of Information Technology
Ho Chi Minh City, Vietnam
19520405@gm.uit.edu.vn

2nd Nguyen Vo Thien An
University of Information Technology
Ho Chi Minh City, Vietnam
19521186@gm.uit.edu.vn

3rd Do Vu Gia Can
University of Information Technology
Ho Chi Minh City, Vietnam
19521271@gm.uit.edu.vn

4th Nguyen Thanh Phuc
University of Information Technology
Ho Chi Minh City, Vietnam
19522040@gm.uit.edu.vn

***Abstract*—Poor air quality due to rapid urbanization and industrialization is causing various lung ailments, which poses a threat to human health. Monitoring, modeling, and forecasting air quality could help promote awareness and protect people from the adversities of air pollution. Air quality is monitored through various air quality monitoring stations located in and around a region. In this paper, we build some models for air quality forecasting. The proposed framework was extensively evaluated for forecasting the real-world air quality data of Ho Chi Minh City. The framework was found to be effective in predicting the concentrations of pollutants in the air.**

I. Introduction

The public's awareness of air pollution has grown recently. Human health is significantly impacted by air pollution, particularly by atmospheric particle matter. Small PM2.5 is sufficient for it to circulate in the blood, which can lead to cardiovascular disease and even death. Countries must cooperate for the environment's long-term protection, create laws to lessen air pollution. Research into air quality has recently concerns have been quickly growing, which indicates that more people are paying attention to the air quality problems. Three key areas have been the subject of studies on air quality: air quality monitoring, causal analysis, forecasting, and prediction. Air Environmental sensors can be simply installed for the collection of high-quality information, or retrieved from public databases. Additionally to this problem, we built some models in order to forecast the air quality in the future based on the current data and the historical data.

II. Plafform

- Pyspark: It is majorly used for processing structured and semi-structured datasets. It also provides an optimized API that can read the data from the various data source containing different files formats. Thus, with PySpark you can process the data by making use of SQL as well as HiveQL
- Terraform: HashiCorp Terraform is an infrastructure as code tool that lets you define both cloud and on-prem resources in human-readable configuration files that you can version, reuse, and share. You can then use a consistent workflow to provision and manage all of your infrastructures throughout its lifecycle.
- LocalStack: LocalStack provides an easy-to-use test/mocking framework for developing Cloud applications. It spins up a testing environment on your local machine that provides the same functionality and APIs as the real AWS cloud environment.
- Apache Airflow: Apache Airflow is a powerful and widely-used open-source workflow management system (WMS) designed to programmatically author, schedule, orchestrate, and monitor data pipelines and workflows. Airflow enables you to manage your data pipelines by authoring workflows as Directed Acyclic Graphs (DAGs) of tasks
- PowerBI: Power BI Desktop to build custom usage metrics reports based on the underlying dataset. See Establish a connection to a published dataset for details. Power BI Desktop uses a Live Connection to the Report Usage Metrics Model dataset.
- Snowflake: Snowflake optimizes and stores data in a columnar format within the storage layer, organized into databases as specified by the user. dynamically as resource needs change. When virtual warehouses execute queries, they transparently and automatically cache data from the database storage layer.

III. Workflow

A. Ingest data from API and push to Kinesis Firehose

Data ingested from API are in JSON format like Fig. 2. We then fetch the data to a lambda function called "Data_to_firehose" - a function where data is flattened automatically every 1 hour and pushed to firehose.

B. Data from Firehose to S3 - Localstack

Firehose stream created using Terraform can connect directly to an S3 bucket. Whenever there is data pushed to firehose stream, it's passed to S3 bucket automatically.

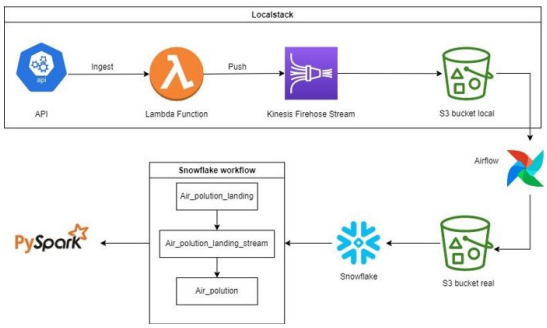

Fig. 1. Data pipeline.


```
{
  "coord":[
    50,
    50
  ],
  "list":[
    {
      "dt":1605182400,
      "main":{
        "aqi":1
      },
      "components":{
        "co":201.94053649902344,
        "no":0.01877197064459324,
        "no2":0.7711350917816162,
        "o3":68.66455078125,
        "so2":0.6407499313354492,
        "pm2_5":0.5,
        "pm10":0.540438711643219,
        "nh3":0.12369127571582794
      }
    }
  ]
}
```


Fig. 2. Raw data.

C. Upload data from the S3 Localstack to the real S3 using Airflow

Include 3 tasks in order:

- Download_from_s3localstack: Download data in s3 bucket localstack.
- Upload_to_s3real: Upload file downloaded to aws s3 bucket.
- Remove_place: Delete file in s3 bucket localstack

D. S3 to SnowFlake

To be able to import data from S3 bucket to SnowFlake, we need to set up these following things:

- Create external stage: Point to folder in a specified bucket in S3.
- Create table: Air_polution_landing and Air_polution with schema like Table I .
- Snowpipe: Update the SQS of pipe in S3 bucket Create Event notifications. In short, send notification whenever there is data upload to specific bucket.

TABLE I
AIR_POLUTION TABLE DESCRIPTION

Attributes	Data type	Descrition
RECORD-DATE	DATE-TIME	Date and time of record
CO	FLOAT	concentration of CO
NO	FLOAT	concentration of NO
NO2	FLOAT	concentration of NO
O3	FLOAT	concentration of O3
SO2	FLOAT	concentration of SO2
PM2_5	FLOAT	concentration of PM2_5
PM10	FLOAT	concentration of PM10
NH3	FLOAT	concentration of NH3
AIR-QUALITY	INT	Air Quality Index. Possible values: 1, 2, 3, 4, 5. Where 1 = Good, 2 = Fair, 3 = Moderate, 4 = Poor, 5 = Very Poor

Using Snowflake task and stream, we execute the following tasks

1) Snowpipe check: Check if there is new data in bucket or not every 3 minutes. If there is, process to step 2, otherwise skip.
2) Data from S3 bucket to Air_polution_landing: Read JSON file and store data in a table called "Air_polution_landing". It play as an temporary table before data is transfered to main table.
3) Stream on Air_polution_landing: When data is read into Air_polution_landing table, a stream on this table also receive the same data
4) Stream_to_main_table: Data on stream will be transfered to main table Air_polution and deleted automatically once the process is done.
5) Truncate_table: Truncate table Air_polution_landing to save space.
6) Src_file_remove: Remove file in S3 bucket

E. Snowflake to Pyspark Dataframe

Snowflake works with both Python and Spark, allowing developers to leverage Pyspark capabilities in the platform. The Snowflake Connector for Spark ("Spark connector") brings Snowflake into the Apache Spark ecosystem, enabling Spark to read data from, and write data to, Snowflake.The Snowflake Connector for Python provides an interface for developing Python applications that can connect to Snowflake and perform all standard operations. Using SparkSQL, we query data from table Air_polution and read it as a Spark dataframe

IV. Models

In this section, we present two models which are applied to train the time series dataset having from the above collection.

A. Autoregressive Integrated Moving Average (ARIMA)

An autoregressive integrated moving average, or ARIMA, is a statistical analysis model that uses time series data to either better understand the data set or to predict future trends. A statistical model is autoregressive if it predicts future values based on past values.

The model's goal is to predict future by examining the differences between values in the series instead of through actual values.

This model include three components:

- Autogression (AR): refer to a model which reflects a changing variable self-regression to its own previous value.
- Integrated (I): represent for the different of raw observations to data values are replaced by the different between simple data values and previous values.
- Moving Average (MA): combine the dependency between an observation and a residual error from a moving average model and applied to the lagged observations.

However, this model is not good under certain market conditions such as financial crises or periods of rapid technological change.

B. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks are a special type of recurrent neural network (RNN) capable of learning long-term dependence in sequential prediction problems. This is a behavior required in complex problem domains like machine translation, speech recognition, and more.

The main features of this model is that remembering data information through many stages without going through the training step.

LSTM has the sequence architecture, instead of having a neural layer, it has four layer to interact with each other specially.

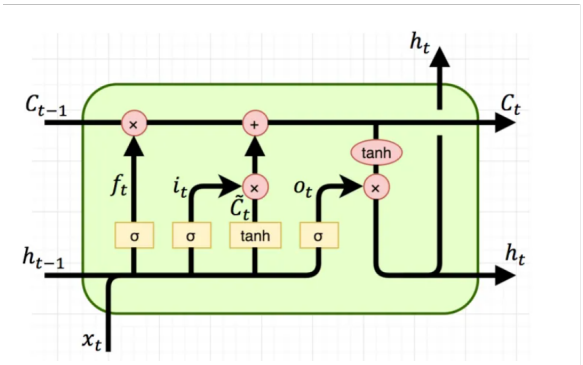

Fig. 3. An example for LSTM Model

A common LSTM unit is generated by a cell, an input gate, an output gate and a forget gate. The cell is responsible for remembering values over arbitrary time intervals and three remaining gates regulate the flow of information into and out of the cell.

V. Results

Now, we present the experimental results which helps us evaluate the performance of two models for forecasting air pollution problems.

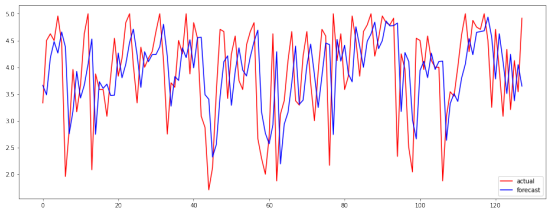

Fig. 4. Predict vs actual of ARIMA model.

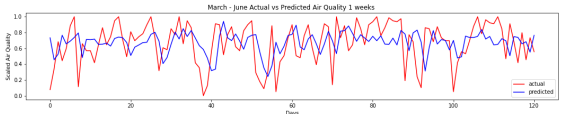

Fig. 5. Predict vs actual of LSTM with 1 week time interval.

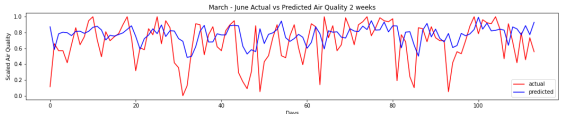

Fig. 6. Predict vs actual of LSTM with 2 weeks time interval.

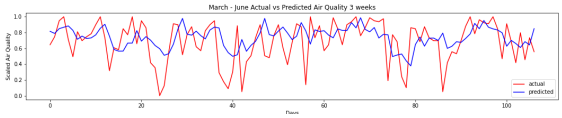

Fig. 7. Predict vs actual of LSTM with 3 weeks time interval.

As can be seen from the figures 4,5 and 6, when applied for the short period of time, LSTM hands out the quite low results, which is proven when the prediction is quite different from the actual value. In contrast, when applied for one month time interval, the results are pretty good. This says that LSTM operates better over the long time interval, thanks to its long-term dependency learning. In generally, The performance of the LSTM model is quite fit with the time-series data.

VI. Conclusion

This paper presents a model to predict air pollutant concentrations based on historical air pollutant concentration data, meteorological data, and time stamp data. The LSTM model is well-suited for modeling time series with long time dependencies, and can automatically determine the optimum point. Experiments showed that the proposed LSTM algorithm

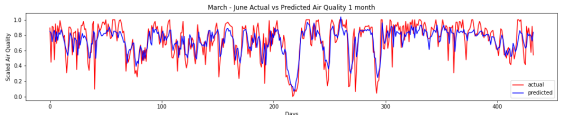

Fig. 8. Predict vs actual of LSTM with 1 month time interval.

performed better than other algorithms, with lower error rates, greater accuracy, and smaller mean and median error values. Our model can be used to predict air pollutant concentrations or air quality at different scales. Although the long-term prediction performance of prediction tasks was reduced, the proposed model was suitable for longer period prediction tasks.

REFERENCES