# Real-time Inhouse Weather Forecast Applying Apache Spark

Phan Le Chi Trung[1†], Tran Nguyen Man[1†], Do Trong Hop[1*]

[1*]University of Information Technology, Vietnam National University, Ho Chi Minh City, Vietnam.

*Corresponding author(s). E-mail(s): hopdt@uit.edu.vn;
Contributing authors: 21522725@gm.uit.edu.vn; 21522325@gm.uit.edu.vn;
[†]These authors contributed equally to this work.

**Abstract**

Trong thực tế cuộc sống ngày nay, ngành nông nghiệp đang ngày càng phát triển theo hướng số hoá, hiện đại hoá. Việc áp dụng công nghệ kỹ thuật vào việc cải thiện chất lượng nông trại cụ thể là việc trồng trọt trong nhà là quan trọng và cần thiết. Đặc biệt đối với một số nông sản có tính chất phụ thuộc vào các yếu tố môi trường thì việc nuôi trồng trong nhà sẽ cần có sự theo dõi nghiêm ngặt về các yếu tố xung quanh. Tuy nhiên, hiện nay ta có thể thấy được rất nhiều mô hình được xây dựng để dự đoán các yếu tố môi trường bên ngoài trời nhưng lại có rất ít các mô hình dự đoán các yếu tố này bên trong các phòng kín như các nhà nuôi trồng, nhà kính,vv. Ngoài ra, các dữ liệu liên quan đến các yếu tố này sẽ gia tăng kích thước dữ liệu tăng dần theo thời gian nên không thể thực hiện huấn luyện các mô hình theo phương thức thông thường. Nhận thấy được tiềm năng cũng như cơ hội trong nhu cầu này, nhóm quyết định xây dựng một mô hình có khả năng dự đoán liên tục theo thời gian thực các yếu tố xung quanh ứng dụng công nghệ Big Data và gia tăng tốc độ cũng như giảm bớt độ trễ dự đoán để tăng tính thiết thực và ứng dụng của mô hình.

**Keywords:** Inhouse Weather Forecast, Apache Spark, Spark Streaming, Big Data, Deep Learning, Forecaster, BigDL

## 1 Giới thiệu

Mô hình của nhóm sẽ được xây dựng để có thể được huấn luyện và hoạt động trên hệ thống phân tán và dự đoán với thời gian nhanh nhất. Nhận thấy việc các yếu tố

bên trong nhà có thể bị ảnh hưởng bởi các yếu tố ở bên ngoài. Nhóm quyết định đầu vào của mô hình sẽ là cả 2 yếu tố bên trong phòng kín và bên ngoài trời để có thể đánh giá một cách chính xác và khách quan nhất kết quả dự đoán của mô hình. Đầu ra của mô hình sẽ là nhiệt độ, độ ẩm của căn phòng trong 1 phút tiếp theo. Do tính chất của bài toán cần sự nhanh chóng và chính xác nên nhóm sẽ cho dữ liệu chạy qua ba mô hình khác nhau để huấn luyện bao gồm: LSTM, Seq2Seq, TCN. Sau khi huấn luyện nhóm sẽ so sánh hiệu suất về thời gian và độ chính xác của mô hình để chọn ra được mô hình hoạt động tốt nhất. Cuối cùng, nhóm sẽ biểu thị kết qua thông qua việc xây dựng một hiển thị Streaming dự đoán dữ liệu đầu vào trực tuyến để có thể dễ dàng quan sát và sử dụng.

# 2 Công trình liên quan

Real-time prediction model for indoor temperature in a commercial building [1] của Gary và các cộng sự đã giới thiệu một mô hình dự đoán nhiệt độ bên trong trung tâm thương mại theo thời gian thực sử dụng mạng neural nonlinear autoregressive network (NAR) kết hợp với hệ thống nhận diện dựa trên các yếu tố ngoại sinh. Bài báo cũng đã trình bày phương pháp dự đoán cho các khu vực gần kề nhau ở trong tòa nhà. Prediction of Indoor Air Temperature Using Weather Data and Simple Building Descriptors [2] của Jose và các công sự trình bày một phương pháp phân loại nhiệt độ trong nhà thành thang điểm từ 1 đến 7 ứng với độ nóng tăng dần. Mô hình sử dụng cây phân loại để huấn luyện và dự đoán nhiệt độ.

# 3 Bộ dữ liệu

Theo dự định ban đầu, dữ liêu huấn luyện mô hình sẽ được thu thập trực tiếp từ các thiết bị đo các thông số yếu tố môi trường cung cấp. Tuy nhiên, do hiện chưa có đủ điều kiện và thời gian để thiết lập và cài đặt thiết bị nên nhóm sẽ sử dụng bộ dữ liệu đã có sẵn được thu thập từ các thiết bị kết hợp với các dữ liệu trực tuyến trên web để xây dựng dữ liệu huấn luyện mô hình này.

## 3.1 Dữ liệu các thuộc tính trong nhà

Dữ liệu trong nhà được nhóm sử dụng từ bộ dataset: **Indoor Temperature and Relative Humidity Dataset of Controlled and Uncontrolled Environments** [3]. Dữ liệu bao gồm các số đo nhiệt độ và độ ẩm ở bên trong một viện bảo tàng tại San Jose (Costa Rica). Dữ liệu được thu thập từ 12 thiết bị đo giống nhau được đặt tại các vị trí khác nhau trong bảo tàng và sẽ tự động thu thập sau một khoảng thời gian trung bình từ 2 đến 3 giây. Các điểm dữ liệu trải dài xuyên suốt từ tháng 10 đến hết tháng 11 năm 2019 bao gồm 3 thuộc tính (Table 1).

## 3.2 Dữ liệu thuộc tính ngoài trời

Đối với dữ liệu ngoài trời, nhóm tiến hành thu thập dữ liệu được cung cấp mỗi ngày tại trang web: www.timeanddate.com. Trang web cung cấp dữ liệu thời tiết mỗi giờ một lần. Để có thể tương thích với dữ liệu trong nhà về vị trí địa lý và thời gian,

| Name | Unit | Description |
|---|---|---|
| Timestamp | Milisecond | The time when data is collected |
| Humidity | Percent | the humidity inside the museum |
| Temperature | Celsius | The temperature inside the museum |

**Table 1**: Inhouse data description

nhóm tập trung thu thập dữ liêu tử tháng 10 đến hết tháng 11 năm 2019 tại San Jose Costa Rica. Dữ liệu ngoài trời sẽ bao gồm các thuộc tính (Table 2). Nhóm chọn sử dụng Selenium và BeautifulSoup4 làm công cụ để thu thập, tổ chức và lưu trữ dữ liệu. Cuối cùng nhóm sẽ xuất dữ liệu thành 1 file json với cú pháp yyyymmdd.json và lưu trữ vào thiết bị đang sử dụng (Fig. 1).

| Name | Unit | Description |
|---|---|---|
| time | hh:mm:ss | The time when the data is collected |
| Temp | Degree celsius | The temperature at a point of time |
| Weather | | The description of the weather at a point of time |
| Wind | km/h | The speed of the wind at a point of time |
| Humidity | percent | The humidity at a point of time |
| Barometer | mbar | The barometer at a point of time |
| Visibility | km | The longest distance a person can see at a point of time |

**Table 2**: Outdoor data description

### 3.3 Dữ liệu hoàn chỉnh

Sau khi đã hoàn tất việc thu thập dữ liệu ngoài trời và trong nhà. Hai bộ dữ liệu sẽ được kết hợp với nhau dựa trên thuộc tính thời gian đo đạt nhiệt độ ngoài trời(Fig.2). Để đảm bảo dữ liệu ổn định về thời gian, nhóm thực hiện lấy các dự đoán theo từng phút sau đó tính trung bình các giá trị của mỗi phút và tạo ra được bộ dữ liệu cuối cùng.

## 4 Tổng quan kiến trúc

Tổng quan về hệ thống dự báo nhiệt độ trong nhà theo thời gian thực có hai thành phần chủ yếu: một mô hình dự đoán được huấn luyện dựa trên dữ liệu đã thu thập trước đó (ngoại tuyến), một hệ thống dự đoán nhiệt độ trong nhà trong thời gian thực (trực tuyến), hệ thống được thể hiện như Fig. 3.

### 4.1 Apache Spark

Apache Spark là một framework mã nguồn mở được phát triển vào năm 2009 bởi AMPLab để giải quyết các vấn đề của các hệ thống xử lý dữ liệu truyền thống. Việc tính toán trên nhiều máy khác nhau và sử dụng bộ nhớ trong làm cho Spark có tốc độ xử lý nhanh. Spark cung cấp khả năng xử lý dữ liệu trực tuyến theo thời gian thực thông qua Spark Streaming.

```
- Weather data/
  ├── Inhouse_data/
  │   ├── November
  │   └── October
  ├── Outdoor_data/
  │   ├── November/
  │   │   ├── 20191101.json
  │   │   └── 20191102.json
  │   └── October/
  │       ├── 20191001.json
  │       └── 20191001.json
  └── Combined/
      └── final_data.csv
```

**Fig. 1**: Cấu trúc cây thư mục lưu trữ dữ liệu

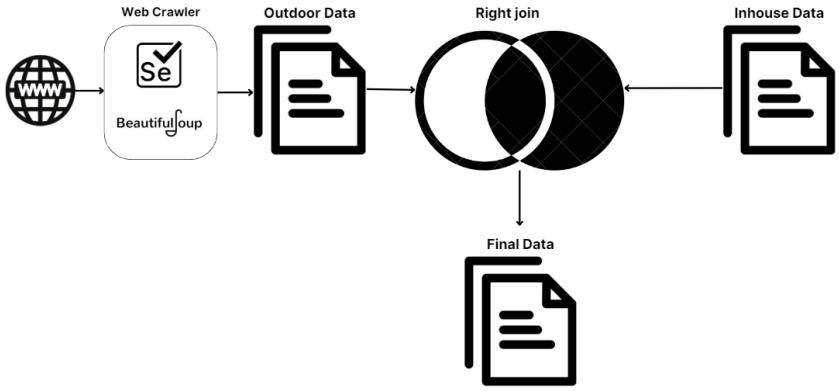


**Fig. 2**: Quy trình tạo dữ liệu

Apache Spark bao gồm hai thành phần chính là Spark Core và Bộ thư viện. Spark Core là thành phần thực thi và quản lý công việc, cung cấp trải nghiệm người dùng liền mạch. Người dùng gửi công việc tới Spark Core thông qua API của Spark Core bằng các ngôn ngữ lập trình như Scala, Python, Java và R. Bộ thư viện Spark cung cấp các công cụ cao cấp như Spark SQL, MLlib, GraphX và Structured Streaming.

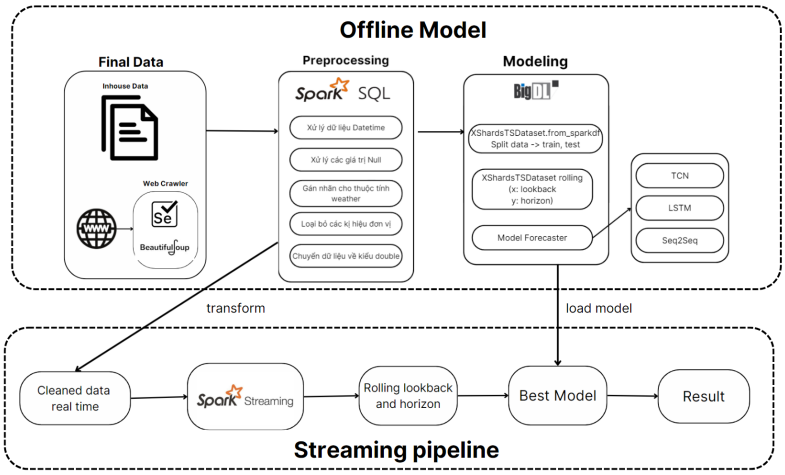


**Fig. 3**: Tổng quan hệ thống dự đoán nhiệt độ trong nhà theo thời gian thực

Trong đồ án này, nhóm sử dụng Spark SQL và Structured Streaming để xử lý dữ liệu từ số liệu của các máy đo trong nhà và dữ liệu ngoài trời (mô phỏng) theo thời gian thực và tiến hành roll dữ liệu liên quan (lookback, horizon) để xây dựng các mô hình dự đoán trong BigDL. Nhóm sử dụng Pyspark, một giao diện cho Apache Spark trong ngôn ngữ Python, để triển khai hệ thống dự đoán nhiệt độ trong nhà theo thời gian thực.

## 4.2 Thành phần ngoại tuyến

### 4.2.1 Bộ dữ liệu huấn luyện

Bộ dữ liệu hoàn chỉnh được trình bày ở mục 3, dữ liệu sẽ được chia ra thành tập train và test theo tỉ lệ 9:1 nối liền nhau. Nghĩa là test sẽ là 1 phần cuối trong 10 phần của dữ liệu, 9 phần đầu sẽ là tập train dùng để huấn luyện mô hình dự đoán.

### 4.2.2 Tiền xử lý dữ liệu

- Xử lý dữ liệu Datetime: từ dữ liệu ban đầu, timestamp là dấu thời gian (một chuỗi kí tự số), ta có thể chuyển đổi timestamp lại để có được thời gian lấy các dòng dữ liệu từ các máy đo thông qua hàm from_unixtime và to_timestamp từ functions của PySparkSQL. Và tần suất lấy thông tin từ các máy đo là không ổn định (khoảng 3 5s sẽ có 1 điểm dữ liệu mới), nên nhóm sẽ gom các điểm dữ liệu chung mốc phút lại với nhau bằng cách chia lấy trung bình cho các điểm dữ liệu có chung mốc phút.
- Xử lý các giá trị Null, các giá trị ngoài lề
  - Các giá trị Null: đến từ cột Visibility, Humidity, Barometer, đôi khi các thuộc tính này sẽ không được xác định nên sẽ có giá trị là NaN, nhóm đã chuẩn chỉnh lại và đưa các dữ liệu này thành giá trị -1.

  - Các giá trị ngoài lề: đến từ cột wind_speed, đôi khi không có gió, nó sẽ được định nghĩa là chuỗi "No wind" thay vì là số, nên nhóm đã đưa dữ liệu này thành giá trị 0 trong dữ liệu.

- Gán nhãn cho thuộc tính weather: nhận thấy thuộc tính weather cũng góp phần ảnh hưởng tới nhiệt độ trong nhà, mà giá trị của cột weather lại là dạng chuỗi, nhóm đã tiến hành gán nhãn lại cho các chuỗi này dựa trên quan điểm của nhóm để phân loại thời tiết với các nhãn sau:
  - Nhãn 0: thời tiết bình thường
  - Nhãn 1: trời nắng
  - Nhãn 2: trời mưa
  - Nhãn 3: trời mưa nhẹ
  - Nhãn 4: trời sắp mưa
- Loại bỏ các kí hiệu đơn vị: đơn giản là các giá trị từ các cột chứa cả số và chữ hoặc kí tự đặc biệt, nhóm tiến hành bỏ các kí hiệu đơn vị đó và lấy các số đo, ví dụ, temp 21°C -> 21 hay barometer 1015 mbar -> 1015, vì các thuộc tính đều dùng kí hiệu đơn vị chung nên loại bỏ đơn vị sẽ không gây ra sự chênh lệch dữ liệu.
- Chuyển dữ liệu về kiểu double

### 4.2.3 Chuẩn bị dữ liệu cho mô hình dự đoán (roll lookback và horizon)

Ở bước chuẩn bị dữ liệu cho mô hình, nhóm sử dụng XShardsTSDataset để roll lookback và horizon, XShardsTSDataset cung cấp nhiều hoạt động xử lý dữ liệu (ví dụ: điền giá trị thiếu, loại bỏ trùng lặp, lấy mẫu lại, tỷ lệ/chỉnh tỷ lệ, lăn) và các phương pháp kỹ thuật đặc trưng (ví dụ: đặc trưng ngày tháng, đặc trưng tổng hợp). XShardsTSDataset có thể được khởi tạo từ xshards của spark dataframe và chuyển đổi sang xshards của numpy theo cách phân tán và song song.

Nhóm đã dùng XShardsTSDataset để chia dữ liệu sang tập train và test theo tỉ lệ 9:1, sau đó, XShardsTSDataset cũng có thể dùng để roll lookback và horizon tự động để đưa vào làm input của mô hình, nhóm đã dùng XShardsTSDataset roll lookback=5 và horizon=1. Nghĩa là dùng 5 dòng dữ liệu trong quá khứ để đoán 1 dòng mới trong tương lai (trong lúc huấn luyện mô hình thì nó sẽ lấy 5 dòng dữ liệu quá khứ và học các đặc trưng và đánh trọng số để đưa ra một dòng dữ liệu mới càng gần với giá trị của horizon càng tốt). Fig. 4 thể hiện cách roll lookback, horizon thành các sample x, y từ mô hình XShardsTSDataset. Cũng như thể hiện cách đặt các target của bài nghiên cứu này, nhóm sẽ đặt toàn bộ các thuộc tính là target nhằm mục đích biến bài toán thành multivariate output, để có thể phát triển thành dự đoán dữ liệu liên tục dựa trên các dữ liệu đã được dự đoán.

### 4.2.4 Mô hình dự đoán (Forecaster from BigDL[4])

- LSTMForecaster: Long short-term memory (LSTM) là một loại đặc biệt của mạng neural recurrent (RNN). Ở mô hình LSTMForecaster của BigDL, họ triển khai

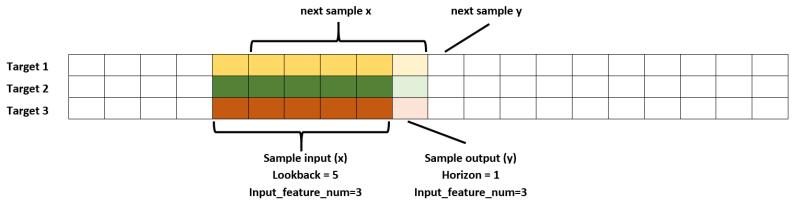


**Fig. 4**: Cách roll lookback, horizon của XShardsTSDataset

phiên bản cơ bản của LSTM - VanillaLSTM cho bộ dự đoán này cho tác vụ dự báo chuỗi thời gian. Nó bao gồm hai lớp LSTM, hai lớp dropout và một lớp dense.

- TCNForecaster: Temporal Convolutional Networks (TCN) là một mạng neural sử dụng kiến trúc convolutional thay vì mạng recurrent. Nó hỗ trợ các trường hợp đa bước và đa biến. Các Convolution Causal cho phép tính toán song song quy mô lớn, giúp TCN có thời gian suy luận ít hơn so với mô hình dựa trên RNN như LSTM.
- Seq2SeqForecaster: Seq2SeqForecaster là một bộ dự đoán (forecaster) dựa trên mô hình chuỗi thành chuỗi (sequence to sequence) sử dụng LSTM, và phù hợp cho việc dự báo chuỗi thời gian đa biến và đa bước.

## 4.3 Thành phần trực tuyến

Thành phần trực tuyến là một pipeline dự đoán nhiệt độ trong nhà trong thời gian thực. Hai công việc chính trong phần này là thu thập dữ liệu trực tuyến và dự đoán nhiệt độ trong nhà theo thời gian thực. Trong bài báo cáo này, nhóm sử dụng dữ liệu đã được chuẩn bị từ trước thay cho dữ liệu trực tuyến như là một dữ liệu mô phỏng.

### 4.3.1 Dữ liệu mô phỏng cho dữ liệu trực tuyến

Theo mục 4.2.3 thì ta có thể thấy dữ liệu đầu vào của mô hình là các numpy array, với x là lookback và y là horizon, thì trong dữ liệu mô phỏng này, dữ liệu được train trong mô hình là học từ 5 lookback và đoán ra 1 horizon. Với mô phỏng dữ liệu, ta cần ít nhất 5 dòng dữ liệu từ file để mô hình có thể cho ra kết quả cuối cùng.

### 4.3.2 Dự đoán nhiệt độ trong nhà theo thời gian thực

Spark Streaming xử lý các dữ liệu được thu thập theo thời gian thực (tiền xử lý dữ liệu mô phỏng); roll lookback và horizon để làm đầu vào cho mô hình tốt nhất đã được huấn luyện ở phần ngoại tuyến, sau đó mô hình này sẽ dự đoán nhiệt độ trong nhà dựa trên dữ liệu mô phỏng.

# 5 Đánh giá mô hình

## 5.1 Phương pháp đánh giá

### 5.1.1 MSE

Mean Squared Error (MSE) - sai số toàn phương trung bình, là một phương pháp được sử dụng để đo độ chính xác của một mô hình dự báo. MSE tính toán sự khác biệt trung bình bình phương giữa giá trị dự báo và giá trị thực tế:

$$\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

Giá trị MSE luôn không âm, nhưng giá trị này càng tiến về 0 thì phản ánh được mô hình chạy càng chính xác. Kỳ vọng của nhóm là MSE càng thấp càng tốt để tiếp tục phát triển bài toán từ univariate output sang multivariate output.

### 5.1.2 Thời gian thực thi

Đơn giản là so sánh thời gian thực thi của các mô hình qua nhiều lần chạy thử, mô hình có thời gian thực thi càng thấp phản ánh mô hình chạy ít tốn tài nguyên và chi phí tính toán, điều này sẽ giúp tiết kiệm thời gian khi nâng cấp dữ liệu và giảm số trường hợp mô hình bị quá tải so với những mô hình có thời gian thực thi lâu hơn.

## 5.2 Kết quả thực nghiệm

Nhóm đã thực hiện train các mô hình với lookback=5, horizon=1 và train trên 3 epochs, kết quả thu được cho thấy 3 model đều trả ra MSE có giá trị gần như là giống nhau, nên nhóm quyết định chọn mô hình có thời gian thực thi nhanh nhất là LSTMForecaster, kết quả được thể hiện ở Table 3 và các hình thể hiện như bên dưới:

- LSTM: Fig. 5
- TCN: Fig. 6
- Seq2Seq: Fig. 7

| Forecaster Model | MSE | Time |
|---|---|---|
| TCN | 0.20735644 | 213.591s |
| LSTM | 0.20776777 | 102.732s |
| Seq2Seq | 0.20515943 | 152.614s |

**Table 3**: Kết quả thử nghiệm

# 6 Kết luận và hướng phát triển

Trong bài nghiên cứu này, nhóm đã áp dụng các mô hình học sâu để dự đoán nhiệt độ trong nhà từ nguồn dữ liệu thu thập từ máy đo trong nhà và dữ liệu ngoài trời (mô phỏng từ web), với các mô hình được áp dụng đều cho ra độ đo MSE đạt mức ổn (khoảng 0.2) và mô hình LSTMForecaster là mô hình có thời gian thực thi nhanh nhất.

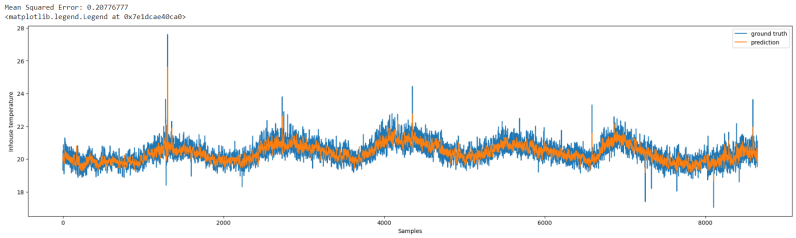


**Fig. 5**: LSTMForecaster

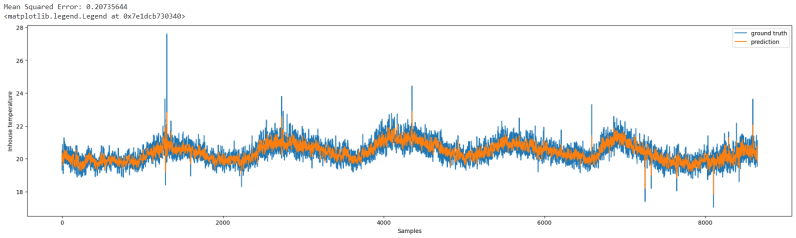


**Fig. 6**: TCNForecaster

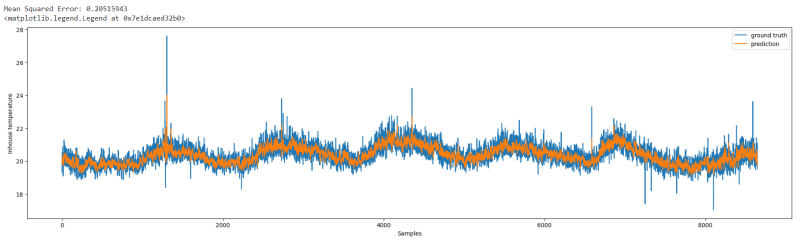


**Fig. 7**: TCNForecaster

Nhóm cũng đã thử nghiệm với bộ dữ liệu hiện tại, với việc đưa bài toán từ univariate output thành multivariate output với giá trị MSE của mô hình LSTM với các thuộc tính như Table 4.

Có thể phát triển bài toán thành dự đoán tương lai liên tục, dựa vào Spark Streaming, ta có thể lấy output của bài toán để đem thành một dòng dữ liệu mới để tiếp tục cho hệ thống dự đoán thời gian thực đưa ra kết quả tiếp theo.

Qua Fig. 8 ta cũng có thể thấy được hạn chế của dữ liệu, ban đầu mô hình sẽ dự đoán đúng được xu hướng tăng/giảm của nhiệt độ, nhưng về sau mô hình đoán lại chưa được như mong đợi, để khắc phục điều này, chúng ta cần phải nâng cấp thêm dữ liệu, khắc phục được hạn chế nguồn dữ liệu mô phỏng, từ những việc trên có thể làm giảm đi MSE và mô hình sẽ dự đoán chính xác hơn ở các điểm dữ liệu về sau. Giá trị MSE cao phản ánh mô hình dự đoán không chính xác, nếu đem kết quả đó làm đầu vào để dự đoán dữ liệu mới sẽ càng không chính xác, việc khắc phục được hạn chế của dữ liệu, giảm MSE sẽ cải thiện đáng kể hướng phát triển này.

| inhouse_temp | inhouse_humidity | temp | weather | wind_speed | humidity | barometer | visibility |
|---|---|---|---|---|---|---|---|
| 0.20776778 | 1.0711561 | 0.018307418 | 0.010216303 | 0.5746452 | 0.5208447 | 0.01093557 | 0.9211925 |

**Table 4**: MSE của các thuộc tính từ mô hình LSTM

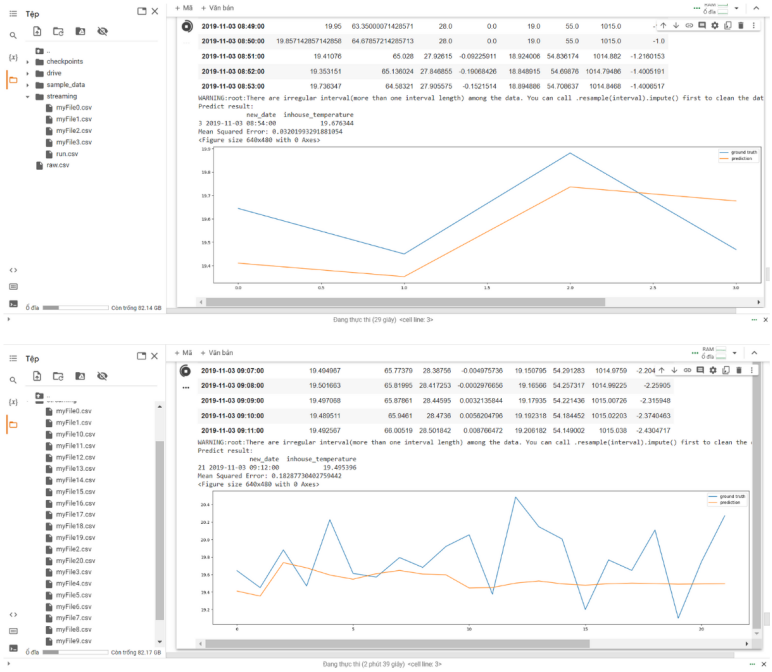

**Fig. 8**: Sử dụng Streaming Spark để liên tục nhận dữ liệu từ các dòng đã được dữ đoán

# References


[1] Afroz, Z., Urmee, T., Shafiullah, G., Higgins, G.: Real-time prediction model for indoor temperature in a commercial building. Applied Energy **231**, 29–53 (2018) https://doi.org/10.1016/j.apenergy.2018.09.052

[2] Aguilera, J.J., Andersen, R., Toftum, J.: Prediction of indoor air temperature using weather data and simple building descriptors. International journal of environmental research and public health **16**, 4349 (2019) https://doi.org/10.3390/ijerph16224349

[3] Botero-Valencia, L.M.-V.D. Juan; Castano-Londono: Indoor temperature and relative humidity dataset. Mendeley Data, V2 (2022) https://doi.org/10.17632/dxyvxk6h96.2

[4] Dai, J.J., Wang, Y., Qiu, X., Ding, D., Zhang, Y., Wang, Y., Jia, X., Zhang, L.C.,

Wan, Y., Li, Z., Wang, J., Huang, S., Wu, Z., Wang, Y., Yang, Y., She, B., Shi, D., Lu, Q., Huang, K., Song, G.: Bigdl: A distributed deep learning framework for big data. In: Proceedings of the ACM Symposium on Cloud Computing. SoCC'19, pp. 50–60. Association for Computing Machinery, ??? (2019). https://doi.org/10.1145/3357223.3362707 . https://arxiv.org/pdf/1804.05839.pdf