ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Các vấn đề hiện đại trong KHMT (INT3011E 21)

BÁO CÁO ƯỚC TÍNH LƯỢNG MƯA SỬ DỤNG DỮ LIỆU ĐA NGUỒN

NHÓM 28

Nguyễn Đình Chính - 20021307

HÀ NỘI - 2023

Contents

DATASET 1

1. Giới thiệu

-Một số lý do sử dụng thuật toán học máy để áp dụng cho việc mô hình hóa và dự đoán hiện tượng mưa:

+Khả năng xử lý lượng lớn dữ liệu

+Khả năng khám phá hành vi các mẫu hoặc mối quan hệ không rõ ràng giữa các dữ liệu được xử lý mà không thể nhìn thấy trực tiếp

+Khả năng sử dụng mô hình đại diện cho các hiện tượng mưa, từ đó đưa ra dự đoán về dữ liệu mới thu được từ nó.

-Mục tiêu bài báo cáo là thực hiện dự đoán lượng mưa ở Thanh Hóa sử dụng 1 số mô hình học máy như MLR (Multivariate Linear Regression), Random Forest và Long short-term memory (LSTM)

2. Khám phá dữ liệu

2.1 Bộ dữ liệu

- Bài báo cáo sử dụng tập dữ liệu được cung cấp sẵn dưới dạng file xlxs

- Bộ dữ liệu này bao gồm một mẫu có 180.518 dữ liệu với thông tin về 25 biến nghiên cứu. Dữ liệu mô tả thông tin khí tượng trên toàn tỉnh Thanh Hóa được thu thập hàng ngày trong 2 tháng 9, 10. Đặc biệt, biến value là biến mục tiêu dự đoán lượng mưa.

Tên Feature	Mô tả	Dữ liệu bị thiếu	Loại
B04B, B05B, B06B, B09B, B10B, B11B, B12B, B14B, B16B, I2B, I4B, IRB, VSB, WVB	Giá trị của các band phổ vệ tinh Himawari8 thu được tại vị trí ứng với tọa độ trạm đo mưa	8752 cho mỗi feature(riêng VSB, WVB là 9364)	float
CAPE, TCC, TCW, TCWV	Giá trị các sản phẩm phân tích ERA5 tại vị trí tương ứng với tọa độ trạm đo mưa	3565 cho mỗi feature	float

Bảng mô tả dataset.

-Thống kê về các biến cũng được xem xét (trừ biến IMERG):

	value	B04B	B05B	B06B	B09B	B10B	B11B	B12B	B14B
count	180518.000000	171766.000000	171766.000000	171766.000000	171766.000000	171766.000000	171766.000000	171766.000000	171766.000000
mean	0.387904	0.133540	0.077556	0.050540	243.502103	250.592091	274.064079	256.134847	274.628023
std	2.397454	0.187717	0.109586	0.077957	14.899867	18.556099	24.904239	16.909618	25.495011
min	0.000000	0.000000	0.000000	0.000000	102.969421	0.000000	0.000000	108.106400	102.687599
25%	0.000000	0.000000	0.000000	0.000000	242.323166	249.953152	269.419724	253.358822	267.970413
50%	0.000000	0.002344	0.001302	0.001133	248.047241	256.707825	284.037796	262.912018	285.626617
75%	0.000000	0.250186	0.148750	0.079585	251.273190	259.683624	288.887268	265.999298	290.813080
max	87.000000	0.924143	0.581414	0.397904	259.936646	267.162018	300.364532	273.108063	303.339874

B16B	I2B	I4B	IRB	VSB	WVB	CAPE	TCC	TCW	TCWV
171766.000000	171766.000000	171766.000000	171766.000000	171154.000000	171154.000000	176953.000000	176953.000000	176953.000000	176953.000000
260.324232	271.936995	283.350739	275.968026	0.088642	234.835579	513.735719	0.598203	45.496486	45.196144
20.991708	24.779519	21.914715	24.972113	0.152033	12.631280	790.716142	0.365685	10.696034	10.338895
102.958733	101.696297	108.100006	103.016319	0.000000	103.262726	0.000000	0.000000	12.393962	12.391813
256.899040	266.061417	278.533875	270.148064	0.000000	233.463367	18.625000	0.240662	37.741257	37.649521
269.646912	282.561310	290.480652	286.580933	0.001953	238.242348	169.187500	0.658529	45.551437	45.355362
272.564087	287.231873	295.951935	291.622902	0.106726	241.200607	669.687500	0.985042	53.084641	52.795776
279.133881	298.516541	309.721527	303.998779	0.912455	249.843307	8496.551758	1.000008	88.392822	73.954330

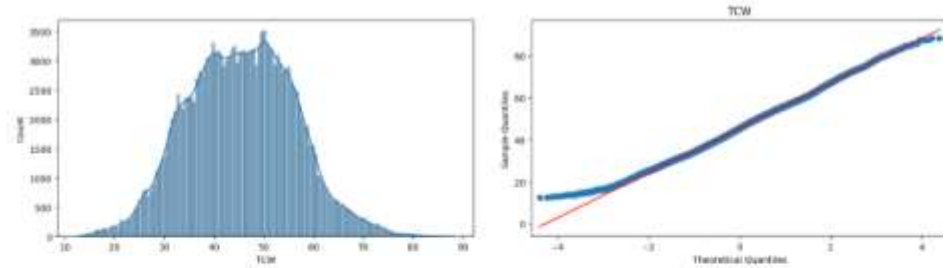
+Có thể thấy ở cột “value” các phân vị 25%, 50%, 75% đều mang giá trị 0 và có giá trị “count” nhiều hơn toàn bộ các cột còn lại, chứng tỏ có khoảng 9000 dữ liệu bị khuyết ở đây

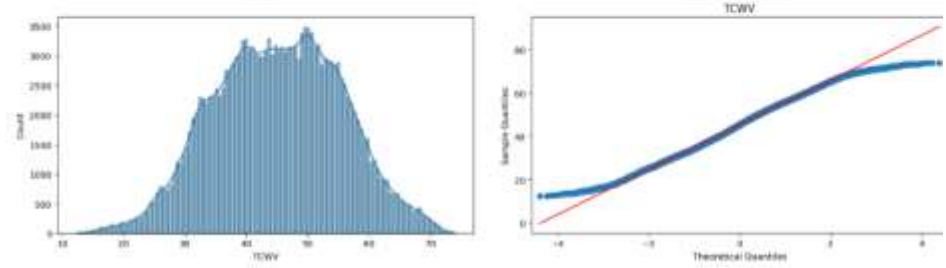
+Cột “CAPE” có độ lệch chuẩn cao cho thấy dữ liệu được trải rộng hơn trong khi cột “B06B” có độ lệch chuẩn thấp nghĩa là dữ liệu dao động xung quanh giá trị trung bình

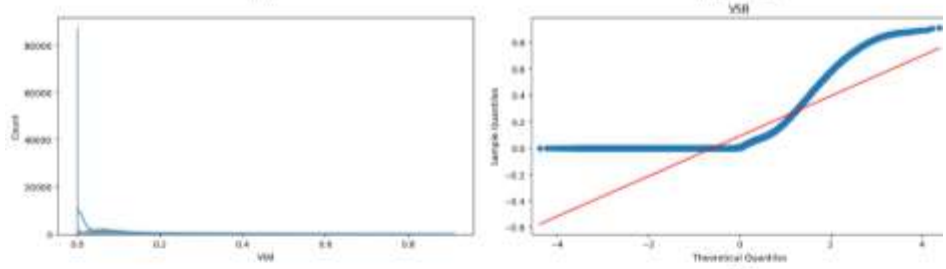
2.2 Minh họa dữ liệu

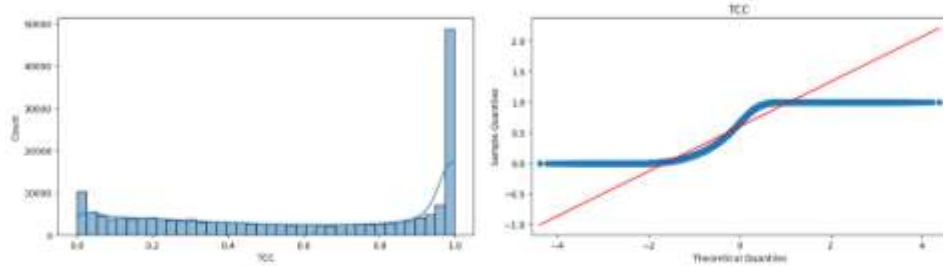
2.2.1 Đối với các feature

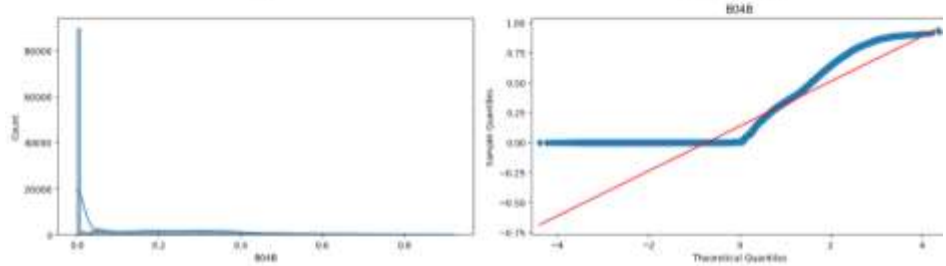
-Để có cái nhìn tốt hơn về các phân phối của từng trường dữ liệu, mỗi cột dữ liệu feature sẽ được minh họa bằng biểu đồ histogram và biểu đồ QQ-plot để quan sát những giá trị không nằm trên đường thẳng kì vọng của phân phối chuẩn:

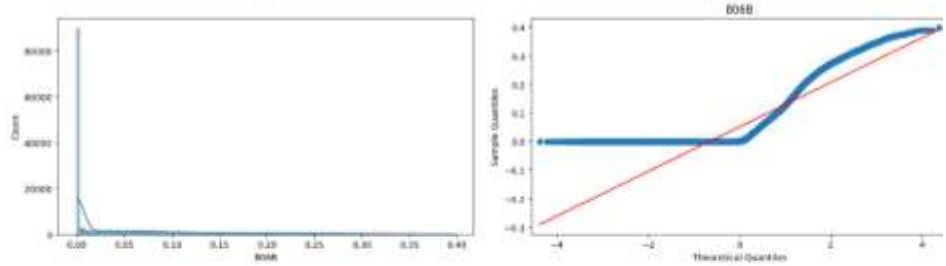

TCW
Count
TCW
Sample Quantiles
Theoretical Quantiles

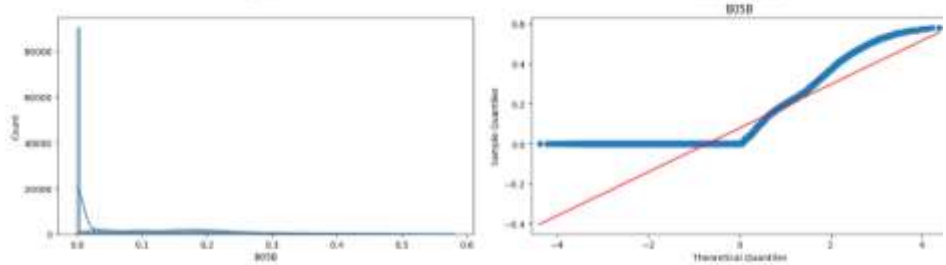

TCWV
Count
TCWV
Sample Quantiles
Theoretical Quantiles


VSI
Count
VSI
Sample Quantiles
Theoretical Quantiles

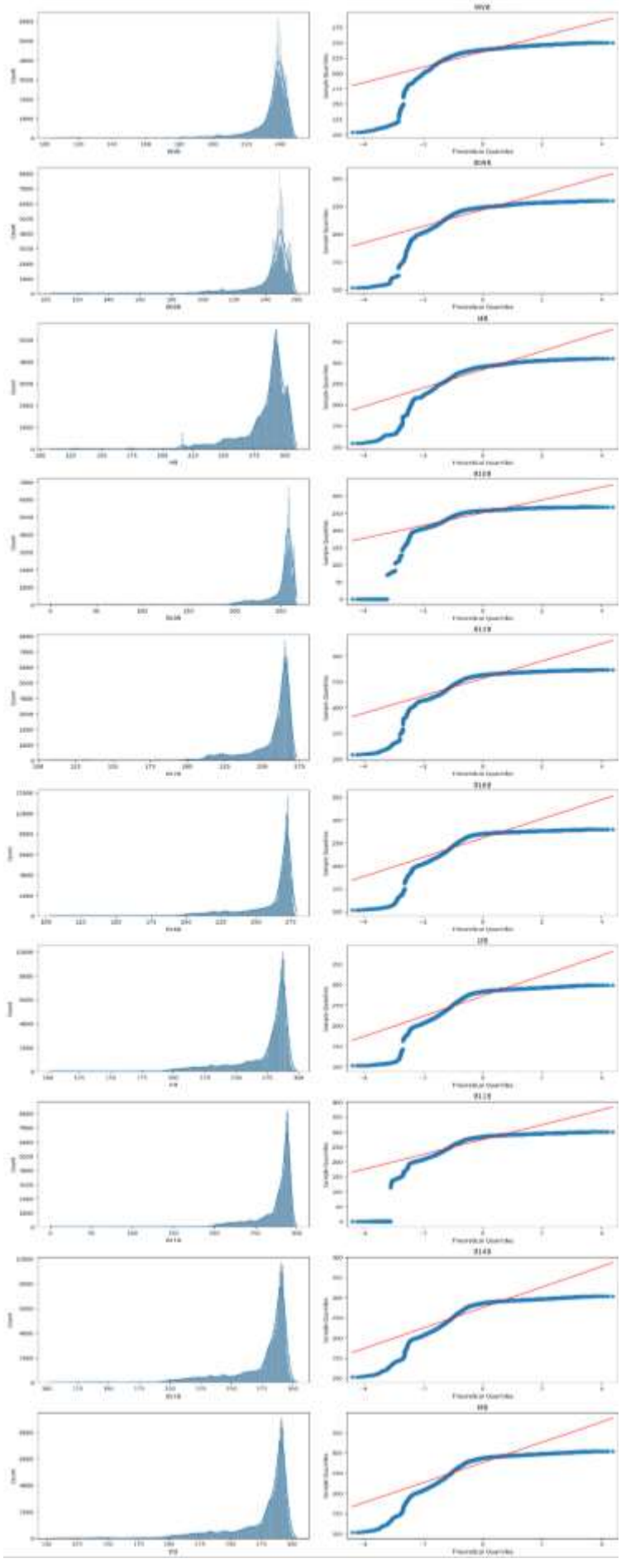

TCC
Count
TCC
Sample Quantiles
Theoretical Quantiles


B04B
Count
B04B
Sample Quantiles
Theoretical Quantiles


B06B
Count
B06B
Sample Quantiles
Theoretical Quantiles


B07B
Count
B07B
Sample Quantiles
Theoretical Quantiles


CAPE

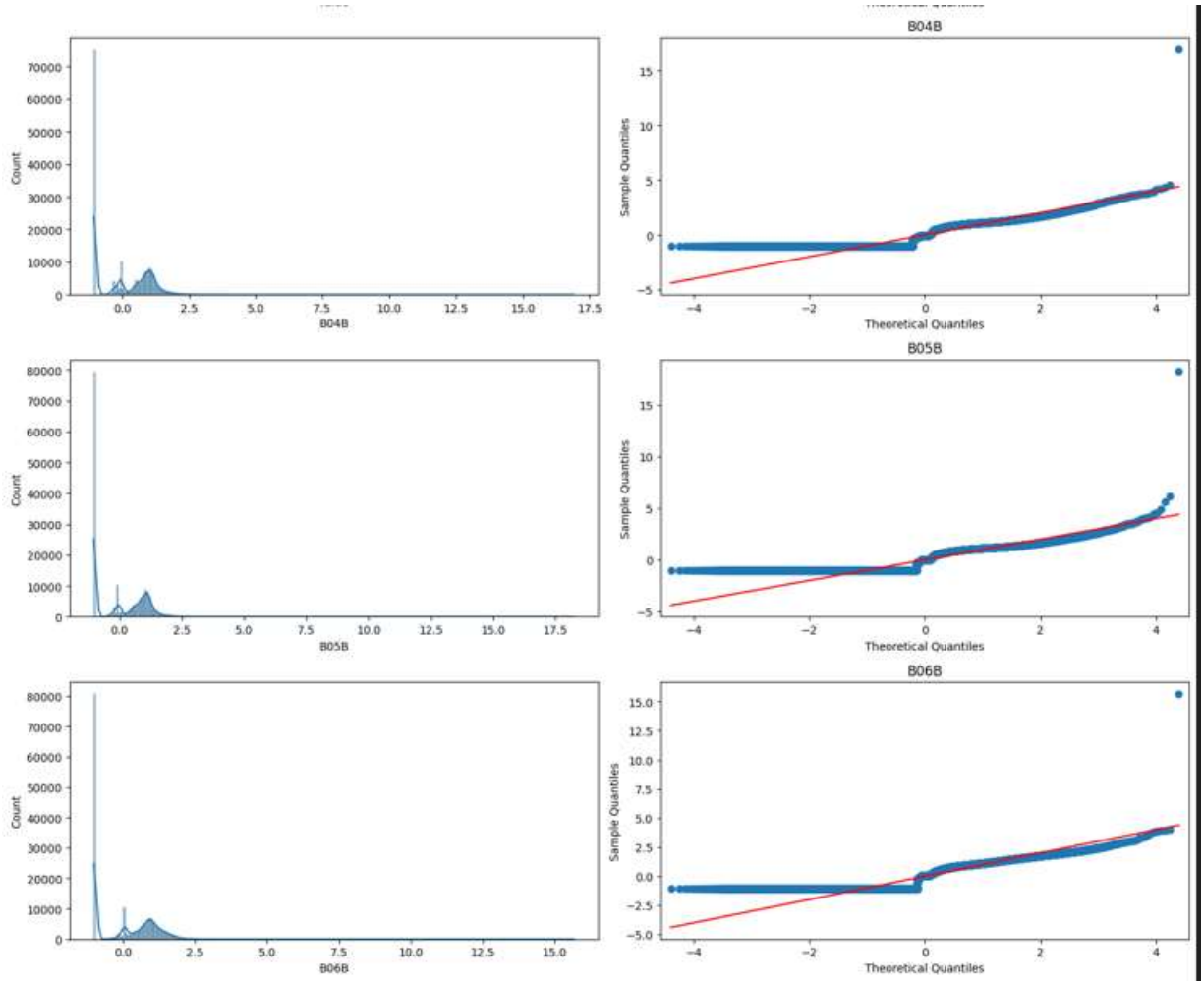


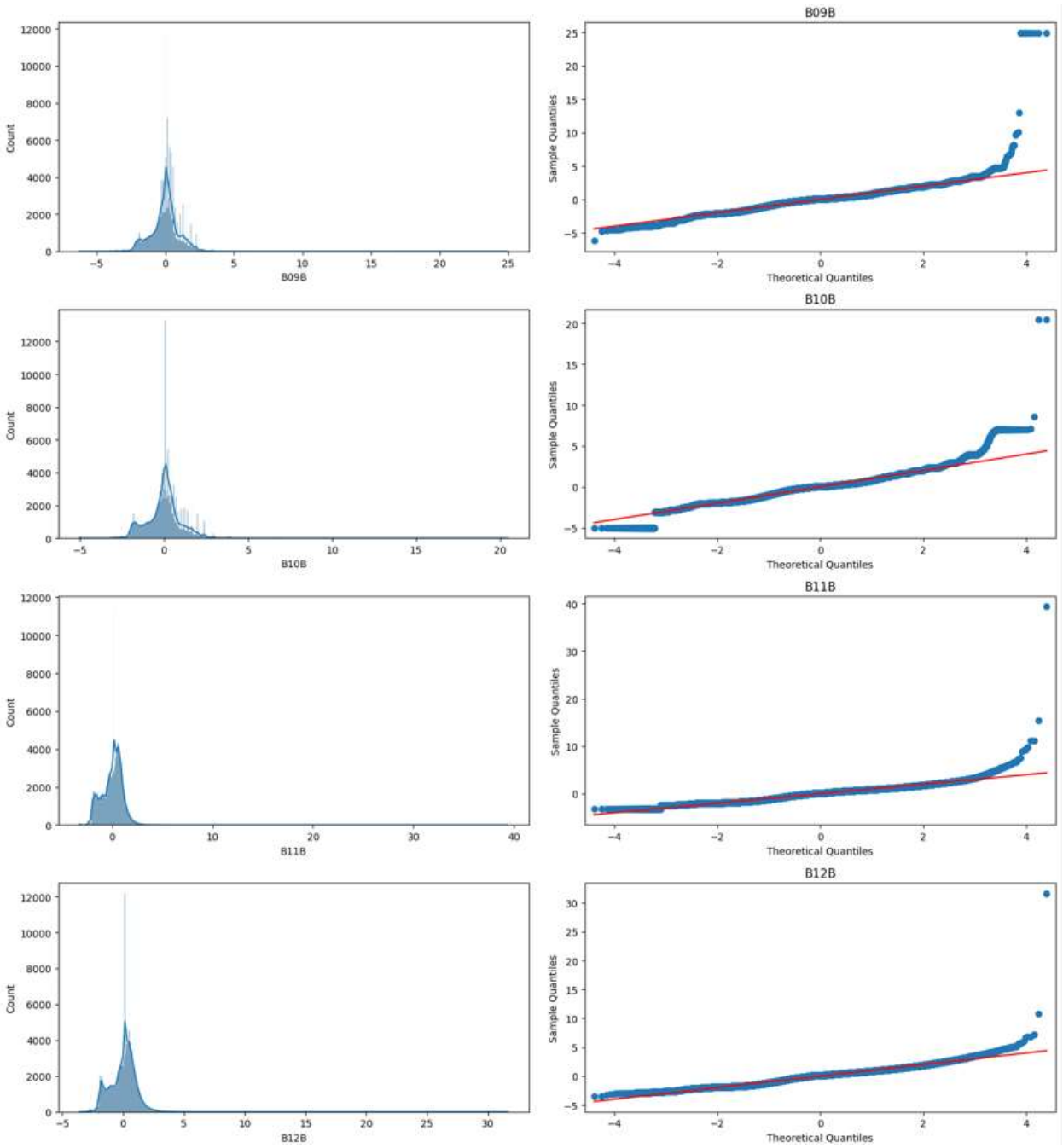
+Có thể thấy các feature đầu vào không tuân theo phân phối chuẩn trừ TCW, TCWV. Các biểu đồ dường như sai lệch rất nhiều so với mong muốn để phân phối

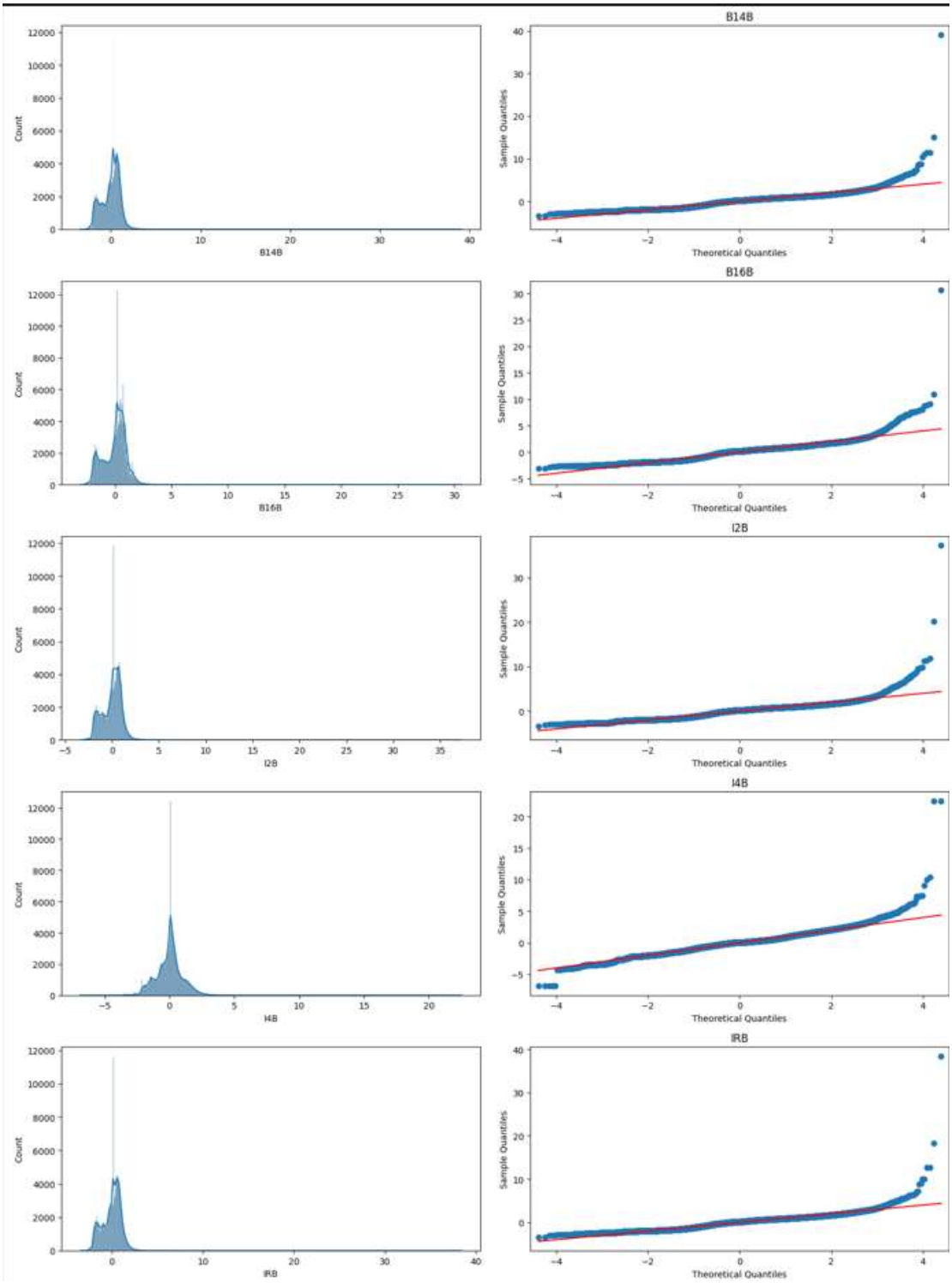
+ Các nhóm biến có phân phối tương tự nhau là:

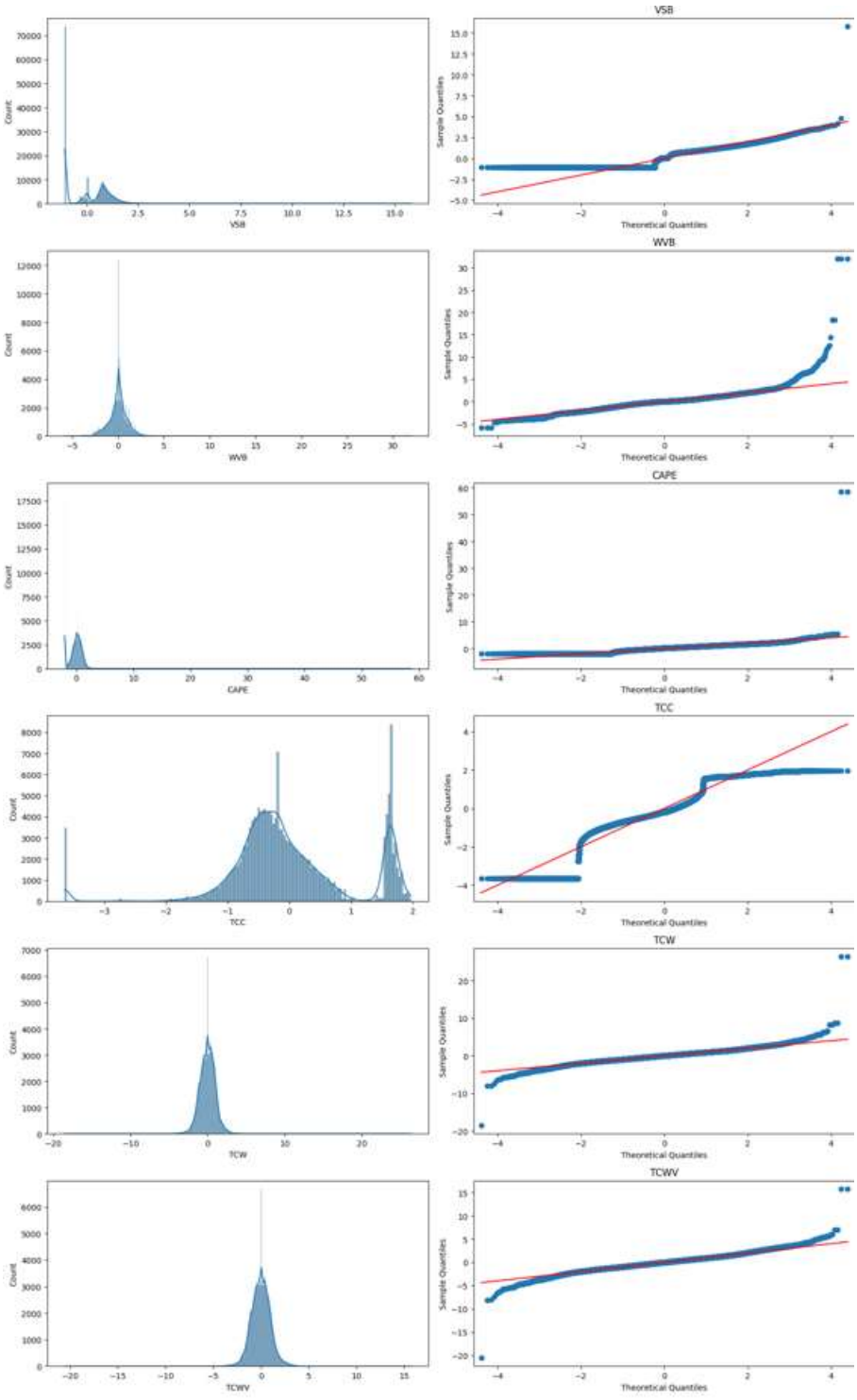
- VSB, TCC, B04B, B06B, B05B
- Các biến còn lại

-Vì vậy, để giúp mô hình học máy đưa ra dự đoán tốt hơn, cần đưa dữ liệu trở nên bình thường. Do vậy, chuyển đổi phân phối trên thành phân phối thường.

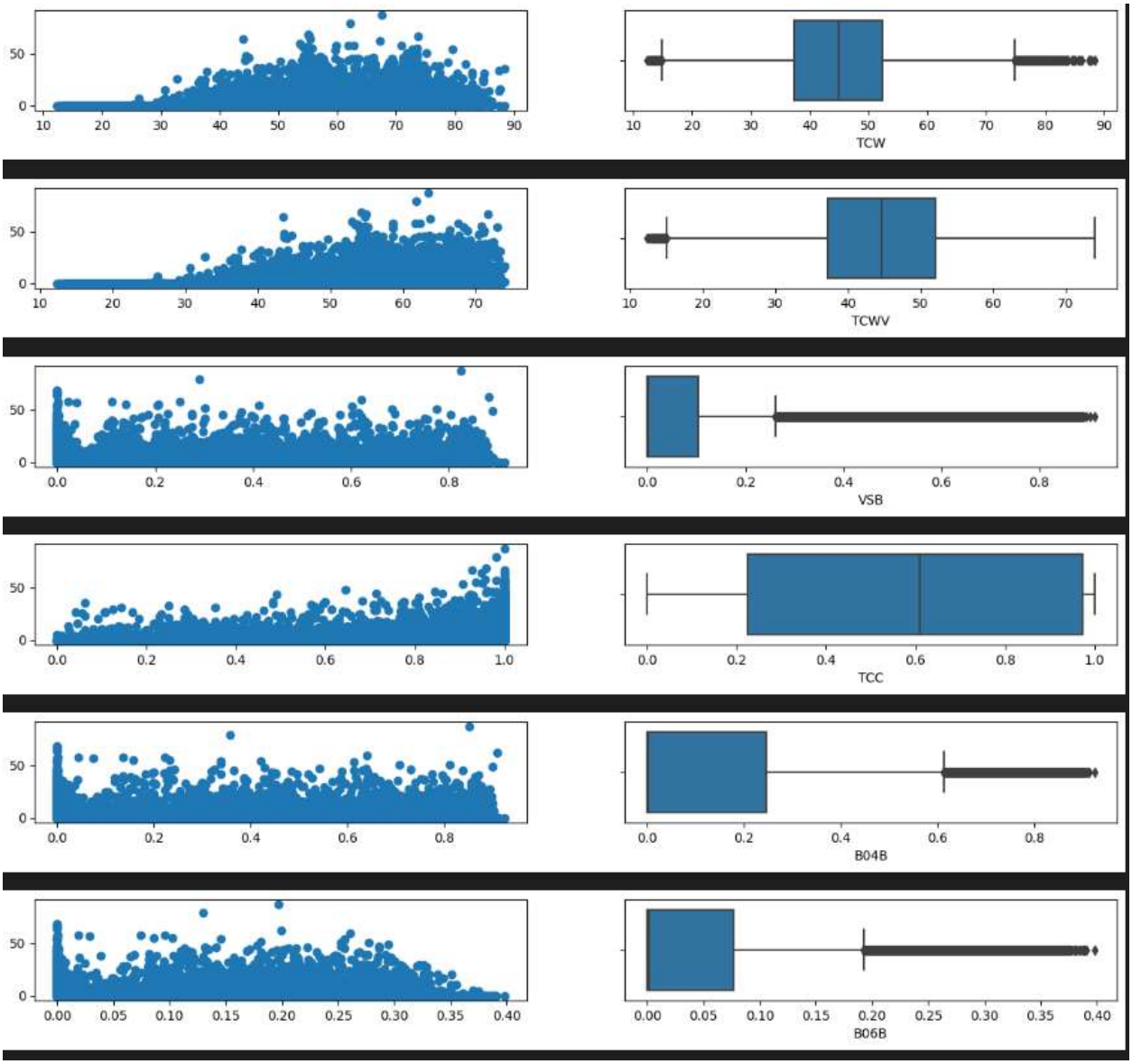


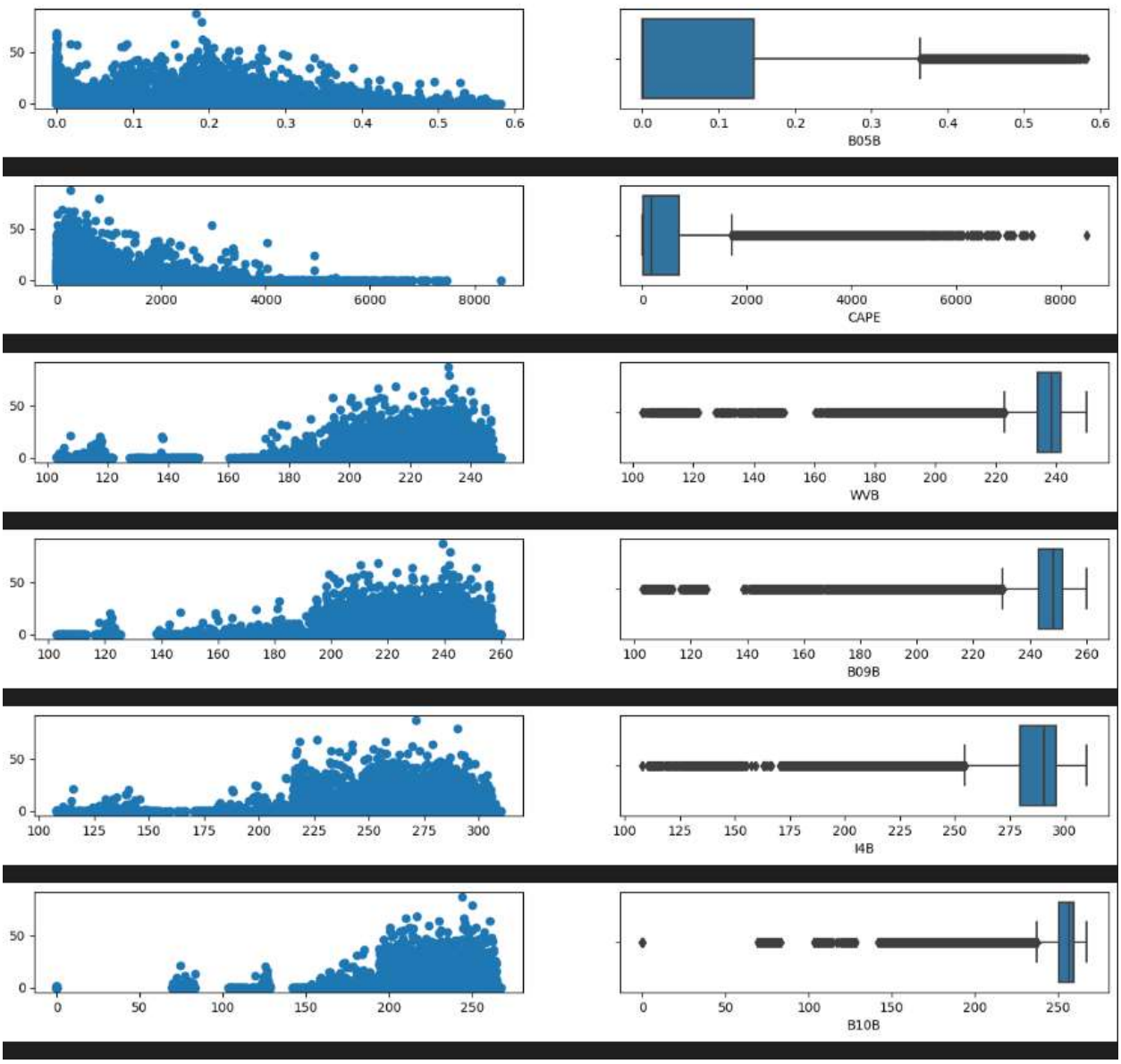

B09B
Count
B09B
Sample Quantiles
Theoretical Quantiles
B10B
Count
B10B
Sample Quantiles
Theoretical Quantiles
B11B
Count
B11B
Sample Quantiles
Theoretical Quantiles
B12B
Count
B12B
Sample Quantiles
Theoretical Quantiles

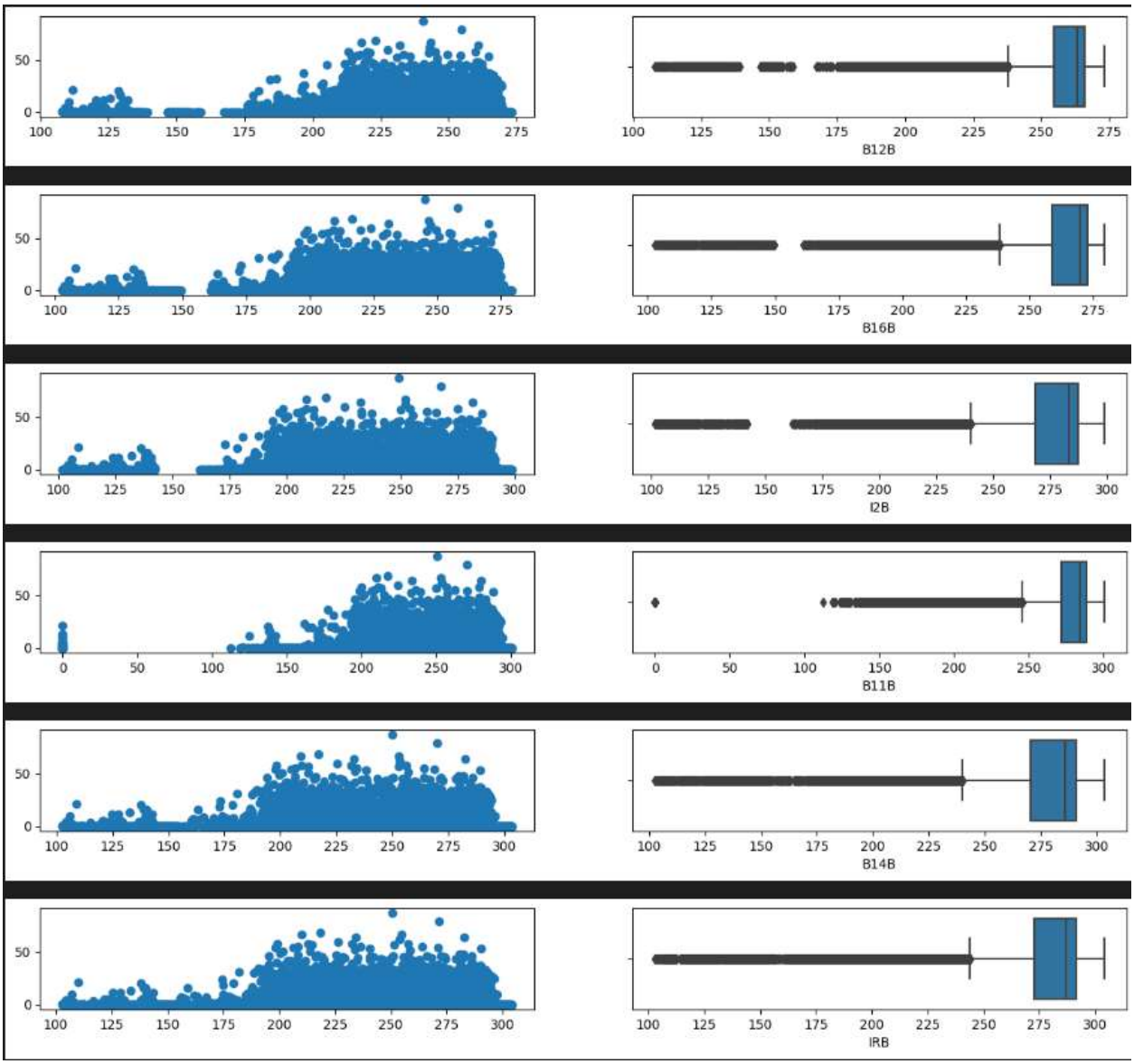

B14B
Count
B14B
Sample Quantiles
Theoretical Quantiles
B16B
Count
B16B
Sample Quantiles
Theoretical Quantiles
I2B
Count
I2B
Sample Quantiles
Theoretical Quantiles
I4B
Count
I4B
Sample Quantiles
Theoretical Quantiles
IRB
Count
IRB
Sample Quantiles
Theoretical Quantiles



-Tiếp đến, ta xem xét phân bố của các điểm dữ liệu thông qua biểu đồ boxplot để có những biện pháp phù hợp để xử lý dữ liệu phù hợp.


TCW
TCWV
VSB
TCC
B04B
B06B


B05B
CAPE
WVB
B09B
I4B
B10B



+TCWV, TCC là những biến có ít những điểm ngoại lệ, trong khi đó các biến còn lại xuất hiện rất nhiều điểm ngoại lệ, điều này sẽ ảnh hưởng đến hiệu suất của mô hình.

+Nhìn chung, nếu xóa bỏ các ngoại lệ có thể bỏ đi 1 số lượng lớn dữ liệu, hoặc chúng ta vô tình có thể xóa bỏ những điểm dữ liệu quan trọng sẽ ảnh hưởng đến khả năng dự đoán của mô hình sau này.

+Vì vậy, ta cần cách biến đổi dữ liệu ngoại lệ này về những giá trị hợp lí hơn.

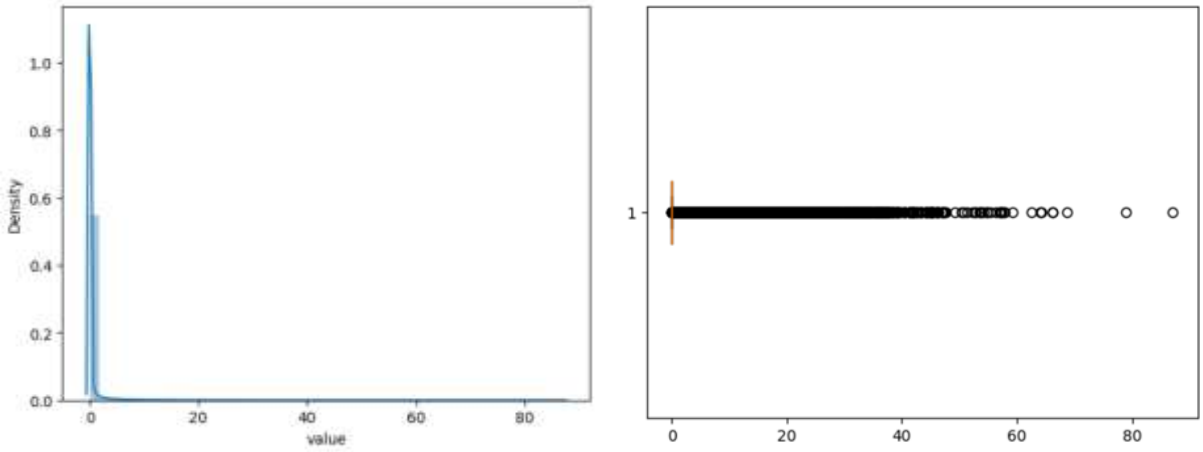
2.2.2 Đối với biến target

-Đầu tiên, ta xem qua số liệu thống kê của biến value trong bộ dữ liệu:

count	180518.000000
mean	0.387904
std	2.397454
min	0.000000
25%	0.000000
50%	0.000000
75%	0.000000
max	87.000000

+Có thể thấy các ngày không mưa (tương ứng với giá trị bằng 0) chiếm phần lớn trong bộ dữ liệu và ngày có mưa có giá trị cao nhất là 87

-Tiếp theo, ta xem xét sự phân phối và phân bố của biến này:



+Tương tự, value không tuân theo phân hối chuẩn, xuất hiện nhiều điểm ngoại lệ lệch về phía bên phải của biểu đồ.

+Ngoài ra, cũng có thể thấy giá trị trung bình, giá trị thấp nhất, các phân vị đều thu hẹp lại thành 1 đường thẳng tại giá trị 0.

3. Tiền xử lý và phương pháp

3.1 Chuẩn bị dữ liệu

1. Xử lý ngoại lệ: Đầu tiên điều chỉnh tỉ lệ các feature để chúng tuân theo phân phối chuẩn (như phần minh họa dữ liệu), áp dụng quy tắc 3σ cho phân phối chuẩn

- Trong phân phối chuẩn, giả sử μ là kỳ vọng và σ là độ lệch chuẩn. Quy tắc 3σ cho phân phối chuẩn nói rằng:

- 68% các điểm dữ liệu nằm trong khoảng μ±σ
- 95% các điểm dữ liệu nằm trong khoảng μ±2σ

- 99.7% các điểm dữ liệu nằm trong khoảng μ±3σ

- Với một điểm dữ liệu x, z score của nó được tính bởi:

$$\frac{x - \mu}{\sigma}$$

- Những điểm có z score nằm ngoài đoạn [−3,3] có thể được coi là các điểm ngoại lệ. Biến đổi toán học một chút, việc này tương đương với việc các điểm nằm ngoài đoạn [μ−3σ, μ+3σ] được coi là các điểm ngoại lệ.

2. Dữ liệu bị thiếu: số lượng mẫu của từng biến rỗng vẫn được phân tích, để không loại bỏ 1 lượng lớn dữ liệu. Kết quả phân tích các biến rỗng:

- Những mẫu bị rỗng ở 1 số biến do mất dữ liệu sẽ được thay thế bằng giá trị trung bình hàng tháng của biến đó ở các cột tương ứng tương ứng.

id	1329
value	0
datetime	0
B04B	8752
B05B	8752
B06B	8752
B09B	8752
B10B	8752
B11B	8752
B12B	8752
B14B	8752
B16B	8752
I2B	8752
I4B	8752
IRB	8752
VSB	9364
WVB	9364
CAPE	3565
TCC	3565
TCW	3565
TCWV	3565
IMERG	0

Số lượng biến có giá trị "NA"

3.2 Phương pháp

3.2.1 Thuật toán MLR (multivariate linear regression)

-Tương tự với mô hình hồi quy tuyến tính đơn giản nhưng với nhiều biến độc lập tham gia vào biến phụ thuộc và do đó có nhiều hệ số để xác định và tính toán phức tạp hơn do các biến được thêm vào

-Phương trình hồi quy tuyến tính đa biến:

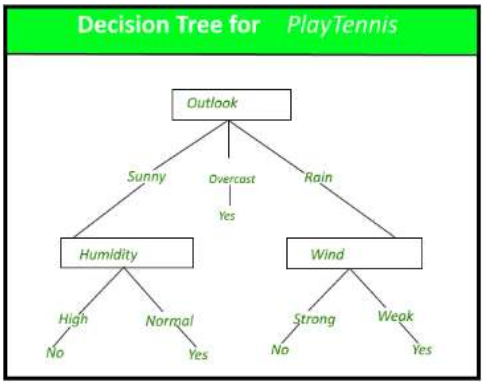
$$Y_i = \alpha + \beta_1 x_i^{(1)} + \beta_2 x_i^{(2)} + \ldots + \beta_n x_i^{(n)}$$

Trong đó Y_i là ước tính thứ i của các biến phụ thuộc Y, có n biến độc lập và x_i^j biểu thị thành phần thứ i của biến độc lập/feature

3.2.2 Rừng ngẫu nhiên.

* Cây quyết định:

-Là thuật toán học máy có giám sát, phổ biến để phân loại và dự đoán. Thuật toán sử dụng cấu trúc cây để mô hình hóa quan hệ giữa các biến. Cây quyết định bắt đầu từ 1 gốc và chia thành nhiều nhánh. Trong đó, mỗi nút bên trong biểu thị 1 phép thử trên 1 thuộc tính, 1 nhánh biểu thị 1 kết quả của phép thử và mỗi nút lá giữ 1 lớp nhãn.



-Một cây có thể được "học" bằng cách chia tập hợp nguồn thành các tập con dựa trên việc kiểm tra giá trị thuộc tính. Qúa tình này được lặp lại trên mỗi tập con dẫn xuất theo cách đệ quy được gọi là phân vùng đệ quy. Đệ quy được hoàn thành khi tất cả các tập con tại 1 nút có cùng giá trị của biến mục tiêu hoặc khi việc tách không còn thêm giá trị cho dự đoán.

- Một khó khăn của thuật toán này bao gồm việc xác định việc phân chia cây sẽ được thực hiện từ biến nào, điều cần thiết là dữ liệu phải chứa một lớp duy nhất.

Để xác định ứng cử viên phân chia tốt nhất, hai biện pháp được sử dụng: entropy và chỉ số Gini

**Rừng ngẫu nhiên*

- Mọi cây quyết định đều có phương sai cao, nhưng khi kết hợp song song tất cả chúng lại với nhau thì phương sai tổng hợp sẽ thấp vì mỗi cây quyết định được huấn luyện hoàn hảo trên dữ liệu mẫu cụ thể đó và do đó đầu ra không phụ thuộc vào một cây quyết định mà phụ thuộc vào nhiều cây quyết định. Trong trường hợp vấn đề hồi quy, đầu ra cuối cùng là giá trị trung bình của tất cả các đầu ra.

- Random Forest là một mô hình học tổng hợp có khả năng thực hiện tác vụ hồi quy bằng cách sử dụng nhiều cây quyết định. Ý tưởng cơ bản đằng sau điều này là kết hợp nhiều cây quyết định để xác định đầu ra cuối cùng thay vì dựa vào các cây quyết định riêng lẻ.

3.2.3 LSTM

-Mạng LSTM là 1 phần mở rộng của mạng nơ-ron hồi quy (RNN-Recurrent Neural Network) khắc phục được các tình huống khi RNN bị lỗi.

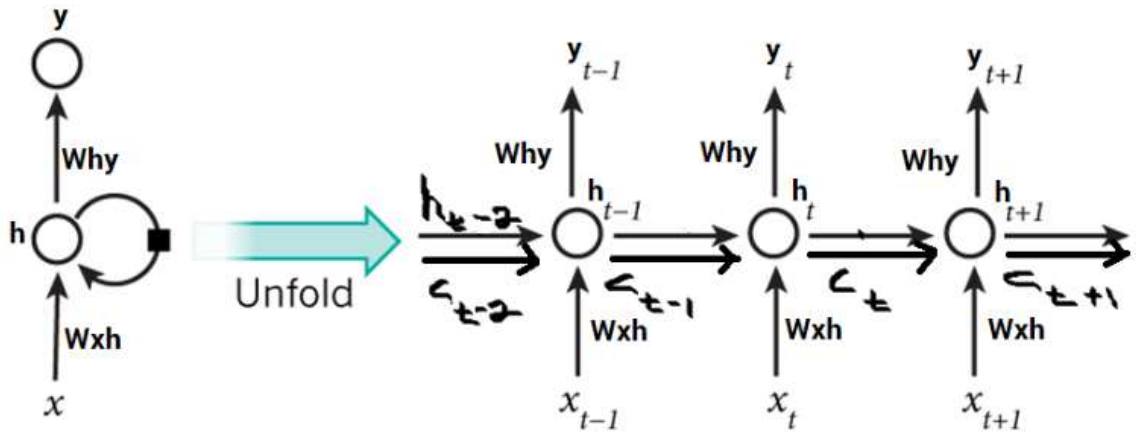
-Nói về RNN, đó là 1 mạng hoạt động trên đầu vào hiện tại bằng cách xem xét đầu ra trước đó (feedback) và lưu trữ trong bộ nhớ của nó trong 1 khoảng thời gian ngắn (short-term memory). Tuy nhiên có những hạn chế đối với RNN, đầu tiên là nó không thể lưu trữ thong tin trong thời gian dài; thứ hai không kiểm soát được thông tin nào tốt và thông tin nào cần được “lãng quên”. Ngoài ra, các vấn đè khác với RNN là bùng nổ và biến mất độ dốc trong quá trình đào tạo mạng thông qua quay lui (backtracking).

-Do đó, LSTM được thiết kế để độ dốc được loại bỏ hoàn toàn, trong khi mô hình được đào tạo không bị thay đổi, có khả năng bỏ đi hoặc thêm các thông tin cần thiết, được điều chỉnh bởi các nhóm được gọi là cổng (gate): Input, Output và Forget. Ngoài ra, độ phức tạp để cập nhật mỗi trọng số giảm xuống O(1) với LSTM

-Kiến trúc LSTM: Sự khác biệt cơ bản giữa kiến trúc của RNN và LSTM là lớp ẩn của LSTM là một đơn vị có cổng hoặc ô có cổng. Nó bao gồm bốn lớp tương tác với nhau và tạo ra đầu ra của ô đó cùng với trạng thái của ô. Hai đầu ra này sau đó được chuyển vào lớp ẩn tiếp theo. Không giống như các RNN chỉ có một lớp

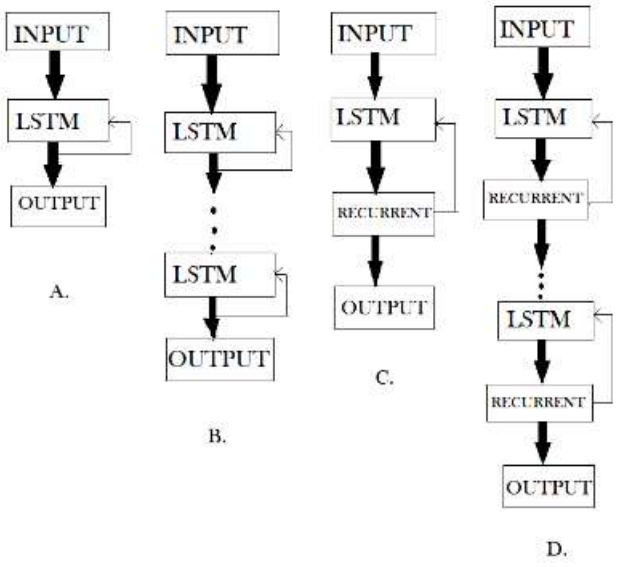
tanh mạng thần kinh duy nhất, các LSTM bao gồm ba cổng sigmoid logistic và một lớp tanh. Các cổng đã được giới thiệu để hạn chế thông tin được truyền qua ô. Chúng xác định phần thông tin nào cần cho ô tiếp theo và phần nào sẽ bị loại bỏ. Đầu ra thường nằm trong phạm vi 0-1 trong đó '0' có nghĩa là 'từ chối tất cả' và '1' có nghĩa là 'bao gồm tất cả'.

-Lớp ẩn của LSTM:



-Mỗi ô LSTM có ba đầu vào h_{t-1}, C_{t-1} và x_t; hai đầu ra h_t và C_t. Trong một thời gian nhất định t, h_t là trạng thái ẩn, C_t là trạng thái ô hoặc bộ nhớ, x_t là điểm dữ liệu hoặc đầu vào hiện tại. Lớp sigmoid đầu tiên có hai đầu vào h_{t-1} và x_t trong đó h_{t-1} là trạng thái ẩn của ô trước đó. Nó được gọi là cổng quên vì đầu ra của nó chọn lượng thông tin của ô trước đó sẽ được đưa vào. Đầu ra là một số trong [0,1] được nhân (theo điểm) với trạng thái ô trước đó C_{t-1}.

-1 số biến thể của LSTM:



+Hình A là mạng LSTM cơ bản: chỉ có 1 lớp LSTM giữa lớp đầu vào và đầu ra.

+Hình B là Deep LSTM bao gồm 1 số lớp giữa đầu vào và đầu ra.

+Hình C là mạng LSTM với 1 lớp hồi quy lặp lại

+Hình D deep LSTM với nhiều lớp hồi quy lặp lại

-Trong báo cáo này, deep LSTM đực sử dụng.

3.3 Các chỉ số đánh giá hiệu quả:

- Lỗi tính khoảng cách của các dự đoán so với đầu ra thực tế. Có hai công thức: Lỗi tuyệt đối trung bình (MAE) và Lỗi bình phương trung bình (MSE):

+MSE (Mean Squared Error): nó tìm thấy sai số bình phương giữa các giá trị được dự đoán và thực tế. MSE là thước đo chất lượng của một công cụ ước tính - nó luôn không âm và các giá trị càng gần 0 càng tốt.

+MAE (Mean Absolute Error): đo độ lớn trung bình của các lỗi trong một tập hợp các dự đoán mà không cần xem xét hướng của chúng. Đó là giá trị trung bình trên mẫu thử nghiệm về sự khác biệt tuyệt đối giữa dự đoán và quan sát thực tế, trong đó tất cả các khác biệt riêng lẻ có trọng số bằng nhau.

$$\mathrm{MAE} = \frac{1}{N}\sum_{k=1}^{N}|y_k - \widehat{y_k}|$$

$$\mathrm{MSE} = \frac{1}{N}\sum_{j=1}^{N}\left(y_k - \widehat{y_k}\right)^2,$$

Trong đó:

+N tương ứng với tổng số lượng mẫu

+y_k tương ứng với giá trị dự đoán

+$\hat{y}_k$ tương ứng giá trị thực tế

- RMSE: là thước đo mức độ phù hợp của đường hồi quy với các điểm dữ liệu. RMSE cũng có thể được hiểu là Độ lệch chuẩn trong phần dư. Được tính bằng căn bậc 2 của MSE

- R-squared là một thước đo thống kê thể hiện mức độ phù hợp của mô hình hồi quy. Giá trị lý tưởng cho r-square là 1. Giá trị của r-square càng gần 1 thì mô hình càng phù hợp.

$$R^2 = 1 - \frac{RSS}{TSS}$$

Trong đó: RSS là tổng bình phương của số dư, TSS là tổng bình phương.

4. Kết quả

4.1 MLR:

-Thuật toán này được triển khai bằng cách sử dụng hàm LinearRegression trong thư viện scikitLearn của Python

-Dữ liệu được chia thành 2 tập: train và test (tỉ lệ 8:2)

-Huấn luyện mô hình trên tập train và đánh giá trên tập test.

- Kết quả đánh giá thu được trên tập test như sau:

```
The test MAE is:  0.65

The test MSE is:  5.05

R2 score: 0.09947884943649576

The test RMSE is:  2.2473024511514117
```

- Kết quả đánh giá thu được trên tập train như sau:

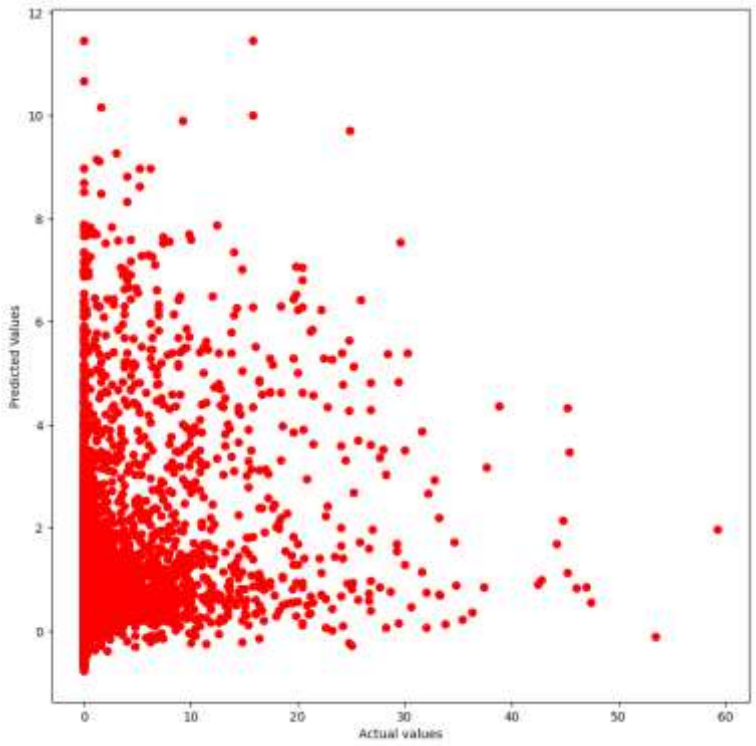
```
The training MAE is:  0.67

The training MSE is:  5.05

R2 score: 0.09647552996491537

The training RMSE is:  2.2473024511514117
```

-Biểu đồ trực quan hóa kết quả của mô hình trên tập test:



+Kết quả dự đoán và thực tế không nằm trên 1 đường thẳng cho thấy độ chính xác của kết quả dự đoán không cao.

-Phương trình tuyến tính của mô hình:

```
value  = 0.39
+ (0.27 * B04B)
+ (-0.37 * B05B)
+ (-0.19 * B06B)
+ (0.09 * B09B)
+ (0.02 * B10B)
+ (-0.01 * B11B)
+ (-0.02 * B12B)
+ (-0.1 * B14B)
+ (-0.02 * B16B)
+ (0.05 * I2B)
+ (0.1 * I4B)
+ (-0.03 * IRB)
+ (0.33 * VSB)
+ (-0.1 * WVB)
+ (-0.16 * CAPE)
+ (0.0 * TCC)
+ (6.75 * TCW)
+ (-6.29 * TCWV)
```

4.2 Rừng ngẫu nhiên

-Thuật toán này được triển khai bằng cách sử dụng hàm RandomForestRegressor trong thư viện scikitLearn của Python với các tham số sau được điều chỉnh

+n_estimators: số cây quyết định cần xem xét

+max_depth: độ sâu tối đa của cây quyết định.

+random_state: Kiểm soát cả tính ngẫu nhiên của quá trình bootstrapping của các mẫu được sử dụng khi xây dựng cây

-Dữ liệu được chia thành 2 tập: train và test (tỉ lệ 8:2)

-Trước hết, với các tham số là mặc định(n_estimators = 100, max_depth = None) và random_state =42.

- Kết quả đánh giá trên tập train:

```
The training MAE is:  0.21

The training MSE is:  0.87

R2 score: 0.8545818990422661

RMSE: 0.9319347646250888
```

- Kết quả đánh giá trên tập test:

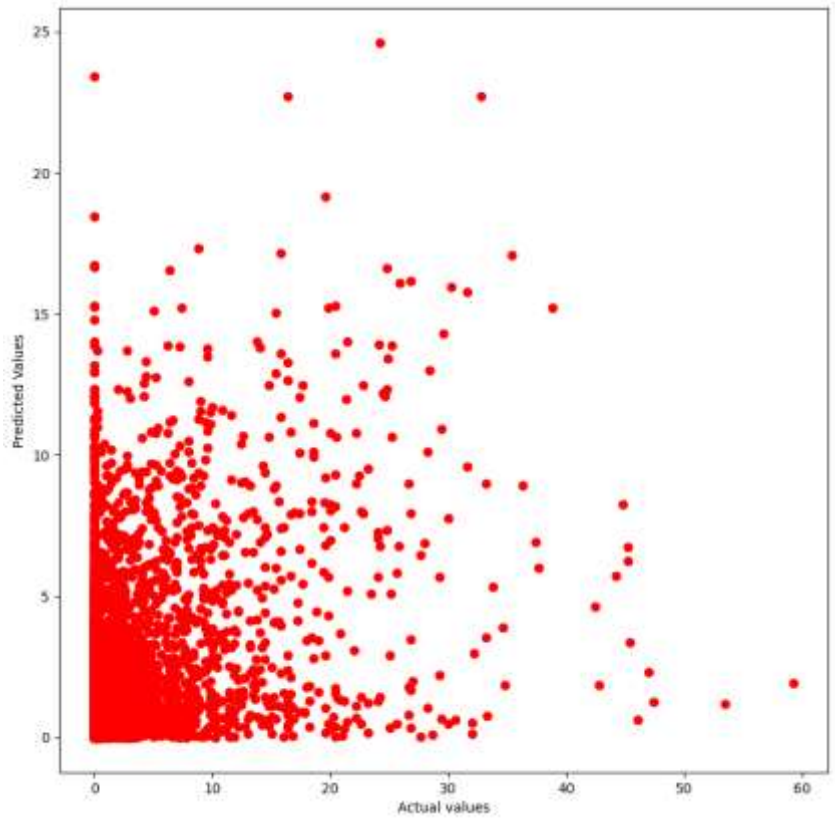
```
The test MAE is:  0.56

The test MSE is:  3.9

R2 score: 0.19534620161694827

The test RMSE is:  1.9751606345066939
```

+ Biểu đồ trực quan hóa kết quả của mô hình trên tập test:



-Điều chỉnh các tham số bằng gridSearch với n_estimators = [20, 30, 40, 50, 100] và max_depth = [8, 12, 16], random_state = 42; từ đó chọn ra tham số tốt nhất để huấn luyện mô hình

- Kết quả đánh giá trên tập train:

```
The training MAE is:  0.38

The training MSE is:  1.97

R2 score: 0.6707087095889184

The training RMSE is:  1.402382231212662
```

- Kết quả đánh giá mô hình trên tập test:

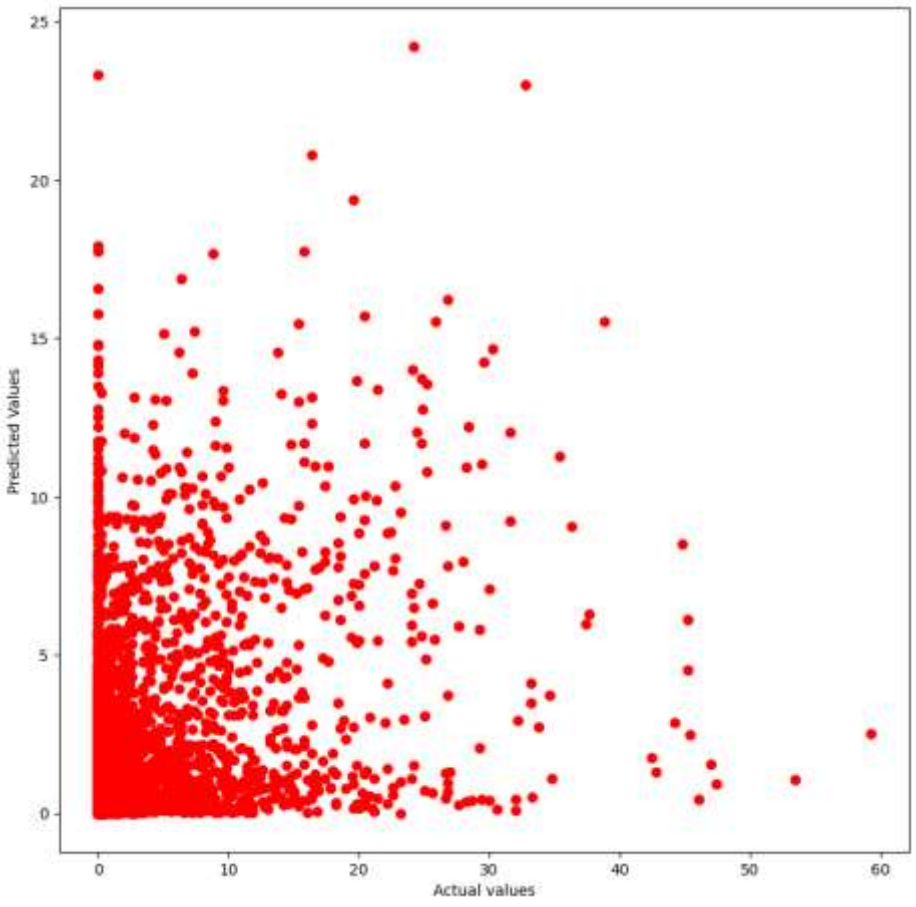
```
The test MAE is:  0.54

The test MSE is:  3.94

R2 score: 0.1883059175068389

The test RMSE is:  1.9837826080716472
```

+ Biểu đồ trực quan hóa kết quả của mô hình trên tập test:



4.3 LSTM

-Mô hình LSTM gồm 2 lớp ẩn nằm giữa lớp đầu vào và đầu ra (tương tự hình B trong phần phương pháp)

-Đầu vào của mô hình: dùng của sổ trượt window_size = 5 trên các giá trị value theo thứ tự lần lượt.

-Đầu ra: dùng giá trị của value tại thời điểm t-1 đến t-5 để dự đoán tại thời điểm t

-Dữ liệu được chia thành 2 tập: train và val (tỉ lệ 8:2)

-Lớp ẩn được triển khai bằng cách sử dụng hàm LSTM trong thư viện keras của Python với các tham số sau được điều chỉnh:

+Units: số chiều không gian đầu ra-50.

+Activation: Hàm activation cho lớp ẩn- 'relu': rectified linear function

+ return_sequences: Trả về đầu ra cuối cùng trong chuỗi đầu ra hay toàn bộ chuỗi- True (ở trong lớp ẩn đầu tiên)

-Hàm mất mát được sử dụng là: MSE, trình tối ưu hóa: Adam với learning_rate = 0.0001, epochs = 10.

-Số lượng tham số trong mạng:

Layer (type)	Output Shape	Param #
lstm_16 (LSTM)	(None, 5, 50)	10400
lstm_17 (LSTM)	(None, 50)	20200
dense_14 (Dense)	(None, 1)	51

Total params: 30,651
Trainable params: 30,651
Non-trainable params: 0

-Chọn mô hình tốt nhất và thực hiện dự đoán trên tập train.

+Kết quả đánh giá:

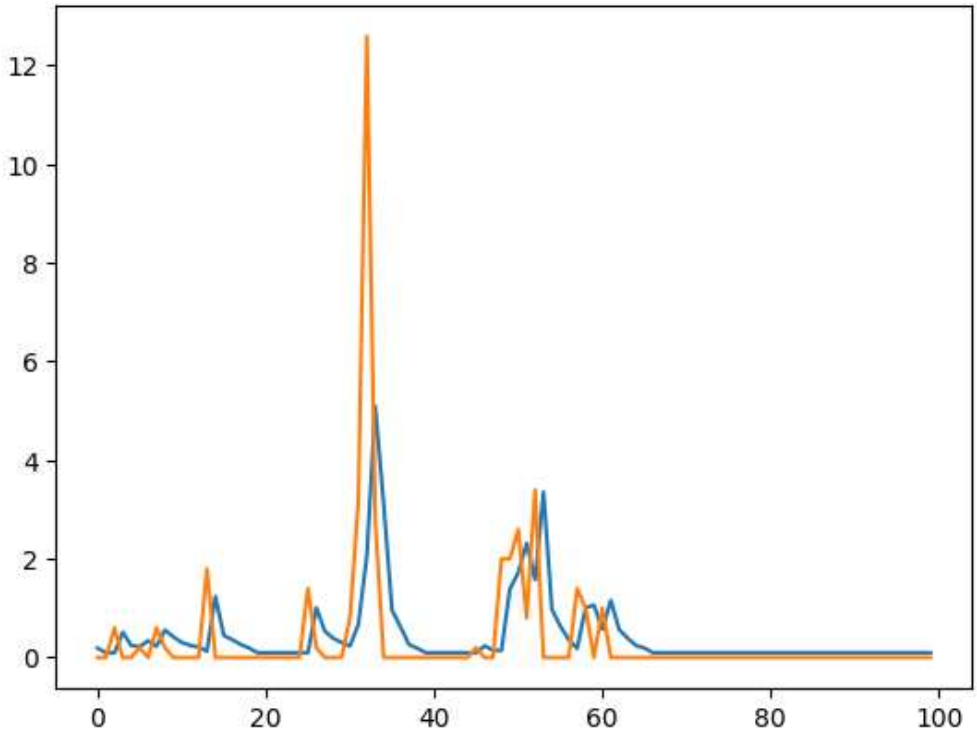
```
The training MAE is:  0.46

The training MSE is:  3.72

R2 score: 0.32471449948695863

The training RMSE is:  1.929954787087298
```

+ Biểu đồ trực quan hóa kết quả của mô hình trên tập train:



-Kết quả của mô hình trên tập val:

+Kết quả đánh giá:

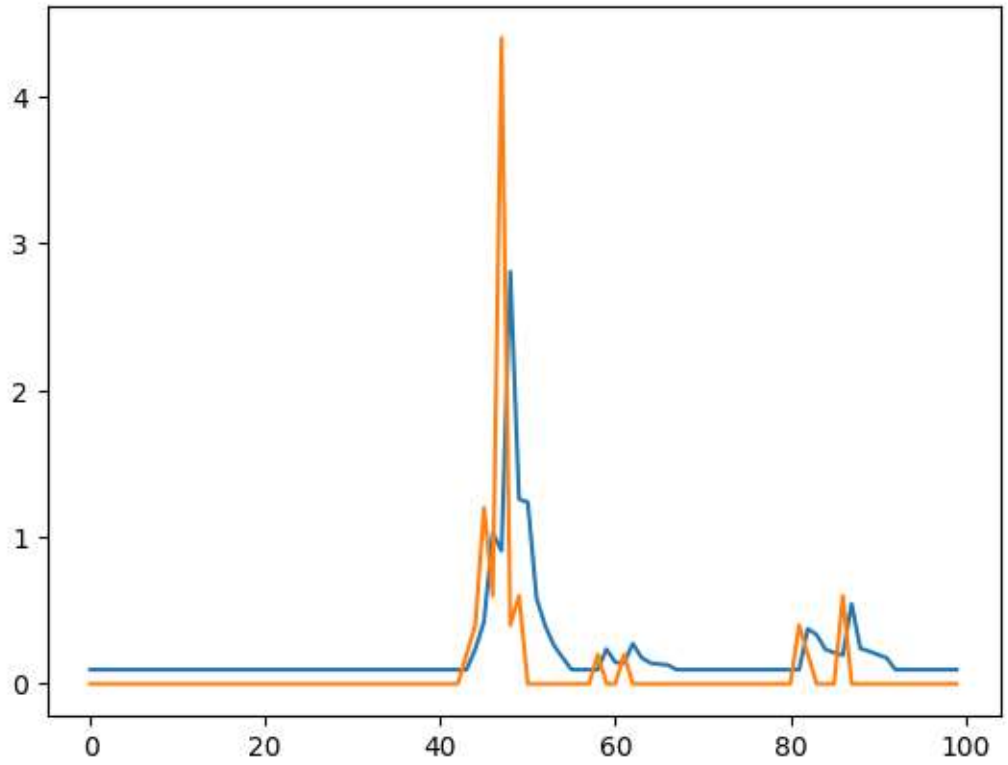
```
The test MAE is:  0.52

The test MSE is:  5.28

R2 score: 0.19318704485687344

The test RMSE is:  2.2987985969581275
```

+ Biểu đồ trực quan hóa kết quả của mô hình trên tập val:



5. Tổng kết

-Các mô hình được phân tích mô để xem xét mô hình nào phù hợp nhất để dự đoán lượng mưa.

-Sau đây là bảng đánh giá mô hình trên dựa trên các chỉ số đánh giá tập train (được làm tròn):

Mô hình	MAE	MSE	R2	RMSE
MLR	0.67	5.05	0.096	2.25
RF	0.21	0.87	0.85	0.93
RF(Grid)	0.38	1.97	0.67	1.4
LSTM	0.46	3.72	0.32	1.93

(chú thích: RF- rừng ngẫu nhiên không dùng grid; RF(Grid)-rừng ngẫu nhiên dùng gridSearch)

+Các chỉ số đánh giá tốt nhất được tô đỏ trên mỗi cột: chỉ số MAE, MSE, RMSE càng nhỏ thì càng tốt; trong khi R2 các giá trị càng gần 1 thì càng tốt/ các chỉ số tệ nhất được tô màu xanh.

+Có thể thấy mô hình rừng ngẫu nhiên (không dùng grid) cho kết quả tốt nhất, trong khi mô hình MLR cho kết quả thiếu chính xác nhất.

+Ngoài ra, việc chọn các tham số trong gridSearch có thể chưa thực sự tối ưu nên chỉ số có thể thấp hơn mặc định ban đầu)

-Tiếp theo, bảng đánh giá mô hình trên dựa trên các chỉ số tập test (được làm tròn):

Mô hình	MAE	MSE	R2	RMSE
MLR	0.65	5.05	0.099	2.25
RF	0.56	3.9	0.195	1.97
RF(Grid)	0.54	3.94	0.188	1.98
LSTM	0.52	5.28	0.193	2.3

(chú thích: RF- rừng ngẫu nhiên không dùng grid; RF(Grid)-rừng ngẫu nhiên dùng gridSearch)

+Các chỉ số đánh giá tốt nhất được tô đỏ trên mỗi cột: chỉ số MAE, MSE, RMSE càng nhỏ thì càng tốt; trong khi R2 các giá trị càng gần 1 thì càng tốt/ các chỉ số tệ nhất được tô màu xanh.

+Đến tập test các chỉ số đánh giá có sự thay đổi; có thể thấy mô hình tốt nhất vẫn là rừng ngẫu nhiên, trong khi đó LSTM bộc lỗ rõ điểm yếu khi chạy trên tập test.

-Tóm lại kết quả này cho thấy mô hình rừng ngẫu nhiên là mô hình tốt nhất. Vì vậy việc sử dụng mô hình này với tham số tối ưu nhất có thể cho kết quả tốt hơn.

-Trong quá trình xây dựng mô hình, do chưa có kinh nghiệm điều chỉnh cac tham số và việc thực hiện các tham số chủ yếu do mò mẫm, vì vậy các chỉ số đánh giá vẫn chưa thực sự là tốt nhất cho mỗi mô hình.