

Thử nghiệm

Thiết lập:

Cho phép tối đa 2 cloud vm chạy 1 lúc (1 monitor, 1 scale tối đa).

Dữ liệu:

Sinh kịch bản số lượng request theo thời gian, chu kỳ 8h, jmeter bắn request theo kịch bản này.

Dữ liệu train có độ dài khoảng 4 ngày.



Mô hình:

Mỗi điểm dữ liệu lấy trung bình trong 4 phút, cứ 4 phút lấy được điểm mới rồi dự báo điểm của 4 phút sau và kiểm tra điều kiện scale.

window size = 4 điểm.

update mô hình lại sau 120 điểm, = 8h.

Điều kiện scale:

- Scale thêm: dự đoán 4 phút sau > 0.55 thì scale

- Scale bớt:

1 máy cần ít hơn 0.55, 2 máy cần ít hơn khoảng 0.27, tức là khi đã scale và đang có 2 máy, nếu monitor được điểm $< 0.27 * k$ thì scale bớt (k đang để 0.8).

2 điều kiện:

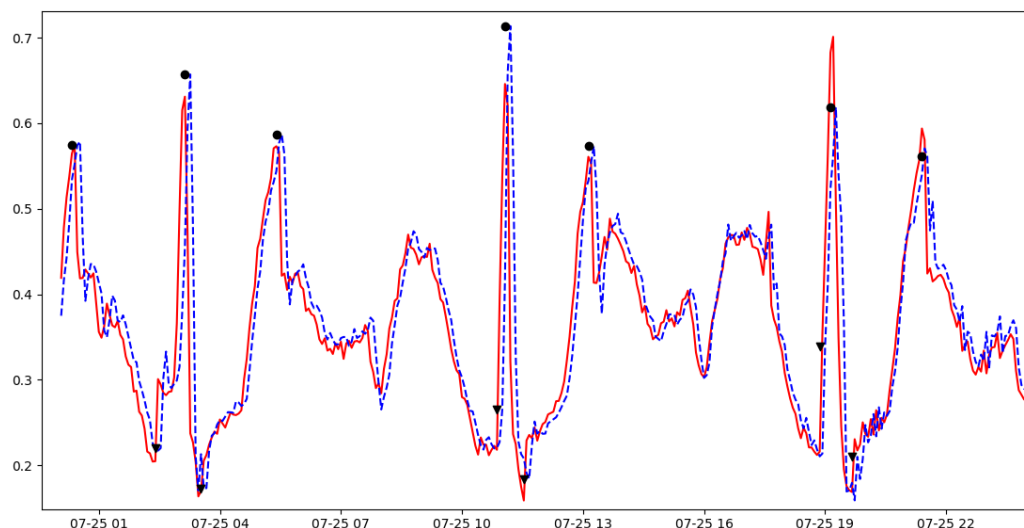
- + dự đoán 4 phút sau cũng $< 0.27 * k$

- + trung bình m điểm đã có $< 0.27 * k$, ở đây $m = 3$.

Kịch bản:

- Cho jmeter bắn request theo kịch bản trên.

- Trong mỗi chu kỳ 8h, em bắn thêm request trong khoảng 40 phút cuối.



Bảng kết quả thử nghiệm: Đường đỏ: dữ liệu thật, Đường xanh nét đứt: dữ liệu dự đoán, Hình tròn đen: thời điểm tạo thêm máy ảo, Hình tam giác đen: thời điểm xóa bớt máy ảo.

Độ chính xác theo từng lần update

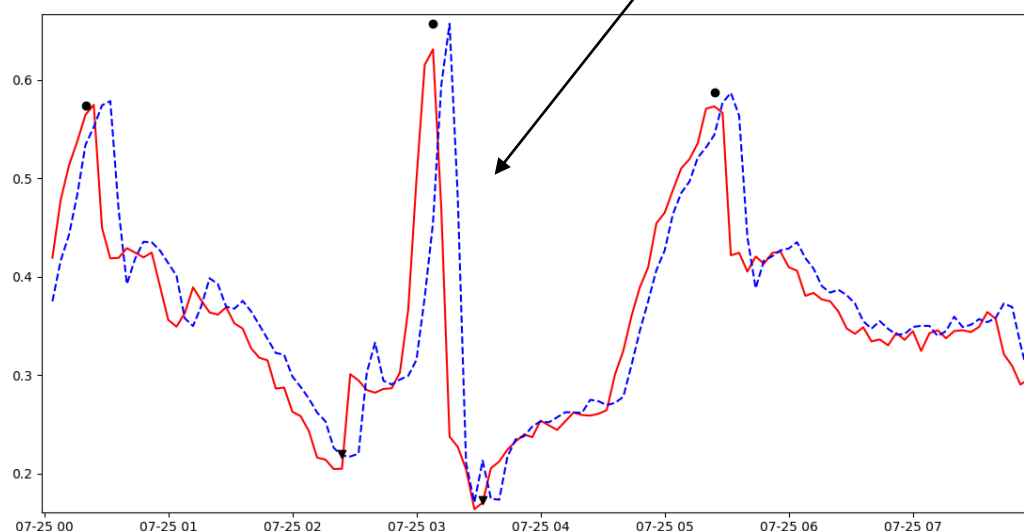
	Update 1	Update 2	Update 3
MAE	0.0389492	0.0337054	0.0309157
RMSE	0.0688761	0.0667234	0.0525199
MAPE	0.1135399	0.0983195	0.0897212

Bảng MAE, RMSE, MAPE

Thử nghiệm chạy được 3 lần update

Update 1:

Chu kỳ dữ liệu train thuật toán
 AUTO SCALE phát hiện được: không có



Ở đây đỉnh thứ 2 là do bắn thêm request vào. Cũng tương tự với 2 lần update còn lại.

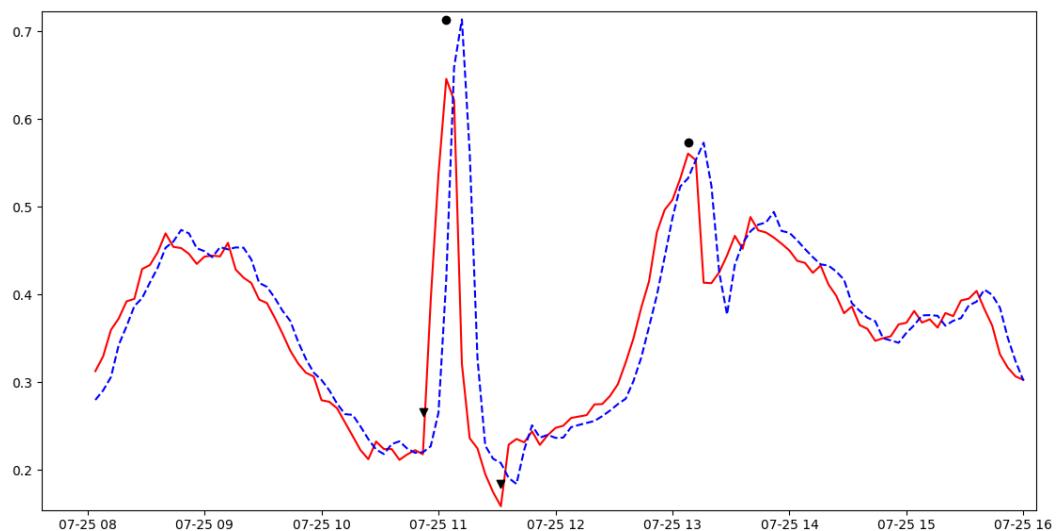
Các điểm phát hiện scale up:

Điều kiện scale phụ thuộc vào giá trị được dự đoán (4 phút sau).

	Giá trị thật	Giá trị dự đoán 4 phút tiếp
1	0.564515617227	0.552029
2	0.631060618395	0.594336
3	0.573064421474	0.576719

Update 2:

Chu kỳ dữ liệu train: không có



Các điểm phát hiện scale up:

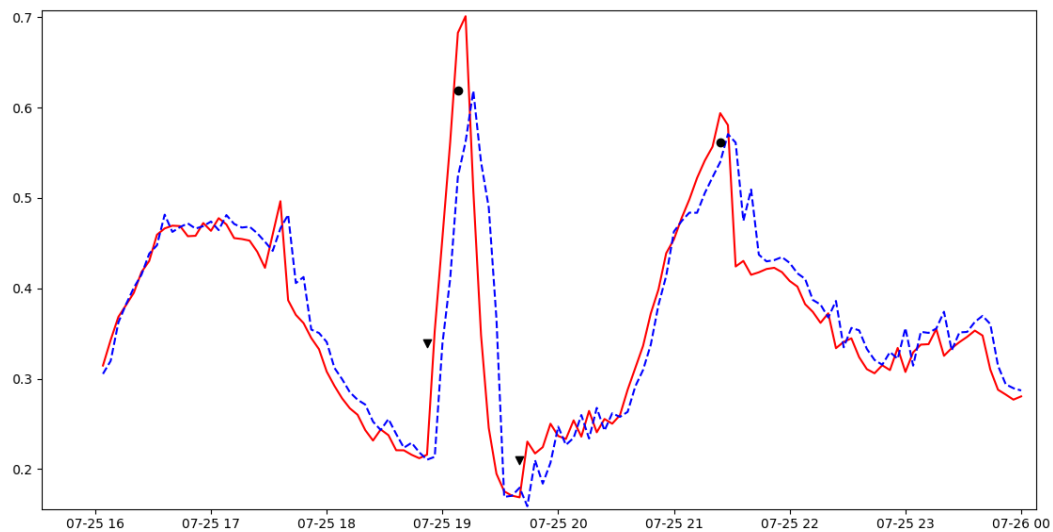
	Giá trị thật	Giá trị dự đoán
1	0.645878836204	0.659165
2	0.560714454337	0.553439

Update 3:

Chu kỳ dữ liệu train: không có

Các điểm phát hiện scale up:

	Giá trị thật	Giá trị dự đoán
1	0.645878836204	0.659165
2	0.560714454337	0.553439



Các điểm phát hiện scale up:

	Giá trị thật	Giá trị dự đoán
1	0.68267326644	0.56223
2	0.593808467451	0.570563

Kết luận:

- Chứng minh được dự đoán tốt hơn sau nhiều lần update (các sai số giảm dần).
- Vấn đề:
 - Khi bắt thêm request (đỉnh thứ 2), mô hình không bám sát sườn đi lên. Cái này em nghĩ quan trọng vì nó ảnh hưởng tới mục đích chính của dự đoán là đoán được thời điểm trước khi quá tải để scale. Nhưng theo kết quả thì kết quả dự đoán trường hợp bắt thêm request chậm hơn dữ liệu thật nên scale vẫn chậm hơn máy ảo bị quá tải. Lần Update 2 dự đoán sườn đi lên tốt hơn Update 1, Update 3 tốt hơn nữa nhưng Update 3 giá trị dự đoán ít hơn giá trị thật nên kết quả scale vẫn chưa tốt.

Em định thử nghiệm vòng lặp dự đoán ít hơn (2,3 phút 1 lần, dự đoán trước 4,6 phút tương lai) xem kết quả scale có tốt hơn không.

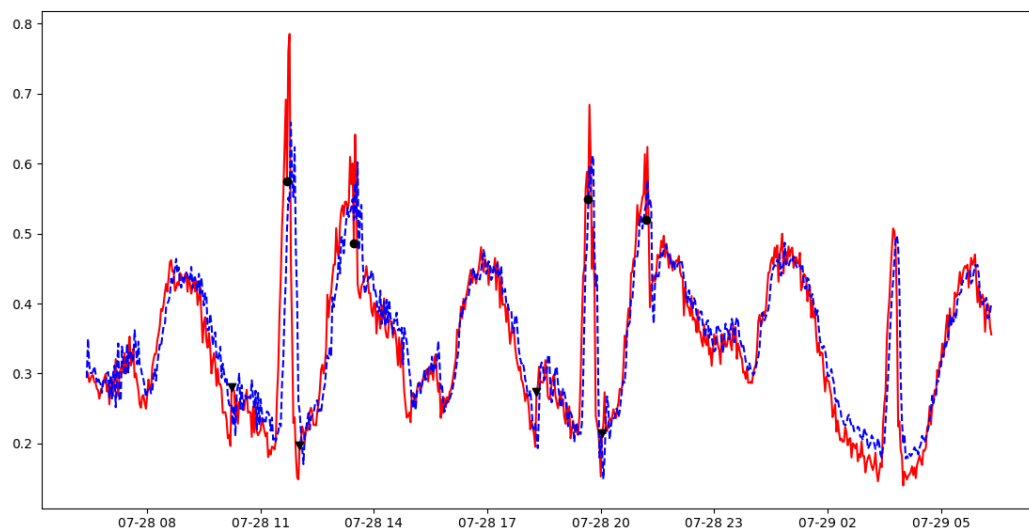
Thử nghiệm mô hình dữ liệu mỗi 2 phút, dự đoán trước 4 phút.

Giống thử nghiệm trên, update sau 8h, bổn thêm request trong khoảng 40 phút nhưng khác thời điểm so với thử nghiệm trên.

Dữ liệu:



Kết quả:



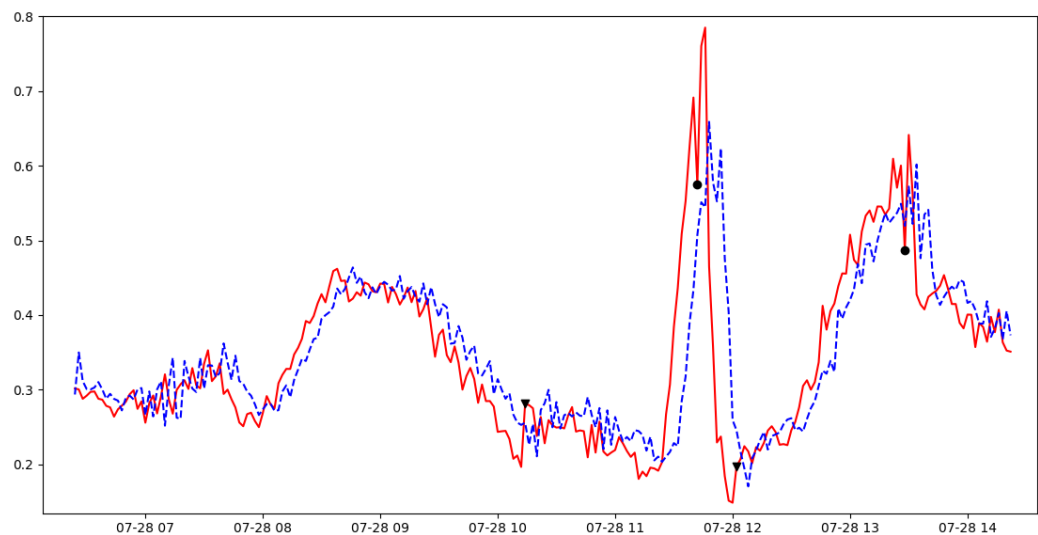
Bảng kết quả thử nghiệm: Đường đỏ: dữ liệu thật, Đường xanh nét đứt: dữ liệu dự đoán, Hình tròn đen: thời điểm tạo thêm máy ảo, Hình tam giác đen: thời điểm xóa bớt máy ảo.

Độ chính xác theo từng lần update

	Update 1	Update 2	Update 3
MAE	0.0441327	0.0280163	0.0261965
RMSE	0.0724164	0.0431355	0.0369108
MAPE	0.1390710	0.0869374	0.0977987

Bảng MAE, RMSE, MAPE

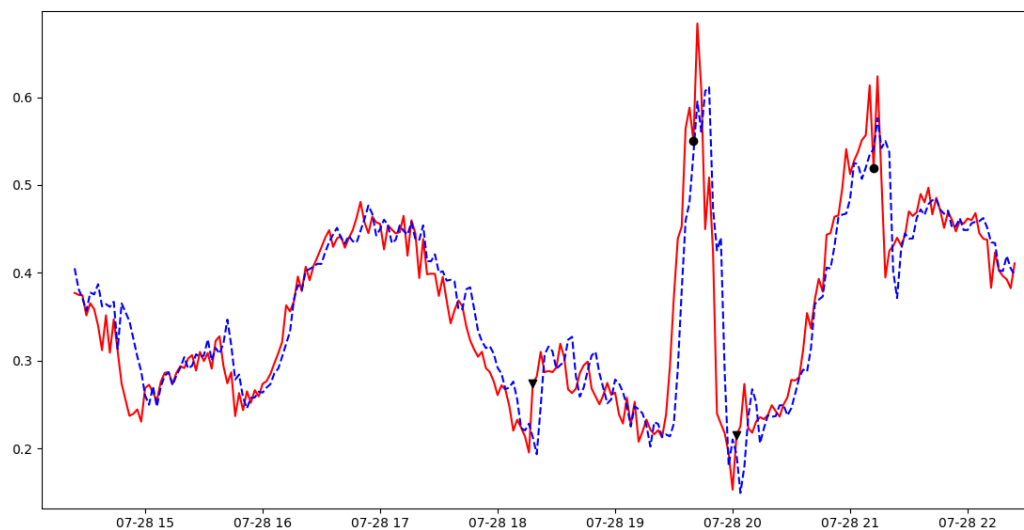
Update 1:
Chu kỳ: không



Các điểm phát hiện scale up:

	Giá trị thật	Giá trị dự đoán
1	0.621239427179	0.567302
2	0.550086988867	0.556884

Update 2:
Chu kỳ: không



Các điểm phát hiện scale up:

	Giá trị thật	Giá trị dự đoán
1	0.691480386024	0.551197
2	0.600309219959	0.572176

Update 3:

Chu kỳ: không

