

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG



NIÊN LUẬN NGÀNH
KHOA HỌC MÁY TÍNH

Đề tài

DỰ BÁO THỜI TIẾT VỚI THUẬT TOÁN RNN

Sinh viên thực hiện: Nguyễn Việt Hào

MSSV: B1812338

Khóa: 44

Cần Thơ, 05/2022

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG



NIÊN LUẬN NGÀNH
KHOA HỌC MÁY TÍNH

Đề tài

DỰ BÁO THỜI TIẾT VỚI THUẬT TOÁN RNN

Giáo viên hướng dẫn:

Th.S.Phạm Nguyên Hoàng

Sinh viên thực hiện:

Nguyễn Việt Hào

MSSV: B1812338

Khóa: 44

Cần Thơ, 05/2022

NHẬN XÉT CỦA GIẢNG VIÊN

LỜI CẢM ƠN

Đầu tiên em xin gửi lời cảm ơn chân thành đến thầy Phạm Nguyên Hoàng đã hướng dẫn em hoàn thành học phần Niên luận ngành này. Học phần đã giúp em mở rộng thêm nhiều kiến thức mới trong quá trình làm và vận dụng tốt những kiến thức đó. Cảm ơn thầy đã tận tình giảng dạy hướng dẫn cũng như tạo điều kiện thuận lợi nhất cho em trong suốt quá trình làm đề tài. Do hạn chế về thời gian và trình độ, bản báo cáo này chắc chắn sẽ có những thiếu sót. Mong thầy góp ý cho bài báo cáo của em để có thể làm hành trang quý giá cho tương lai. Em xin chân thành cảm ơn !

Cần Thơ, ngày 12 tháng 5 năm 2022

MỤC LỤC

NHẬN XÉT CỦA GIẢNG VIÊN	1
LỜI CẢM ƠN	2
MỤC LỤC	3
DANH MỤC HÌNH	6
DANH MỤC BẢNG	8
ABSTRACT	9
TÓM TẮT	10
PHẦN GIỚI THIỆU	11
1. Đặt vấn đề	11
2. Lịch sử giải quyết vấn đề	11
2.1 Daily Temperature Prediction Using Recurrent Neural Networks and Long-Short Term Memory.....	11
2.2 Machine Learning – Thủ làm Nhà Thiên Văn Dự báo thời tiết.	13
3. Mục tiêu đề tài	15
4. Đối tượng và phạm vi nghiên cứu.....	15
5. Phương pháp nghiên cứu.....	16
6. Kết quả đạt được	17
7. Bố cục luận	17
PHẦN NỘI DUNG	18
Chương I. MÔ TẢ BÀI TOÁN	18
1. Mô tả chi tiết bài toán	18
2. Các vấn đề và giải pháp liên quan	18
2.1 Mạng nơ ron nhân tạo (neural network)	18
2.2 Mạng nơ ron truyền thẳng (feed forward neural network)	18
2.3 Mạng nơ ron hồi quy (RNN)	19
2.3.1 Ví dụ về RNN mô hình hóa trong xử lý ngôn ngữ.....	20
2.3.2 Quá trình lưu thông tin trong RNN	21

2.4 Thuật toán lan truyền ngược (BBTT – Backpropagation Through Time).....	22
2.5 Ưu điểm và nhược điểm của RNN	22
2.5.1 Ưu điểm.....	23
2.5.2 Nhược điểm.....	23
2.6 Vấn đề và giải pháp liên quan đến bài toán	23
2.6.1 Mô hình tuần tự Sequential	23
2.6.2 Long-short term memory – LSTM.....	24
3. Các công cụ sử dụng và một số thư viện.	25
3.1 Google Colaboratory.....	25
3.2 Thư viện Keras.....	26
3.3 Thư viện Pandas.....	27
3.4 Thư viện Numpy.....	27
3.5 Thư viện Sklearn.....	28
3.6 Thư viện Matplotlib	28
Chương II. THIẾT KẾ VÀ CÀI ĐẶT.....	30
1. Thiết kế hệ thống.....	30
1.1 Đọc tập dữ liệu.....	30
1.2 Tiền xử lý dữ liệu.....	31
1.3 Xây dựng mô hình (thiết kế mạng nơ ron)	31
2. Cài đặt hệ thống	32
2.1 Đọc tập dữ liệu.....	32
2.1.1 Thông tin cơ bản của tập dữ liệu.....	32
2.1.2 Thống kê cơ bản về dữ liệu.....	32
2.1.3 Kiểm tra các giá trị rỗng	33
2.2 Tiền xử lý dữ liệu.....	33
2.2.1 Thay đổi chỉ mục cho tập dữ liệu.....	33
2.2.2 Xử lý các giá trị bị thiếu (missing values)	34
2.2.3 Xử lý bằng Label Encoder	34
2.2.4 Xóa các cột không cần thiết	35
2.2.5 Hiển thị dữ liệu.....	36
2.2.5.1 Biểu đồ Ta (nhiệt độ trung bình-độ C)	36
2.2.5.2 Biểu đồ Tx (biểu đồ nhiệt độ cao nhất-độ C)	36
2.2.6 Scale dữ liệu	36

2.3 Xây dựng mô hình (model).....	37
2.3.1 Thiết kế mạng.....	37
2.3.2 Tạo tập dữ liệu huấn luyện.....	38
2.3.3 Tập dữ liệu kiểm tra	38
2.3.4 Biên dịch mô hình và huấn luyện mô hình	38
2.4 Dự đoán thời tiết Cần Thơ với tập dữ liệu kiểm tra dựa trên mô hình đã xây dựng.....	39
Chương III. KẾT QUẢ THỰC NGHIỆM.....	44
1. Kiểm thử	44
1.1 Trường hợp kiểm thử 1: Scale bằng hàm RobustScaler vs Min-Max Scaler (Cần Thơ).....	44
1.2 Trường hợp kiểm thử 2: Tăng số noron ở tầng LSTM	44
1.3 Trường hợp kiểm thử 3: Thủ mới Station Sapa và Đà Nẵng.....	45
1.3.1 Biểu diễn dự đoán Ta (average temperature):.....	46
1.3.2 Biểu diễn dự đoán rH (relative humidity)	47
1.3.3 Biểu diễn dự đoán Sh (hours of sunshine)	47
1.4 Trường hợp kiểm thử 4: với timestep = 24 (Cần Thơ)	48
2. Đánh giá	49
PHẦN KẾT LUẬN.....	50
1. Kết quả đạt được	50
1.1 Kỹ năng.....	50
1.2 Chương trình	50
2. Hướng phát triển	50
TÀI LIỆU THAM KHẢO	51

DANH MỤC HÌNH

Hình 1: Mô hình LSTM	12
Hình 2: Dữ liệu tóm tắt của Hà Nội.....	13
Hình 3: Ví dụ mô tả tập dữ liệu huấn luyện.....	14
Hình 4: Kết quả dự đoán khá tốt với 15 ngày	15
Hình 5: Mạng RNN (bên trái), mạng FFNN (bên phải)	19
Hình 6: Mô hình mạng RNN	20
Hình 7: Ví dụ về xử lý ngôn ngữ bằng RNN	20
Hình 8: Quá trình lưu thông tin trong cell	21
Hình 9: Mô hình tuần tự Sequential.....	24
Hình 10: Recurrent Neural Network.....	24
Hình 11: Long short term memory	25
Hình 12: Giao diện của Google Colab.....	26
Hình 13: Một Pandas dataframe	27
Hình 14: Hàm reshape trong Numpy	28
Hình 15: Các dạng biểu đồ trong Matplotlib0	29
Hình 16: Tập dữ liệu thời tiết ban đầu của Việt Nam.....	30
Hình 17: Dữ liệu tệp dulieuthoitiet.csv.....	31
Hình 18: Thông tin của tập dữ liệu	32
Hình 19: Thống kê tập dữ liệu ban đầu.....	33
Hình 20: Hàm strftime() trong datetime.....	34
Hình 21: Thêm cột mã hóa của cột station	34
Hình 22: Thống kê số tháng của station	35
Hình 23: Tập dữ liệu Cần Thơ sau khi xóa các cột không cần thiết.....	35
Hình 24: Biểu đồ nhiệt độ trung bình	36
Hình 25: Biểu đồ nhiệt độ cao nhất	36
Hình 26: Dữ liệu sau khi đã scale	37
Hình 27: Mô hình Stacked LSTM	37
Hình 28: Dữ liệu minh họa x_train và y_train	38
Hình 29: Kết quả của quá trình huấn luyện	39
Hình 30: Loss của model	39
Hình 31: Biểu diễn giá trị Nhiệt độ Trung bình huấn luyện, giá trị dự đoán và thực tế.....	40
Hình 32: Biểu diễn giá trị Nhiệt độ Trung bình dự đoán và thực tế	41

Hình 33: Biểu diễn giá trị Độ ẩm tương đối huấn luyện, giá trị dự đoán và thực tế	41
Hình 34: Biểu diễn giá trị Độ ẩm tương đối dự đoán và thực tế.....	42
Hình 35: Biểu diễn giá trị Độ ẩm tuyệt đối huấn luyện, giá trị dự đoán và thực tế	42
Hình 36: Biểu diễn giá trị Độ ẩm tuyệt đối dự đoán và thực tế	43
Hình 37: Biểu diễn dự đoán Ta của Sapa	46
Hình 38: Biểu diễn dự đoán Ta của Đà Nẵng	46
Hình 39: Biểu diễn dự đoán rH của Sapa	47
Hình 40: Biểu diễn dự đoán rH của Đà Nẵng	47
Hình 41: Biểu diễn dự đoán Sh của Sapa	48
Hình 42: Biểu diễn dự đoán Sh của Đà Nẵng	48

DANH MỤC BẢNG

Bảng 1: So sánh loss và accuracy của SGD và Adam	12
Bảng 2: Bảng so sánh loss và accuracy khi thay đổi lượng dữ liệu được huấn luyện.....	12
Bảng 3: Bảng so sánh loss và accuracy khi thay đổi cách chia dữ liệu huấn luyện và kiểm tra	13
Bảng 4: Thống kê giá trị Null	33
Bảng 5: RMSE và MAE của mô hình dự đoán cho Cần Thơ.....	40
Bảng 6: Kết quả kiểm thử thay đổi phương pháp Scale	44
Bảng 7: Kết quả kiểm thử thay đổi số noron	45
Bảng 8: Kết quả kiểm thử với Station khác	46
Bảng 9: Kết quả kiểm thử thay đổi time_step.....	49

ABSTRACT

Weather forecasting is one of the most important aspects of modern life. Weather forecasting will let us know the weather only in the future, so that people can prepare for the future, for travels, ... or even organizations that emigrate from disaster will exist, thanks to the advancements of current science and technology.

Because weather is a time series, we may design a weather prediction model for the future utilize data acquired in the past with the advancement of in the contemporary world. As a result, in this topic, I'll use recurrent neural networks in general, and LSTM (long short term memory), which is a type of RNN in particular, as well as the Stacked LSTM model to predict weather parameters like temperature, humidity, rainfall, and so on, using accessible weather data.

Research and apply data preprocessing techniques such as: to encode numeric data, research and implement data preparation techniques such as Label Encoder... The neural network model will next be generated using the Keras library's Sequential model. This model is quite powerful with deep learning, and it is well suited for time series problems, such as weather forecasting.

The project's outcomes include a successful neural network model for time series forecasting in this topic, weather indicators, with high model evaluation indexes (rmse, mae) unexpected, but some data fields may not yield good results because there are weather indicators with a lot of variation between the past and the future. As a result, the built prediction model will be unable to predict these data fields accurately.

TÓM TẮT

Dự báo thời tiết là một trong những điều cần thiết trong thế giới hiện đại ngày nay. Với sự phát triển của khoa học công nghệ hiện đại, dự báo thời tiết sẽ giúp cho chúng ta biết được các chỉ số thời tiết trong tương lai, nhờ vào đó con người có thể lập kế hoạch cho tương lai, cho các chuyến đi chơi,... hay thậm chí là tổ chức di tán người dân từ những dự báo thiên tai sẽ xảy ra.

Thời tiết có tính chất chuỗi thời gian, nên với sự phát triển của hiện tại ta có thể xây dựng một mô hình dự đoán thời tiết cho tương lai bằng các số liệu thu thập được từ quá khứ. Vì thế trong đề tài này em sẽ sử dụng mạng nơ ron hồi quy nói chung và LSTM (long short term memory) là một dạng của RNN nói riêng, cùng với áp dụng mô hình Stacked LSTM để dự đoán cho các thông số thời tiết như nhiệt độ, độ ẩm, lượng mưa, ... bằng những số liệu thời tiết có sẵn.

Nghiên cứu và áp dụng các kỹ thuật tiền xử lý dữ liệu như Label Encoder để mã hóa dữ liệu về kiểu số, ... Mô hình mạng nơ ron sau đó sẽ được xây dựng nhờ vào mô hình tuần tự Sequential của thư viện Keras rất mạnh mẽ với mô hình học sâu, mô hình này được rất thích hợp cho bài toán chuỗi thời gian và ở đề tài sẽ là bài toán dự báo thời tiết.

Kết quả của đề tài đã thu được một mô hình mạng noron hiệu quả cho dự báo chuỗi thời gian ở đề tài này là dự báo các chỉ số thời tiết, đã đạt được các chỉ số đánh giá mô hình (rmse, mae) cao ngoài mong đợi, nhưng đối với một số trường dữ liệu có thể sẽ cho kết quả không cao vì tồn tại các chỉ số thời tiết có sự biến thiên rất cao giữa quá khứ và tương lai dẫn đến mô hình dự đoán đã xây dựng sẽ không thể dự đoán tốt cho các trường dữ liệu này.

PHẦN GIỚI THIỆU

1. Đặt vấn đề

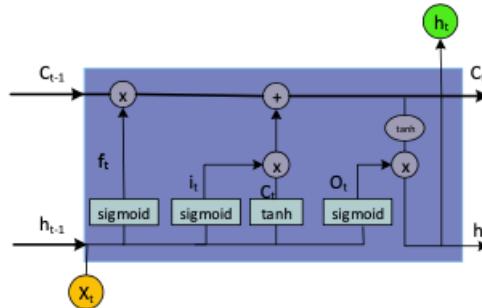
Hiện nay, với sự phát triển của kỹ thuật khoa học, công nghệ tiên tiến thì việc dự báo thời tiết cho mọi địa điểm trên địa cầu đã không còn quá xa lạ. Con người cũng đã nỗ lực trong việc dự báo thời tiết không chính thức trong nhiều thiên niên kỷ trước và mới bắt đầu chính thức từ thế kỷ mươi chín. Nhờ sự phát triển của khoa học công nghệ, các thiết bị hiện đại ngày nay thì các dữ liệu thu thập một cách dễ dàng hơn từ các trạm thời tiết, vệ tinh được sử dụng để phân tích và dự đoán các trạng thái, chỉ số thời tiết. Con người dựa trên các quan sát trong quá khứ về thời tiết để có thể dự đoán được thời tiết trong tương lai. Và việc áp dụng máy học đã được sử dụng rộng rãi và có thể hoạt động tốt trong lĩnh vực này.

2. Lịch sử giải quyết vấn đề.

2.1 Daily Temperature Prediction Using Recurrent Neural Networks and Long-Short Term Memory

Đây là bài viết viết về chủ đề dự báo nhiệt độ với RNN và LSTM, được viết bởi Ike Sri Rahayu, Esmeralda C Djamal, Ridwan Ilyas của khoa tin học, đại học Jenderal Achmad Yani, Indonesia. Dữ liệu của bài báo được lấy từ Cơ quan Địa Vật lý và khí tượng (BMKG) ở Bandung từ năm 2000 đến năm 2019. Bài báo này xây dựng một mô hình dự đoán nhiệt độ hằng ngày trong ba ngày tới bằng 5 lớp: lạnh, mát, bình thường, ấm và nóng bằng cách sử dụng mạng Nơ ron hồi quy và bộ nhớ ngắn hạn dài hạn (LSTM). Nghiên cứu được tiến hành do nhiệt độ là một trong những chỉ số thời tiết quan trọng, nhiệt độ thay đổi có thể ảnh hưởng đến cơ thể con người, đặc biệt là nếu sử dụng quần áo không phù hợp, điều này khiến con người phải thay đổi chất liệu quần áo để phù hợp hơn. Bài báo nghiên cứu xây dựng một mô hình có thể dự đoán nhiệt độ hằng ngày bằng cách sử dụng RNN. Với dữ liệu đầu vào là 4 chỉ số thời tiết là nhiệt độ, độ ẩm, lượng mưa và tốc độ gió của 20 năm. Dữ liệu trước khi được huấn luyện sẽ được tiền xử lý (tính trung bình cho các giá trị bị thiếu, scale dữ liệu, xử lý các giá trị NaN...) và sau đó huấn luyện bằng mô hình RNN và LSTM dự đoán nhiệt độ cho ba ngày tiếp theo. Sử dụng kỹ thuật phân đoạn, tức là nhóm theo mỗi ba ngày nhóm thành 1 bộ (gồm đầy đủ chỉ số thời tiết), kết quả sẽ tạo ra 2406 bộ dữ liệu, mỗi bộ là một tháng được sắp xếp theo thời gian. Đầu ra của mô hình là hàm Softmax để phân loại cho nhiệt độ đầu ra. Song đó thì RNN là một bộ nhớ ngắn hạn nên sẽ có hạn chế trong xử lý nhiều dữ liệu nên LSTM là một mô hình có thể khắc phục

được những hạn chế đó, có khả năng học các phụ thuộc dài hạn và các quá trình tính toán sẽ phức tạp hơn.



Hình 1: Mô hình LSTM

Bài báo này xây dựng mô hình dự đoán nhiệt độ mỗi ba ngày với dữ liệu của 20 năm, 14 năm và 4 năm. Mô hình sử dụng 2 bộ tối ưu hóa (optimizer) là Stochastic Gradient Descent (SGD) và Adaptive Moment Estimation (Adam). Trong đó mô hình Adam qua thử nghiệm sẽ cho kết quả tốt hơn mô hình SGD. Sau đó các thử nghiệm sẽ được thử nghiệm với bộ optimizer Adam

No	Optimization Model	Training Data		Testing Data	
		Loss	Accuracy (%)	Loss	Accuracy (%)
1.	SGD	0.01271	87.24	0.01346	76.48
2.	Adam	0.01041	90.92	0.01079	80.36

Bảng 1: So sánh loss và accuracy của SGD và Adam

Kiểm thử với dữ liệu huấn luyện là 20 năm, 12 năm và 4 năm cho ra được độ chính xác cao nhất là 90.92% trên tập dữ liệu huấn luyện và 80.36% trên tập dữ liệu kiểm tra. Kết quả cho thấy được việc giảm lượng tập dữ liệu huấn luyện đi sẽ làm ảnh hưởng đến độ chính xác của mô hình (loss tăng lên accuracy giảm xuống)

No	Dataset	Training Data		Testing Data	
		Loss	Accuracy (%)	Loss	Accuracy (%)
1.	20 years	0.01041	90.92	0.01079	80.36
2.	12 years	0.01127	89.34	0.01292	78.26
3.	4 years	0.01525	83.33	0.01325	76.54

Bảng 2: Bảng so sánh loss và accuracy khi thay đổi lượng dữ liệu được huấn luyện

Và cách chia lượng tập dữ liệu huấn luyện và kiểm tra cũng sẽ ảnh hưởng đến độ chính xác của mô hình:

Training (%)	Testing (%)	Training Data		Testing Data	
		Loss	Acc (%)	Loss	Acc (%)
80	20	0.01041	90.92	0.01079	80.36
70	30	0.01266	86.94	0.01166	79.14
60	40	0.01375	85.26	0.01471	75.34

Bảng 3: Bảng so sánh loss và accuracy khi thay đổi cách chia dữ liệu huấn luyện và kiểm tra

Qua các thử nghiệm, có thể kết luận rằng mô hình tối ưu hóa, lượng dữ liệu và cách chia dữ liệu có thể ảnh hưởng đến kết quả thu được.

2.2 Machine Learning – Thủ làm Nhà Thiên Văn Dự báo thời tiết.

Bài viết này được tác giả Lavender nói về việc sử dụng các thông tin thời tiết trong quá khứ để dự đoán cho thời tiết hôm sau. Dữ liệu của bài viết này là dữ liệu của thời tiết Hà Nội được lấy từ <https://www.meteoblue.com/> với dữ liệu miễn phí thì chỉ có thể download được dữ liệu thời tiết của 15 ngày gần nhất và có thể trả phí để download dữ liệu trong 30 năm gần nhất.

Year	Month	Day	Hour	Minute	Temperature	Total_Precipitation	Wind_Speed	Wind_Direction
0	2018	1	15	0	0	16.40	0.0	6.73
1	2018	1	15	1	0	16.27	0.0	6.62
2	2018	1	15	2	0	16.14	0.0	6.30
3	2018	1	15	3	0	16.04	0.0	6.30
4	2018	1	15	4	0	15.97	0.0	6.19
5	2018	1	15	5	0	15.90	0.0	5.90

Hình 2: Dữ liệu thời tiết của Hà Nội

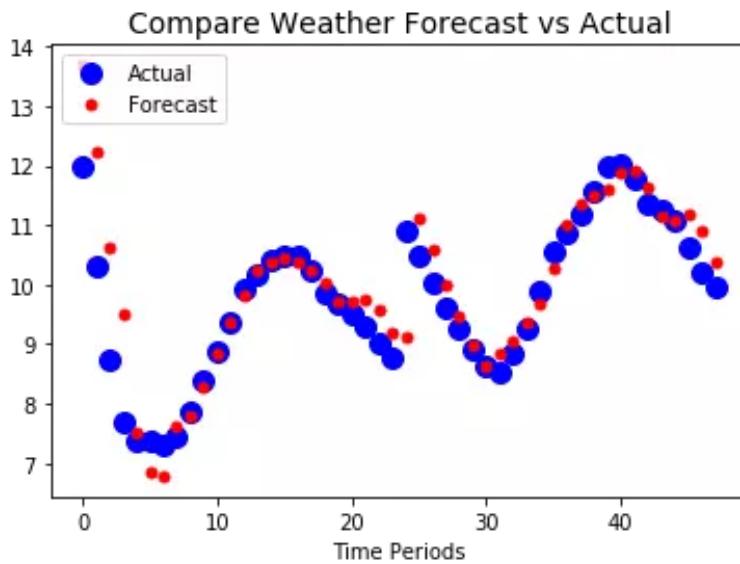
Tác giả sử dụng thư viện Tensorflow để hỗ trợ việc huấn luyện vì đã được tích hợp nhiều thuật toán khác nhau giúp tối ưu được thời gian xây dựng hệ thống học sâu (deep learning). Cùng với các thư viện Pandas, Numpy để tiền xử lý dữ liệu, cũng như đưa ra cái nhìn tổng quan về dữ liệu bằng cách vẽ các đồ thị bằng thư viện Matplotlib. Vì dữ liệu thời tiết là dữ liệu chuỗi thời gian nên các dữ liệu ở tương lai sẽ phụ thuộc vào các dữ liệu trong quá khứ, mang trong mình tính chất chuỗi thời gian, nên tác giả sử dụng các dữ liệu trong thời gian trước cụ thể là 1 giờ trước làm X (tập dữ liệu huấn luyện) và dữ liệu trong tương lai 1 giờ liền kề làm Y

x	y
15.97	15.9
15.9	15.83
15.83	15.85
15.85	16.24
16.24	16.69
16.69	17.22
17.22	17.72
17.72	18.25

Hình 3: Ví dụ mô tả tập dữ liệu huấn luyện

Do đó, tác giả sẽ sử dụng giá trị thuộc tính Temperature (nhiệt độ) độ C từ 15-01-2018 00:00:00 đến 29-01-2018 00:00:00 làm tập X_train với y_train sẽ là giá trị từ 15-01-2018 01:00:00 và hai ngày cuối cùng làm tập X_test (tương đương với 48 giờ) – mô hình xây dựng sẽ dự đoán chỉ số thời tiết cho 48 giờ này. Biến đổi tập dữ liệu về dạng (samples, time_steps, feature) để phù hợp với đầu vào của mô hình mạng nơron ở đây sẽ là (14, 24, 1) tương đương 14 ngày, 1 ngày 24 giờ, và 1 giờ ứng với 1 thuộc tính thời tiết làm dữ liệu đầu vào.

Tác giả lựa chọn thuật toán Recurrent Neural Network (RNN) hay mạng nơ ron hồi quy do tính chất sử dụng các dữ liệu trong quá khứ để dự đoán tương lai, có khả năng nhớ được các thông tin trước đó (do đầu ra của dữ liệu sẽ phụ thuộc vào các tính toán trước đó). Tác giả sử dụng nơ ron mô hình RNN truyền thống với 100 nơ ron đầu ra của tầng này, hàm activation là relu, phương pháp huấn luyện là Adam và hàm mất mát là MSE. Sau đó tác giả huấn luyện với 1000 epoch và đã thu được kết quả khá tốt chỉ với 15 ngày làm dữ liệu đầu vào và có thể sẽ hiệu quả hơn với dữ liệu đầu vào nhiều hơn thay vì 15 ngày.



Hình 4: Kết quả dự đoán khá tốt với 15 ngày

3. Mục tiêu đề tài

Dự báo thời tiết, trong đó thời tiết có tính chất chuỗi thời gian. Cùng với sự phát triển của các ngành như khoa học dữ liệu, các công nghệ hiện đại để thu thập dữ liệu thì chúng ta có thể xây dựng một mô hình dự đoán trạng thái thời tiết như nhiệt độ, độ ẩm... từ các dữ liệu ta thập thập được. Mục tiêu của đề tài sẽ sử dụng dữ liệu thời tiết đã có sẵn (nhiệt độ, độ ẩm,...), xây dựng một mô hình mạng nơ ron hồi quy, dự đoán và đánh giá mô hình.

4. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu của đề tài :

- Mạng nơ ron hồi quy (RNN).
- Mô hình Long short term memory của mạng Mạng nơ ron hồi quy.
- Các kỹ thuật tiền xử lý dữ liệu.
- Thư viện Keras.
- Dữ liệu chuỗi thời gian.
- Vẽ biểu đồ bằng thư viện Matplotlib

Phạm vi nghiên cứu:

- Tiền xử lý dữ liệu (xử lý dữ liệu bị thiếu)

- Nghiên cứu áp dụng thư việc Keras
- Xây dựng mô hình dự báo thời tiết với LSTM
- Đánh giá mô hình bằng các chỉ số như rmse, mae

5. Phương pháp nghiên cứu

- Xác định đề tài: Để tránh mất nhiều thời gian đọc các đề tài nội dung không liên quan hoặc không phục cho mục đích đề tài của mình thì việc đầu tiên cần làm là xác định chỉ rõ đề tài cần nghiên cứu. Để có được hướng đi cũng như cái nhìn tổng quan cho bài báo cáo.
- Xác định các từ khoá: Xác định từ khoá liên quan đến đề tài để có thể thu hẹp phạm vi, dễ dàng tìm kiếm xây dựng chủ đề tránh mất nhiều thời gian.
 - o RNN, Time series, Sequential, Long short term memory
- Tìm các đề tài tương tự và chọn lọc: Chọn lọc lại các đề tài, tài liệu tìm được, đặc biệt là đọc phần tóm tắt xem có giúp ta thu được giá trị phù hợp liên quan không?
- Đề ra các hướng làm: Xây dựng mô hình dự báo thời tiết theo cách tiếp cận với thuật toán RNN
- Tìm hiểu về RNN, LSTM, Timeseries
- Đọc tập dữ liệu, hiển thị để có cái nhìn tổng quan về dữ liệu.
- Tiền xử lý dữ liệu
- Xây dựng tập dữ liệu huấn luyện và tập dữ liệu kiểm thử.
- Xây dựng mô hình LSTM
- Đánh giá mô hình (rmse, vẽ biểu đồ)
- Nhận góp ý từ người khác để có nhiều góc nhìn hơn để hoàn thiện chỉnh sửa.
- Chính sửa lại từ các góp ý
- Viết báo cáo

6. Kết quả đạt được

ây dựng được một mô hình dự báo thời tiết từ tập dữ liệu thời tiết có sẵn bằng mô hình mạng nơ ron hồi quy long short term memory – một mô hình nâng cao của mạng nơ ron hồi quy. Đánh giá mô hình đã được xây dựng.

7. Bố cục niên luận

Bố cục gồm 3 phần với nội dung như sau:

PHẦN GIỚI THIỆU

Giới thiệu tổng quát về đề tài

PHẦN NỘI DUNG

Chương 1: Mô tả bài toán

Chương 2: Thiết kế, cài đặt giải thuật.

Chương 3: Kết quả thực nghiệm

PHẦN KẾT LUẬN

Kết quả đạt được và hướng phát triển

TÀI LIỆU THAM KHẢO

PHẦN NỘI DUNG

Chương I. MÔ TẢ BÀI TOÁN

1. Mô tả chi tiết bài toán

Chuỗi thời gian là một chuỗi các phép đo được thực hiện theo thời gian, thường thu được ở các khoảng cách đều nhau, có thể là hàng ngày, hàng tháng, hàng quý hoặc hàng năm. Nói cách khác, chuỗi thời gian là một chuỗi các điểm dữ liệu được ghi lại tại các thời điểm cụ thể. Các điểm dữ liệu này thường bao gồm các phép đo liên tiếp được thực hiện trong một thời gian và được sử dụng để theo dõi sự thay đổi theo thời gian như lưu lượng dòng chảy hàng năm, dữ liệu dân số hàng năm, lãi suất hàng tuần,... Ở bài toán này cũng sẽ giải quyết bài toán dữ liệu chuỗi thời gian, tức sẽ dự đoán cho chỉ số thời tiết cho một số tỉnh thành của Việt Nam trong tập dữ liệu 65 tỉnh thành được thu thập theo hàng tháng (ứng với mỗi dòng dữ liệu) trong 50 năm (từ 1960 đến 2010) bằng cách sử dụng mô hình mạng nơron hồi quy kết hợp ứng dụng Stacked LSTM để dự đoán cho 7 chỉ số thời tiết (nhiệt độ trung bình, nhiệt độ cao nhất, nhiệt độ thấp nhất, lượng mưa, độ ẩm tương đối, thời lượng nắng, độ ẩm tuyệt đối) để dự đoán cho một số tỉnh thành (Cần Thơ, Đà Nẵng, Sapa).

2. Các vấn đề và giải pháp liên quan

2.1 Mạng nơ ron nhân tạo (neural network)

Mạng nơ ron nhân tạo (neural network) là một mô hình toán học hay mô hình tính toán được xây dựng dựa trên mạng nơ ron sinh học, là một hệ thống tính toán lấy cảm hứng từ sự hoạt động của các nơ-ron trong hệ thần kinh của con người.

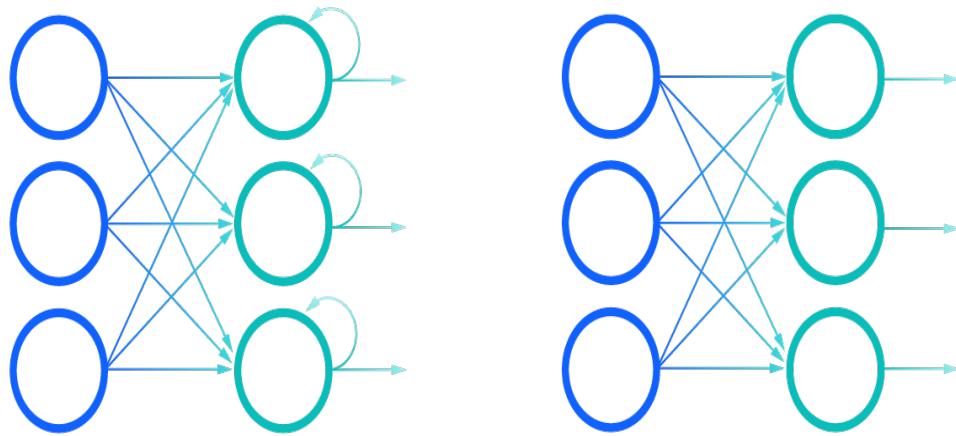
2.2 Mạng nơ ron truyền thẳng (feed forward neural network)

Trước hết nói về mạng nơron truyền thẳng (feed forward neural network), mạng nơ ron sẽ gồm 3 thành phần chính là: tầng đầu vào (input layer), tầng ẩn (hidden layer), tầng đầu ra (output layer)

- Thông tin chỉ chảy theo một hướng: từ tầng đầu vào, qua các lớp ẩn, đến lớp đầu ra
- Các mạng nơ ron này chuyển tiếp nguồn cấp dữ liệu mà không giữ lại bộ nhớ về các đầu vào mà chúng đã xử lý.
- Đầu vào và đầu ra của mạng nơ ron truyền thẳng độc lập với nhau.

Mạng nơ ron truyền thẳng (FFNN) không phù hợp với những bài toán dạng chuỗi (mô tả nội dung, hoàn thành câu, chuỗi thời gian...) vì các dự đoán tiếp theo sẽ phụ thuộc và tính toán dựa trên các dữ liệu trước nó (như vị trí trong câu, từ nào ở trước nó...)

Do đó, mạng nơ ron hồi quy (RNN) đã được ra đời để xử lý các dạng dữ liệu mà FFNN sẽ làm không tốt như dữ liệu chuỗi thời gian, dữ liệu có tính phụ thuộc nhau...

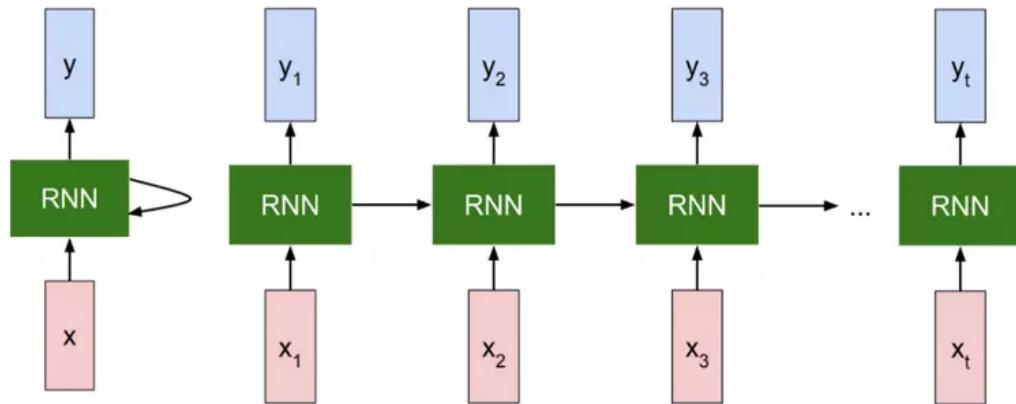


Hình 5: Mạng RNN (bên trái), mạng FFNN (bên phải)

2.3 Mạng nơ ron hồi quy (RNN)

Với mạng nơ ron thông thường thì dữ liệu chúng ta cho vào cùng một lúc, nhưng đôi khi dữ liệu chúng ta lại có quan hệ trình tự với nhau, lúc này khi chúng ta thay đổi vị trí, trình tự của dữ liệu làm mất đi ý nghĩa trình tự sẽ dẫn đến kết quả sai khác. Ví dụ: “Bạn đi học chưa” và “Bạn chưa đi học” khi tách từ ta sẽ được bộ [‘Bạn’, ‘đi’, ‘học’, ‘chưa’], ta thấy sẽ không có sự phân biệt giữa 2 câu trên. Lúc này chúng ta một mô hình mạnh mẽ hơn là RNN.

Mạng nơ ron hồi quy (RNN) sử dụng bộ nhớ để lưu giữ lại thông tin, tính toán trước đó phục vụ cho các tính toán dự đoán hiện tại và tương lai để đạt được kết quả tối ưu nhất. Như hình bên dưới, các x đại diện cho đầu vào được chia theo các bước thời gian (time_step), $x(t)$ là đầu vào cho time_step t , $y(t)$ là đầu ra của time_step t (x_3 sẽ là vector đầu vào đại diện cho từ thứ 3 trong câu)

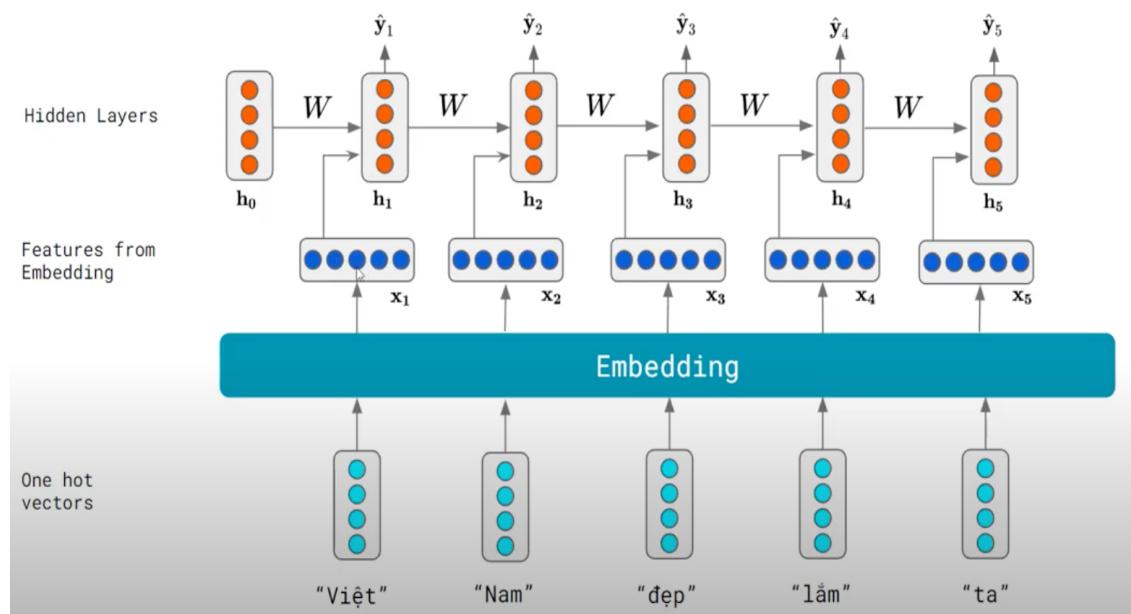


Hình 6: Mô hình mạng RNN

Một ví dụ nhỏ khi chưa có RNN về việc hoàn thành câu sử dụng mô hình đếm n-grams language model: Để dự đoán được từ “xe” khi biết từ “tôi” và từ “lái”, sẽ bằng cách lấy số lượng câu “tôi lái xe” xuất hiện trong từ điển chia (/) số lượng câu “tôi lái” trong từ điển $\Leftrightarrow P(\text{"học"} | \text{"tôi"}, \text{"lái"}) = \text{count}(\text{"tôi lái xe"}) / \text{count}(\text{"tôi lái"})$. Mô hình đếm như trên có một vài vấn đề là nếu câu “tôi lái xe” chưa bao giờ xuất hiện trong từ điển sẽ cho xác suất bằng 0, nếu từ “tôi lái” không xuất hiện trong từ điển sẽ cho xác suất không tồn tại. Cho nên sau này với sự phát triển của Deep learning thì RNN đã ra đời.

2.3.1 Ví dụ về RNN mô hình hóa trong xử lý ngôn ngữ

“Ơi” ?



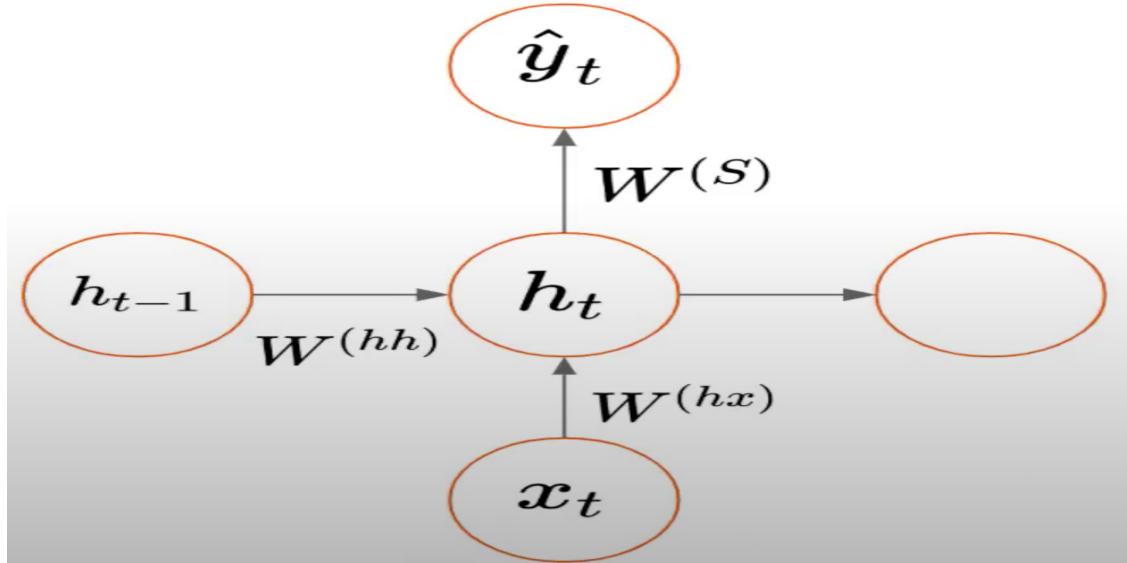
Hình 7: Ví dụ về xử lý ngôn ngữ bằng RNN

Trong ví dụ trên, đầu tiên vì chưa có giá trị đầu tiên nên có thể sẽ khởi tạo $h(0) = 0$, sau đó sẽ học giá trị của $x(1)$ - từ “Việt” sau khi word embedding (mỗi từ sẽ có các giá trị riêng của nó trong vector embedding), cho ra được xác suất $y(1)$ là xác suất của từ nào đó có khả năng xuất hiện sau từ “Việt” là cao nhất (ở đây sẽ là từ “Nam”).

Bước tiếp theo $h(2)$ lúc này sẽ học giá trị của $h(1)$ và $x(2)$ sẽ là “Việt” và “Nam”, sau đó tính xác suất của từ nào mà có xác suất đứng sau hai từ “Việt” và “Nam” cao nhất. Tương tự với các từ còn lại.

2.3.2 Quá trình lưu thông tin trong RNN

Quá trình lưu thông tin được diễn ra trong một cell được thể hiện như hình bên dưới:



Hình 8: Quá trình lưu thông tin trong cell

Trong đó,

- $x(t)$: có kích thước x
- $h(t-1)$ và $h(t)$: có kích thước h .
- $W(hh)$: $h \times h$
- $W(hx)$: $h \times x$
- $y(t)$: có kích thước y .
- $W(S)$: $S = h \times y$

Như hình bên trên, mong muốn của chúng ta là từ chuỗi lịch sử $h(t-1)$ kết hợp với $x(t)$ cho ra được output có xác suất mong muốn nhất. Công thức tính $h(t)$ trong xử lý ngôn ngữ:

$$h_t = \sigma(W^{(hh)}h_{t-1} + W^{hx}x_{[t]})$$

Đầu ra $y(t)$ sẽ được tính theo công thức: softmax là một activation function khá là phổ biến thường được dùng cho các bài toán phân loại (classification) nhiều lớp. Công thức tính $y(t)$ trong xử lý ngôn ngữ:

$$\hat{y}_t = \text{softmax}(W^{(S)}h_t)$$

Trọng số W ở đây chỉ có một và sẽ được dùng cho tất cả các cell của mô hình RNN, chỉ khác kích thước để khi kết hợp với các giá trị tạo ra vector có số chiều nhất quán phục vụ cho các phép toán trên vector.

Có thể nói một cell ở đây được coi như là mô phỏng cho việc lưu trữ thông tin để tạo ra một lịch sử mới là $h(t)$, có thể nói các mô hình nâng cao sau này như LSTM đều chỉ tìm ra hàm $h(t)$ tốt nhất, phát triển cho hàm $h(t)$ phức tạp hơn để cho ra được kết quả mong muốn hơn.

2.4 Thuật toán lan truyền ngược (BBTT – Backpropagation Through Time)

Trong quá trình training thì chúng ta có ba tham số cần phải tìm là $W(hh)$, $W(hx)$, $W(S)$ hay $W(hy)$, chúng ta cần tính đạo hàm của L theo các W trên (L là loss function) để thực hiện Gradient Descent. Đạo hàm của L với W :

$$\begin{aligned} \frac{\partial L_t}{\partial W_{hh}} &= \frac{\partial L_t}{\partial h_t} \frac{\partial h_t}{\partial h_{t-1}} \cdots \frac{\partial h_1}{\partial W_{hh}} \\ &= \frac{\partial L_t}{\partial h_t} \left(\prod_{t=2}^T \frac{\partial h_t}{\partial h_{t-1}} \right) \frac{\partial h_1}{\partial W_{hh}} \\ &= \frac{\partial L_t}{\partial h_t} \left(\prod_{t=2}^T \tanh'(W_{hh}h_{t-1} + W_{xh}x_t)W_{hh}^{T-1} \right) \frac{\partial h_1}{\partial W_{hh}} \end{aligned}$$

2.5 Ưu điểm và nhược điểm của RNN

2.5.1 Ưu điểm

- Có thể áp dụng với các dữ liệu chuỗi nói chung và dữ liệu chuỗi thời gian nói riêng.
- Mạnh mẽ hơn mô hình mạng nơ ron truyền thống.

2.5.2 Nhược điểm

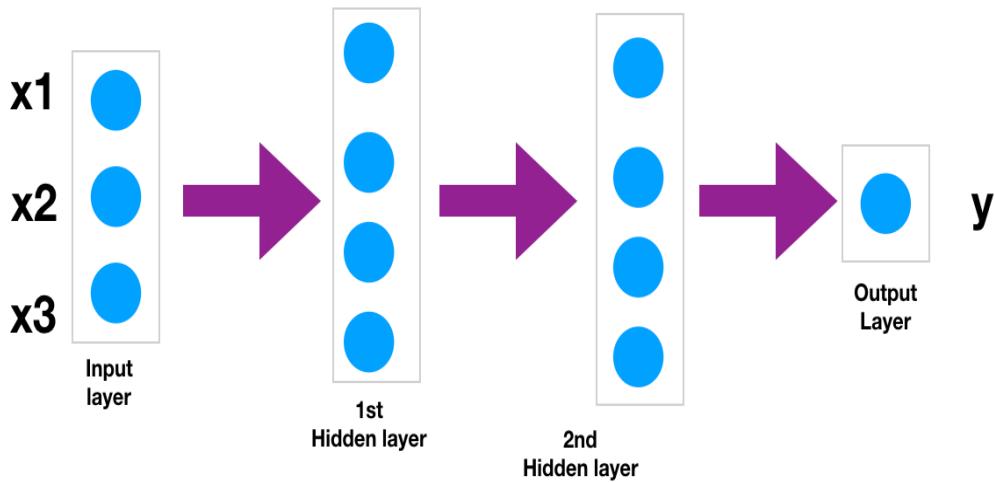
- Các cell phải tuần tự, phụ thuộc với nhau, nghĩa là muốn tính $h(t)$ là phải tính xong $h(t-1)$, muốn tính $h(t+1)$ phải tính xong $h(t)$ nên mô hình có thể chậm.
- Hiện tượng vanishing: nếu chuỗi (sequence) quá dài thì sẽ có quá nhiều phép nhân và khi trọng số w bé hơn một (do các hàm activation function trong thuật toán lan truyền ngược BBTT sẽ có giá trị bé hơn một) thì tích của nhiều số bé hơn một sẽ xấp xỉ không, dẫn đến việc cập nhật trọng số sẽ trở nên vô nghĩa.
- Hiện tượng Exploding Gradient: tùy thuộc vào hàm activation function mà làm cho ma trận trở nên lớn hơn.

2.6 Vấn đề và giải pháp liên quan đến bài toán

2.6.1 Mô hình tuần tự Sequential

Mô hình Sequential được gọi là “sequential” vì nó liên quan đến việc xác định một kiến trúc sequential và trong kiến trúc ta có thể thêm vào từng tầng theo mô hình tuyến tính từ đầu vào cho đến đầu ra theo thứ tự.

Recurrent Neural Network được sử dụng phổ biến ở mô hình Sequential này.

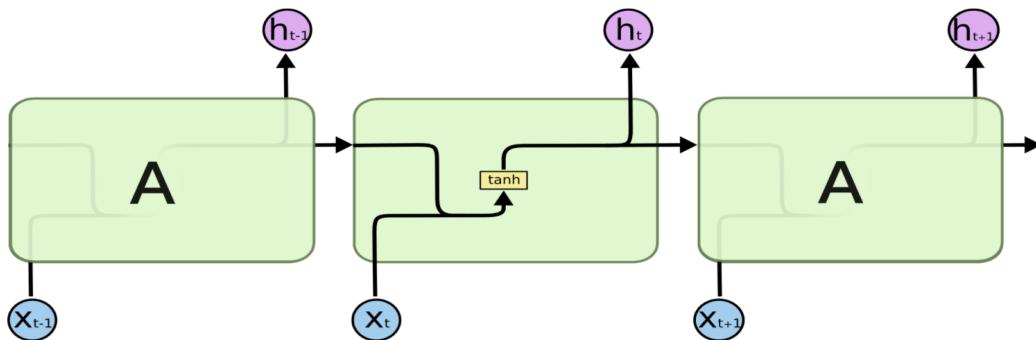


Hình 9: Mô hình tuần tự Sequential

2.6.2 Long-short term memory – LSTM

LSTM là mạng trí nhớ ngắn hạn định hướng dài hạn là một mô hình nâng cấp của RNN có khả năng học đc sự phụ thuộc trong dài hạn.

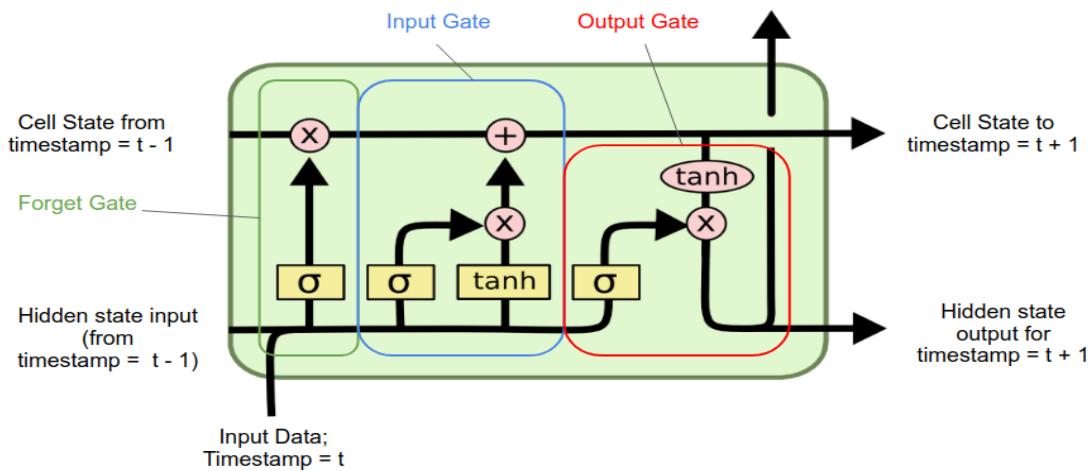
Bản chất vẫn là RNN nhưng hàm các cell trong LSTM sẽ phức tạp hơn để cso thê nhận đc $h(t)$ tốt nhất.



Hình 10: Recurrent Neural Network

LSTM khắc phục được những hạn chế của RNN vì RNN mang thông tin từ tầng (layer) trước ra layer sau nhưng chỉ mang qua được một số trạng thái (state) nhất định sau đó sẽ bị vanishing gradient => RNN là short term memory, LSTM sẽ có thêm các cổng (gate) để có thể giữ lại các thông tin cần thiết sẽ được dùng sau (có thể hạn chế được khả năng bị vanishing gradient) khắc phục được những hạn

chế của RNN => LSTM sẽ mang được thông tin ở xa hơn => Long short term memory.



Hình 11: Long short term memory

LSTM có các cổng (gates) bên trong cho phép mô hình huấn luyện tốt hơn với thuật toán BPTT so với với RNN, điều chỉnh luồng thông tin được đưa qua (thông tin nào được lưu trữ, thông tin nào bị xóa đi). LSTM nổi bật khi có thêm một trạng thái ô được sử dụng như một đường dẫn (pathway) để kết nối luồng dữ liệu từ mỗi cổng. Một khía cạnh LSTM vẫn có thể bị vanishing gradient nhưng mô hình LSTM này sẽ mạnh mẽ hơn, hạn chế được hiện tượng đó.

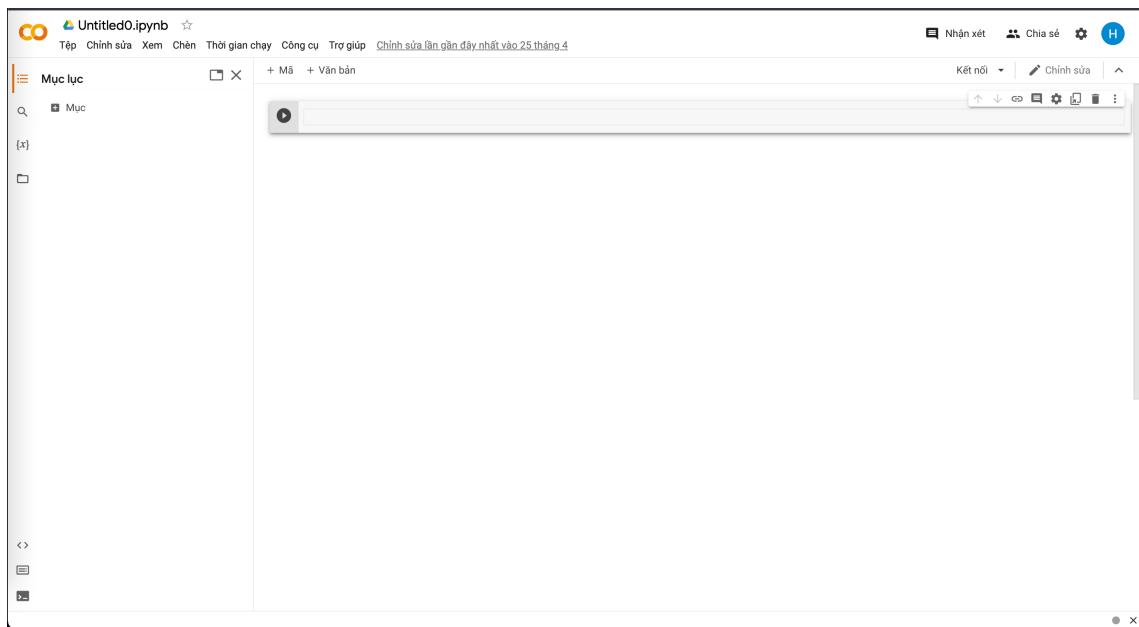
3. Các công cụ sử dụng và một số thư viện.

3.1 Google Colaboratory

Google Colaboratory được gọi tắt là Google Colab là một thành quả của Google Research. Mục đích Google Colab là giúp chúng ta chạy code Python trực tiếp thông qua trình duyệt, chúng phù hợp với các công việc liên quan đến phân tích dữ liệu (Data Analysis), máy học (Machine Learning).

Đặc biệt đối với lĩnh vực AI – Deep learning thì sẽ cần đến hàng trăm nghìn phép tính, việc đó đòi hỏi máy phải có cấu hình trung bình đến cao. Để giải quyết vấn đề đó, Google Colab giúp chúng ta chạy thông qua trình duyệt và các tài nguyên cần sử dụng như CPU, GPUs, TPUs sẽ được hệ thống Google Colab cung cấp, được cài đặt sẵn rất nhiều thư viện phổ biến như Pandas, Numpy..., có thể chạy theo từng ô (block) rất thuận tiện cho việc kiểm thử và đánh giá và có thể kết nối trực tiếp với Google Drive.

Hạn chế lớn nhất của hệ thống này sẽ chỉ làm việc liên tục được 12 giờ và với bản Google Colab Pro sẽ là 24 giờ và có thể truy cập RAM lên đến 32 GB. Tóm lại thì Google Colab sẽ phù hợp với những người đang tiếp cận đến lĩnh vực Data, Machine learning và đặc biệt là Deep learning mà chưa đủ kinh phí để đầu tư cấu hình.



Hình 12: Giao diện của Google Colab

3.2 Thư viện Keras

Keras là một thư viện Python mạnh mẽ và dễ sử dụng được xây dựng dựa trên các thư viện học sâu phổ biến như TensorFlow, Theano... để tạo các mô hình học sâu. Cấu trúc tối thiểu, đơn giản, cộng đồng hỗ trợ lớn. Được phát triển vào năm 2005 bởi Francois Chollet, là một kỹ sư nghiên cứu Deep learning.

Thư viện Keras có hai mô hình là Sequential và API function, với bài toán dạng chuỗi nói chung và dạng chuỗi thời gian nói riêng (được sử dụng cho bài toán dự báo thời tiết) sẽ sử dụng mô hình Sequential (mô hình tuần tự) rất hiệu quả.

Thư viện Keras giúp xây dựng một mô hình học sâu (deep learning) nhanh và dễ dàng. Giúp giúp ta tạo một mô hình theo từng tầng một (layer by layer)

Hỗ trợ xây dựng Convolution Neural Network – CNN (mạng nơ ron tích chập) và cả Recurrent Neural Network – RNN

- CNN là chủ yếu dành cho các vấn đề thị giác máy tính, mạnh mẽ trong xử lý ảnh, giúp xác định các đối tượng, vị trí của các đối tượng, quan hệ của các đối tượng trong hình ảnh.

- RNN chủ yếu xử lý dữ liệu dạng chuỗi, có tính tuần tự như chuỗi thời gian.

3.3 Thư viện Pandas

Thư viện Pandas là một thư viện Python cung cấp cấu trúc dữ liệu nhanh, mạnh, linh hoạt, là một thư viện mã nguồn mở được sử dụng rộng rãi bởi khả năng hỗ trợ mạnh mẽ trong các thao tác với dữ liệu. Là một công cụ giúp phân tích và xử lý dữ liệu với ngôn ngữ Python.

Khai báo thư viện sau khi cài đặt một cách dễ dàng: import pandas

Pandas sử dụng một cấu trúc dữ liệu riêng là Dataframe, cung cấp nhiều chức năng xử lý và làm việc trên cấu trúc dữ liệu này.

Là một công cụ cho phép đọc/ghi (read/write) dữ liệu ở nhiều dạng tập tin như csv, text, excel, sql database,... vào dataframe của Pandas.

Dễ dàng xử lý các giá trị bị thiếu, thêm, sửa, xóa với các trường dữ liệu. Thông kê đánh giá dễ dàng tập dữ liệu bằng hàm info() hoặc describe(), tìm kiếm, sắp xếp, truy xuất có điều kiện với các dữ liệu.

	year	month	station	Ta	Tx	Tm	Rf	rH	Sh	aH
0	1960	1	Honggai	NaN	NaN	NaN	NaN	81.0	NaN	NaN
1	1960	2	Honggai	NaN	NaN	NaN	NaN	79.0	NaN	NaN
2	1960	3	Honggai	NaN	NaN	NaN	NaN	90.0	NaN	NaN
3	1960	4	Honggai	NaN	NaN	NaN	NaN	83.0	NaN	NaN
4	1960	5	Honggai	NaN	NaN	NaN	NaN	81.0	NaN	NaN
...
32731	2010	12	Vinh	20.3	24.0	18.0	35.0	83.0	86.0	20.1
32732	2010	12	Vinhyen	19.1	23.0	16.6	29.0	80.0	87.0	17.9
32733	2010	12	Vungtau	26.7	30.4	24.3	1.0	79.0	142.0	27.6
32734	2010	12	Xuanloc	NaN	NaN	NaN	NaN	NaN	NaN	25.4
32735	2010	12	Yenbai	18.0	21.5	15.8	65.0	87.0	61.0	18.2

32736 rows × 10 columns

Hình 13: Một Pandas dataframe

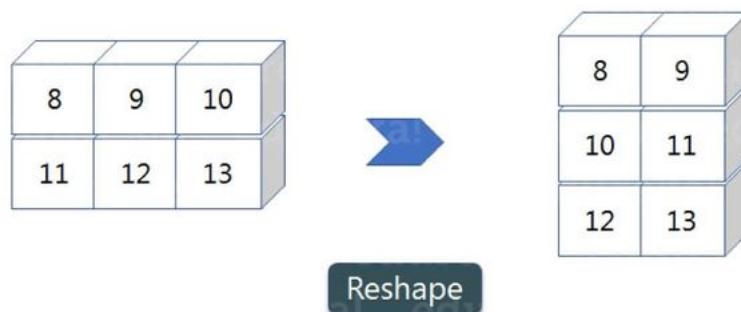
3.4 Thư viện Numpy

Thư viện Numpy là một thư viện mạnh mẽ, được sử dụng rộng rãi của Python. Cho phép người dùng làm việc hiệu quả với các ma trận (matrix) và mảng (array) với tốc độ xử lý nhanh hơn rất nhiều.

Khai báo thư viện sau khi cài đặt: import numpy

Numpy hỗ trợ tính toán trên các mảng nhiều chiều có kích thước lớn bằng các hàm đã được tối ưu đối với các mảng đó. (tính tổng của mảng, tìm giá trị lớn nhất, nhỏ nhất trong mảng theo từng chiều...)

Ví dụ: đổi mảng 2x3 thành mảng 3x2 như hình bên dưới:



Hình 14: Hàm reshape trong Numpy

3.5 Thư viện Sklearn

Sklearn hay Scikit-learn là một thư viện Python mã nguồn mở dành cho lĩnh vực máy học (machine learning). Nó cung cấp các công cụ hiệu quả để học máy, các mô hình phân lớp, hồi quy, gom cụm.

Sklearn được phát triển bởi David Cournapeau trong một dự án mùa hè năm 2007, sau đó được một nhóm nghiên cứu trong Viện nghiên cứu Khoa học máy tính và Tự động hóa của Pháp phát triển thêm và công bố bản phát hành đầu tiên vào đầu năm 2010.

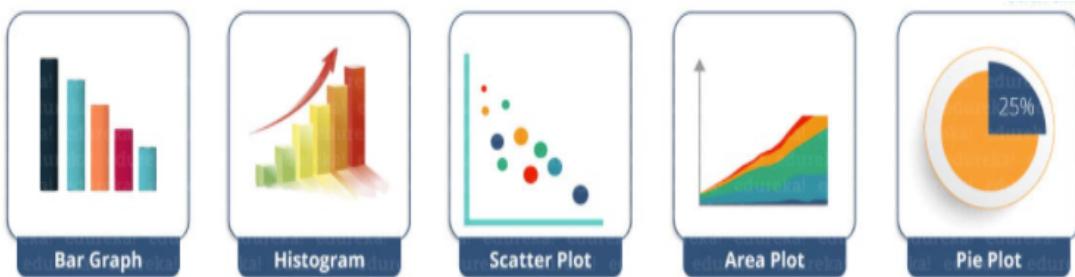
Sklearn hỗ trợ các thuật toán hiện đại như KNN, random forest, SVM, neural network,... Các tập dữ liệu được tích hợp sẵn trong thư viện (iris,...), các phương pháp tiền xử lý dữ liệu như PCA (Principal component analysis), đánh giá mô hình (metrics) và con nhiều công cụ khác.

3.6 Thư viện Matplotlib

Vẽ biểu đồ là một trong những điều cực kỳ cần thiết trong khi làm việc với dữ liệu, đặc biệt là các lập trình hay không phải lập trình viên muốn phát triển trong lĩnh vực Data Analysis hay Data Science. Thì thư viện Matplotlib là một thư viện không thể bỏ qua.

Matplotlib là một thư viện giúp chúng ta trực quan hóa dữ liệu mạnh mẽ, vẽ đồ họa 2D bằng ngôn ngữ Python. Matplotlib cung cấp một module mạnh mẽ để vẽ biểu đồ là Pyplot Có thể vẽ các biểu đồ để có cái nhìn tổng quát về dữ liệu

bằng thư viện này, một số biểu đồ như: bar graph, histogram, scatter plot, area plot, pie plot.



Hình 15: Các dạng biểu đồ trong Matplotlib

Chương II. THIẾT KẾ VÀ CÀI ĐẶT

1. Thiết kế hệ thống

1.1 Đọc tập dữ liệu

Tập dữ liệu ban đầu có phần mở rộng là rdata, chứa thông tin thời tiết của ba quốc gia trong khoảng thời gian quá khứ là Lào, Thái Lan và Việt Nam

Ta chỉ quan tâm đến tập dữ liệu thời tiết của Việt Nam, tập dữ liệu này được ghi lại trong khoảng thời gian từ 1960 đến năm 2010 của Việt Nam ở nhiều tỉnh thành.

Trong đề tài này ta chỉ quan tâm đến bảy thuộc tính của tập dữ liệu này:

- Ta (average temperature in °C): Nhiệt độ trung bình ở độ C
- Tx (maximal temperature in °C): Nhiệt độ cao nhất ở độ C
- Tm (minimal temperature in °C): Nhiệt độ thấp nhất ở độ C
- Rf (rainfall in mm): Lượng mưa đơn vị mm
- rH (relative humidity in %): Độ ẩm tương đối đơn vị %
- Sh (hours of sunshine): Thời lượng có nắng đơn vị giờ
- aH (absolute humidity in mm Hg): Độ ẩm tuyệt đối.

	year	month	station	Ta	Tx	Tm	Rf	rH	Sh	aH
0	1960	1	Honggai	NaN	NaN	NaN	NaN	81.0	NaN	NaN
1	1960	2	Honggai	NaN	NaN	NaN	NaN	79.0	NaN	NaN
2	1960	3	Honggai	NaN	NaN	NaN	NaN	90.0	NaN	NaN
3	1960	4	Honggai	NaN	NaN	NaN	NaN	83.0	NaN	NaN
4	1960	5	Honggai	NaN	NaN	NaN	NaN	81.0	NaN	NaN
...
32731	2010	12	Vinh	20.3	24.0	18.0	35.0	83.0	86.0	20.1
32732	2010	12	Vinhyen	19.1	23.0	16.6	29.0	80.0	87.0	17.9
32733	2010	12	Vungtau	26.7	30.4	24.3	1.0	79.0	142.0	27.6
32734	2010	12	Xuanloc	NaN	NaN	NaN	NaN	NaN	NaN	25.4
32735	2010	12	Yenbai	18.0	21.5	15.8	65.0	87.0	61.0	18.2

32736 rows × 10 columns

Hình 16: Tập dữ liệu thời tiết ban đầu của Việt Nam

Tập dữ liệu thời tiết Việt Nam có 32736 dòng dữ liệu, mỗi dòng đại diện cho một tháng của một tỉnh thành nào đó và có các thuộc tính ghi lại các chỉ số thời tiết.

Để thuận tiện cho việc đọc tập dữ liệu và xử lý ta xuất dữ liệu ra tệp mới với phần mở rộng là csv, với tên là “dulieuthoitiet.csv”

	year	month	station	Ta	Tx	Tm	Rf	rH	Sh	aH
0	1960	1	Honggai	NaN	NaN	NaN	NaN	81.0	NaN	NaN
1	1960	2	Honggai	NaN	NaN	NaN	NaN	79.0	NaN	NaN
2	1960	3	Honggai	NaN	NaN	NaN	NaN	90.0	NaN	NaN
3	1960	4	Honggai	NaN	NaN	NaN	NaN	83.0	NaN	NaN
4	1960	5	Honggai	NaN	NaN	NaN	NaN	81.0	NaN	NaN
...
32731	2010	12	Vinh	20.3	24.0	18.0	35.0	83.0	86.0	20.1
32732	2010	12	Vinhyen	19.1	23.0	16.6	29.0	80.0	87.0	17.9
32733	2010	12	Vungtau	26.7	30.4	24.3	1.0	79.0	142.0	27.6
32734	2010	12	Xuanloc	NaN	NaN	NaN	NaN	NaN	NaN	25.4
32735	2010	12	Yenbai	18.0	21.5	15.8	65.0	87.0	61.0	18.2

32736 rows × 10 columns

Hình 17: Dữ liệu tệp dulieuthoitiet.csv

1.2 Tiên xử lý dữ liệu

Tiến hành đọc tập dữ liệu thời tiết, sau đó biến đổi sao cho dễ xử lý (gộp cột month và year thành cột chỉ mục(index) của dữ liệu. cái nhìn tổng quan, xử lý các giá trị bị rỗng (NaN), chuẩn hóa dữ liệu (rất cần thiết cho mô hình mạng nơ ron). Tạo tập dữ liệu huấn luyện (training set) và tập dữ liệu kiểm tra (testing set), các tập dữ liệu này phải ở dạng mảng ba chiều (3D array) mà mô hình LSTM có thể xử lý được.

1.3 Xây dựng mô hình (thiết kế mạng nơ ron)

Sử dụng mô hình Sequential (mô hình tuần tự) thiết kế một kiến trúc gồm các tầng ẩn (hidden layers) bên trong mô hình, ứng dụng Stacked LSTM để tạo “độ sâu” cho mô hình, làm mô hình phức tạp hơn (giúp đối tượng cần dự đoán “rõ nét” hơn). Việc làm này sẽ giảm thiểu đi việc sử dụng nhiều nơ ron, giúp mô hình chạy nhanh và chính xác hơn và trong một số nghiên cứu thì “độ sâu” của mạng sẽ quan trọng hơn số lượng nơron.

Mô hình LSTM hoạt động trên dữ liệu tuần tự (sequence data), việc bổ sung thêm các tầng ẩn sẽ làm tăng thêm mức độ trừu tượng hóa của các quan sát đầu vào theo thời gian.

2. Cài đặt hệ thống

2.1 Đọc tập dữ liệu

Sử dụng pyreadr một gói của Python cho phép đọc và ghi các tệp RData vào hoặc từ khung dữ liệu (dataframe) của thư viện Pandas

Sử dụng hàm read_r và truyền vào đường dẫn của tệp có phần mở rộng là rdata.

Vì tập dữ liệu này chứa dữ liệu thời tiết của ba quốc gia, nên ta sử dụng hàm keys() để kiểm tra ta được: odict_keys(['data_laos', 'data_thailand', 'data_vietnam', 'stations_laos', 'stations_vietnam']), trong đề tài này ta sẽ làm việc với dữ liệu thời tiết của Việt Nam (data_vietnam) và lưu lại thành tệp csv.

2.1.1 Thông tin cơ bản của tập dữ liệu.

Sử dụng hàm info() để xem thông tin của dataframe (sau khi xử lý gộp cột year và month làm chỉ mục):

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 32736 entries, 1960-01-01 to 2010-12-01
Data columns (total 8 columns):
 #   Column   Non-Null Count  Dtype  
--- 
 0   station   32736 non-null   object 
 1   Ta         31656 non-null   float64
 2   Tx         31355 non-null   float64
 3   Tm         31356 non-null   float64
 4   Rf         31752 non-null   float64
 5   rH         31596 non-null   float64
 6   Sh         30840 non-null   float64
 7   aH         31932 non-null   float64
dtypes: float64(7), object(1)
memory usage: 2.2+ MB
```

Hình 18: Thông tin của tập dữ liệu

Tập dữ liệu cho thấy có 32736 quan sát, có chỉ mục từ 1960-01-01 đến 2010-12-01, tập dữ liệu gồm có tám cột, trong bảng thông tin trên còn cho biết đối kiểu dữ liệu (Dtype) và số giá trị không rỗng (Non-Null Count) của từng cột.

2.1.2 Thông kê cơ bản về dữ liệu.

Sử dụng hàm describe() trong thư viện pandas để tạo một bản thống kê dữ liệu, giúp ta có cái nhìn tổng quan ban đầu về dữ liệu, qua thống kê ta thấy độ lệch chuẩn (std) của Rf (lượng mưa) và Sh (thời lượng có nắng) khá cao, có thể sẽ ảnh hưởng đến mô hình chúng ta xây dựng.

	Ta	Tx	Tm	Rf	rH	Sh	aH
count	31656.000000	31355.000000	31356.000000	31752.000000	31596.000000	30840.000000	31932.000000
mean	24.136056	28.399084	21.296563	157.060837	82.946891	160.728402	25.417506
std	4.352241	4.679795	4.352521	176.779286	5.263266	69.031249	5.832405
min	2.500000	0.000000	0.000000	0.000000	49.000000	0.000000	2.900000
25%	21.000000	25.200000	18.200000	27.600000	80.000000	113.700000	20.700000
50%	25.500000	30.000000	22.600000	103.000000	84.000000	163.600000	26.700000
75%	27.500000	31.900000	24.700000	234.000000	86.810000	207.825000	30.300000
max	35.800000	39.300000	39.000000	2451.700000	99.000000	674.000000	39.900000

Hình 19: Thống kê tập dữ liệu ban đầu

- Bảng dữ liệu cho biết các số liệu theo từng cột như:
- Count: số lượng giá trị không rỗng.
- Mean: giá trị trung bình.
- Std (standard deviation): độ lệch chuẩn của giá trị.
- Min: giá trị nhỏ nhất.
- 25%, 50%, 75%: các khoảng giá trị.
- Max: giá trị lớn nhất.

2.1.3 Kiểm tra các giá trị rỗng

Sử dụng hàm isna() để kiểm tra xem liệu rằng trong ô đó có giá trị hay không, nếu không có giá trị thì trả về TRUE, ngược lại thì FALSE. Sau đó kết hợp với hàm sum() để đếm tổng số giá trị TRUE của mỗi cột dữ liệu ta được số lượng các giá trị rỗng:

Columns	Null values count
station	0
Ta	1080
Tx	1381
Tm	1380
Rf	984
rH	1140
Sh	1896
aH	804

Bảng 4: Thống kê giá trị Null

2.2 Tiền xử lý dữ liệu

2.2.1 Thay đổi chỉ mục cho tập dữ liệu.

Nhận thấy dữ liệu có hai cột là year và month quá rườm rà, ta tiến hành gộp hai cột này thành một và lấy chúng làm chỉ mục (index) cho tập dữ liệu. Ta sử dụng hàm `strptime(string, format code)` tạo đối tượng thời gian (datetime object) từ chuỗi đã cho, nhận vào hai đối số: string, format code:

```
date_string = "21 June, 2018"
...
date_object = datetime.strptime(date_string, "%d %B, %Y")
```

Hình 20: Hàm `strptime()` trong `datetime`

2.2.2 Xử lý các giá trị bị thiếu (missing values)

Nhận thấy các giá trị NaN không đáng kể nên ta tiến hành xóa các dòng nào có ghi nhận giá trị NaN bằng hàm `dropna()` của thư viện Pandas với `inplace = True` để thao tác trực tiếp với `dataframe`. Sau khi xóa các giá trị NaN thì dữ liệu sẽ còn lại 30479 dòng so với 32736 ban đầu.

2.2.3 Xử lý bằng Label Encoder

Chúng ta tiến hành thêm một cột là “station_encoded” là dữ liệu mã hóa cột dữ liệu “station” về dạng số bằng hàm `LabelEncoder()`, hàm này mã hóa dữ liệu về dữ liệu số có giá trị từ 0 đến n (n là số lớp của dữ liệu):

	station	Ta	Tx	Tm	Rf	rH	Sh	aH	station_encoded
date									
1961-01-01	Bacgiang	15.7	19.7	12.7	2.7	74.0	102.4	13.6	2
1961-01-01	Hanam	15.9	19.6	13.7	10.2	82.0	94.2	15.3	22
1961-01-01	Hatinh	17.0	19.9	14.9	76.0	90.0	77.3	17.8	23
1961-01-01	Hoabinh	15.3	20.2	11.7	2.3	81.0	89.8	14.2	24
1961-01-01	Honggai	15.3	19.6	12.6	2.7	74.0	117.1	13.4	25
...
2010-12-01	Viettri	18.9	22.2	16.5	25.0	80.0	78.0	17.7	59
2010-12-01	Vinh	20.3	24.0	18.0	35.0	83.0	86.0	20.1	60
2010-12-01	Vinhuyen	19.1	23.0	16.6	29.0	80.0	87.0	17.9	61
2010-12-01	Vungtau	26.7	30.4	24.3	1.0	79.0	142.0	27.6	62
2010-12-01	Yenbai	18.0	21.5	15.8	65.0	87.0	61.0	18.2	64

30479 rows × 9 columns

Hình 21: Thêm cột mã hóa của cột station

Sử dụng hàm `values_count()` để thống kê dữ liệu của mỗi “station” xem có đủ số tháng từ năm 1961 đến năm 2010 không:

month_count	
station	
Bacgiang	600
Phulien	600
Yenbai	600
Vinhuyen	600
Vinh	600
...	...
Dongphu	360
Hue	360
Phuoclong	348
Xuanloc	168
Bacninh	132

65 rows x 1 columns

Hình 22: Thống kê số tháng của station

Thông kê trên cho thấy dữ liệu ở các địa điểm (station) có thể bị thiếu do quá trình ghi nhận và xử lý giá trị NaN phục vụ cho mô hình, có 17 địa điểm là đủ 600 tháng (50 năm) và 48 địa điểm không ghi nhận đủ 50 năm. Sẽ sử dụng dữ liệu của Cần Thơ với ghi nhận là 384 tháng để đánh giá trước.

2.2.4 Xóa các cột không cần thiết

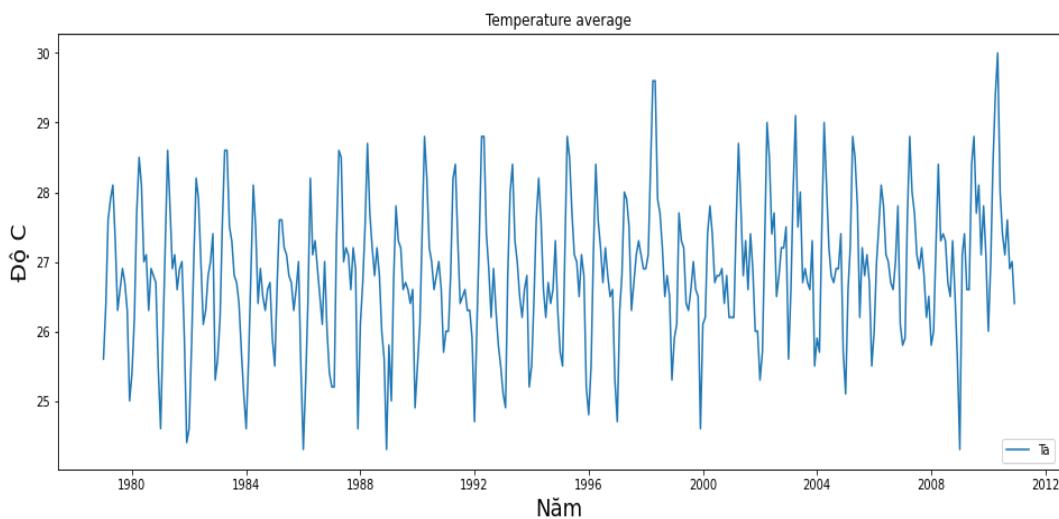
Sau khi sử dụng các cột station để để thống kê và phân tích, ta tiến hành xóa các cột bằng hàm drop() với inplace = True. Dữ liệu lúc này chỉ còn 7 thuộc tính dùng để huấn luyện mô hình và index là chỉ thời gian ghi nhận của các thuộc tính đó.

Ta	Tx	Tm	Rf	rH	Sh	aH
date						
1979-01-01	25.6	30.2	22.3	0.0	78.0	288.7
1979-02-01	26.4	31.4	22.7	0.0	76.0	265.5
1979-03-01	27.6	32.9	23.6	0.1	77.0	284.8
1979-04-01	27.9	32.9	24.5	117.9	83.0	238.0
1979-05-01	28.1	32.5	25.0	87.1	85.0	202.0

Hình 23: Tập dữ liệu Cần Thơ sau khi xóa các cột không cần thiết

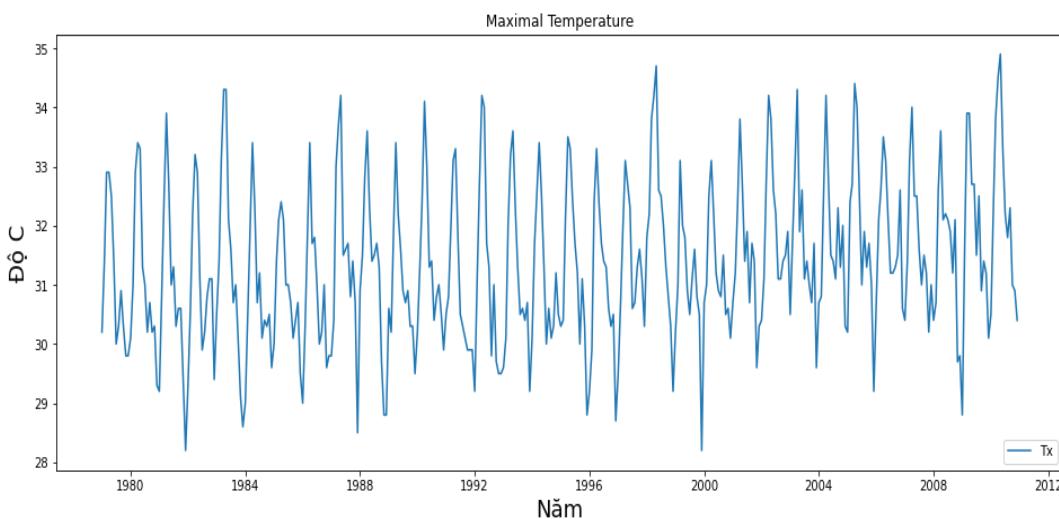
2.2.5 Hiển thị dữ liệu.

2.2.5.1 Biểu đồ Ta (nhiệt độ trung bình-độ C)



Hình 24: Biểu đồ nhiệt độ trung bình

2.2.5.2 Biểu đồ Tx (biểu đồ nhiệt độ cao nhất-độ C)



Hình 25: Biểu đồ nhiệt độ cao nhất

2.2.6 Scale dữ liệu

Scale dữ liệu là một thao tác rất cần thiết trong các mô hình mạng nơron, vì trong dữ liệu có thể các thuộc tính chênh lệch nhau rất nhiều, ví dụ một thuộc tính nằm trong khoảng 0 đến 1000 còn một cột có giá trị trong khoảng 0 đến 1 thì sẽ làm cho quá trình huấn luyện không ổn định cho ra mô hình có hiệu quả không cao.

Ta tiến hành scale dữ liệu bằng hàm MinMaxScaler() của thư viện sklearn về trong khoảng 0 tới 1 với feature_range=(0, 1), ta được:

```
array([[0.22807018, 0.29850746, 0.37142857, ..., 0.33333333, 0.42833828,
       0.10638298],
      [0.36842105, 0.47761194, 0.42857143, ..., 0.23809524, 0.39391691,
       0.19148936],
      [0.57894737, 0.70149254, 0.55714286, ..., 0.28571429, 0.42255193,
       0.41489362],
      ...,
      [0.45614035, 0.41791045, 0.64285714, ..., 0.71428571, 0.24035608,
       0.61702128],
      [0.47368421, 0.40298507, 0.64285714, ..., 0.66666667, 0.27002967,
       0.56382979],
      [0.36842105, 0.32835821, 0.51428571, ..., 0.52380952, 0.25074184,
       0.39361702]])
```

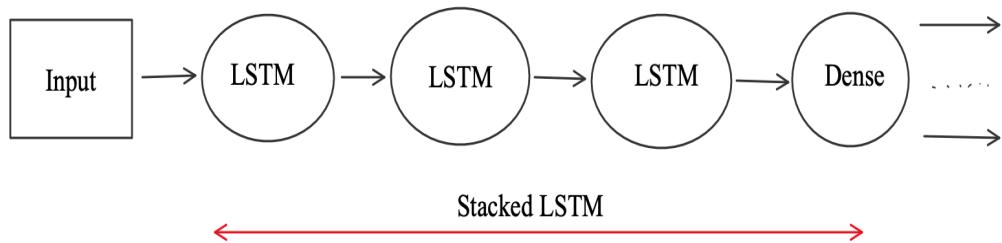
Hình 26: Dữ liệu sau khi đã scale

2.3 Xây dựng mô hình (model).

2.3.1 Thiết kế mạng

Khởi tạo model bằng hàm Sequential() để tạo mô hình tuần tự từng tầng một cùng với mô hình Stacked LSTM (xếp chồng các tầng LSTM lại với nhau như giới thiệu ở phần trên)

Mỗi tầng mạng sẽ đòi hỏi một kích thước đầu vào khác nhau, ở đây ta sử dụng hai tầng mạng là LSTM và Dense, có kiến trúc như sau: [input] LSTM (64 noron) => LSTM (32 noron) => LSTM (16 noron) => Dense (7 noron) [output]

*Hình 27: Mô hình Stacked LSTM*

Chú ý kiến trúc Stacked LSTM này thì hai tầng LSTM đầu tiên return_sequential phải là True để có thể tiếp tục tạo ra mảng ba chiều làm đầu vào cho tầng LSTM tiếp theo. LSTM đòi hỏi đầu vào là một mảng ba chiều (3D array) có dạng (samples, time_steps, features) nên các đầu vào của mô hình (x_train và x_test) phải ở dạng mảng ba chiều. Đầu vào của tầng LSTM đầu tiên sẽ được xác định bằng input_shape(time_steps, features). LSTM cần xử lý các samples, mỗi samples là một chuỗi thời gian. Tầng Dense cuối cùng yêu cầu đầu vào là một mảng hai chiều (2D array), nên tầng LSTM trước đó sẽ return_sequential =

Flase, đầu ra của tầng Dense này sẽ là bảy noron tương ứng với 7 thuộc tính cần dự đoán.

2.3.2 Tạo tập dữ liệu huấn luyện

Từ tập dữ liệu thời tiết của Cần Thơ, ta dùng 80% dữ liệu làm tập huấn luyện (training) và 20% dữ liệu còn lại làm tập kiểm tra (testing).

Ta có 80% dữ liệu sẽ ứng với 307 dòng, và 20% còn lại sẽ là 77 dòng.

Vì mô hình dự báo thời tiết là mô hình học có giám sát, thuật toán ta sử dụng là LSTM nên ta sẽ dùng 24 tháng trước để dự đoán cho tháng 25 liền kề tương ứng với x_{train} và y_{train} . Sử dụng vòng lặp từ 24 đến 307, trong từng vòng lặp sử dụng hàm append để thêm dữ liệu vào từng list, sau đó chuyển về dạng numpy array, với sự mạnh mẽ của thư viện Numpy nó giúp ta tự động chuyển về mảng ba chiều. Lúc này dữ liệu của bộ dữ liệu huấn luyện chỉ còn 283 bộ.

Bộ dữ liệu huấn luyện: x_{train} : (283, 24, 7), y_{train} (283, 7)

```
[4.91228070e-01, 4.17910448e-01, 7.14285714e-01, 2.10707767e-01,
 8.09523810e-01, 2.61721068e-01, 7.55319149e-01],
[3.50877193e-01, 2.98507463e-01, 6.28571429e-01, 4.43723383e-01,
 8.57142857e-01, 2.37388724e-01, 6.48936170e-01],
[4.56140351e-01, 3.73134328e-01, 6.71428571e-01, 3.71932671e-01,
 7.61904762e-01, 2.54154303e-01, 7.34042553e-01],
[4.38596491e-01, 2.98507463e-01, 6.85714286e-01, 6.91340499e-01,
 8.09523810e-01, 2.18249258e-01, 7.44680851e-01],
[4.21052632e-01, 3.13432836e-01, 6.71428571e-01, 3.35226120e-01,
 6.66666667e-01, 2.82492582e-01, 5.95744681e-01],
[2.10526316e-01, 1.64179104e-01, 4.42857143e-01, 8.61894139e-02,
 5.71428571e-01, 3.12611276e-01, 3.29787234e-01],
[5.26315789e-02, 1.49253731e-01, 1.85714286e-01, 5.88116001e-03,
 3.80952381e-01, 3.55637982e-01, 4.25531915e-02]]]
y_train:
[array([0.05263158, 0.14925373, 0.18571429, 0.00588116, 0.38095238,
       0.35563798, 0.04255319]), array([0.29824561, 0.40298507, 0.41428571, 0. , 0.38095238,
       0.37477745, 0.19148936])]
(283, 24, 7)
(283, 24, 7)
(283, 7)
```

Hình 28: Dữ liệu minh họa x_{train} và y_{train}

2.3.3 Tập dữ liệu kiểm tra.

Tương tự với dữ liệu của tập huấn luyện ta tiến hành tạo tập dữ liệu kiểm tra và sẽ lấy lùi về sau 24 dữ liệu để dự đoán cho tháng đầu tiên của tập dữ liệu kiểm tra và chú ý phải chuyển dữ liệu về mảng Numpy tương đương với mảng ba chiều để phù hợp với mô hình.

2.3.4 Biên dịch mô hình và huấn luyện mô hình

Biên dịch mô hình với thuật toán để tối ưu hóa mô hình: bộ tối ưu hóa (optimizer) là Adam – một dạng của SGD (Stochastic Gradient Descent) và hàm mất mát (loss function) là RMSE (Root Mean Squared Error), huấn luyện mô hình với batch_size = 1 và epochs = 50. Kết quả của quá trình huấn luyện khá tốt trên tập dữ liệu huấn luyện, tức cho ra loss = 0.0078 ở epochs thứ 50. Mô hình có

tổng cộng 34103 tham số và tất cả 34103 tham số này đều được dùng để huấn luyện mô hình.

```

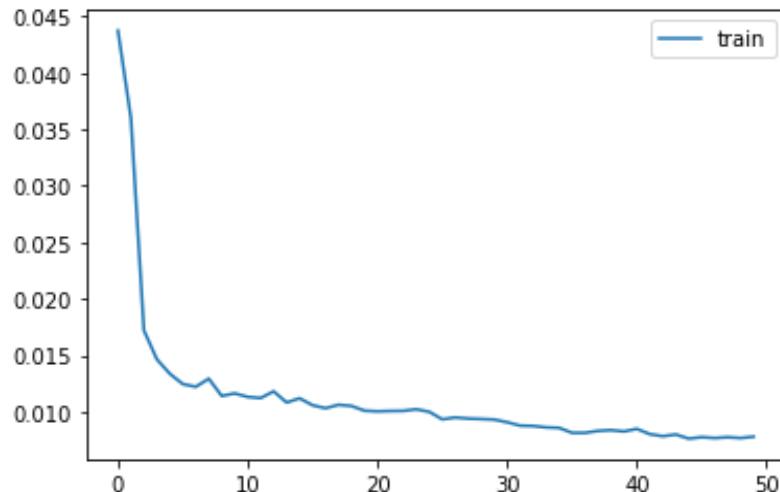
Epoch 48/50
283/283 [=====] - 5s 19ms/step - loss: 0.0078
Epoch 49/50
283/283 [=====] - 5s 19ms/step - loss: 0.0077
Epoch 50/50
283/283 [=====] - 5s 19ms/step - loss: 0.0078
Model: "sequential_17"



| Layer (type)             | Output Shape   | Param # |
|--------------------------|----------------|---------|
| <hr/>                    |                |         |
| lstm_51 (LSTM)           | (None, 24, 64) | 18432   |
| lstm_52 (LSTM)           | (None, 24, 32) | 12416   |
| lstm_53 (LSTM)           | (None, 16)     | 3136    |
| dense_17 (Dense)         | (None, 7)      | 119     |
| <hr/>                    |                |         |
| Total params: 34,103     |                |         |
| Trainable params: 34,103 |                |         |
| Non-trainable params: 0  |                |         |


```

Hình 29: Kết quả của quá trình huấn luyện



Hình 30: Loss của model

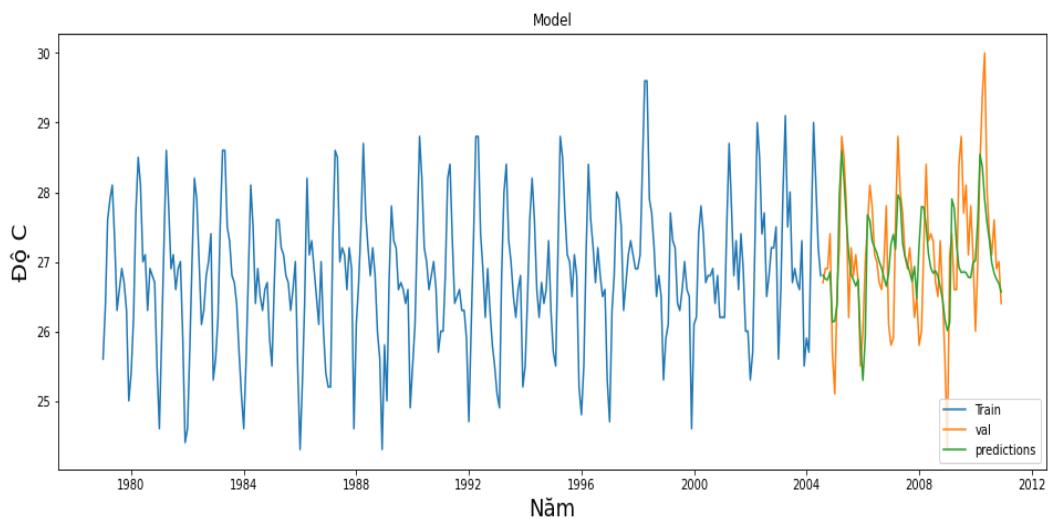
2.4 Dự đoán thời tiết Cần Thơ với tập dữ liệu kiểm tra dựa trên mô hình đã xây dựng

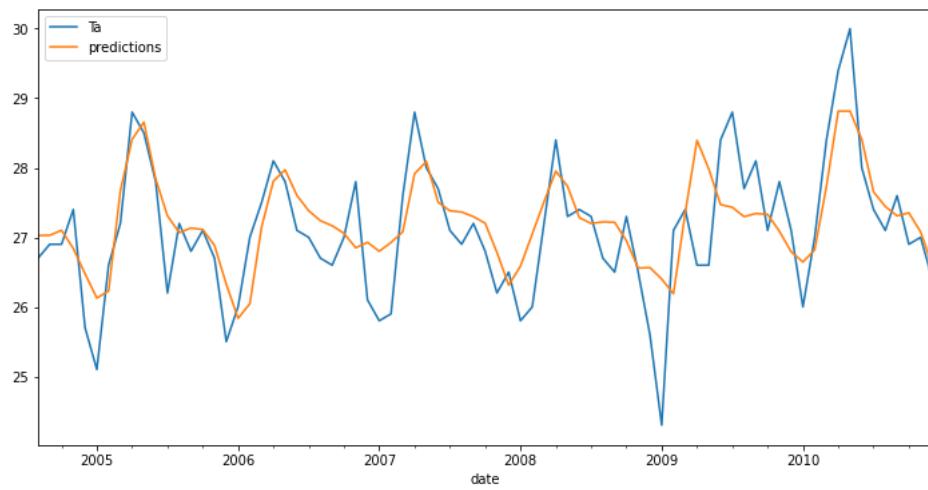
Attribute	RMSE	MAE
Ta	0.6820195674395932	0.5068000050334189
Tx	0.8564067502419317	0.6827068824272651

Tm	0.49036516061814206	0.37873618386008523
Rf	67.53338178414319	51.27342019383009
rH	3.2687604855167676	2.4607227315531146
Sh	114.97666349530546	73.52516249619521
aH	0.9395616201391798	0.6787397409414317
Loss	0,0078	

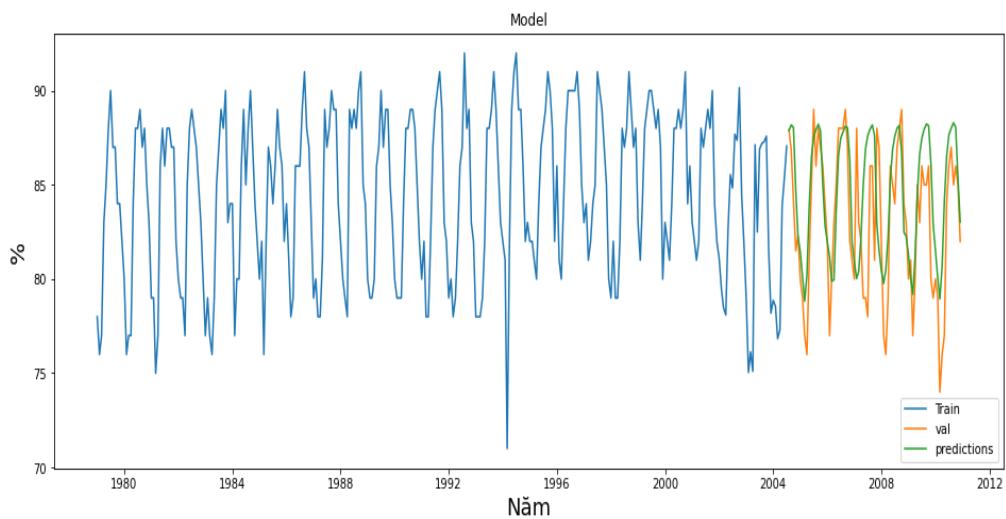
Bảng 5: RMSE và MAE của mô hình dự đoán cho Cần Thơ

Kết quả cho thấy mô hình dự đoán rất tốt cho các thuộc tính Ta, Tx, Tm, rH và aH. Đối với hai thuộc tính Rf và Sh lại cho kết quả không cao do hai chỉ số thời tiết này có độ lệch chuẩn quá lớn \Leftrightarrow độ biến thiên cao, dẫn đến mô hình đã dự báo không tốt cho hai chỉ số này.

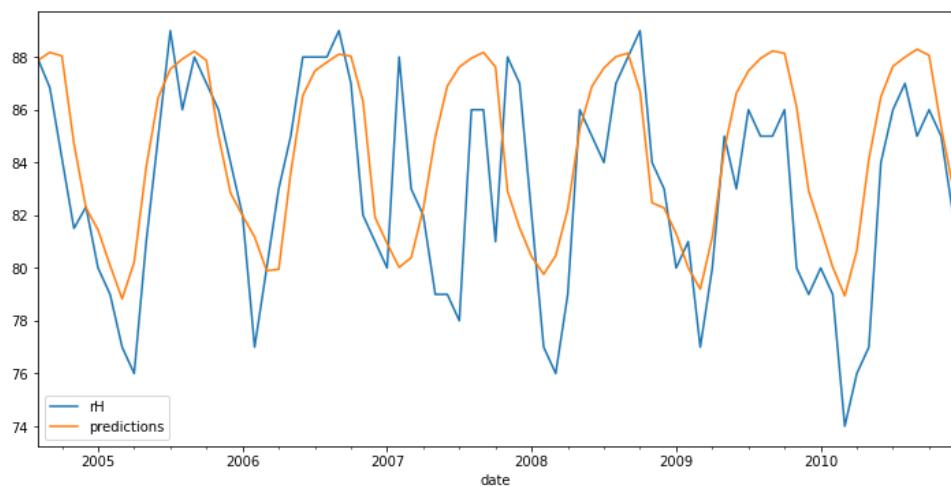
*Hình 31: Biểu diễn giá trị Nhiệt độ Trung bình huấn luyện, giá trị dự đoán và thực tế*



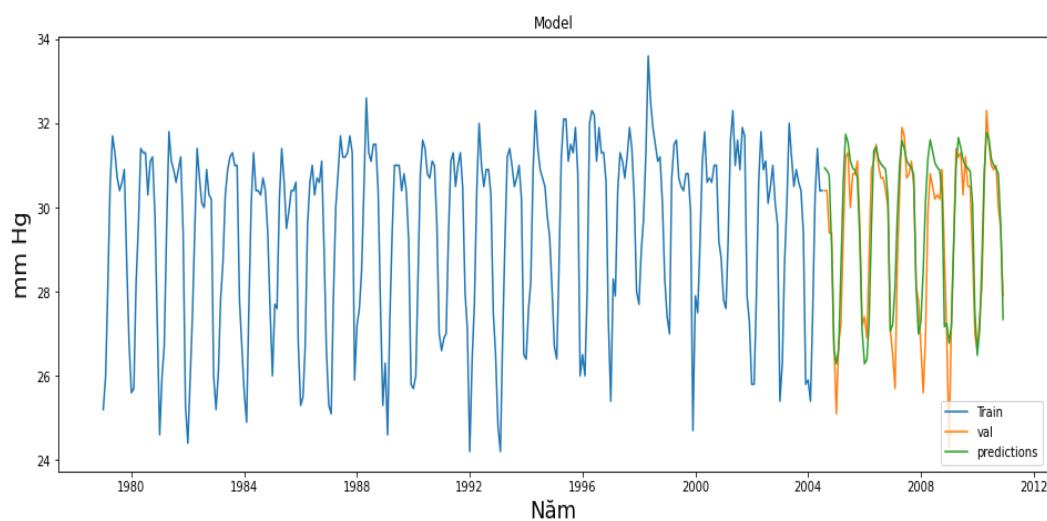
Hình 32: Biểu diễn giá trị Nhiệt độ Trung bình dự đoán và thực tế



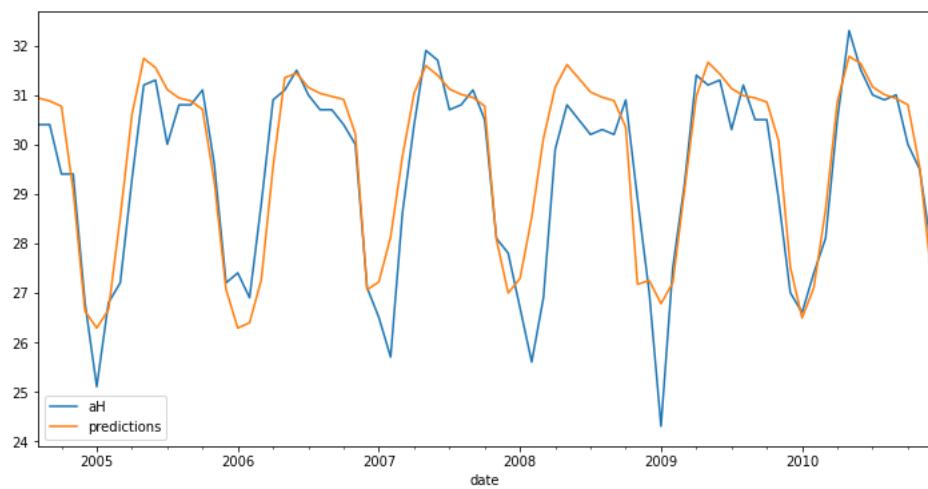
Hình 33: Biểu diễn giá trị Độ ẩm tương đối huấn luyện, giá trị dự đoán và thực tế



Hình 34: Biểu diễn giá trị Độ ẩm tương đối dự đoán và thực tế



Hình 35: Biểu diễn giá trị Độ ẩm tuyệt đối huấn luyện, giá trị dự đoán và thực tế



Hình 36: Biểu diễn giá trị Độ ẩm tuyệt đối dự đoán và thực tế

Chương III. KẾT QUẢ THỰC NGHIỆM

1. Kiểm thử

1.1 Trường hợp kiểm thử 1: Scale bằng hàm RobustScaler vs Min-Max Scaler (Cần Thơ)

Attrib ute	RobustScaler		Min-Max Scaler	
	RMSE	MAE	RMSE	MAE
Ta	0.80055392750 29574	0.61191522177 16366	0.682019567439 5932	0.506800005033 4189
Tx	1.05537503878 34868	0.84275327905 43197	0.856406750241 9317	0.682706882427 2651
Tm	0.77324629607 58721	0.53848009976 47372	0.490365160618 14206	0.378736183860 08523
Rf	105.095678875 38918	81.2790336509 8037	67.53338178414 319	51.27342019383 009
rH	4.05706108619 72005	3.09640741521 6619	3.268760485516 7676	2.460722731553 1146
Sh	112.780986577 6253	75.4327329957 64	114.9766634953 0546	73.52516249619 521
aH	1.30327139023 49096	0.91686494505 25062	0.939561620139 1798	0.678739740941 4317
Loss (50 epochs)	0.0633		0.0078	

Bảng 6: Kết quả kiểm thử thay đổi phương pháp Scale

1.2 Trường hợp kiểm thử 2: Tăng số nơron ở tầng LSTM

Attrib ute	256-128-64-7		128-64-32-7	
	RMSE	MAE	RMSE	MAE
Ta	0.80846155117 44739	0.64113709090 59203	0.639119951051 3551	0.462619563511 43974

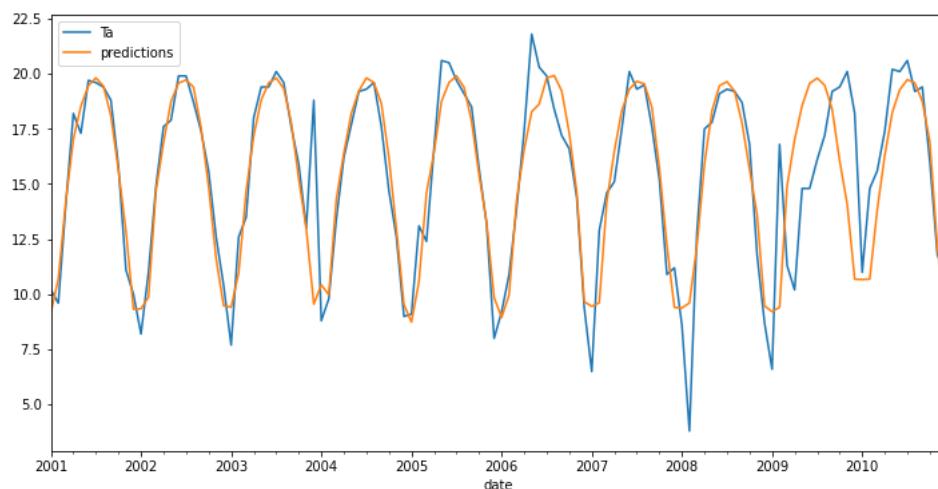
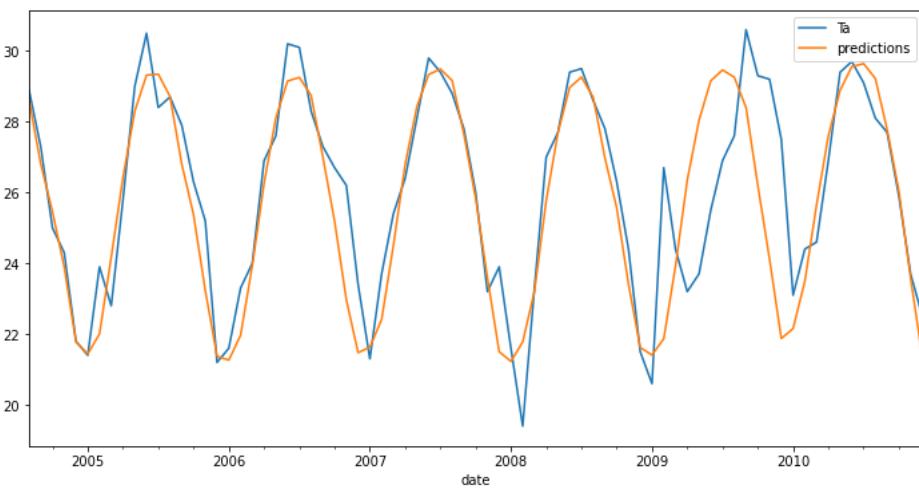
Tx	0.95939270302 09705	0.78518884832 2088	0.742306215607 0208	0.582739396528 764
Tm	0.65874896528 5054	0.49486336150 72672	0.481242961449 01113	0.361614361057 0337
Rf	90.1983123254 8658	68.5609094378 236	65.07490345473 119	47.37608209616 178
rH	3.35115814547 76502	2.49566238601 4864	2.813973656383 875	2.002885540058 0863
Sh	122.156254657 75132	79.4842832094 6631	121.7413191742 295	78.07914829006 442
aH	0.93747223338 46823	0.71341735542 5946	0.860335807270 8774	0.641740734546 2156
Loss (50 epochs)	0.0072		0.0078	

Bảng 7: Kết quả kiểm thử thay đổi số noron

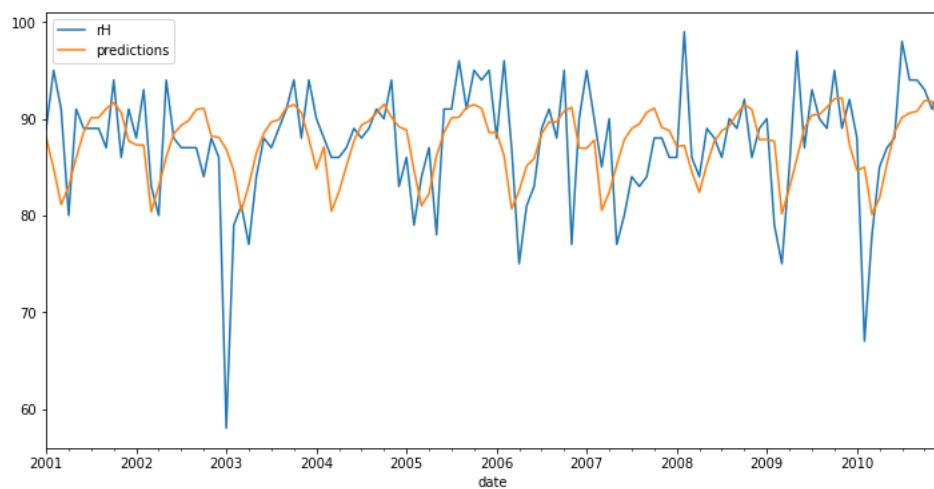
1.3 Trường hợp kiểm thử 3: Thủ mới Station Sapa và Đà Nẵng

Attrib ute	Sapa		Đà Nẵng	
	RMSE	MAE	RMSE	MAE
Ta	2.13277122267 75692	1.34169998168 9453	1.64950300218 13497	1.09763374824 02851
Tx	1.59047166540 1553	1.11159315903 98154	1.20617246870 66815	0.95506023555 60694
Tm	2.26839166748 3213	1.38086248079 93573	4.28717725119 3953	2.32569189690 92433
Rf	107.717390674 81766	81.5522211647 0337	195.540433879 5344	111.508073633 36736
rH	5.49792040382 4476	3.92279624938 96484	3.11985689519 97365	2.32404528332 99514
Sh	51.4942612551 6092	30.9220969645 1823	40.6630364727 855	32.2770713112 571

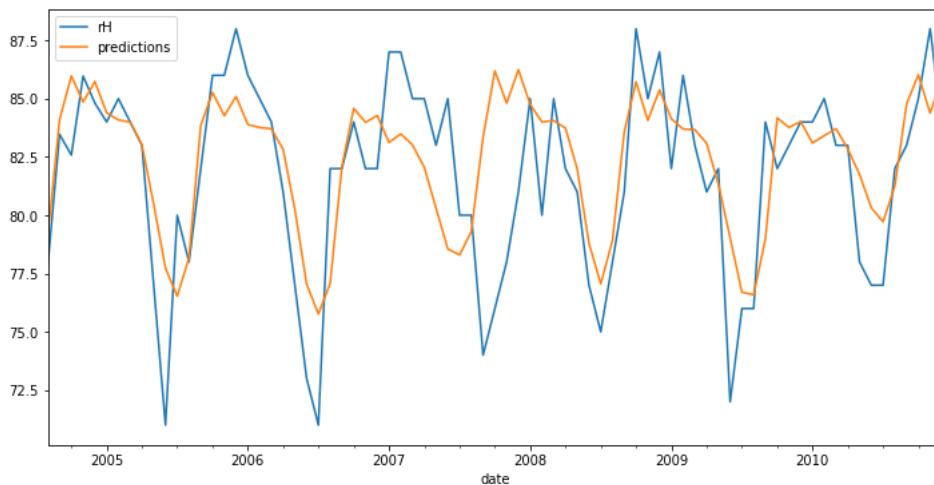
aH	1.05471346260 85792	0.83637183984 12069	1.39945978997 80283	1.06255179566 2224
Loss (50 epochs)	0.0065 (7 phút)			0.0079 (4 phút)

*Bảng 8: Kết quả kiểm thử với Station khác***1.3.1 Biểu diễn dự đoán Ta (average temperature):***Hình 37: Biểu diễn dự đoán Ta của Sapa**Hình 38: Biểu diễn dự đoán Ta của Đà Nẵng*

1.3.2 Biểu diễn dự đoán rH (relative humidity)

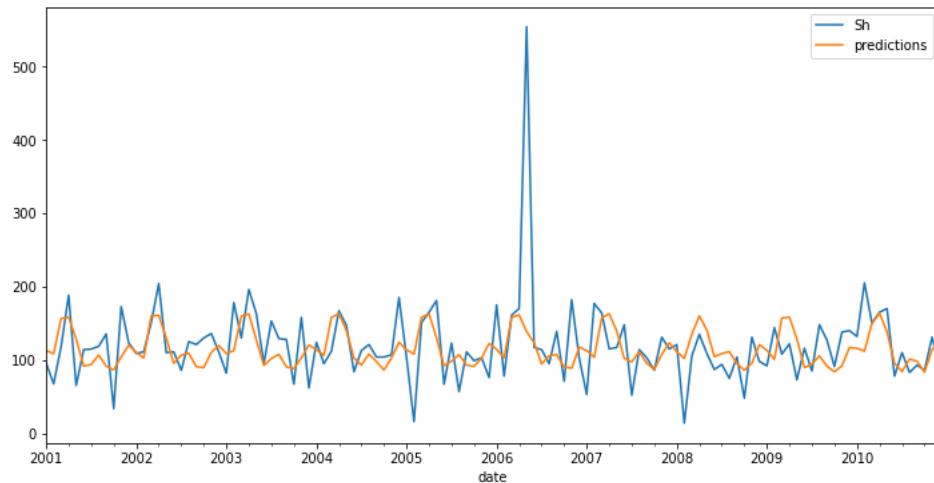


Hình 39: Biểu diễn dự đoán rH của Sapa

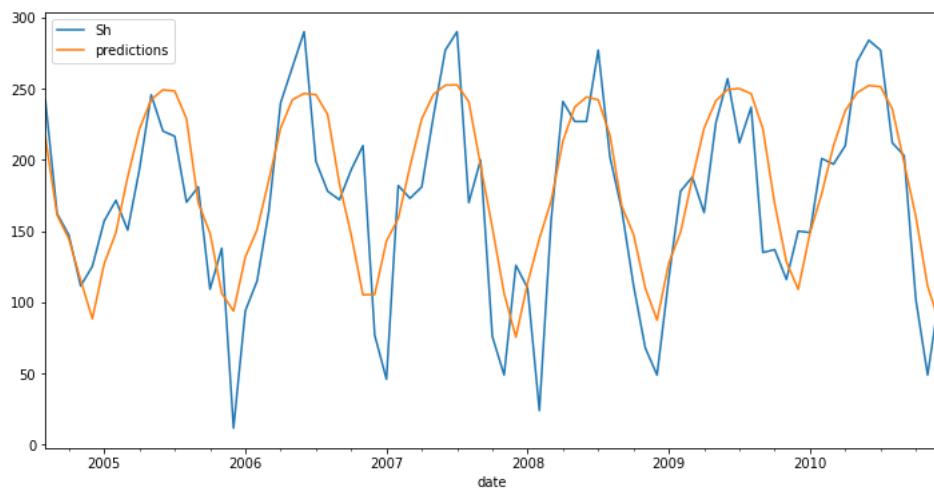


Hình 40: Biểu diễn dự đoán rH của Đà Nẵng

1.3.3 Biểu diễn dự đoán Sh (hours of sunshine)



Hình 41: Biểu diễn dự đoán Sh của Sapa



Hình 42: Biểu diễn dự đoán Sh của Đà Nẵng

1.4 Trường hợp kiểm thử 4: với timestep = 24 (Cần Thơ)

Attribute	Time_step=12		Time_step=24		Time_step=36	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
Ta	0.800553 92750295 74	0.611915 22177163 66	0.640473 91583770 26	0.4674148 46048726 8	0.6645146 4299506 26	0.517505 83896389 26
Tx	1.055375 03878348	0.842753 27905431	0.776088 37856389	0.6191423 88678216	0.8204805 29100291	0.665071 76981343

	68	97	66	2	6	85
Tm	0.773246	0.538480	0.510991	0.3936946	0.4994966	0.399318
	29607587	09976473	92320668	22485668	84660310	76938064
	21	72	68	56	16	37
Rf	105.0956	81.27903	56.91604	42.090163	71.384651	53.98196
	78875389	36509803	58494148	06837658	96030575	90803428
	18	7	3			7
rH	4.057061	3.096407	3.396531	2.5738469	3.6995220	2.894262
	08619720	41521661	79917499	26998782	36890844	64378931
	05	9	37	3	6	6
Sh	112.7809	75.43273	113.5751	72.758633	115.14351	75.07292
	86577625	2995764	89197412	66065086	74880586	42894556
	3		31		8	6
aH	1.303271	0.916864	0.870583	0.6459246	0.8989355	0.704344
	39023490	94505250	68789370	84598848	88191733	19706270
	96	62	37	4	6	31
Loss	0.0079		0.0078		0.0078	

Bảng 9: Kết quả kiểm thử thay đổi time_step

2. Đánh giá

- Đối với kiểm thử tăng số noron sẽ làm cho model huấn luyện lâu hơn (với mô hình 256-128-64-7 thời gian huấn luyện với 50 epoch sẽ mất 9 phút 39 giây) mà kết quả cũng không thay đổi nhiều.
- Trường hợp kiểm thử 1: thay đổi phương pháp scale, kết quả cho thấy phương pháp scale bằng Min-Max Scaler sẽ cho kết quả tốt hơn khá rõ rệt.
- Thay đổi time_step sẽ cho kết quả của các chỉ số đánh giá có thay đổi tốt ở time_step bằng 24, cụ thể là ở chỉ số thời tiết Rf cho ra RMSE khoảng 57. Và ở các chỉ số khác cũng cho chỉ số RMSE tốt hơn so với 2 trường hợp time_step = 12 và time_step = 36
- Thay đổi các địa điểm dự đoán sẽ cho kết quả khác nhau vì mỗi tỉnh thành sẽ có thời tiết khác nhau nhưng nhìn chung các tỉnh thành đều có độ lệch chuẩn cao ở Rf (Rainfall) và Sh (Hours of sunshine). Trong 3 địa điểm dự đoán trong đề tài này thì Đà Nẵng có độ lệch chuẩn của Rf cao nhất (249.152402) và thấp nhất là Cần Thơ (119.470364), và độ lệch chuẩn của Sh ở Sapa sẽ là thấp nhất (44.285443), Cần Thơ và Đà Nẵng thì tương đương nhau (khoảng 66)

PHẦN KẾT LUẬN

1. Kết quả đạt được

1.1 Kỹ năng

- Chủ động sắp xếp thời gian để xây dựng đề tài đúng yêu cầu đề ra hằng tuần.
- Tham khảo được nhiều bài báo khoa học, bài luận trong nước và ngoài nước trong quá trình tìm kiếm tài liệu.
- Hiểu được cách hoạt động của mô hình RNN, cũng như LSTM
- Nâng cao khả năng tự học, tự tìm kiếm tài liệu, phân tích tài liệu, cũng như định hướng được mục tiêu sẽ làm cho chủ đề.

1.2 Chương trình

- Hiểu và xây dựng được mô hình mạng nơ ron.
- Khả năng phân tích dữ liệu tốt hơn.
- Ứng dụng được các kiến thức đã học ở môn Khai khoáng dữ liệu và Nguyên lý máy học.

2. Hướng phát triển

- Xây dựng xử lý các dữ liệu có độ lệch chuẩn lớn.
- Xây dựng thêm ứng dụng (ứng dụng web) để demo mô hình
- Thay đổi thiết kế mạng khác: thêm nhiều tầng (layers) hơn thay đổi số neuron khác nhau.
- Tìm hiểu và ứng dụng các pre-trained model.

TÀI LIỆU THAM KHẢO

- [1] Wikipedia – Bách khoa toàn thư mở. Bộ nhớ dài-ngắn hạn
- [2] Jason Brownlee (2017), Multivariate Time Series Forecasting with LSTMs in Keras, Deep Learning for Time Series,
<https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/>
- [3] Jason Brownlee (2017), How to Convert a Time Series to a Supervised Learning Problem in Python, Deep Learning for Time Series,
<https://machinelearningmastery.com/convert-time-series-supervised-learning-problem-python/>
- [4] Quy Nguyen (2021), 11. Các phương pháp scale dữ liệu trong machine learning, <https://ndquy.github.io/posts/cac-phuong-phap-scaling/>
- [5] To Duc Thang (2020), Làm quen với Keras, <https://viblo.asia/p/lam-quen-voi-keras-gJ59mxJ5X2>
- [6] FuchsMichaelAndi (2019), Feature Scaling with Scikit-Learn,
<https://michael-fuchs-python.netlify.app/2019/08/31/feature-scaling-with-scikit-learn/#normalize-or-standardize>
- [7] Santhoopa Jayawardhana (2020), Sequence Models & Recurrent Neural Networks (RNNs), <https://towardsdatascience.com/sequence-models-and-recurrent-neural-networks-rnns-62cadeb4f1e1#:~:text=Sequence>
- [8] Tanvir (2020), Word Embedding and One Hot Encoding,
<https://medium.com/intelligentmachines/word-embedding-and-one-hot-encoding-ad17b4bbe111>
- [9] Christopher Olah (2015), Understanding LSTM Networks,
<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- [10] Huynh Chi Chung (2020), Giới thiệu về Deep learning, thư viện Keras,
<https://viblo.asia/p/gioi-thieu-ve-deep-learning-thu-vien-keras-63vKjDGAl2R>
- [11] Jason Brownlee (2017), Difference Between Return Sequences and Return States for LSTMs in Keras,

<https://machinelearningmastery.com/return-sequences-and-return-states-for-lstms-in-keras/>

- [12] Jason Brownlee (2017), Stacked Long Short-Term Memory Networks, <https://machinelearningmastery.com/stacked-long-short-term-memory-networks/>
- [13] Trần Trung Trực (2020), Optimizer- Hiểu sâu về các thuật toán tối ưu (GD,SGD,Adam,...), <https://viblo.asia/p/optimizer-hieu-sau-vecac-thuat-toan-toi-uu-gdsgdadam-Qbq5QQ9E5D8>
- [14] Tek4 (2021), API Mô Hình Tuần Tự – Sequential (Đơn Giản) – Keras Cơ Bản, <https://tek4.vn/api-mo-hinh-tuan-tu-sequential-don-gian-keras-co-ban>
- [15] Shipra Saxena (2021), Introduction to Long Short Term Memory (LSTM), <https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/>
- [16] Aakarsha Chug (2021), Deep Learning | Introduction to Long Short Term Memory, <https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/>
- [17] Ike Sri Rahayu, Esmeralda C Djamal, Ridwan Ilyas (2020), Jenderal Achmad Yani university, Indonesia, Daily Temperature Prediction Using Recurrent Neural Networks and Long-Short Term Memory, <http://www.ieomsociety.org/detroit2020/papers/540.pdf>
- [18] What is time series data?, influxdata, <https://www.influxdata.com/what-is-time-series-data/>
- [19] Tuan Nguyen (2019), Long short term memory (LSTM), <https://nttuan8.com/bai-14-long-short-term-memory-lstm/>