DỰ BÁO MỨC ĐỘ CHỈ SỐ CHẤT LƯỢNG KHÔNG KHÍ TẠI BA THÀNH PHỐ CỦA VIỆT NAM SỬ DỤNG PHƯƠNG PHÁP HỌC MÁY VÀ HỌC SÂU

1st Cao Hoài Sang
Khoa Hệ Thống Thông Tin
TP.HCM, Việt Nam
21522541@gm.uit.edu.vn

2nd Nguyễn Trần Gia Kiệt
Khoa Hệ Thống Thông Tin
TP.HCM, Việt Nam
21522258@gm.uit.edu.vn

3rd Thi Thành Công
Khoa Hệ Thống Thông Tin
TP.HCM, Việt Nam
21521897@gm.uit.edu.vn

4th Nguyễn Hoàng Đăng Khoa
Khoa Hệ Thống Thông Tin
TP.HCM, Việt Nam
21520999@gm.uit.edu.vn

5th Cù Ngọc Hoàng
Khoa Hệ Thống Thông Tin
TP.HCM, Việt Nam
21522086@gm.uit.edu.vn

***Tóm tắt nội dung*—Mục tiêu chính của nghiên cứu này là dự đoán chất lượng không khí tai ba thành phố được chỉ định (Hà Nội, Việt Trì và Đà Nẵng) của Việt Nam bằng cách kết hợp các thuật toán học máy và học sâu. Các mô hình bao gồm Gauss Newton Method Non-Linear, Residual Convolutional Neural Networks (ResCNN), Neural Hierarchical Interpolation for Time Series Forecasting (N-HiTS), Dynamic Linear Model (DLM), Simple Exponential Smoothing (SES), Linear Regression (LR), Autoregressive Integrated Moving Average (ARIMA), Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), Long Short Term Memory (LSTM). Hiệu quả của tất cả các mô hình được đề cập trên được đo lường bằng Mean Absolute Percentage Error (MAPE), Root Mean Squared Error, nhằm đạt được độ chính xác tối đa trong dự báo chuỗi thời gian chất lượng không khí chính xác.**

***Index Terms*—Nonlinear regression, Gauss-Newton, generalized least squares, iteratively reweighted least squares**

I. Giới Thiệu

Với sự gia tăng dân số nhanh chóng ở Việt Nam cùng với sự công nghiệp hóa ngày càng mạnh mẽ ở các khu vực trọng yếu, vấn đề chất lượng không khí nhanh chóng trở thành mối quan tâm chính, đặc biệt là ở các thành phố lớn của Việt Nam. Sự suy giảm chất lượng không khí không chỉ gây ra những rủi ro lớn đối với sức khỏe của cư dân mà còn đe dọa sự cân bằng sinh thái mong manh của khu vực. Trong bối cảnh này, việc dự đoán chính xác và kịp thời các mức độ chất lượng không khí là rất cần thiết cho các chiến lược giảm thiểu hiệu quả và can thiệp y tế công cộng. Nghiên cứu của nhóm chúng tôi sử dụng sự kết hợp của các mô hình học máy và học sâu để dự đoán chất lượng không khí, bằng cách tận dụng sức mạnh của các thuật toán phức tạp như Gauss Newton Method Non-Linear, Resilient Convolutional Neural Networks (ResCNN), Neural Basis Expansion Analysis for Time Series Forecasting (N-BEATS), Dynamic Linear Model (DLM), Simple Exponential Smoothing (SES), Linear Regression (LR), Autoregressive Integrated Moving Average (ARIMA), Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), và Long Short Term Memory (LSTM).

Việc bao gồm một tập hợp đa dạng các mô hình cho phép chúng tôi khám phá các khía cạnh khác nhau của dữ liệu và so sánh hiệu suất của các phương pháp khác nhau. Chẳng hạn, các mô hình thống kê truyền thống như SES, LR, và ARIMA cung cấp tính giải thích và sự đơn giản, làm cho chúng trở thành các mô hình cơ sở có giá trị để so sánh. Mặt khác, các kiến trúc học sâu như RNN, GRU, và LSTM xuất sắc trong việc nắm bắt các phụ thuộc phi tuyến tính và phụ thuộc thời gian dài hạn, những đặc điểm thường có trong dữ liệu chuỗi thời gian chất lượng không khí. Ngoài ra, các phương pháp sáng tạo như ResCNN và N-HiTS mang lại những lợi thế độc đáo trong việc xử lý các cấu trúc không gian và thứ bậc trong dữ liệu, nâng cao độ chính xác của các dự đoán.

Identify applicable funding agency here. If none, delete this.

II. Nghiên cứu liên quan

A. Gauss-Newton nonlinear method

Vào năm 2023 , Gauss-Newton được ứng dụng để tối ưu hóa một biến thể của sai phân bình phương trung bình Bellman (MSBE). Cụ thể, Gauss-Newton được sử dụng trong mỗi vòng lặp của phương pháp học Tăng Cường Sai Phân Gauss-Newton (Gauss-Newton Temporal Difference - GNTD) để thực hiện các bước cập nhật tham số của mô hình xấp xỉ hàm phi tuyến. [1]

B. ResCNN

Phân loại ECG bằng bộ Ensemble của Residual CNNs với Cơ chế Chú ý.

Trong phân loại ECG, sử dụng ResNet kết hợp với cơ chế chú ý đa đầu đã chứng minh hiệu quả. Giải pháp của Đội ISIBrno-AIMT trong Cuộc thi PhysioNet 2021 đã thể hiện sự vượt trội bằng việc phân loại ECG thành 26 nhóm khác nhau. Phương pháp này tích hợp các hàm mất mát và tối ưu hóa tiến hóa, mang lại đóng góp quan trọng cho lĩnh vực này. [4]

C. Dynamic linear models

Tháng 2 năm 2001, Monica Chioga cùng với Carlo Gaetan từ Università di Padova, Ý đã đề xuất mô hình hóa các tác động của phơi nhiễm chất ô nhiễm ngắn hạn lên sức khỏe bằng cách sử dụng các mô hình tuyến tính tổng quát động, nhằm tìm ra mối quan hệ giữa số ca tử vong không do tai nạn hàng ngày và ô nhiễm không khí ở thành phố Birmingham, Alabama. [5]

Hiểu biết về động lực trong vành đai bức xạ là rất quan trọng. Các electron có năng lượng cao có thể gây hại cho các vệ tinh. Do đó, việc cung cấp cho các nhà vận hành vệ tinh dự báo chính xác về cường độ dòng electron với thời gian dự báo có thể hành động là rất quan trọng. Vì vậy, vào năm 2014, D. Osthus, P. C. Caragea, D. Higdon, S. K. Morley, G.D. Reeves, và B. P. Weaver đã sử dụng các mô hình tuyến tính động để dự báo electron trong vành đai bức xạ bằng cách so sánh độ chính xác dự báo trước 1 ngày của một mô hình tuyến tính động đơn giản với mô hình dự báo electron tương đối hiện tại (REFM). [6]

D. N-BEATS

Năm 2022, một nhóm gồm năm nhà nghiên cứu, sau khi xem xét các phương pháp dự báo chuỗi thời gian truyền thống như VAR và các biến thể của nó, nhận thấy chúng không hiệu quả do tính chất không ổn định của dữ liệu chất lượng không khí. Do đó, họ đã chọn sử dụng kiến trúc N-BEATS [8] để xây dựng mô hình dự báo cho nhiều chất ô nhiễm không khí khác nhau ở Thành phố Hồ Chí Minh, cụ thể là giá trị của NO2, SO2, CO và O3. Mô hình này đã xem xét các biến động theo thời gian, điều kiện khí tượng, và các yếu tố ảnh hưởng nội bộ giữa các chất ô nhiễm. Kết quả của nghiên cứu bao gồm phân tích không gian-thời gian và phân tích tương quan để hiểu rõ hơn về sự biến động và mối quan hệ giữa các biến liên quan.

Năm 2021, một nhóm nghiên cứu tại Đại học Khoa học và Công nghệ Hà Nội và Đại học FPT đã áp dụng mô hình NBEATS để dự báo nhu cầu điện ngắn hạn ở Việt Nam [9]. Sau khi quan sát thấy mô hình hoạt động tốt với dữ liệu chuỗi thời gian trong các lĩnh vực khác nhau như công nghiệp, tài chính, và nhân khẩu học, nhóm nghiên cứu cũng đã so sánh nó với các phương pháp khác như mạng LSTM với hàm mất mát Pinball và các mô hình tự hồi quy. Tuy nhiên, những phương pháp này có nhược điểm là thời gian chạy tương đối chậm, làm cho việc triển khai chúng trong các ứng dụng thực tế gặp khó khăn.

E. Recurrent Neural Network

Trong nghiên cứu về hiểu biết ô nhiễm không khí, Brian S. Freemana, Graham Taylora, Bahram Gharabaghia, và Jesse Thé từ Trường Kỹ thuật, Đại học Guelph, Guelph, Ontario, Canada dự báo chuỗi thời gian chất lượng không khí bằng cách sử dụng mô hình học sâu mạng nơ-ron hồi quy (RNN) với bộ nhớ dài hạn (LSTM). [10]

F. SES

Nhóm tác giả từ University of Karachi, Pakistan đã sử dụng mô hình ARIMA và SES để dự đoán lượng khí thải CO2 từ một số nước châu Á như Japan, Bangladesh, China, Pakistan, India, Sri Lanka, Iran, Singapore, và Nepal trong khoảng từ năm 1971 đến 2014. Dựa trên FMAE, SES phù hợp để dự đoán lượng CO2 ở Pakistan và Sri Lanka. Trong khi đó, ARIMA thì phù hợp với Japan, China, India, Iran và Singapore. Đối với Nepal và Bangladesh, cả hai mô hình đều cho kết quả tương đương nhau. [11]

G. Linear Regression

Năm 2022, tại Đại học Manonmaniam Sundaranar, A. Loganathan, Sumithra Palraj, và Deneshkumar V đã công bố một nghiên cứu về việc sử dụng hồi quy tuyến tính để dự báo chỉ số Chất lượng không khí (AQI). Nghiên cứu tập trung vào phân tích dữ liệu AQI thu thập từ trạm giám sát ở Chennai, Ấn Độ, và xây dựng một mô hình hồi quy tuyến tính đa biến. Đánh giá về tính hợp lệ của mô hình được thực hiện thông qua phân tích dư thừa. [18]

H. LSTM

Ricardo Navares và José L. Aznarte đã sử dụng mô hình LSTM để dự đoán chất lượng không khí tại thành phố Marid. Nhóm tác giả dựa vào các chỉ số không khí gây hại như CO, NO2, O3, PM10, SO2 được thu thập tại các địa điểm khác nhau của Madrid. Sau đó sử dụng các cấu hình khác nhau của mô hình LSTM để dự đoán chất lượng không khí và so sánh cấu hình LSTM nào sẽ là tốt nhất. [12]

I. ARIMA

Hiện nay, hầu hết các nghiên cứu đều dựa trên việc dự đoán xu hướng thị trường chứng khoán bằng mạng nơ-ron dựa trên ARIMA [13] ARIMA được sử dụng làm cả mô hình phân tích và dự báo trong cơ sở dữ liệu PACAP-CCER của Trung Quốc, được phát triển bởi Trung tâm Nghiên cứu Thị trường Vốn Thái Bình Dương (PACAP) tại Đại học Rhode Island (Mỹ) và Công ty Dịch vụ Thông tin SINOFIN, liên kết với Trung tâm Nghiên cứu Kinh tế Trung Quốc (CCER) của Đại học Bắc Kinh (Trung Quốc). [14] ARIMA đã được áp dụng để giải quyết các vấn đề thực tế trong thị trường chứng khoán bằng cách dự báo giá cổ phiếu của bốn công ty hàng đầu trong chỉ số Nifty Midcap-50 bằng MATLAB kèm theo các chỉ số hiệu suất. [15] Kết hợp mô hình hồi quy mờ và mô hình ARIMA, mô hình ARIMA mờ (FARIMA) đã được phát triển để dự đoán tỷ giá đồng Đài tệ sang Đô la Mỹ. [16] Một mục đích khác mà mô hình ARIMA đã được sử dụng là để dự đoán hoặc dự báo giá cả, cụ thể là giá điện của ngày mai.

III. Tài nguyên

A. Bộ dữ liệu

Dự báo chất lượng không khí là rất quan trọng để giảm thiểu các tác động tiêu cực của ô nhiễm lên sức khỏe con người và môi trường. Phân tích chuỗi thời gian nổi lên như một công cụ mạnh mẽ trong lĩnh vực này, cho phép dự đoán các mức độ chất lượng không khí trong tương lai dựa trên dữ liệu lịch sử.

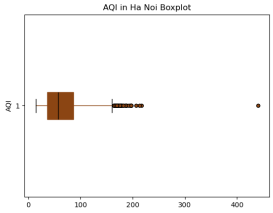
Do các vấn đề phổ biến liên quan đến bụi mịn ở khu vực phía Bắc, nhóm đã chọn một bộ dữ liệu chi tiết về chất lượng không khí của ba thành phố lớn ở miền Bắc Việt Nam: Bắc Ninh, Hà Nội và Quảng Ninh.

Bộ dữ liệu chủ yếu kéo dài từ năm 2021 đến năm hiện tại, 2024, bao gồm 6 cột tương ứng với các thành phần khác nhau trong không khí và chất lượng không khí được đánh giá thông qua Chỉ số Chất lượng Không khí (AQI): nồng độ PM2.5, nồng độ PM10, nồng độ O3, nồng độ NO2, nồng độ SO2 và nồng độ CO.

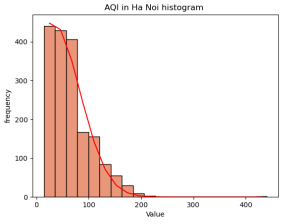
B. Thống kế mô tả

Bảng I
HaNoi, DaNang, VietTri's Descriptive Statistics

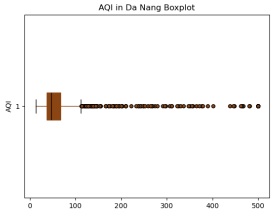
	HaNoi	DaNang	VietTri
Count	2840	3136	2348
Mean	75.351	78.721	60.160
Std	42.285	73.749	41.381
Min	11	13	15
25%	43.0	43.0	31.0
50%	66.0	59.0	47.0
75%	101.0	88.0	81.0
Max	498	500	828



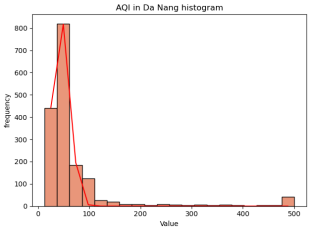
Hình 1. HaNoi AQI's boxplot



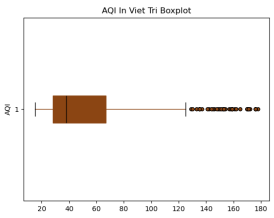
Hình 2. HaNoi AQI's histogram



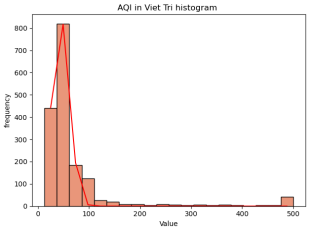
Hình 3. DaNang AQI's boxplot



Hình 4. DaNang AQI's histogram



Hình 5. VietTri AQI's boxplot



Hình 6. VietTri AQI's histogram

IV. Phương pháp luận

A. Gauss newton method nonlinear

1) Least Squares: Khoảng cách giữa một đường cong được khớp và một quan sát được gọi là sai số dư, hoặc lỗi.

$$Residuals = y_i - \widehat{y}_i$$

Tổng của các sai số bình phương được tính bằng phương trình sau:

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

Trong đó:
- y_i là các giá trị quan sát được
- $\widehat{y}_i$ là các giá trị khớp

2) Phương pháp Newton: Với hàm $y = y_0 e^{-kt}$, chúng ta tìm giá trị nhỏ nhất của SSE. Chúng ta tìm giá trị k bằng phương pháp Newton.

$$SSE = \sum_{i}^{n}(y_i - y_0 e^{-kt_i})^2$$

$$k_{\text{new}} = k_{\text{old}} - \frac{f'(k_{\text{old}})}{f''(k_{\text{old}})}$$

Hoặc chúng ta có thể giải thích bằng ma trận Hessian như sau:

$$\begin{pmatrix} k_{\text{new}} \\ y_{0,\text{new}} \end{pmatrix} = \begin{pmatrix} k_{\text{old}} \\ y_{0,\text{old}} \end{pmatrix} - H^{-1}G$$

Trong đó:
- $H = \begin{bmatrix} \frac{\partial^2 f}{\partial k^2} & \frac{\partial^2 f}{\partial k \partial y_0} \\ \frac{\partial^2 f}{\partial y_0 \partial k} & \frac{\partial^2 f}{\partial y_0^2} \end{bmatrix}$
- $G = \begin{bmatrix} \frac{\partial f}{\partial k} \\ \frac{\partial f}{\partial y_0} \end{bmatrix}$

Vấn đề với phương pháp Newton trong hồi quy phi tuyến là việc tính toán ma trận Hessian và nghịch đảo của nó gặp khó khăn. Để giải quyết vấn đề này, phương pháp Gauss-Newton thay thế bằng cách xấp xỉ ma trận Hessian.

Chúng ta có thể viết lại SSE như sau:

$$SSE = \sum_{i}^{n} r_i^2 = r^T r$$

Trong đó:
- r là một vector chứa các sai số dư

3) Phương pháp Gauss-Newton: Chúng ta lấy đạo hàm của SSE theo các tham số trong mô hình bằng quy tắc chuỗi, chúng ta có được phương trình sau:

$$\frac{\partial SSE}{\partial \beta_j} = 2\sum_i^n r_i \frac{\partial r_i}{\partial \beta_j}$$

Sau đó, bỏ số hai vì nó sẽ không ảnh hưởng đến việc ước tính các tham số. Tương ứng với ma trận Jacobian.

$$J_r = \begin{bmatrix} \frac{\partial r_1}{\partial \beta_1} & \frac{\partial r_1}{\partial \beta_2} \\ \frac{\partial r_2}{\partial \beta_1} & \frac{\partial r_2}{\partial \beta_2} \\ \vdots & \vdots \\ \frac{\partial r_n}{\partial \beta_1} & \frac{\partial r_n}{\partial \beta_2} \end{bmatrix}$$

Với phương trình sau cho tổng các sai số bình phương (SSE):

$$SSE = \sum_{i=1}^{n} (y_i - y_0 e^{-kt_i})^2$$

Chúng ta có:

$$\frac{\partial^2 SSE}{\partial \beta_j \partial \beta_k} = \sum_i^n \left(\frac{\partial r_i}{\partial \beta_j} \frac{\partial r_i}{\partial \beta_k} + r_i \frac{\partial^2 r_i}{\partial \beta_j \partial \beta_k} \right)$$

Sự khác biệt chính giữa phương pháp Newton và Gauss-Newton là phương pháp Gauss-Newton bỏ qua $r_i \frac{\partial^2 r_i}{\partial \beta_j \partial \beta_k}$. Do đó, đạo hàm bậc hai được xấp xỉ bằng hàm sau:

$$\frac{\partial^2 SSE}{\partial \beta_j \partial \beta_k} \approx \sum_i^n \left(\frac{\partial r_i}{\partial \beta_j} \frac{\partial r_i}{\partial \beta_k} \right) = J_r^T J_r$$

Sử dụng quy tắc cập nhật sau trong phương pháp Newton. Đối với Gauss-Newton, đơn giản chỉ cần cắm vào xấp xỉ cho ma trận Hessian và gradient.

$$\begin{pmatrix} k_{\text{new}} \\ y_{0,\text{new}} \end{pmatrix} = \begin{pmatrix} k_{\text{old}} \\ y_{0,\text{old}} \end{pmatrix} - (J_r^T J_r)^{-1} J_r^T r$$

Với β là một vector cột với các tham số được ước tính. Đối với ví dụ đơn giản chỉ ước tính hai tham số, phương trình trông như sau:

$$\beta_{\text{new}} = \beta_{\text{old}} - (J_r^T J_r)^{-1} J_r^T r(\beta_{\text{old}})$$

$$\begin{pmatrix} k_{\text{new}} \\ y_{0,\text{new}} \end{pmatrix} = \begin{pmatrix} k_{\text{old}} \\ y_{0,\text{old}} \end{pmatrix} - (J_r^T J_r)^{-1} J_r^T r \begin{pmatrix} k_{\text{old}} \\ y_{0,\text{old}} \end{pmatrix}$$

B. AUTOREGRESSIVE INTEGRATED MOVING AVERAGE (ARIMA)

ARIMA là viết tắt của "Autoregressive Integrated Moving Average". Mô hình ARIMA thường được sử dụng để dự báo dữ liệu chuỗi thời gian đơn biến. Mô hình ARIMA có thể xử lý một chuỗi thời gian nếu chuỗi đó là dừng và không có dữ liệu bị thiếu. Phương pháp này được sử dụng trong nhiều nghiên cứu để dự báo.

ARIMA là sự kết hợp của 3 thành phần, Auto-Regressive – AR, Integrated – I và Moving Average – MA tương ứng với các tham số p, d và q đại diện cho ba thành phần chính của mô hình, trong đó:

• $p - AR(p)$: Tham số p đại diện cho số lượng các quá trình tự hồi quy trong thành phần tự hồi quy (AutoRegressive) của mô hình ARIMA. Nó chỉ ra số lượng ngày quá khứ của chuỗi dữ liệu mà được sử dụng để dự đoán giá trị hiện tại. Mỗi giá trị quá khứ được sử dụng là một hệ số trong mô hình tự hồi quy. Giá trị của p phụ thuộc vào sự phụ thuộc tạm thời trong chuỗi dữ liệu và có thể được xác định bằng cách sử dụng các phương pháp như đồ thị tự tương quan (ACF - AutoCorrelation Function) hoặc hàm tương quan một lệnh (PACF - Partial AutoCorrelation Function). Phương trình tự hồi quy AR được tổng quát như sau:

$$Y_t = c + \varphi_1 Y_{t-1} + \varphi_2 Y_{t-2} + \ldots + \varphi_p Y_{t-p} + \epsilon_t$$

Trong đó:
– Y_t đại diện cho giá trị dữ liệu tại thời điểm t
– c là hằng số chặn
– φ là hệ số AutoRegressive(AR)
– ϵ_t là sai số ngẫu nhiên
– p là số bậc

• $q - MA(q)$: Tham số q đại diện cho số lượng các thành phần trung bình động (Moving Average) trong mô hình ARIMA. Nó chỉ ra số lượng giá trị trung bình động được sử dụng để dự đoán giá trị hiện tại.

$$Y_t = c + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \ldots + \theta_p \epsilon_{t-p} + \epsilon_t$$

Trong đó:
– Y_t đại diện cho giá trị dữ liệu tại thời điểm t
– c là hằng số chặn
– θ là hệ số Moving Average(MA)
– ϵ_t là hệ số tương quan
– p là số bậc

• $d - I(d)$: Tham số d đại diện cho số lần lấy đạo hàm-sai phân (differencing) trên chuỗi dữ liệu ban đầu để loại bỏ xu hướng (trend) và/hoặc thành phần mùa vụ (seasonality) hay chuyển đổi dữ liệu thành chuỗi dừng. chuỗi dừng là chuỗi có trung bình, phương sai và tự tương quan không đổi theo thời gian. Một chuỗi thời gian được coi là chuỗi dừng nếu nó có trung bình không đổi, phương sai không đổi và tự tương quan không đổi. Chuỗi dừng là Công thức tính sai phân tại thời điểm t như sau

$$\Delta y_t = y_t - y_{t-1}$$

Sau khi kết hợp tất cả, ta có ARIMA(p, d, q) được biểu diễn như sau:

$$\Delta y_t = c + \varphi_1 \Delta y_{t-1} + \ldots + \varphi_p \Delta y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q}$$

Trong đó:
– Y_t đại diện cho giá trị dữ liệu tại thời điểm t
– c là hằng số chặn
– θ là hệ số Moving Average(MA)
– ϵ_t là hệ số tương quan
– φ là hệ số AutoRegressive(AR)

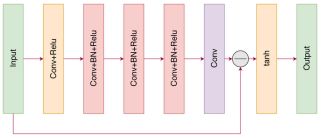
C. LSMT ResCNN

1) Tổng quan: Mô hình LSTM ResCNN là một mô hình dự đoán chuỗi thời gian, là sự kết hợp giữa LSTM và CNN 1D sử dụng kết nối bỏ qua.

2) ResCNN được đề xuất với cách tiếp cận dựa trên độ dốc:

a) Giới thiệu mô hình: Trong bài báo này, chúng tôi đề xuất một mô hình ResCNN, đó là mô hình với một kết nối dư thêm trên các lớp CNN 1D. Chúng tôi đã bao gồm một kết nối dư để tránh mất thông tin quan trọng trong quá trình áp dụng nhiều lớp tích chập 1D liên tục.

b) Thiết kế mô hình: Hình dưới đây mô tả cấu trúc của mô hình mạng nơ-ron tích chập sử dụng kết nối nối bỏ qua(ResCNN) có thể dùng trong các bài toán chuỗi thời gian.



Hình 7. Mô hình ResCNN được đề xuất

c) Các thành phần chính:

- **Input (Đầu vào)**: Đây là nơi dữ liệu chuỗi thời gian đầu vào được đưa vào mạng. Dữ liệu đầu vào thường là các chuỗi thời gian, có thể bao gồm các đặc trưng như giá trị thời gian, chỉ số kinh tế, hoặc dữ liệu cảm biến.
- **LSTM (Long Short-Term Memory)**: Lớp LSTM có khả năng xử lý và ghi nhớ thông tin tuần tự trong dữ liệu chuỗi thời gian. LSTM giúp nắm bắt các mối quan hệ dài hạn trong dữ liệu, là yếu tố quan trọng trong việc dự báo và nhận dạng mẫu trong chuỗi thời gian.
- **Conv + ReLU**:
  - **Conv (Convolution)**: Lớp tích chập giúp mạng trích xuất các đặc trưng không gian từ đầu vào đã được xử lý qua LSTM. Trong ngữ cảnh chuỗi thời gian, lớp này có thể giúp nhận diện các mẫu tuần hoàn hoặc các mối quan hệ phức tạp trong dữ liệu.
  - **ReLU (Rectified Linear Unit)**: Hàm kích hoạt ReLU giúp mạng học các đặc trưng phi tuyến tính bằng cách thay thế các giá trị âm bằng 0.
- **Conv + BN + ReLU**:
  - **Conv (Convolution)**: Lớp tích chập tiếp tục trích xuất các đặc trưng từ dữ liệu.
  - **BN (Batch Normalization)**: Lớp chuẩn hóa theo batch giúp ổn định quá trình huấn luyện và tăng tốc độ hội tụ bằng cách chuẩn hóa các đầu ra của lớp tích chập.
  - **ReLU**: Hàm kích hoạt ReLU giúp mạng học các đặc trưng phi tuyến tính.
- **Conv (Convolution)**: Lớp tích chập khác, không có hàm kích hoạt sau đó, tiếp tục trích xuất các đặc trưng cuối cùng từ dữ liệu.
- **Skip Connection**: Sử dụng phép cộng để thêm đầu vào ban đầu vào kết quả của lớp tích chập cuối cùng. Đây là một dạng kết nối còn lại (Residual Connection) giúp bảo toàn thông tin từ đầu vào ban đầu và cải thiện quá trình huấn luyện.
- **Tanh**: Hàm kích hoạt tanh được sử dụng để đưa đầu ra về một khoảng giá trị nhất định (ở đây là từ -1 đến 1), giúp chuẩn hóa đầu ra của mạng.
- **Output (Đầu ra)**: Đây là nơi mạng trả về kết quả cuối cùng sau khi đã qua tất cả các lớp và các phép biến đổi. Đầu ra có thể là các giá trị dự báo hoặc các kết quả phân loại.

Tóm lại, mô hình này sử dụng các lớp tích chập (convolutional layers) để trích xuất các đặc trưng không gian phức tạp. Hàm kích hoạt ReLU và lớp chuẩn hóa theo batch giúp tăng tính phi tuyến và ổn định trong quá trình huấn luyện. Kết nối tắt và hàm kích hoạt tanh giúp cải thiện hiệu quả học tập và chuẩn hóa đầu ra.

3) 1D-CNN: CNN phổ biến nhất được biết đến là CNN 2D, chủ yếu được sử dụng trong xử lý ảnh. Ngoài ra còn có CNN 1D được sử dụng trong xử lý ngôn ngữ tự nhiên và phân tích chuỗi thời gian, nó trích xuất các đặc trưng bằng cách di chuyển kernel dọc theo trục thời gian. Lớp CNN 1D đóng góp vào việc trích xuất các đặc trưng vốn có trong dữ liệu, và số lượng bộ lọc xác định số lượng đặc trưng cần được học, phù hợp với kích thước của không gian đầu ra.
Mỗi lớp tích chập áp dụng các bộ lọc lên đầu vào, sau đó là một hàm kích hoạt. Công thức tổng quát cho một lớp tích chập một chiều là:

$$\text{out}(N_i, C_{out_j}, L_{out_l}) = \text{bias}(C_{out_j}) + \sum_{k=0}^{C_{in}-1} \text{weight}(C_{out_j}, k) * \text{input}(N_i, k, L_{l-k})$$

Trong đó:

- **out**$(N_i, C_{out_j}, L_{out_l})$: Giá trị đầu ra của mẫu thứ N_i tại kênh đầu ra C_{out_j} và vị trí L_{out_l}.
- **bias**(C_{out_j}): Giá trị bias tương ứng với kênh đầu ra C_{out_j}.
- **weight**(C_{out_j}, k): Trọng số của bộ lọc tương ứng với kênh đầu ra C_{out_j} và kênh đầu vào k.
- **input**(N_i, k, L_{l-k}): Giá trị đầu vào của mẫu thứ N_i tại kênh đầu vào k và vị trí L_{l-k}.
- $*$: Đại diện cho phép tích chập.

4) Lớp Chuẩn hóa Theo Batch: Chuẩn hóa đầu ra của lớp tích chập để cải thiện quá trình huấn luyện. Công thức cho một lớp chuẩn hóa là:

$$\hat{x} = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}}$$

Trong đó:

- x là đầu vào của lớp chuẩn hóa.
- μ là giá trị trung bình của đầu vào.
- σ^2 là phương sai của đầu vào.
- ϵ là một giá trị rất nhỏ để tránh chia cho 0.
- γ và β là các tham số học được (scale và shift).

5) Lớp kích hoạt cuối cùng: Lớp kích hoạt cuối cùng sử dụng hàm kích hoạt `tanh` để biến đổi đầu ra thành khoảng $(-1, 1)$. Công thức của hàm `tanh` là:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

6) Kết nối còn lại: Kết nối còn lại thêm đầu vào ban đầu vào đầu ra của lớp tích chập cuối cùng:

$$\textbf{out}_{res}(x) = x + \text{Conv}(x)$$

Trong đó:

- $\textbf{out}_{res}(x)$ là đầu ra của kết nối còn lại.
- x là đầu vào ban đầu.
- $\textbf{Conv}(x)$ là đầu ra của lớp tích chập cuối cùng.

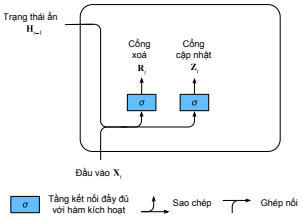
D. Gated Recurrent Unit (GRU)

1) GRU là gì?: GRU là một biến thể của LSTM

Sự khác biệt chính giữa GRU và LSTM nằm ở cách xử lý trạng thái ẩn và ô nhớ:

- **LSTM**: Sử dụng ba cổng (input gate, output gate và forget gate) để duy trì trạng thái của ô nhớ riêng biệt.
- **GRU**: Không có ô nhớ riêng biệt, thay vào đó, nó sử dụng một "vector kích hoạt ứng viên" và hai cổng (reset gate và update gate).
- **Cổng thiết lập lại (reset gate)**: Quyết định bao nhiêu trạng thái ẩn trước đó cần quên.
- **Cổng cập nhật (update gate)**: Quyết định bao nhiêu của vector kích hoạt ứng viên cần tích hợp vào trạng thái ẩn mới.

2) Cổng thiết lập lại và cổng cập nhật: Đầu tiên ta giới thiệu thiết lập lại và cổng cập nhật. Ta thiết kế chúng thành các vector có các phần tử trong khoảng (0,1) để có thể biểu diễn các tổ hợp lồi. Chẳng hạn, một biến xóa cho phép kiểm soát bao nhiêu phần của trạng thái trước đây được giữ lại. Tương tự, một biến cập nhật cho phép kiểm soát bao nhiêu phần của trạng thái mới sẽ giống trạng thái cũ.



Hình 8. Cổng thiết lập lại và cổng cập nhật trong GRU

Ta bắt đầu bằng việc thiết kế các cổng tạo ra các biến này. Hình 8 minh họa các đầu vào cho cả cổng thiết lập lại và cổng cập nhật trong GRU, với đầu vào ở bước thời gian hiện tại $\mathbf{X}_t$ và trạng thái ẩn ở bước thời gian trước đó $\mathbf{H}_{t-1}$. Đầu ra được tạo bởi một tầng kết nối đầy đủ với hàm kích hoạt sigmoid.

Tại bước thời gian t, với đầu vào minibatch là $\mathbf{X}_t \in \mathbb{R}^{n\times d}$ (số lượng mẫu: n, số lượng đầu vào: d) và trạng thái ẩn ở bước thời gian gần nhất là $\mathbf{H}_{t-1} \in \mathbb{R}^{n\times h}$ (số lượng trạng thái ẩn: h), cổng thiết lập lại $\mathbf{R}_t \in \mathbb{R}^{n\times h}$ và cổng cập nhật $\mathbf{Z}_t \in \mathbb{R}^{n\times h}$ được tính như sau:

$$\mathbf{R}_t = \sigma(\mathbf{X}_t\mathbf{W}_{xr} + \mathbf{H}_{t-1}\mathbf{W}_{hr} + \mathbf{b}_r)$$

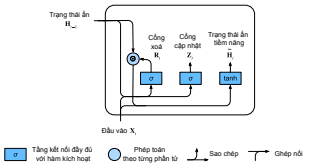
$$\mathbf{Z}_t = \sigma(\mathbf{X}_t\mathbf{W}_{xz} + \mathbf{H}_{t-1}\mathbf{W}_{hz} + \mathbf{b}_z).$$

Ở đây, $\mathbf{W}_{xr}, \mathbf{W}_{xz} \in \mathbb{R}^{d\times h}$ và $\mathbf{W}_{hr}, \mathbf{W}_{hz} \in \mathbb{R}^{h\times h}$ là các ma trận trọng số và $\mathbf{b}_r, \mathbf{b}_z \in \mathbb{R}^{1\times h}$ là các hệ số điều chỉnh. Ta sẽ sử dụng hàm sigmoid để biến đổi các giá trị đầu vào nằm trong khoảng $(0, 1)$.

3) Hoạt động của cổng thiết lập lại: Trong RNN thông thường, ta cập nhật trọng thái cổng thiết lập lại theo công thức

$$\mathbf{H}_t = \tanh(\mathbf{X}_t\mathbf{W}_{xh} + \mathbf{H}_{t-1}\mathbf{W}_{hh} + \mathbf{b}_h).$$

Hình 9 minh họa luồng tính toán sau khi áp dụng cổng thiết lập lại. Ký hiệu $\odot$ biểu thị phép nhân theo từng phần tử giữa các tensor.

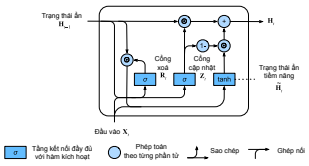


Hình 9. Tính toán trạng thái tiềm năng trong một GRU. Phép nhân được thực hiện theo phần tử

$$\tilde{\mathbf{H}}_t = \tanh(\mathbf{X}_t\mathbf{W}_{xh} + (\mathbf{R}_t \odot \mathbf{H}_{t-1})\mathbf{W}_{hh} + \mathbf{b}_h).$$

4) Hoạt động của cổng cập nhật: Cổng này xác định mức độ giống nhau giữa trạng thái mới $\mathbf{H}_t$ và trạng thái cũ $\mathbf{H}_{t-1}$ cũng như mức độ trạng thái tiềm năng $\widetilde{\mathbf{H}}_t$ được sử dụng. Cổng cập nhật $\mathbf{Z}_t$ được sử dụng cho mục đích này bằng cách áp dụng tổ hợp lồi giữa trạng thái cũ và trạng thái tiềm năng. Ta có phương trình cập nhật cuối cho GRU.

$$\mathbf{H}_t = \mathbf{Z}_t \circ \mathbf{H}_{t-1} + (1 - \mathbf{Z}_t) \circ \tilde{\mathbf{H}}_t.$$



Hình 10. Tính toán trạng thái ẩn trong một GRU. Phép nhân được thực hiện cho từng phần tử

Nếu các giá trị trong cổng cập nhật $\mathbf{Z}_t$ bằng 1, chúng ta chỉ đơn giản giữ lại trạng thái cũ. Trong trường hợp này, thông tin từ $\mathbf{X}_t$ về cơ bản được bỏ qua, tương đương với việc bỏ qua bước thời gian t trong chuỗi phụ thuộc. Ngược lại, nếu $\mathbf{Z}_t$ gần giá trị 0, trạng thái ẩn $\mathbf{H}_t$ sẽ gần với trạng thái ẩn tiềm năng $\widetilde{\mathbf{H}}_t$. Những thiết kế trên có thể giúp chúng ta giải quyết vấn đề tiêu biến gradient trong các mạng RNN và nắm bắt tốt hơn sự phụ thuộc xa trong chuỗi thời gian. Tóm lại, các mạng GRU có hai tính chất nổi bật sau:

- Cổng xoá giúp nắm bắt các phụ thuộc ngắn hạn trong chuỗi thời gian.
- Cổng cập nhật giúp nắm bắt các phụ thuộc dài hạn trong chuỗi thời gian.

E. Dynamic Linear Model (DLM)

Dynamic Linear Model là một trường hợp đặt biệt của mô hình trạng thái không gian - mô hình biểu hiện trạng thái của một hệ thống. Model mô tả hành động của một hệ thống động bằng cách sử dụng một bộ biến, được gọi là biến trạng thái để miêu tả trạng thái hiện tại của hệ thống. Những biến này thể hiện các đặc điểm bên trong của trạng thái biến đổi qua thời gian. Một model DLM đơn gian có hai phương trình, mô hình đầu tiên là mô hình quan sát (observation model):

$$y_t = X_i^T \theta_t + e_i$$

và mô hình trạng thái (state equation) quan sát. Mô tả sự tiến hóa của trạng thái ẩn (không quan sát được) qua thời gian:

$$\theta_t = G_t \theta_(t-1) + w_t$$

Trong đó:

• γ_t dữ liệu quan sát tại thời điểm t.

• χ_t ma trận quan sát.

• θ_t là vector trạng thái chứa yếu tố xu hướng và mùa vụ.

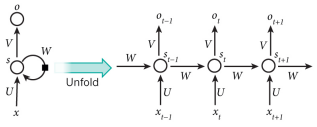
• ϵ_t là nhiễu loạn quan sát.

• G là ma trận giao chuyển để mô tả lại quá trình tiến hóa của trạng thái.

• ω_t là nhiễu loạn trạng thái.

F. Recurrent Neural Networks (RNN)

Ý tưởng chính của RNN (Recurrent Neural Network) là sử dụng chuỗi các thông tin. Trong các mạng nơ-ron truyền thống tất cả các đầu vào và cả đầu ra là độc lập với nhau. Tức là chúng không liên kết thành chuỗi với nhau. Nhưng các mô hình này không phù hợp trong rất nhiều bài toán. RNN được gọi là hồi quy (Recurrent) bởi thực hiện cùng một tác vụ cho tất cả các phần tử của một chuỗi với đầu ra phụ thuộc vào cả các phép tính trước đó. Nói cách khác, RNN có khả năng nhớ các thông tin được tính toán trước đó.



Hình 11. Một mạng lưới thần kinh tái diễn và sự diễn ra theo thời gian của quá trình tính toán liên quan đến tính toán chuyển tiếp.

Mô hình trên mô tả phép triển khai nội dung của một RNN. Triển khai ở đây có thể hiểu đơn giản là ta vẽ ra một mạng nơ-ron chuỗi tuần tự. Ví dụ ta có một câu gồm 5 chữ, thì mạng nơ-ron được triển khai sẽ gồm 5 tầng nơ-ron tương ứng với mỗi chữ một tầng. Lúc đó việc tính toán bên trong RNN được thực hiện như sau:

- x_t là đầu vào tại bước t. Ví dụ, x_t là một vec-tơ one-hot tương ứng với từ thứ 2 của câu.
- s_t à trạng thái ẩn tại bước t. Nó chính là bộ nhớ của mạng. s_t được tính toán dựa trên cả các trạng thái ẩn phía trước và đầu vào tại bước đó: $s_t = f(Ux_t + Ws_{t-1})$. Hàm f thường là một hàm phi tuyến tính như tang hyperbolic (tanh) hay ReLu. Để làm phép toán cho phần tử ẩn đầu tiên ta cần khởi tạo thêm θ.
- o_t à đầu ra tại bước t. Ví dụ, ta muốn dự đoán từ tiếp theo có thể xuất hiện trong câu thì o_t chính là một vec-tơ xác xuất các từ trong danh sách từ vựng của ta: $o_t = softmax(Vs_t)$

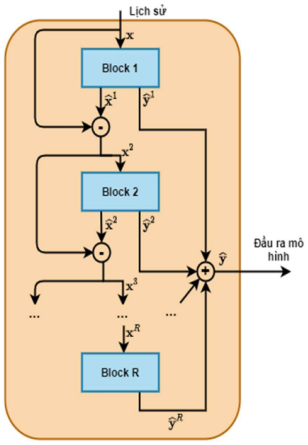
G. Neural Basis Expansion Analysis for Time Series Forecasting (N-BEATS)

1) Tổng quan: N-BEATS được phát triển bởi một nhóm nghiên cứu của Element AI, và được giới thiệu trong bài báo "N-BEATS: Neural Basis Expansion Analysis for Time Series Forecasting"bởi Boris Oreshkin, Dmitri Carpov, Nicolas Chapados và Yoshua Bengio vào năm 2019.

2) Backcast và Forecast: Mỗi khối trong N-BEATS tạo ra hai đầu ra: một backcast và một forecast. Backcast là quá trình tái tạo dữ liệu đầu vào của mô hình, giúp network học tập bằng cách trừ phần tái tạo này khỏi đầu vào, cho phép khối tiếp theo học từ phần dư còn lại. Forecast là đầu ra của khối cho các giá trị dự đoán trong tương lai, đóng góp vào dự báo cuối cùng.

3) Dữ liệu đầu vào: Xét bài toán dự báo đơn chuỗi thời gian:
Từ chuỗi thời gian có độ dài t là $\mathbf{x} = [y_{T-t+1}, y_{T-t+2}, \ldots, y_T] \in \mathbb{R}^T$ ta cần dự báo giá trị của H bước tiếp theo là $\mathbf{y} = [y_{T+1}, y_{T+2}, \ldots, y_{T+H}] \in \mathbb{R}^H$. Ta kí hiệu $\hat{\mathbf{y}}$ là dự đoán của mô hình cho vectơ $\mathbf{y}$. Kích thước của dữ liệu đầu vào là $t = nH$ (n thường nằm trong khoảng từ $[2, 7]$) được gọi là khoảng thời gian xem lại (lookback period). Mô hình N-BEATS sẽ học và tìm hiểu chuỗi thời gian từ khoảng thời gian xem lại để dự đoán giá trị của $\mathbf{H}$ điểm tiếp theo.

4) Kiến trúc tổng quát của mô hình: Kiến trúc tổng quát của mô hình N-BEATS được mô tả như trong hình 12. Mô hình bao gồm các block được xếp chồng lên nhau. Đầu vào của block đầu tiên là đầu vào tổng thể của mô hình hay còn gọi là khoảng thời gian xem lại $\mathbf{x}$. Mỗi block có hai đầu ra. Một đầu ra (backcast) sẽ được làm đầu vào cho block tiếp theo. Đầu ra còn lại (forecast) sẽ được tổng hợp để đưa ra kết quả cuối cùng.



Hình 12. Kiến trúc tổng quát của N-BEATS.

Chi tiết hơn, với block thứ i trong mô hình:

- Nếu $i = 1$, tức là block đầu tiên, đầu vào chính là cửa sổ dữ liệu $\mathbf{x}$;
- Nếu $i > 1$, đầu vào của block thứ i là:

$$\mathbf{x}^i = \mathbf{x}^{i-1} - \hat{\mathbf{x}}^{i-1} \quad (1)$$

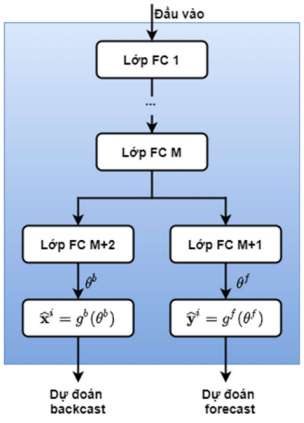
Block thứ i gồm hai đầu ra:

- Đầu ra backcast $\hat{\mathbf{x}}^i$ được làm đầu vào cho block tiếp theo.
- Đầu ra forecast $\hat{\mathbf{y}}^i$ dùng để tổng hợp cho kết quả dự đoán cuối cùng:

$$\hat{\mathbf{y}} = \sum_{i=1}^{R} \hat{\mathbf{y}}^i \quad (2)$$

Từ đây, ta thấy được đầu vào của block sẽ không chứa mà block trước đó đã dự đoán được. Điều này giúp các block sau tập trung vào những phần thông tin chưa được dự đoán.

5) Kiến trúc của block: Hình 13 mô tả kiến trúc trong block thứ $\mathbf{i}$ của mô hình. Kiến trúc của block đơn giản chỉ bao gồm các lớp kết nối đầy đủ FC (fully connected). Các lớp kết nối đầy đủ đơn giản là một lớp ánh xạ tuyến tính với hàm kích hoạt RELU. Ví dụ:



Hình 13. Kiến trúc của block

$$FC(\mathbf{x}^i) = RELU(W\mathbf{x}^i + b) (3)$$

trong đó $\mathbf{W}$ và $\mathbf{b}$ lần lượt là các ma trận sẽ được học trong quá trình huấn luyện. Hàm RELU là hàm $\mathbf{f(x) = max(0, x)}$. Block bao gồm M lớp FC chung (M thường là 4). Sau đó, sẽ có hai lớp FC là M+1 và M+2 để chia đầu ra thành 2 nhánh. Đầu ra của hai lớp trên lần lượt là hai vectơ hệ số θ^f và θ^b. Cuối cùng, ta sẽ ánh xạ hai vectơ trên thành các đầu ra backcast và forecast thông qua các hàm $\mathbf{g}^f$ và $\mathbf{g}^b$. Tuỳ vào dạng hàm $\mathbf{g}$ khác nhau mà chúng ta sẽ có các loại block với vai trò khác nhau. Cụ thể, mô hình N-BEATS giới thiệu ba loại block: trend block, seasonality block, generic block.

Generic block là kiến trúc không mang tính diễn giải được trong chuỗi thời gian. Generic block đơn giản đặt các ánh xạ $\mathbf{g}^f$ và $\mathbf{g}^b$ là các ánh xạ tuyến tính đối với đầu ra của lớp trước.

Trend block là loại block để nhận biết và phân tích được tính xu hướng của dữ liệu. Một đặc điểm của xu hướng thường phần lớn là các hàm đơn điệu hoặc thay đổi chậm. Để tìm hiểu được tính xu hướng của dữ liệu, các hàm $\mathbf{g}^f$ và $\mathbf{g}^b$ sẽ là các hàm đa thức với bậc $\mathbf{p}$ nhỏ theo thời gian. Khi đó, các hệ số θ^f và θ^b đóng vai trò là các hệ số của đa thức. Ví dụ đầu ra forecast của trend block thứ $\mathbf{i}$ là:

$$\hat{\mathbf{y}^i} = \sum_{j=0}^{p} \theta_j^f t^j \quad (4)$$

Trong đó θ_j^f là phần tử thứ j trong vectơ hệ số $\mathbf{t} = \frac{[\mathbf{0},\mathbf{1},\mathbf{2},\ldots,\mathbf{H-2},\mathbf{H-1}]^\mathbf{T}}{\mathbf{H}}$

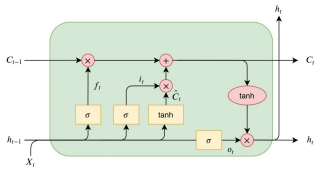
Cuối cùng, loại seasonality block dùng để nhận biết tính mùa của dữ liệu. Đặc điểm chính của tính mùa là dữ liệu thường có tính chu kỳ. Do đó, để mô phỏng tính mùa, các hàm $\mathbf{g}^f$ và $\mathbf{g}^b$ sẽ là các loại hàm tuần hoàn, ví dụ $\mathbf{y}_t = \mathbf{y}_{\Delta+t}$ trong đó $\boldsymbol{\Delta}$ là chu kỳ mùa. Sử dụng chuỗi Fourier để xây dựng các hàm tuần hoàn trên, ta có đầu ra forecast của seasonality block thứ $\mathbf{i}$ là:

$$\hat{y}^i = \sum_{j=0}^{|\frac{H}{2}-1|} \theta_j^f \cos(2\pi it) + \theta_{j+|\frac{H}{2}|}^f \sin(2\pi it) \quad (5)$$

Sau khi giới thiệu ba loại block trên, mô hình N-BEATS được xây dựng dựa trên số lượng và thứ tự của mỗi loại block.

H. Long Short Term Memory (LSTM)

Mạng bộ nhớ dài-ngắn (Long Short Term Memory), thường được gọi là LSTM - là một dạng cải tiến của RNN, nó có khả năng học được các phụ thuộc xa .LSTM gồm trạng thái ẩn h_t (hidden state) và c_t (cell state). LSTM gồm 3 thành phần chính:



Hình 14. Một ô LSTM ở trạng thái t.

Cổng quên(Forget gate):Quản lý việc thông tin đến từ trạng thái trước sẽ được giữ lại hoặc loại bỏ.

$$f_t = \sigma(W_{fx}.x_t + W_{fh}.h_{t-1} + b_f)$$

Cổng vào (Input gate): Quản lý việc học thông tin mới từ đầu vào.

$$i_t = \sigma(W_{ix}.x_t + W_{ih}.h_{t-1} + b_i)$$

$$\widetilde{C}_t = tanh\left(W_{cx}.x_t + W_{ch}.h_{t-1} + b_c\right)$$

$$C_t = f_t \circ c_{t-1} + i_t \circ \widetilde{C}_t$$

Cổng ra (Output gate): Quản lý việc đưa thông tin mới cập nhật vào trạng thái kế tiếp.

$$o_t = \sigma(W_{ox}.x_t + W_{oh}.h_{t-1} + b_o)$$

$$h_t = o_t \circ tanh(C_t)$$

Trong đó:
• x_t : Vector đầu vào các đặc trưng tại thời điểm t.
• h_{t-1} : Trạng thái ẩn đầu ra của tại thời điểm t-1.
• $W_{fx}, W_{fh}, W_{ix}, W_{ih}, W_{cx}, W_{hx}$, : Các ma trận trọng số.
• b_f, b_i, b_c : Các hệ số bias.
• Tanh: Hàm tanh có giá trị tử -1 đến 1.
• i_t : Giá trị cổng đầu vào tại thời điểm t.
• σ : Hàm Sigmoid có giá trị tử 0 đến 1.
• f_t : Giá trị của cổng quên tại thời điểm t.
• $\widetilde{C}_t$: Ô nhớ mới.
• C_t : Ô nhớ tổng thể.
• $\circ$: Phép nhân Hadamard.
• $f_t \circ c_{t-1}$: Quyết định lượng thông tin nào từ trạng thái ô nhớ trước đó sẽ được giữ lại cho ô tiếp theo.
• $i_t \circ \widetilde{C}_t$: Quyết định lượng thông tin nào từ $\widetilde{C}_t$ sẽ được thêm vào C_t.
• o_t : Giá trị đầu ra tại thời điểm t.

I. Simple Exponential Smoothing (SES)

Simple exponential smoothing (SES) hay san bằng hàm mũ đơn giản là một phương pháp dự đoán dữ liệu chuỗi thời gian. SES dựa vào tổng trung bình trọng số của dữ liệu thực tế gần nhất và dữ liệu được dự đoán trước đó để dự đoán dữ liệu tiếp theo trong tương lai. SES phù hợp với dữ liệu chuỗi thời gian không có xu hướng và không có tính mùa vụ.. SES có thể viết ở hai dạng: dạng trung bình trọng số và dạng đối tượng.

Dạng trung bình trọng số:

$$\hat{y}_{t+1|t} = \alpha y_t + (1-\alpha)\hat{y}_{t|t-1}$$

Trong đó:
• $\hat{y}_{t+1|t}$: giá trị dự đoán ở thời điểm t+1 dựa vào giá trị thực tế ở thời điểm t.
• α: hệ số làm trơn $(0 \leq \alpha \leq 1)$.
• $\hat{y}_{t|t-1}$: giá trị dự đoán ở thời điểm t dựa vào giá trị thực tế ở thời điểm t-1.

Giả sử tại thời điểm ban đầu $\hat{y}_1 = y_1$, công thức ở trên có thể viết lại thành:

$\hat{y}_{t+1|t} = \alpha y_t + \alpha(1-\alpha)y_{t-1} + \alpha(1-\alpha)^2 y_{t-2} + \ldots + \alpha(1-\alpha)^{t-1} y_1 + (1-\alpha)^t y_1 = \sum_{i=0}^{t-1} \alpha(1-\alpha)^i y_{t-i} + (1-\alpha)^t y_1$

Dạng đối tượng:
• Phương trình dự đoán: $\hat{y}_{t+h|t} = l_t$, h=1,2,3,....
• Phương trình làm trơn: $l_t = \alpha y_t + (1-\alpha)l_{t-1}$

Trong đó:
• $\hat{y}_{t+h|t}$: giá trị dự đoán tại thời điểm t+h dựa trên giá trị thực tế ở thời điểm t.
• α: hệ số làm trơn $(0 \leq \alpha \leq 1)$.
• l_t: cấp độ (giá trị đã được làm trơn) của chuỗi tại thời điểm t.

Dự báo phẳng:

$$\hat{y}_{t+h|t} = \hat{y}_{t+1|t} = \ell_t, h = 2, 3, \ldots.$$

Trong SES, tất cả các giá trị dự báo không có giá trị thực tế trước đó sẽ dựa vào giá trị dự báo và giá trị thực tế cuối cùng trong tập dữ liệu. Đó gọi là dự báo "phẳng".

J. Linear Regression

Phân tích hồi quy là một công cụ để xây dựng các mô hình toán học và thống kê mô tả mối quan hệ giữa một biến phụ thuộc và một hoặc nhiều biến độc lập, hoặc biến giải thích, tất cả đều là các biến số. Kỹ thuật thống kê này được sử dụng để tìm một phương trình dự đoán tốt nhất cho biến y như một hàm tuyến tính của các biến x.

Một mô hình hồi quy tuyến tính đa biến có dạng:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \varepsilon$$

Trong đó:

- Y là biến phụ thuộc (biến mục tiêu).
- $X_1, X_2, \ldots, X_k$ là các biến độc lập (biến giải thích).
- β_0 là hệ số chặn.
- $\beta_1, \ldots, \beta_k$ là các hệ số hồi quy cho các biến độc lập.
- ε là thuật ngữ lỗi.

V. KẾT QUẢ

A. HaNoi dataset

Dataset's Evaluation				
Model	Training:Testing	RMSE	MAPE (%)	MAE
ARIMA	7:3	49.32	**43.87**	36.28
	8:2	54.44	77.00	48.37
	9:1	**31.09**	49.10	**26.47**
DLM	7:3	5212.21	7.55	0.016
	8:2	1014.97	1.62	0.0005
	9:1	**822.63**	**1.26**	**0.0003**
GNN	7:3	916.692	1.67	0.00055
	8:2	948.341	1.74	0.00057
	9:1	. **761.754**	**1.21**	**0.0003**
GRU	7:3	7847.594	15.278	0.041
	8:2	7501.223	15.14	0.036
	9:1	**3371.058**	**6.414**	**0.006**
NBEATS	7:3	28.31	31.20	21.95
	8:2	**27.26**	**27.06**	**21.62**
	9:1	34.99	29.10	27.17
ResCNN	7:3	28.739	31.41	20.543
	8:2	32.038	30.35	24.295
	9:1	**28.718**	**26.159**	**21.505**
RNN	7:3	7849.6833	15.2872	0.0407
	8:2	7502.4992	15.1483	0.0357
	9:1	**3342.8102**	**6.3561**	**0.0057**
SES	7:3	941.7588	1.7384	0.0005
	8:2	939.7588	1.6546	0.0005
	9:1	**936.8374**	**1.6273**	**0.0005**
LN	7:3	64.77	59.609	51.019
	8:2	65.904	58.715	54.566
	9:1	**40.554**	**33.039**	**30.327**
LSTM	7:3	941.7588	1.7384	0.0005
	8:2	939.7588	1.6546	0.0005
	9:1	**936.8374**	**1.6273**	**0.0005**

Bảng II
HANOI DATASET'S EVALUATION

TÀI LIỆU

[1] Ke, Z., Zhang, J., & Wen, Z. (2023). Gauss-Newton Temporal Difference Learning with Nonlinear Function Approximation.

[2] H. Choi, C. Jung, T. Kang, H. J. Kim, and I. -Y. Kwak, "Explainable time-series prediction using a residual network and gradient-based methods,"in IEEE Access, vol. 10, pp. 108469-108482, 2022.

[3] Doan Vo Duy Thanh, Nguyen Van Cuong, Vo Tan Phat, Le Khac Hong Phuc, Nguyen Duy Du, Nguyen Van Quang, Pham Minh Duc, Pham Hong Vinh, and Nguyen Canh Thuong. *Gated Recurrent Unit (GRU)*. Accessed May 26, 2024. https://d2l.aivivn.com/chapter_recurrent-modern/gru_vn.html.

[4] Nejedly P, Ivora A, Viscor I, Koscova Z, Smisek R, Jurak P, Plesinger F. Classification of ECG using ensemble of residual CNNs with or without attention mechanism. Physiol Meas. 2022 Apr 28;43(4). doi: 10.1088/1361-6579/ac647c. PMID: 35381586.

[5] M. Chiogna and C. Gaetan, "Dynamic Generalized Linear Models with Application to Environmental Epidemiology,"Journal of the Royal Statistical Society Series C: Applied Statistics, vol. 51, no. 4, pp. 453-468, 2002.

[6] D. Osthus, P. C. Caragea, D. Higdon, S. K. Morley, G.D.Reeves and B. P. Weaver, "Dynamic linear models for forecasting of radiation beltelectrons and limitations on physical interpretationof predictive models,"vol. 12, no. 6, pp. 323-446, 2014.

[7] Wang, Junjie & Su, Xiaohong & Zhao, Lingling & Zhang, Jun. (2020). Deep Reinforcement Learning for Data Association in Cell Tracking. Frontiers in Bioengineering and Biotechnology. 8. 298. 10.3389/fbioe.2020.00298.

[8] Rajnish Rakholia , Quan Le, Bang Quoc Ho, Khue Vu, Ricardo Simon Carbajo. "Multi-output machine learning model for regional air pollution forecasting in Ho Chi Minh City, Vietnam" (2023). doi: https://doi.org/10.1016/j.envint.2023.107848. URL: https://www.sciencedirect.com/science/article/pii/S0160412023001216

[9] Nguyen Anh Tuan, Le Anh Ngoc . "APPLYING N-BEATS MODEL FOR SHORT-TERM LOAD FORECASTING IN VIETNAM" (2022). URL: https://khcncongthuong.vn/tin-tuc/t18804/ung-dung-mo-hinh-n-beats-cho-du-bao-phu-tai-dien-ngan-han-o-viet-nam.html

[10] B. S, Freemana, G. Taylora, B. Gharabaghia and J. Thé, "Forecasting air quality time series using deep learning,"Journal of the air & waste management association, vol. 68, no. 8, pp. 866-886, 2017-2018.

[11] Fatima, Samreen & Saad, Sayed & Zia, Syeda & Hussain, Ehtesham & Fraz, Tayyab & Shafi, Mehwish & Khan,. (2019). Forecasting Carbon Dioxide Emission of Asian Countries Using ARIMA and Simple Exponential Smoothing Models. International Journal of Economic and Environment Geology. 10. 10.46660/ojs.v10i1.219.

[12] Navares, Ricardo & Aznarte, José. (2019). Predicting air quality with deep learning LSTM: Towards comprehensive models. Ecological Informatics. 55. 101019. 10.1016/j.ecoinf.2019.101019.

[13] Jarrett, Jeffrey E., and Eric Kyper. "ARIMA modeling with intervention to forecast and analyze chinese stock prices."International Journal of Engineering Business Management 3.3 (2011): 53-58.

[14] Devi, B. Uma, D. Sundar, and P. Alli. "An Effective Time Series Analysis for Stock Trend Prediction Using ARIMA Model for Nifty Midcap-50."

[15] Tseng, Fang-Mei, et al. "Fuzzy ARIMA model for forecasting the foreign exchange market."Fuzzy sets and systems 118.1 (2001): 9-19

[16] Contreras, J., Espinola, R., Nogales, F. J., & Conejo, A. J. (2003). ARIMA models to predict next-day electricity prices. Power Systems, IEEE Transactions on, 18(3), 1014-1020.

[17] Oreshkin, B. N., Carpov, D., Chapados, N., & Bengio, Y. (2019). N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. arXiv preprint arXiv: 1905.10437. https://doi.org/10.48550/arXiv.1905.10437.

[18] Loganathan, A. & Palraj, Sumithra & V, Deneshkumar. (2022). Estimation of Air Quality Index Using Multiple Linear Regression. Applied Ecology and Environmental Sciences. 10. 717-722. 10.12691/aees-10-12-3.