Dự báo sản lượng khai thác hải sản sử dụng các kỹ thuật phân tích chuỗi thời gian

1st Thái Minh Triết
IS403.N21
Trường đại học Công nghệ thông tin
19522397@gm.uit.edu.vn

2nd Nguyễn Ngọc Hiền
IS403.N21
Trường đại học Công nghệ thông tin
20520496@gm.uit.edu.vn

3rd Nguyễn Tô Đức Tài
IS403.N21
Trường đại học Công nghệ thông tin
20520743@gm.uit.edu.vn

4th Nguyễn Thị Kim Liên
IS403.N21
Trường đại học Công nghệ thông tin
20520909@gm.uit.edu.vn

5th Trần Ngọc Linh
IS403.N21
Trường đại học Công nghệ thông tin
20521538@gm.uit.edu.vn

***Tóm tắt nội dung*—Ngành công nghiệp đánh bắt thủy sản đóng vai trò quan trọng trong nền kinh tế và cung cấp nguồn thực phẩm quan trọng cho hàng triệu người trên toàn thế giới. Tuy nhiên, hoạt động đánh bắt đối mặt với nhiều thách thức và khó khăn. Một trong những thách thức lớn là khả năng dự báo và quản lý hiệu quả sản lượng khai thác hải sản. Việc dự báo sản lượng khai thác hải sản là cần thiết để đảm bảo sự bền vững của nguồn tài nguyên cá và đáp ứng nhu cầu lương thực toàn cầu. Trước đây, quy trình đánh cá dựa vào kinh nghiệm và tri thức thông qua quan sát và theo dõi di chuyển của cá. Tuy nhiên, quy trình này không hiệu quả và mất nhiều thời gian và nguồn lực. Đối với công nghiệp đánh cá hiện đại, việc sử dụng các kỹ thuật phân tích dữ liệu và dự báo sản lượng khai thác hải sản trở nên quan trọng. Bằng cách áp dụng các phương pháp phân tích chuỗi thời gian và mô hình hóa dữ liệu, chúng ta có thể phân tích xu hướng và mô hình hóa diễn biến sản lượng khai thác hải sản theo thời gian. Nghiên cứu này áp dụng 10 thuật toán để dự báo sản lượng khai thác của ba loài cá có kinh tế cao và đóng góp quan trọng trong ngành công nghiệp đánh cá. Thông qua nghiên cứu này, có thể chỉ ra rằng các đặc điểm về thời gian và thuộc tính của loài cá đó có thể sử dụng để dự báo sản lượng khai thác trong tương lai, nhằm cung cấp thông tin cần thiết và giúp cải thiện quản lý tài nguyên hải sản trong lĩnh vực kinh doanh.**

***Index Terms*—Dự báo sản lượng khai thác hải sản, chuỗi thời gian, học máy, học sâu.**

I. Giới thiệu chung

Trong ngành đánh cá và nuôi trồng thủy sản, dự báo sản lượng khai thác hải sản là một trong những việc quan trọng để đảm bảo sự bền vững và tăng cường hiệu quả kinh doanh. Hiện nay, việc sử dụng các kỹ thuật phân tích dữ liệu và phân tích chuỗi thời gian đã trở thành một công cụ hữu ích để dự báo sản lượng khai thác hải sản. Trong nghiên cứu này, chúng tôi tập trung vào việc dự báo sản lượng khai thác của ba loài cá có giá trị kinh tế cao là cá tuyết vùng đông bắc Bắc cực (Northeast Arctic Cod), cá tuyết chấm đen vùng đông bắc Bắc cực (Northeast Arctic Haddock) và cá bơn lưỡi ngựa Đại Tây Dương (Atlantic Halibut).

Với một khu vực đánh cá rộng 2,1 triệu mét vuông, Na Uy được coi là quốc gia đánh cá và nuôi trồng thủy sản lớn nhất châu Âu. Thế nên, việc dự báo sản lượng khai thác hải sản của Na Uy đóng một vai trò quan trọng trong việc quản lý tài nguyên cá và phát triển ngành công nghiệp đánh cá.

Nghiên cứu này tập trung vào việc dự báo sản lượng khai thác hải sản sử dụng các kỹ thuật phân tích chuỗi thời gian. Chúng tôi đã nghiên cứu và áp dụng một loạt các thuật toán để dự báo sản lượng khai thác của ba loài cá kể trên, cụ thể là: ARIMA, Exponential Smoothing, Linear Regression, Random Forest, K-Nearest Neighbor, Recurrent Neural Network (RNN), Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU), Sequence-to-Sequence Network và Temporal Convolutional Network. Bằng cách áp dụng những kỹ thuật phân tích chuỗi thời gian và đánh giá hiệu suất của các mô hình dự báo, nghiên cứu này nhằm cung cấp thông tin cần thiết và giúp cải thiện quản lý tài nguyên hải sản trong lĩnh vực kinh doanh. Điều này giúp nhà quản lý và các doanh nghiệp trong ngành cá có thể dự đoán và lập kế hoạch sản xuất, vận chuyển và tiếp thị một cách hiệu quả. Việc dự báo sản lượng khai thác hải sản cũng giúp giảm thiểu rủi ro, tối ưu hóa hoạt động và đảm bảo tính linh hoạt trong quản lý nguồn tài nguyên cá.

II. Các công trình nghiên cứu liên quan

Trong những năm qua, một số phương pháp dự đoán chuỗi thời gian đã được đề xuất. Phương pháp ứng dụng những thuật toán học máy như dự đoán sản lượng lúa mì ở năm bang của Ấn Độ sử dụng hai thuật toán Autoregressive Integrated Moving Average (ARIMA) and Exponential Smoothing (ETS) [1], sử dụng thuật toán Support Vector Machines (SVM) và Random Forest (RF) để dự đoán thị trường cổ phiếu [2] kết quả thực nghiệm cho thấy thuật toán SVM cho kết quả dự đoán tốt hơn thuật toán RF. Reaz Chowdhury và cộng sự [3] đã tiến hành dự báo giá của tiền điện tử sử dụng học máy từ ngày 1/1/2015 đến ngày 1/1/2017 thông qua Gradient Boosted Trees, Neural Network, Ensemble Learning Method và mô hình K-NN. Trong nghiên cứu này, mô hình K-NN hoạt động không hiệu quả như các mô hình khác. [4] Các tác giả đã đề xuất một phương pháp dự đoán nhiệt độ bề mặt biển dựa trên

Long Short Term Memory (LSTM). Bên cạnh đó, tác giả còn thực hiện dự đoán trên mô hình Support Vector Regression (SVR) để so sánh. Để đánh giá hiệu suất của mô hình tác giả sử dụng Mean Squared Error (RMSE) và ACC (giá trị được xác định để đánh giá độ chính xác của dự đoán). Kết quả thực nghiệm cho thấy mô hình LSTM cho hiệu suất dự đoán tốt nhất. [5] Để cải thiện độ chính xác của dự đoán dữ liệu chuỗi thời gian các tác giả đã đề xuất mô hình dự đoán chuỗi thời gian mới dựa trên khung mã hóa-giải mã trong đó bộ mã hóa tự động, mạng thần kinh hồi quy, cơ chế chú ý, mô-đun tích chập và mô-đun kết nối đầy đủ được tích hợp để thực hiện dự đoán dữ liệu cổ phiếu và dự báo nhu cầu về dữ liệu xe đạp dùng chung. Kết quả mô hình đề xuất có độ chính xác dự đoán cao hơn so với những phương pháp nghiên cứu khác.

Về dự báo sản lượng khai thác hải sản có một vài bài nghiên cứu như [6] Trong bài báo này, phương pháp Box-Jenkins đã được sử dụng để xây dựng mô hình Seasonal Autoregressive Integrated Moving Average (SARIMA) cho sản lượng đánh bắt hàng tháng của hai loài cá được tìm thấy ở vùng biển Malaysia trong giai đoạn từ 2007 – 2011. Ngoài ra, còn có nghiên cứu của Thai cùng các đồng sự [7] tập trung vào phát triển các thuật toán khuyến nghị địa điểm đánh bắt trên biển và dự báo sản lượng khai thác hải sản tại vùng đánh cá Na Uy. Các phương pháp dự báo được sử dụng gồm mạng Sequence-to-Sequence (Seq2Seq) và Temporal Convolutional Network (TCN) cho thấy kết quả tốt trong việc dự báo sản lượng loài cá thu và cá tuyết chấm đen từ một đến bốn ngày trong tương lai. Nghiên cứu cũng chỉ ra vai trò quan trọng của các yếu tố môi trường như nhiệt độ bề mặt nước biển đến việc dự báo và mô hình hóa sản lượng khai thác hải sản.

III. Bộ dữ liệu

A. Thông tin bộ dữ liệu

Các tập dữ liệu được sử dụng trong đồ án nghiên cứu này được trích xuất từ một bộ ghi chú tổng hợp về quá trình khai thác hải sản cung cấp bởi Tổng cục Thủy sản của Na Uy[1]. Bộ dữ liệu được biên soạn bằng tiếng Na Uy và chứa thông tin về các chuyến đánh bắt xa bờ của các tàu thuyền đánh cá kể từ năm 2000 tại vùng biển Na Uy thuộc khu vực Bắc Đại Tây Dương và dữ liệu được cập nhật hàng ngày cho đến ngày nay. Các thông tin về chuyến đánh bắt được ghi lại bởi ngư dân sau khi tàu cập cảng, bao gồm 133 trường thuộc tính chứa các dữ kiện quan trọng như: thời gian đánh bắt, địa điểm đánh bắt, các công cụ sử dụng, và sản lượng khai thác được của từng loài thủy-hải sản trong chuyến đánh bắt đó. Với số lượng lớn tàu thuyền hoạt động trong khu vực, mỗi năm có khoảng một triệu ghi chú đánh bắt được thêm mới, với nhiều chủng loài hải sản khác nhau, giúp cho bộ dữ liệu trở thành một nguồn tài nguyên dồi dào cho việc phát triển các phương pháp mô hình hóa và dự báo sản lượng khai thác thủy-hải sản trong tương lai.

Dữ liệu sử dụng cho việc dự báo trong nghiên cứu này được lấy từ ngày 01/01/2000 đến ngày 16/06/2023 về sản lượng khai thác của ba loài cá phổ biến tại khu vực là cá tuyết vùng đông bắc Bắc cực (Northeast Arctic Cod), cá tuyết chấm đen vùng đông bắc Bắc cực (Northeast Arctic Haddock) và cá bơn lưỡi ngựa Đại Tây Dương (Atlantic Halibut). Ba loài hải sản này được lựa chọn cho nghiên cứu do chúng là các loài phổ biến, ít bị khuyết dữ liệu trong thời gian khảo sát. Ngoài ra chúng là những loài cá có tập tính sinh sản và phân bố khác nhau, và có những giá trị kinh tế nhất định, từ đó việc phát triển các mô hình dự báo sản lượng phù hợp với đặc trưng mỗi loài sẽ mang lại ý nghĩa lớn và giúp đỡ cho cuộc sống của người ngư dân. Quá trình rút trích ba tập dữ liệu con từ bộ dữ liệu ban đầu cùng các bước tiền xử lý dữ liệu sẽ được trình bày rõ hơn ở Phần V-A.

[1]https://www.fiskeridir.no/Tall-og-analyse/AApne-data/Fangstdata-seddel-koblet-med-fartoeydata

B. Thống kê mô tả

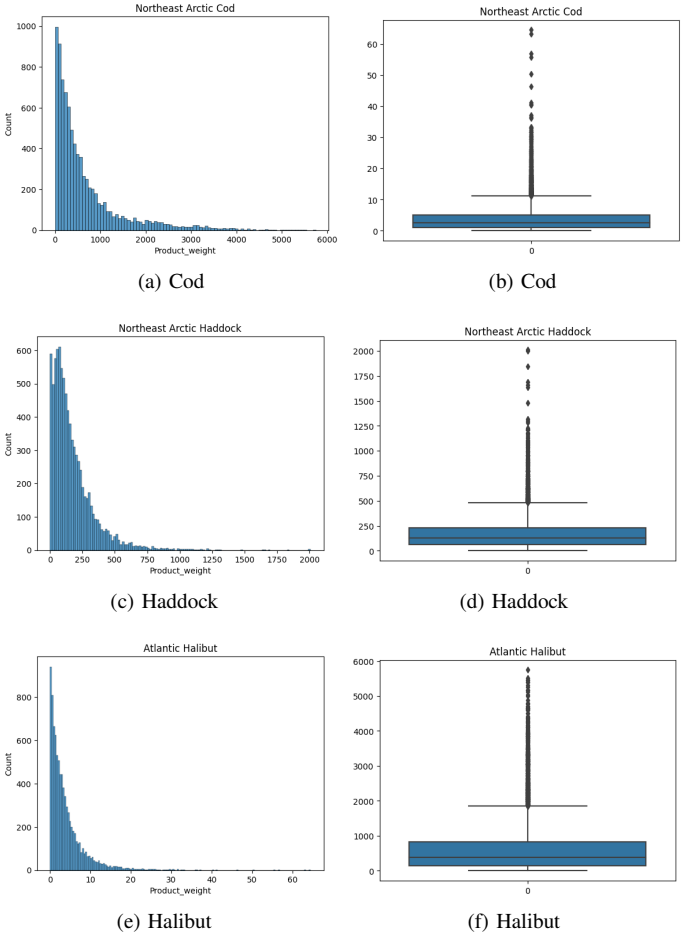
Bảng I: Tổng quan thống kê mô tả ở ba loài hải sản được khảo sát

Giá trị	Cod	Haddock	Halibut
Count	8568	8568	8568
Mean	668.102098	173.369969	3.923664
Variance	649881.1659	28576.40768	20.784164
Standard Deviation	806.152074	169.045579	4.558965
Median	379.3785	126.1202	2.5866
Min	0.007	0.002	0
Max	5744.0752	2009.3079	64.504
Mode	71.441	0.012	0.19
Quantile 1	152.7702	61.759633	1.022075
Quantile 3	834.41875	230.166125	5.117325
Coefficient of Deviation	4.649895	0.975057	0.006824
Kurtosis	5.852337	12.389962	20.912169
Skewness	2.265035	2.621722	3.331709

Theo Bảng I, trong ba loài hải sản, Cod là loài có sản lượng đánh bắt nhiều nhất với sản lượng trung bình khoảng 668,1 tấn/ngày. Ngoài ra, loài này cũng cho thấy sự biến động về sản lượng nhiều hơn so với Haddock và Halibut do hệ số biến thiên của nó (CV = 4.649895) lớn hơn 1. Ta còn có thể thấy Kurtosis của cả ba loài đều khá lớn và chênh lệch nên độ tập trung và phân bố dữ liệu sẽ có sự khác biệt đáng kể nhưng riêng Halibut cho thấy loài này có sự xuất hiện của nhiều giá trị cực đại và cực tiểu trong phân phối của dữ liệu. Bên cạnh đó, giá trị Skewness của Cod, Haddock và Halibut đều lớn hơn 0 nên dữ liệu sẽ có xu hướng lệch về bên phải.

Từ Hình 1, ta thấy được sản lượng của ba loài đều có xu hướng lệch về bên phải và hộp của Halibut có kích thước lớn hơn cũng như đường râu (whiskers) kéo dài xa hộp về phía trên hơn so với hai loài còn lại, do đó, dữ liệu của loài này có sự phân tán và biến động dữ liệu lớn hơn. Ngoài ra, khoảng giá trị của ba loài này cũng khác biệt với nhau, với Cod tập trung chủ yếu trong khoảng từ 0 - 1000, Haddock trong khoảng từ 0 - 250 và Halibut trong khoảng từ 0 - 10.

Nhìn chung, Cod, Haddock và Halibut đều có các đặc tính, biến động và phân tán dữ liệu khác biệt nhau nên việc thực hiện nghiên cứu dựa trên ba loài này là có ý nghĩa.



(a) Cod (b) Cod

(c) Haddock (d) Haddock

(e) Halibut (f) Halibut

Hình 1: Biểu đồ Histogram và Box plot của ba loài hải sản

IV. Phương pháp

A. Linear Regression

Hồi quy tuyến tính (Linear regression) là một phương pháp phân tích thống kê dựa trên việc xác định mối quan hệ giữa hai loại biến bao gồm một biến phụ thuộc (kết quả) và các biến độc lập (dự đoán). Mục đích chính của hồi quy tuyến tính là giúp ta dự đoán được giá trị của biến phụ thuộc dựa trên giá trị của các biến độc lập, kiểm tra xem các biến độc lập có ảnh hưởng thế nào đến giá trị của biến phụ thuộc và các biến độc lập nào là yếu tố quan trọng trong việc dự đoán giá trị của biến phụ thuộc. Hồi quy tuyến tính được sử dụng phổ biến và ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau, có thể kể đến như: Y tế, môi trường, giáo dục, kinh tế...

Phương trình của hồi quy tuyến tính đa biến có dạng như sau:

$$Y = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \ldots + \beta_i \cdot x_i + \varepsilon$$

Trong đó:
Y: là gía trị đầu ra dự đoán của mô hình.
x_1, x_2,..., x_i: là glà các biến độc lập.
ε: là sai số của mô hình.

B. Exponential Smoothing

Exponential Smoothing (ETS) là một phương pháp thống kê được sử dụng để mô hình và dự đoán dữ liệu chuỗi thời gian, được phát triển vào cuối những năm 1950 và được cải tiến bởi Holt (1957), Winters (1960). Nó dựa trên kỹ thuật smoothing mà giả định rằng giá trị hiện tại của chuỗi thời gian phụ thuộc vào các giá trị trước đó. ETS được áp dụng rộng rãi trong các lĩnh vực như tài chính, kinh tế học, và quản lý tồn kho.

Mô hình ETS bao gồm ba thành phần chính: trend (T), seasonal (S) và error (E). Trong đó trend (T) thể hiện sự thay đổi giá trị trong chuỗi, seasonal (S) thể hiện cho chu kỳ của dữ liệu, và error (E) là thành phần không thể dự đoán của chuỗi dữ liệu. Theo đó các giá trị T, S, E bao gồm các giá trị như sau:

- Error: "Additive" (A), "Multiplicative" (M).
- Trend: "None" (N), "Additive" (A), "Additive damped" (Ad), "Multiplicative" (M), "Multiplicative damped" (Md).
- Seasonal: "None" (N), "Additive" (A), "Multiplicative" (M).

Từ những tham số bên trên, ta có tổng số 30 mô hình ETS khác nhau, trong đó có 15 mô hình Additive error và 15 mô hình Multiplicative error. Để xác định mô hình phù hợp nhất trong số 30 mô hình ETS có thể sử dụng một số tiêu chí như Akaike (AIC). Công thức tính AIC như sau:

$$\text{AIC} = -2\left(\frac{LL}{T}\right) + \frac{2t_p}{T}$$

Trong đó:
LL = log likelihood.
t_p = Tổng số tham số.
T = Số lượng quan sát.

C. K-Nearest Neighbors

K-Nearest Neighbors (KNN) là một thuật toán học có giám sát, phi tham số. KNN có thể áp dụng được vào cả hai loại của bài toán Supervised learning là Classification và Regression.

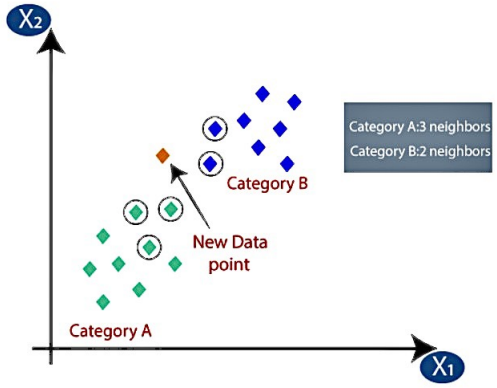
Ý tưởng của thuật toán là các điểm gần nhau trong không gian đặc trưng có xu hướng thuộc cùng một lớp hoặc có tính chất tương tự. Tham số k trong KNN đề cập đến số điểm được dán nhãn (neighbors) để phân loại.

Cách hoạt động của KNN như sau:

- Bước 1: Chọn tham số K.
- Bước 2: Tính khoảng cách từ điểm dữ liệu đến K số neighbors.
- Bước 3: Lấy K số neighbors gần nhất theo khoảng cách đã tính.
- Bước 4: Trong số K neighbors này, hãy đếm số điểm dữ liệu trong mỗi nhóm.
- Bước 5: Điểm dữ liệu mới sẽ thuộc nhóm có số lượng neighbors nhiều hơn.

Để tính khoảng cách k (k-distance), có thể sử dụng đo khoảng cách Euclidean, Manhattan, Minkowski,...

Có một số nhược điểm khi sử dụng KNN cho dữ liệu chuỗi thời gian. Một vấn đề chính là thuật toán nhạy cảm với việc



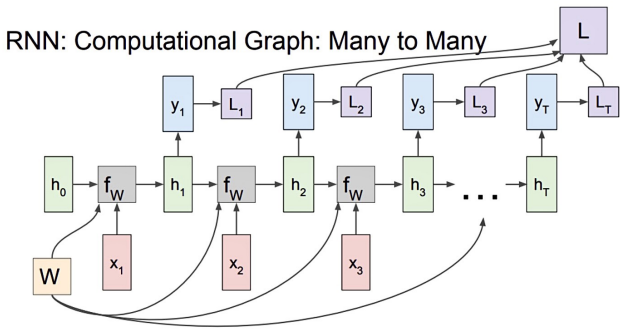
Hình 2: Minh họa cách hoạt động của K-Nearest Neighbors

lựa chọn số liệu khoảng cách, điều này có thể ảnh hưởng lớn đến hiệu suất của mô hình. Việc chọn tham số k cũng rất quan trọng và việc xác định giá trị tối ưu cho k có thể khó khăn. Ngoài ra, do thuật toán KNN lưu trữ tất cả dữ liệu đào tạo nên thuật toán này có thể tốn nhiều bộ nhớ và có thể không mở rộng tốt cho các tập dữ liệu lớn.

D. Recurrent Neural Networks

Ý tưởng chính của RNN (Recurrent Neural Networks) [8] 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 lẽ chúng 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. Trên lý thuyết, RNN có thể sử dụng được thông tin của một văn bản rất dài, tuy nhiên thực tế thì nó chỉ có thể nhớ được một vài bước trước đó mà thôi.



Hình 3: Computational Graph Many to Many

Nếu như mạng Neural Network chỉ là input layer x đi qua hidden layer h và cho ra output layer y với full connected giữa các layer thì trong RNN, x_t sẽ được kết hợp với hidden layer h_{t-1} bằng hàm f_W để tính toán ra hidden layer h_t hiện tại và output y_t sẽ được tính ra từ h_t , W là tập các trọng số và nó được xuất hiện ở tất cả các cụm, các L_1, L_2, ..., L_t là các hàm mất mát. Như vậy kết quả từ các quá trình tính toán trước đã được "nhớ" bằng cách kết hợp thêm h_{t-1} tính ra h_t để tăng độ chính xác cho những dự đoán ở hiện tại. Cụ thể quá trình tính toán được viết dưới dạng toán như sau:

$$h_t = f_W(h_{t-1}, x_t)$$

Hàm f_W chúng ta sẽ sử dụng hàm **tanh**, công thức trên sẽ trở thành:

$$h_t = \tanh(\mathbf{W_{hh}h_{t-1}} - \mathbf{W_{xh}x_t})$$
$$\mathrm{y}_t = W_{hy}h_t$$

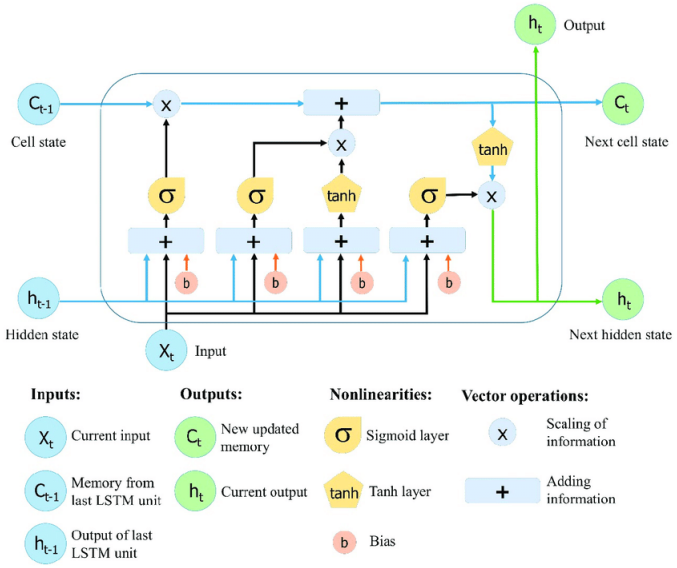
Đối với mạng Neural Network chỉ sử dụng một ma trận trọng số W duy nhất thì với RNN nó sử dụng 3 ma trận trọng số cho 2 quá trình tính toán: W_{hh} kết hợp với "bộ nhớ trước" h_{t-1} và W_{xh} kết hợp với x_t để tính ra "bộ nhớ của bước hiện tại" h_t từ đó kết hợp với W_{hy} để tính ra y_t.

Ngoài mô hình Many to Many như ta thấy ở trên thì RNN còn có các dạng khác như One to One, One to Many, Many to One và Many to Many.

E. Long Short Term Memory Network

Long Short Term Memory (LSTM) [9] là một loại mạng nơ-ron hồi quy (Recurrent Neural Network) được thiết kế đặc biệt nhằm giải quyết các bài toán về phụ thuộc xa (long-term dependency).

Thành phần quan trọng của LSTM là ô nhớ và các cổng (bao gồm cổng quên, cổng đầu vào và cổng đầu ra). Những cổng này quyết định những thông tin nào cần thêm, xóa và xuất ra khỏi ô nhớ.



Hình 4: Kiến trúc của Long Short Term Memory Network

Cổng quên sẽ xóa những thông tin không còn hữu ích khỏi ô nhớ. Phương trình của cổng quên là:

$$f_t = \sigma(\mathbf{W_f}.[\mathbf{h_{t-1}}, \mathbf{x_t}] + \mathbf{b_f})$$

Trong đó:

σ là hàm kích hoạt sigmoid

W_f và b_f lần lượt là trọng số và độ lệch của cổng quên.

Cổng đầu vào sẽ bổ sung những thông tin hữu ích vào ô nhớ. Phương trình của cổng đầu vào là:

$$i_t = \sigma(\mathbf{W_i}.[\mathbf{h_{t-1}}, \mathbf{x_t}] + \mathbf{b_i})$$
$$C_t = tanh(\mathbf{W_C}.[\mathbf{h_{t-1}}, \mathbf{x_t}] + \mathbf{b_C})$$
$$C_t = \mathbf{f_t} * \mathbf{C_{t-1}} + \mathbf{i_t} * \mathbf{C_t}$$

Trong đó:

C_{t-1} và C_t là ô nhớ lần lượt ở thời điểm t-1 và t W_c và b_c lần lượt là trọng số và tham số của ô nhớ.

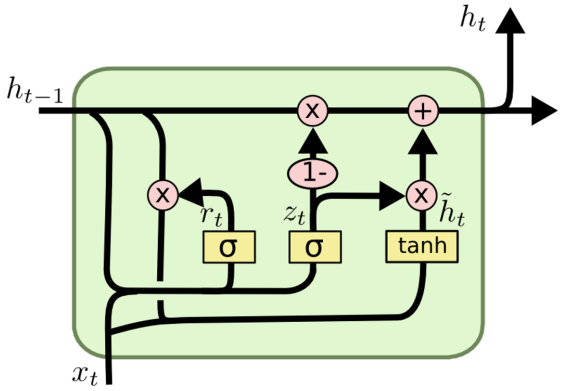
tanh là hàm kích hoạt tanh.

Cổng đầu ra: sẽ trích xuất thông tin hữu ích từ ô nhớ hiện tại để trình bày dưới dạng đầu ra. Phương trình của cổng đầu ra là:

$$o_t = \sigma(\mathbf{W_o}.[\mathbf{h_{t-1}}, \mathbf{x_t}] + \mathbf{b_o})$$

F. Gated Recurrent Unit

Gated Recurrent Unit (GRU) [10] là một biến thể của LSTM được thiết kế nhằm giải quyết vấn đề biến mất gradient đi kèm với mạng RNN mà vẫn cho hiệu suất tương đương. Để giải quyết vấn đề mất mát gradient của mạng RNN truyền thống, GRU sử dụng các cổng cập nhật và cổng cài đặt lại (update gate và reset gate) được mô tả ở Hình 5.



Hình 5: Kiến trúc của mạng Gated Recurrent Unit

Cổng cập nhật giúp mô hình xác định được lượng thông tin trong quá khứ (thông tin ở bước t-1) cần chuyển đến tương lai (bước t) và loại bỏ nguy cơ mất mát gradient. Cổng cài đặt lại ổng này được sử dụng để quyết định lượng thông tin trong quá khứ bị quyên đi. Các cổng cập nhật và cổng cài đặt lại, cũng các trạng thái bộ nhớ được thể hiện ở các công thức sau

$$z_t = \sigma(\mathbf{W_z}.[\mathbf{h_{t-1}}, \mathbf{x_t}])$$
$$r_t = \sigma(\mathbf{W_r}.[\mathbf{h_{t-1}}, \mathbf{x_t}])$$
$$\tilde{h}_t = tanh(\mathbf{W}.[\mathbf{r_t} * \mathbf{h_{t-1}}, \mathbf{x_t}])$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Công thức của cổng cài đặt lại r_t giống với công thức ở cổng update z_t đã nêu ở trên. Sự khác biệt chỉ là ở trọng số và mức sử dụng của cổng này. Thông tin nội dung của bộ nhớ mới sẽ sử dụng cổng cài đặt lại để lưu trữ thông tin có liên quan đến quá khứ. Và ở bước cuối cùng, đầu ra của mạng là vector h_t chứa toàn bộ thông tin ở tại thời điểm t và được truyền đi đến những thành phần tiếp theo.

G. Autoregressive Integrated Moving Average

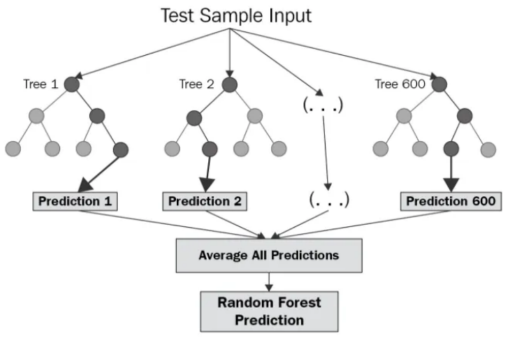
Autoregressive Integrated Moving Average (ARIMA) là một mô hình được sử dụng trong dự báo chuỗi thời gian và chỉ hoạt động tốt nhất nếu dữ liệu phụ thuộc nhiều vào thời gian.

Mô hình ARIMA được chia thành ba phần: quá trình tự hồi quy (AutoRegressive - AR), tích hợp sai phân (Integrated - I) và quá trình trung bình trượt (Moving Average - MA), tương ứng với ba tham số p, d và q là các số không âm. Trong đó:

- AR(q) - AutoRegressive là quá trình tìm mối quan hệ giữa dữ liệu hiện tại và dữ liệu quá khứ.
- MA(q) - Moving Average là quá trình tìm mối quan hệ giữa dữ liệu hiện tại và phần lỗi của quá khứ.
- I(d) - Integrated là hiệu giữa giá trị hiện tại và giá trị trước đó.

H. Random Forest Regression

Random Forest Regression là một thuật toán học có giám sát sử dụng phương pháp ensemble learning trong bài toán hồi quy. Mô hình hoạt động bằng cách kết hợp nhiều cây quyết định độc lập và kết hợp kết quả của chúng để đưa ra giá trị dự đoán cuối cùng.

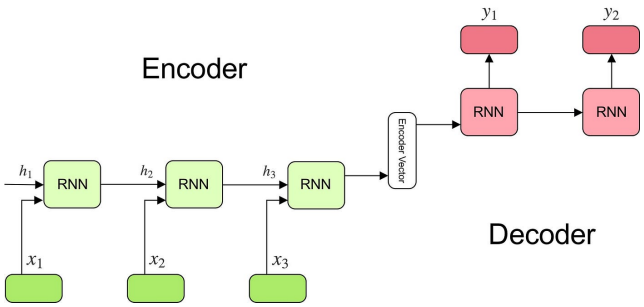


Hình 6: Kiến trúc của Random Forest

Mô hình xây dựng các cây quyết định bằng cách chia dữ liệu huấn luyện theo phương pháp Bootstrap Sampling. Một phần dữ liệu sẽ được lấy ngẫu nhiên từ tập dữ liệu để làm tập huấn luyện và phần còn lại được sử dụng làm tập xác thực của mỗi cây quyết định.

I. Sequence To Sequence

Sequence To Sequence (Seq2Seq) [11] là một loại mô hình mạng nerual trong bài toán học máy sử dụng kiến trúc chuỗi-sang-chuỗi của RNN (Recurrent Neural Network).Được sử dụng rộng rãi trong các tác vụ như dịch ngôn ngữ, tóm tắt văn bản và chú thích hình ảnh.Mô hình gồm 3 phần: bộ mã hóa (encoder), véc tơ trung gian và bộ giải mã (decoder).



Hình 7: Bộ mã hóa-bộ giải mã trong mô hình trình Squence To Sequence

Encoder:

- Đối với mỗi bước thời gian (mỗi đầu vào) t, trạng thái ẩn (vectơ ẩn) h được cập nhật theo đầu vào tại bước thời gian đó X[i].
- Các trạng thái ẩn h_i được tính theo công thức:

$$h_t = \mathbf{f}(\mathbf{W^{(hh)}h_{t-1}} + (\mathbf{W^{(hx)}x_t})$$

Encoder Vector:

- Đây là trạng thái ẩn cuối cùng được tạo ra từ phần mã hóa của mô hình. Được tính bằng công thức trên.
- Vectơ này nhằm đóng gói thông tin cho tất cả các yếu tố đầu vào để giúp bộ giải mã đưa ra dự đoán chính xác.
- Vectơ hoạt động như trạng thái ẩn ban đầu cho phần giải mã của mô hình.

Decoder:

- Bộ giải mã tạo chuỗi đầu ra bằng cách dự đoán đầu ra tiếp theo y_t với trạng thái ẩn h_t.
- Đầu vào cho bộ giải mã là vectơ ẩn cuối cùng thu được ở cuối mô hình bộ mã hóa.
- Mọi trạng thái ẩn h_t được tính theo công thức:

$$h_t = \mathbf{f}(\mathbf{W^{(hh)}h_{t-1}})$$

- Đầu ra y_t tại thời điểm bước t được tính theo công thức:

$$y_t = \mathbf{softmax}(\mathbf{W^S h_t})$$

Kết quả đầu ra được tính bằng cách sử dụng trạng thái ẩn ở bước thời gian hiện tại h_t cùng với trọng số W(S) tương ứng. Softmax được sử dụng để tạo một vectơ xác suất giúp xác định đầu ra cuối cùng.

J. Temporal Convolutional Network

Temporal Convolutional Network (TCN) [12] với kiến trúc mạng neural tích chập được sử dụng nhiều trong bài toán liên quan đên mô hình hóa chuỗi và có các ứng dụng nổi bật trong dự báo chuỗi thời gian. Việc áp dụng kiến trúc tích chập (convolution) giúp mô hình đạt hiệu suất tốt hơn so với các mạng RNN truyền thống, khi giảm thiểu vấn đề về bộ nhớ cũng như sự bùng nổ hoặc biến mất gradient.

Kiến trúc tích chập của mạng TCN mang tính nhân quả (causal), tức là không có sự rò rỉ thông tin từ tương lai về quá khứ. Giống như các biến thể mạng tích chập khác, TCN sử dụng các kernel chạy dọc theo chuỗi để rút trích và ánh xạ chuỗi đầu vào sang chuỗi đầu ra với độ dài tương ứng, giống với mạng RNN. Với phép toán tích chập giãn nở (dilation) cùng tính nhân quả (causal) cho phép mạng TCN ghi được nhiều giá trị trong lịch sử để thực hiện các dự báo cho tương lai.

V. Quá trình thực nghiệm

A. Thu thập và tiền xử lý dữ liệu

Sau khi đã xác định các loài hải sản phù hợp cho nghiên cứu, chúng tôi trích lọc tên của loài hải sản từ bộ dữ liệu ban đầu cùng các thuộc tính về thời gian và sản lượng bao gồm:

- **Last Catch Date**: Ngày đánh bắt cuối cùng trên biển.
- **Gross Weight**: Trọng lượng thô (đơn vị: Kg).
- **Round Weight**: Trọng lượng trước khi xử lý (đơn vị: Kg).
- **Product Weight:** Sản lượng thành phẩm (đơn vị: Kg)

Trong các thuộc tính về khối lượng đánh bắt, chúng tôi chọn thuộc tính Product Weight làm thuộc tính mục tiêu cho bài toán dự báo sản lượng khai thác hải sản, do đây là thuộc tính mang ý nghĩa là sản phẩm khai thác đầu ra và tạo nên lợi nhuận thông qua các hoạt động thương mại. Việc không chọn giá trị đánh bắt làm biến mục tiêu do đây là thông tin nhạy cảm được ẩn đi trong 12 tháng gần nhất, qua đó không phù hợp với yêu cầu của đồ án.

Thời gian đánh bắt thu thập cho đồ án được giới hạn từ ngày 01/01/2000 đến 16/06/2023. Do một loài cá có thể được đánh bắt nhiều lần trong ngày, bởi nhiều tàu khác nhau, nhóm tiếp cận bằng cách gom nhóm và tính tổng sản lượng theo từng ngày. Để thuận tiện cho việc tính toán và trình bày trong báo cáo, giá trị sản lượng khai thác được chuyển đổi từ đơn vị Kilogram (Kg) sang Tấn. Sau đó các điểm dữ liệu bị khuyết được xử lý bằng phép nội suy tuyến tính (linear interpolation) nhằm giảm thiểu những ảnh hưởng không mong muốn đến hình dạng của chuỗi thời gian.

Kết quả của quá trình thu thập và tiền xử lý là dataframe gồm 8568 dòng chứa thông tin đầy đủ về thời gian đánh bắt và sản lượng khai thác theo ngày của loài hải sản tương ứng.

B. Chuẩn bị dữ liệu

Bộ dữ liệu được chia thành 3 tập dữ liệu nhỏ gồm: tập huấn luyện (train), tập xác thực (validation) và tập kiểm thử (test) được chia thành 2 tỉ lệ khác nhau. Tỉ lệ đầu tiên 7:1:2, 70 % cho tập huấn luyện, 10 % cho tập xác thực và 20 % cho tập kiểm thử. Tỉ lệ thứ hai 6:2:2, 60 % cho tập huấn luyện, 20 % cho tập xác thực và 20 % cho tập kiểm thử. Dù với cách chia nào, tập kiểm thử cho nghiên cứu đều bắt đầu từ ngày 07/10/2018 và kết thúc ngày 16/06/2023. Nhằm nâng cao hiệu suất thực nghiệm và tối ưu chi phí tính toán, chúng tôi thực hiện bước chuẩn hóa dữ liệu nhằm biến đổi tập dữ liệu thu được về tập dữ liệu có giá trị trung bình bằng 0 và phương sai bằng 1.

Nhằm phục vu cho việc dự báo nhiều ngày tiếp theo trong tương lai, chúng tôi mô hình hóa bài toán về dạng tự hồi quy (autoregression), theo đó các giá trị trong quá khứ được sử dụng trong quá trình huấn luyện mô hình và dự báo sản lượng. Như vậy giá trị đầu vào và đầu ra cho việc huấn luyện

mô hình của mỗi tập dữ liệu sẽ có dạng lần lượt là (số mẫu, l) và (số mẫu,h), với mỗi mẫu có sản lượng l ngày trong quá khứ sẽ có tương ứng sản lượng h ngày trong tương lai. Trong nghiên cứu này, đối với mỗi tỉ lệ và mỗi loài hải sản, chúng tôi sử dụng các giá trị sản lượng từ 90 ngày gần nhất để dự báo 30 ngày tiếp theo.

C. Thiết lập phương pháp thực nghiệm

Hướng tiếp cận chuẩn bị dữ liệu trình bày ở phần trên được áp dụng ở trong hầu hết quá trình thực nghiệm các mô hình dự báo chuỗi thời gian ngoài trừ hồi quy tuyến tính, ETS và ARIMA nhằm đảm bảo về mặt ý nghĩa của các phương pháp này. Đối với các mô hình không phải Deep Learning, chúng tôi thiết lập và tinh chỉnh tham số mô hình nhằm đạt hiệu suất tối ưu trên tập validation.

Các mô hình học sâu được huấn luyện với batch size cố định là 64 trên 5 epoch, cùng sử dụng thuật toán Adam và hàm mất mát Mean Squared Error (MSE) cho quá trình tối ưu hóa trọng số mô hình. Các mô hình RNN, LSTM , GRU được khởi tạo với hai lớp có số chiều 32 và sử dụng hàm kích hoạt "tanh" nhằm tối ưu chi phí huấn luyện sẵn có. Mô hình Sequence-to-Sequence được thiết lập với hai lớp LSTM có số chiều 64 ở cả bộ Encoder và Decoder, trong khi các tham số mô hình TCN sử dụng kernel có kích thước 3×3 và có số lớp cùng số lượng filter ở mỗi lớp được tinh chỉnh trong quá trình thực nghiệm nhằm đảm bảo sự ổn định trên dữ liệu thực tế.

D. Công cụ sử dụng

Trong quá trình nghiên cứu, nhóm đã sử dụng Jupyter Notebook và Google Colaboratory để viết và thực thi code với ngôn ngữ được sử dụng là Python. Cụ thể hơn, nhóm đã sử dụng những thư viện phổ biến như pandas, numpy, matplotlib, sklearn, statsmodels, keras,... để xử lý dữ liệu, thực hiện phân tích và xây dựng các mô hình dự báo. Ngoài ra, nhóm cũng sử dụng các thư viện bigdl và json để hỗ trợ trong việc đào tạo mô hình và xử lý dữ liệu lớn.

E. Độ đo đánh giá

1) Mean Squared Error (MSE) - Root Mean Squared Error (RMSE): Mean Squared Error (MSE) là trung bình của bình phương các sai số, hoặc là sự khác biệt giữa giá trị dự báo và giá trị thực tế của biến mục tiêu. Giá trị MSE càng thấp, tức là sự khác biệt giữa giá trị dự báo và giá trị thực tế càng nhỏ thì mô hình dự báo càng tốt. MSE được tính theo công thức như sau:

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

Root Mean Squared Error (RMSE) là một biến thể của MSE, được tính bằng cách lấy căn bậc hai của MSE. Do đó, RMSE có đơn vị giống với đơn vị của biến mục tiêu, giúp dễ dàng hiểu và so sánh sai số dự báo trong ngữ cảnh thực tế. RMSE được tính theo công thức sau:

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

2) Mean Absolute Error (MAE): Mean Absolute Error (MAE) là trung bình của sai số tuyệt đối giữa giá trị dự báo và giá trị thực tế. Tương tự như MSE và RMSE, giá trị MAE càng nhỏ thì mô hình sẽ càng tốt. MAE được tính theo công thức sau:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y_i}|$$

3) Mean Absolute Percentage Error (MAPE): Mean Absolute Percentage Error (MAPE) là phần trăm sai số trung bình tuyệt đối giữa giá trị dự báo và giá trị thực tế. Cũng giống với MSE, RMSE hay MAE, giá trị MAPE càng nhỏ thì mô hình sẽ càng tốt. MAPE được tính theo công thức sau:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y_i}}{y_i}\right| \times 100$$

4) R-Squared (R2): R-Squared (R2) đo lường mức độ phù hợp của mô hình dự báo so với dữ liệu thực tế. Giá trị R2 nằm trong khoảng từ 0 đến 1 và càng gần về 1 thì mô hình dự báo càng tốt. R2 được tính theo công thức sau:

$$R2 = 1 - \frac{SSE}{SST}$$

Trong đó:

- SSE (Sum of Squared Errors) là tổng bình phương của sai số của mô hình. SSE được tính theo công thức sau:

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

- SST (Total Sum of Squares) là tổng bình phương của sai số của trung bình mẫu. SST được tính theo công thức sau:

$$SST = \sum_{i=1}^{n}(y_i - \bar{y})^2$$

VI. Kết quả thực nghiệm

Kết quả đánh giá hiệu suất của các phương pháp dự báo chuỗi thời gian trên tập kiểm thử được thể hiện ở Bảng II. Một cách tổng quát, các mô hình mạng học sâu cho thấy hiệu suất nổi trội hơn so với các mô hình dự báo chuỗi thời gian truyền thống dựa trên Học máy. Bên cạnh Linear Regression, mô hình ARIMA tiêu chuẩn cho thấy hiệu suất thực nghiệm thấp nhất trong các phương pháp trên cả 5 độ đo do không sử dụng đặc trưng mùa vụ (seasonal) nên đường dự báo chỉ có dạng đường thẳng. Dựa vào độ đo RMSE, các dự báo của các mô hình khi sử dụng tỉ lệ 7:1:2 cho quá trình thực nghiệm thường cho thấy hiệu suất đánh giá tốt hơn so với khi sử dụng 6:2:2 trên cả ba loài hải sản.

Đối với loài Northeast Arctic Cod, mô hình GRU đạt hiệu suất tốt nhất trên hầu hết các độ đo đánh giá ở cả hai tỉ lệ. Trên tỉ lệ 7:1:2, Mô hình cho kết quả RMSE tối ưu là 600.037 và R2 dương cho thấy sản lượng dự báo tương quan với sản lượng thật. Đây cũng mô hình duy nhất đạt hệ số R2 dương trong các phương pháp thực nghiệm. Với tỉ lệ 6:2:2, mô hình hoạt động kém hiệu quả hơn với độ lỗi RMSE tăng lên 684.755

Bảng II: Kết quả đánh giá hiệu suất thực nghiệm của các mô hình dự báo sản lượng khai thác các loài Northeast Arctic Cod, Northeast Arctic Haddock và Atlantic Halibut trên tập kiểm thử theo tỉ lệ 7:1:2 và 6:2:2

Species	Models	Test set (7:1:2)					Test set (6:2:2)				
		MSE	RMSE	MAE	MAPE	R2	MSE	RMSE	MAE	MAPE	R2
Northeast Arctic Cod	**Linear Regresion**	1034912.628	1017.307	883.749	14.985	-0.173	921784.005	960.096	788.843	12.585	**-0.044**
	ARIMA	1600066.973	1264.938	847.011	2260.829	-8.41E+09	1594772.248	1262.843	843.888	306.154	-3.57E+10
	ETS	726404.000	852.294	542.914	0.879	-11.776	2216239.367	1488.704	1390.549	0.681	-46.151
	Random Forest	431051.329	656.545	433.348	0.634	0.000	495427.628	703.866	450.574	0.686	-0.798
	KNN	548813.310	740.819	460.001	0.785	-0.831	577951.076	760.231	463.888	0.820	-1.766
	RNN	520759.965	721.637	471.312	1.187	-0.104	556621.934	746.071	484.589	0.836	-0.966
	LSTM	431384.417	656.799	435.951	0.611	-0.048	508350.584	712.987	463.613	0.682	-0.777
	GRU	**360044.542**	**600.037**	**417.107**	**0.540**	**0.364**	**468889.095**	**684.755**	**447.902**	**0.667**	-0.541
	Seq2Seq	517789.113	719.576	480.072	0.710	-0.367	563105.047	750.403	484.820	0.749	-1.504
	TCN	486388.940	697.416	487.671	0.937	-0.195	667919.115	817.263	536.107	1.112	-2.334
Northeast Arctic Haddock	**Linear Regresion**	59288.932	243.493	208.287	20.981	**-0.593**	81768.690	285.952	255.945	25.015	**-1.197**
	ARIMA	73773.715	271.613	191.214	533.449	-1.07E+10	73798.531	271.659	191.278	651.508	-1.03E+11
	ETS	36895.922	192.083	126.495	0.655	-25.101	37335.790	193.225	132.291	0.632	-29.424
	Random Forest	34783.812	186.504	118.843	0.645	-6.026	35483.030	188.369	119.037	0.661	-6.476
	KNN	39223.841	198.050	117.005	0.934	-10.091	40171.523	200.428	**117.984**	0.984	-9.979
	RNN	**33708.792**	**183.600**	115.716	0.658	-4.606	34779.758	186.493	119.519	0.661	-5.372
	LSTM	33970.680	184.311	**115.188**	0.662	-5.507	34928.605	186.892	122.370	0.601	-4.123
	GRU	34416.756	185.518	120.847	**0.618**	-4.031	**34127.421**	**184.736**	119.369	0.616	-4.700
	Seq2Seq	34328.160	185.279	116.627	0.672	-5.876	34447.921	185.602	118.996	0.657	-4.179
	TCN	36434.863	190.879	124.662	0.629	-25.711	44435.309	210.797	156.595	**0.597**	-4.114
Atlantic Halibut	**Linear Regresion**	33.641	5.800	4.226	2.134	**0.001**	33.754	5.810	4.304	2.221	**-0.002**
	ARIMA	71.006	8.426	6.112	21.198	-2.21E+07	72.218	8.498	6.208	32.696	-2.32E+08
	ETS	27.460	5.240	3.538	**0.593**	-9.697	28.241	5.314	3.799	**0.568**	-12.753
	Random Forest	31.111	5.578	3.497	0.783	-20.059	33.520	5.790	3.595	0.875	-27.133
	KNN	32.435	5.695	3.588	0.860	-26.290	33.036	5.748	3.588	0.880	-30.634
	RNN	28.288	5.319	3.479	0.838	-6.745	29.216	5.405	3.513	0.971	-10.765
	LSTM	31.591	5.621	3.510	1.053	-7.715	30.894	5.558	3.547	0.843	-10.871
	GRU	**26.349**	**5.133**	**3.331**	0.755	-4.324	30.756	5.546	3.512	0.850	-13.246
	Seq2Seq	28.788	5.365	3.460	0.678	-16.961	30.665	5.538	3.525	0.733	-30.106
	TCN	28.618	5.350	3.813	0.669	-5.884	**25.663**	**5.066**	**3.449**	0.619	-4.385

cho sản lượng dự báo. Dựa trên độ đo RMSE, chúng tôi kết luận mô hình GRU là mô hình tốt nhất cho việc dự báo sản lượng loài Northeast Arctic Cod, theo sau là mô hình Random Forest với RMSE đạt 656.545 ở tỉ lệ 7:1:2 và 450.574 ở tỉ lệ 6:2:2.

Đối với loài Northeast Arctic Haddock, các kết quả độ đo tốt nhất có sự phân bố đa dạng ở các mô hình thực nghiệm và tỉ lệ phân chia tập dữ liệu. Trong đó mô hình RNN đạt hiệu suất tốt nhất trên tỉ lệ 7:1:2 với RMSE đạt 183.600 theo sau là mô hình LSTM với RMSE đạt 184.311. Ở tỉ lệ 6:2:2, mô hình GRU cho kết quả RMSE tối ưu đạt 184.736, theo sau là mô hình Sequence-To-Sequence với RMSE đạt 185.602.

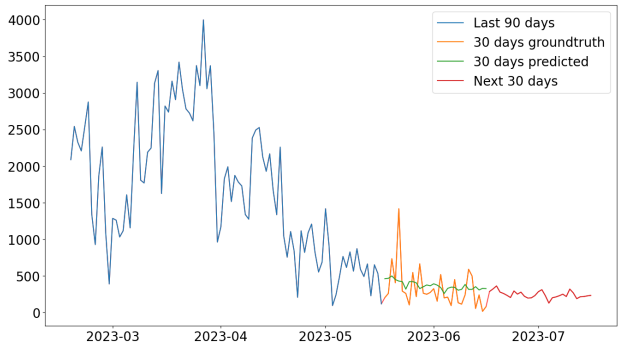
Đối với loài Atlantic Halibut, GRU tiếp tục cho thấy sự thống trị khi đạt kết quả tốt nhất trên ba độ đo MSE, RMSE, MAE trên tỉ lệ 7:1:2, trong đó RMSE tối ưu đạt 5.133. Mô hình cho kết quả cạnh tranh với GRU là ETS với độ lỗi RMSE đạt 5.240, cao hơn 0.107 đơn vị lỗi so với kết quả RMSE của GRU. Khi áp dụng tỉ lệ 6:2:2, Mô hình TCN với kiến trúc mạng tích chập cho kết quả RMSE tốt nhất 5.066, cao hơn hiệu suất của GRU ở tỉ lệ 7:1:2. Mô hình tốt thứ hai là ETS với hiệu suất đạt 5.405 trên RMSE.

Sau khi đã xác định hai mô hình tốt nhất cho mỗi tỉ lệ thực nghiệm ở mỗi loài hải sản, chúng tôi tiến hành áp dụng chúng để dự báo sản lượng khai thác của các loài này trong 30 ngày tiếp theo tính từ 17/06/2023, và trực quan kết quả dự báo dưới dạng biểu đường, trong đó chứa thông tin về sản lượng khai thác trong 90 ngày gần nhất cho dự báo 30 ngày cuối cùng, kết quả dự báo 30 ngày cuối cùng cùng giá trị thực, và sau cùng là kết quả dự báo 30 ngày tiếp theo trong tương lai.

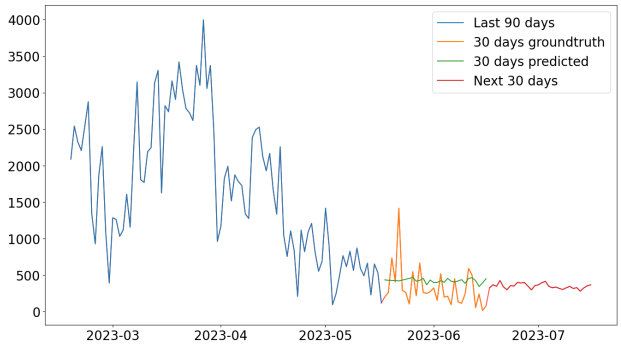
VII. Kết luận và hướng phát triển

Trong bài nghiên cứu về dự báo sản lượng khai thác hải sản sử dụng các kỹ thuật phân tích chuỗi thời gian, chúng tôi đã sử dụng 10 thuật toán khác nhau gồm Linear Regression, KNN, Random Forest, ETS, ARIMA, RNN, LSTM, GRU, Seq2Seq và TCN để thực hiện dự báo sản lượng khai thác của ba loài hải sản là Northeast Arctic Cod, Northeast Arctic Haddock và Atlantic Halibut. Về tổng quát, kết quả nghiên cứu cho thấy các mô hình mạng học sâu cho thấy hiệu suất nổi trội hơn so với các mô hình dự báo chuỗi thời gian truyền thống dựa trên học máy.

Đối với loài Northeast Arctic Cod, mô hình GRU đạt hiệu suất tốt nhất trên hầu hết các độ đo đánh giá ở cả hai tỉ lệ. Đối với loài Northeast Arctic Haddock, mô hình RNN đạt hiệu suất tốt nhất trên tỉ lệ 7:1:2. Ở tỉ lệ 6:2:2, mô hình GRU cho hiệu suất tốt nhất. Đối với loài Atlantic Halibut, mô hình GRU

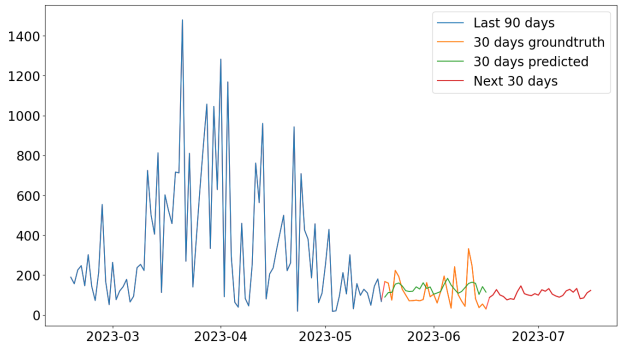


(a) GRU

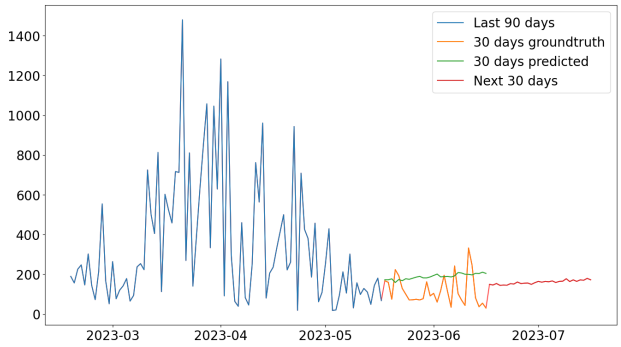


(b) Random Forest

Hình 8: Trực quan hóa sản lượng dự báo trong 30 ngày tiếp theo cho loài *Northeast Arctic Cod* (r = 7:1:2)



(a) RNN



(b) LSTM

Hình 9: Trực quan hóa sản lượng dự báo trong 30 ngày tiếp theo cho loài *Northeast Arctic Haddock* (r = 7:1:2)

đạt hiệu suất tốt nhất trên tỉ lệ 7:1:2. Ở tỉ lệ 6:2:2, mô hình TCN cho hiệu suất tốt nhất.

Bên cạnh các đóng góp trên, nghiên cứu này có một số mặt hạn chế nhất định khi chưa thực hiện các bước tiền xử lý khác như loại bỏ các giá trị ngoại lai ảnh hưởng đến kết quả dự báo. Bên cạnh đó, với sự hạn chế về thời gian và số lượng thuật toán khá nhiều nên các mô hình và phương pháp chưa được tỉnh chỉnh một cách tối ưu và nghiên cứu thiếu đi các thử nghiệm khác sử dụng các khoảng thời gian khác nhau trong quá khứ để đưa ra dự báo tốt hơn.

Trong tương lai, chúng tôi sẽ tiếp tục tìm hiểu và nghiên cứu, giải quyết các hạn chế nêu trên. Bên cạnh đó, có thể tìm hiểu và thêm những đặc trưng về môi trường như nhiệt độ, độ mặn của nước biển vào thực nghiệm để cải thiện hiệu suất dự báo.

Acknowledgment

Nhóm xin được xin gửi lời cảm ơn đến thầy PGS. TS. Nguyễn Đình Thuân, anh Nguyễn Minh Nhựt và chị Nguyễn Thị Việt Hương đã hỗ trợ và hướng dẫn tận tình để nhóm có thể hoàn thành nghiên cứu và đạt được kết quả đề ra.

Tài liệu

[1] S. Shrivastri, K. Alakkari, P. Lal, A. Yonar, and S. Yadav, "A comparative study between (arima -ets) models to forecast wheat production and its importance's in nutritional security," *Journal of Applied Sciences*, vol. 1, pp. 25–37, 07 2022.

[2] M. Kumar and M. Thenmozhi, "Forecasting stock index movement: A comparison of support vector machines and random forest," *SPGMI: Compustat Fundamentals (Topic)*, 2006.

[3] R. Chowdhury, M. A. Rahman, M. S. Rahman, and M. R. C. Mahdy, "An approach to predict and forecast the price of constituents and index of cryptocurrency using machine learning," *Physica A: Statistical Mechanics and its Applications*, 2019.

[4] Q. Zhang, H. Wang, J. Dong, G. Zhong, and X. Sun, "Prediction of sea surface temperature using long short-term memory," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 10, pp. 1745–1749, 2017.

[5] H. Zhang, S. Li, Y. Chen, J. Dai, Y. Yi, and B. Ding, "A novel encoder-decoder model for multivariate time series forecasting," *Intell. Neuroscience*, vol. 2022, jan 2022. [Online]. Available: https://doi.org/10.1155/2022/5596676

[6] H. Y. Bako, M. S. Rusiman, I. L. Kane, and H. M. Matias-Peralta, "Predictive modeling of pelagic fish catch using seasonal arima models," *Agriculture, Forestry and Fisheries*, vol. 2, p. 136, 2013.

[7] M.-T. Thai, T.-N. Chu-Ha, T.-A. Vo, and T.-H. Do, "An approach to recommend fishing location and forecast fish production by using big data analysis and distributed deep learning," in *2022 RIVF International Conference on Computing and Communication Technologies (RIVF)*, 2022, pp. 322–327.

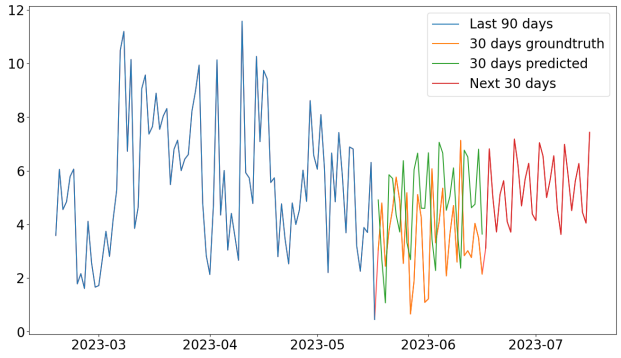
[8] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

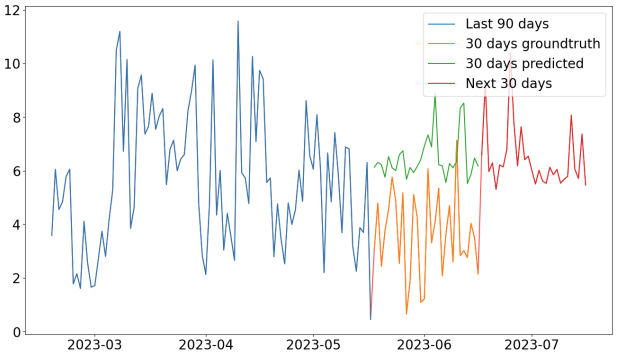
[10] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014. [Online]. Available: http://arxiv.org/abs/1412.3555

[11] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *CoRR*, vol. abs/1409.3215, 2014. [Online]. Available: http://arxiv.org/abs/1409.3215

[12] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *CoRR*, vol. abs/1803.01271, 2018. [Online]. Available: http://arxiv.org/abs/1803.01271

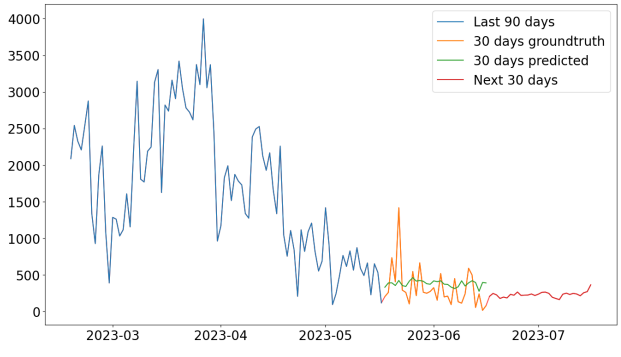


(a) GRU

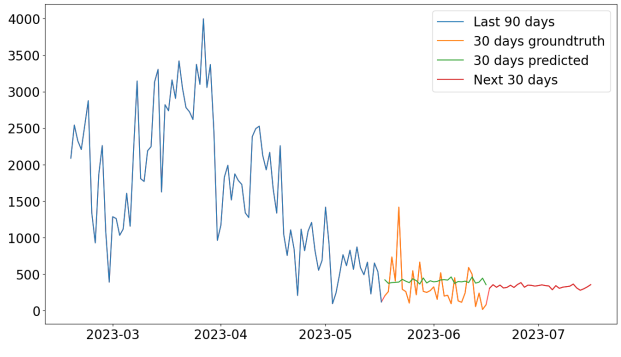


(b) ETS

Hình 10: Trực quan hóa sản lượng dự báo trong 30 ngày tiếp theo cho loài *Atlantic Halibut* (r = 7:1:2)

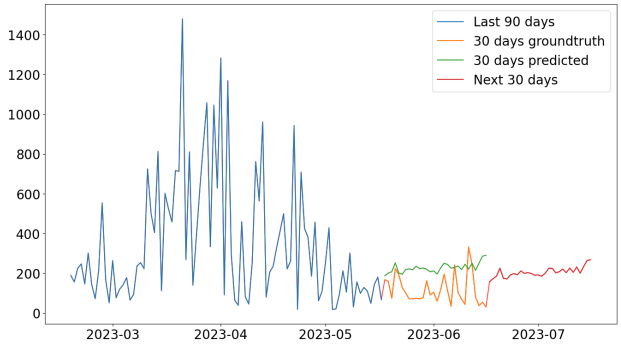


(a) GRU

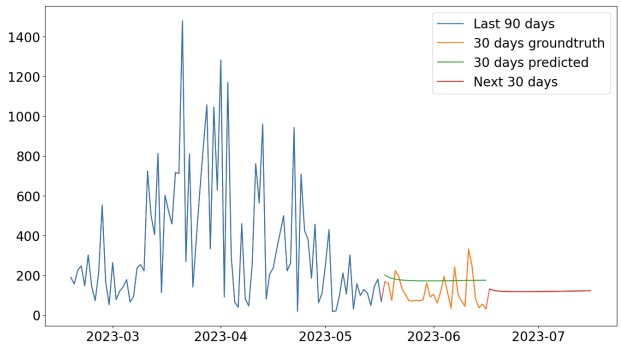


(b) Random Forest

Hình 11: Trực quan hóa sản lượng dự báo trong 30 ngày tiếp theo cho loài *Northeast Arctic Cod* (r = 6:2:2)

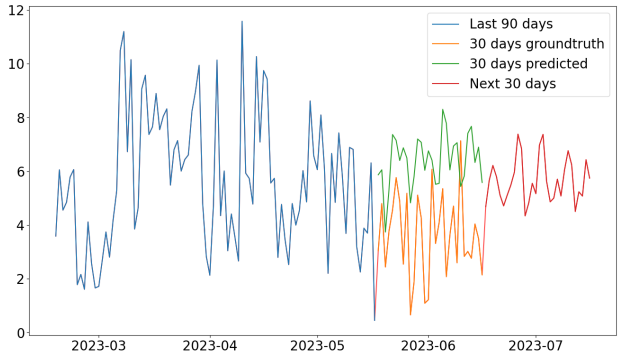


(a) GRU

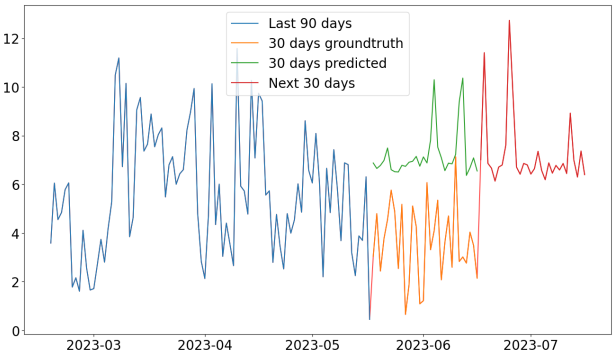


(b) Seq2Seq

Hình 12: Trực quan hóa sản lượng dự báo trong 30 ngày tiếp theo cho loài *Northeast Arctic Haddock* (r = 6:2:2)



(a) TCN



(b) ETS

Hình 13: Trực quan hóa sản lượng dự báo trong 30 ngày tiếp theo cho loài *Atlantic Halibut* (r = 6:2:2)