MỤC LỤC

DANH SÁCH HÌNH VẼ

DANH SÁCH BẢNG

LỜI CẢM ƠN

Đầu tiên, chúng em xin gửi lời cảm ơn chân thành đến Trường Đại học Khoa học Tự nhiên, Khoa Toán - Tin học và Bộ môn Ứng dụng Tin học đã tổ chức, sắp xếp chương trình giảng dạy một cách khoa học và logic, giúp chúng em có được kiến thức chuyên môn vững vàng để hoàn thành khóa luận này. Đặc biệt, chúng em xin bày tỏ lòng biết ơn sâu sắc tới hai giảng viên hướng dẫn - Thầy Ngô Minh Mẫn và Thầy Vũ Đức Thịnh. Sự tận tâm và những kiến thức quý báu mà các Thầy đã truyền đạt cho chúng em trong suốt thời gian qua là nguồn động lực to lớn để chúng em đạt được kết quả như ngày hôm nay. Chúng em cũng xin gửi một lời cảm ơn chân thành đến các anh chị khóa K2019 và các bạn cùng khóa K2020 đã luôn sẵn sàng hỗ trợ, giúp đỡ chúng em trong suốt quá trình thực hiện khóa luận này.

Trải qua khoảng thời gian học tập tại Trường và tham gia khóa luận, chúng em đã học hỏi được nhiều kiến thức bổ ích và phát triển tinh thần học tập nghiêm túc, hiệu quả. Những kiến thức và kinh nghiệm này chắc chắn sẽ là hành trang quý báu giúp chúng em vững bước trong tương lai.

Ngành Khoa học Dữ liệu là một ngành học rất thực tế và bổ ích, đáp ứng nhu cầu thực tiễn của sinh viên. Tuy nhiên, do kiến thức còn hạn chế và khả năng tiếp thu thực tế còn nhiều bỡ ngỡ, chúng em đã cố gắng hết sức nhưng chắc chắn khóa luận này vẫn khó tránh khỏi những thiếu sót. Chúng em mong nhận được sự xem xét và góp ý từ quý Thầy/Cô để bài làm của chúng em được hoàn thiện hơn.

Nhóm chúng em xin chân thành cảm ơn!

GIỚI THIỆU

1. Lý do chọn đề tài

Trong bối cảnh thị trường tài chính, dữ liệu chứng khoán luôn biến động liên tục, đòi hỏi sự chú ý và phân tích sâu sắc để các nhà đầu tư có thể đưa ra những quyết định đúng đắn. Sự biến động này không chỉ thể hiện qua những thay đổi nhỏ hàng ngày mà còn bao gồm cả những điểm bất thường, hay còn gọi là những điểm thay đổi (change-point), là những điểm bắt đầu có sự thay đổi trong xu hướng tăng/giảm của thị trường tài chính, chúng có thể gây ra những biến động lớn và không lường trước được.

Việc xác định chính xác các điểm bất thường này đóng vai trò vô cùng quan trọng trong việc dự báo xu hướng và quản trị rủi ro trên thị trường của các nhà đầu tư. Khi nhận diện được những thời điểm mà xu hướng của thị trường thay đổi đột ngột, họ có thể đưa ra các chiến lược phù hợp nhằm tối ưu hóa lợi nhuận và giảm thiểu rủi ro.

Nhận thấy tầm quan trọng và ứng dụng thực tiễn của bài toán này, nhóm chúng tôi quyết định chọn "Phát hiện điểm bất thường trong thị trường tài chính Việt Nam" làm đề tài khóa luận của mình. Nghiên cứu này không chỉ mang lại lợi ích cho bản thân trong việc nâng cao kiến thức chuyên môn mà còn có thể đóng góp vào việc phát triển các phương pháp phân tích dữ liệu hiệu quả, hỗ trợ cho các quyết định đầu tư trên thị trường tài chính..

2. Phạm vi nghiên cứu và phương hướng tiếp cận

Phạm vi nghiên cứu

Phạm vi nghiên cứu của đề tài "Phát hiện điểm bất thường trong thị trường tài chính Việt Nam" là để tìm kiếm hướng tiếp cận mới và hiệu quả trong việc dự báo và phát hiện các điểm bất thường trong thị trường tài

chính Việt Nam, cụ thể ở đây là áp dụng trên chỉ số VN30-Index, một chỉ số được tính toán dựa trên tập hợp cổ phiếu VN30. Trong đó, VN30 là tập hợp 30 cổ phiếu hàng đầu trên thị trường chứng khoán Việt Nam. Được xây dựng dựa trên chỉ số vốn hóa và tính thanh khoản, VN30 đại diện cho những công ty có quy mô lớn và ảnh hưởng lớn đến thị trường chứng khoán. Các công ty trong danh mục VN30 được chọn lựa kỹ lưỡng dựa trên các tiêu chuẩn nghiêm ngặt và được triển khai bởi Sở Giao dịch Chứng khoán TP. Hồ Chí Minh (HOSE).

Phương hướng tiếp cận

Do dữ liệu chuỗi thời gian thường có kích thước lớn và đặc điểm phức tạp, các nhà nghiên cứu đã tích cực phát triển các mô hình học sâu đặc biệt dành riêng cho dữ liệu chuỗi thời gian. Với ưu điểm của việc xử lý và tìm hiểu được cấu trúc dữ liệu phi tuyến, các mô hình học sâu như Transformer đã được sử dụng để xử lý dữ liệu chuỗi thời gian trong việc dự báo và phát hiện điểm bất thường, cũng như xử lý dữ liệu của thị trường chứng khoán VN30 và chỉ số VN30-Index.

3. Ý nghĩa khoa học và thực tiễn của đề tài

- Nghiên cứu và đưa ra tài liệu tổng hợp liên quan đến vấn đề dự báo và phát hiện điểm bất thường trong dữ liệu chuỗi thời gian cho thị trường tài chính Việt Nam.
- Cung cấp một quy trình phân tích, xử lý dữ liệu chứng khoán và xây dựng mô hình dự báo, phát hiện điểm bất thường.

4. Cấu trúc bài báo cáo

Bố cục của bài báo cáo được trình bày như sau:

- **Chương 1 - Kiến thức liên quan:** Giới thiệu các kiến thức có liên quan đến đề tài, bao gồm các bài toán trong chuỗi thời gian, các mô

hình học sâu thường được sử dụng.

- **Chương 2 - Phương pháp đề xuất:** Trình bày phương pháp được đề xuất để giải quyết vấn đề đưa ra của đề tài.
- **Chương 3 - Thực nghiệm:** Thông tin về bộ dữ liệu được sử dụng sử dụng và quá trình thực nghiệm phương pháp đề xuất, đánh giá mô hình.
- **Chương 4 - Kết luận:** Tổng kết bài nghiên cứu và đưa ra các nhận định và hướng pháp triển tương lai.

CHƯƠNG 1:

KIẾN THỨC LIÊN QUAN

1.1 Dữ liệu chuỗi thời gian

1.1.1 Định nghĩa và phân loại

Dữ liệu chuỗi thời gian là một loại dữ liệu với hình thức phổ biến nhất là một chuỗi các quan sát được ghi nhận liên tục và được lập chỉ mục hoặc sắp xếp theo thứ tự thời gian. Đây cũng chính là đặc điểm tạo nên sự khác biệt quan trọng của loại dữ liệu này khi so sánh với các loại dữ liệu khác.

Dữ liệu chuỗi thời gian thường được chia thành hai loại: Chuỗi thời gian đơn biến (Univariate Time Series) và chuỗi thời gian đa biến (Multivariate Time Series)[7]

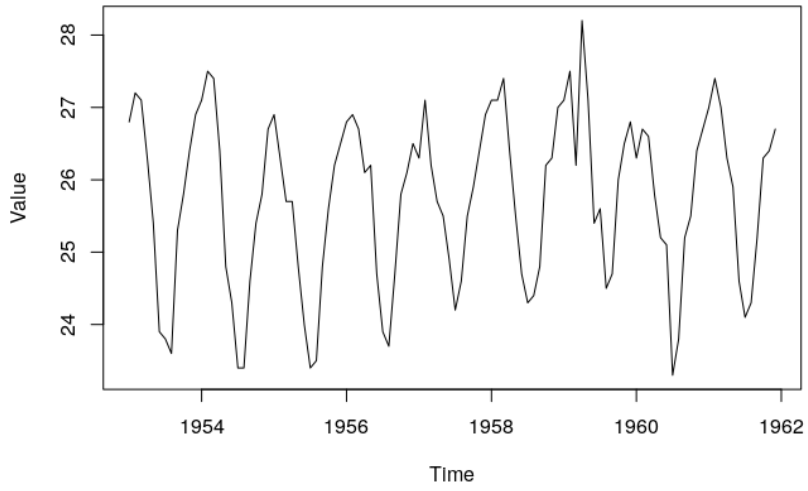
Chuỗi thời gian đơn biến

Chuỗi thời gian đơn biến (Univariate Time Series) là một chuỗi dữ liệu biểu thị sự thay đổi theo thời gian của một biến duy nhất [7], như minh họa trong Hình 1.1. Chúng ta có thể nhắc đến việc ghi lại nhiệt độ trung bình theo từng năm như là một ví dụ cho chuỗi thời gian đơn biến.

Một chuỗi thời gian đơn biến X với t bước thời gian có thể được biểu diễn dưới dạng một chuỗi dữ liệu có thứ tự như sau:

$$X = (x_1, x_2, ..., x_t)$$

trong đó x_i đại diện cho dữ liệu tại bước thời gian thứ $i \in T$ và $T = \{1, 2, \ldots, t\}$.

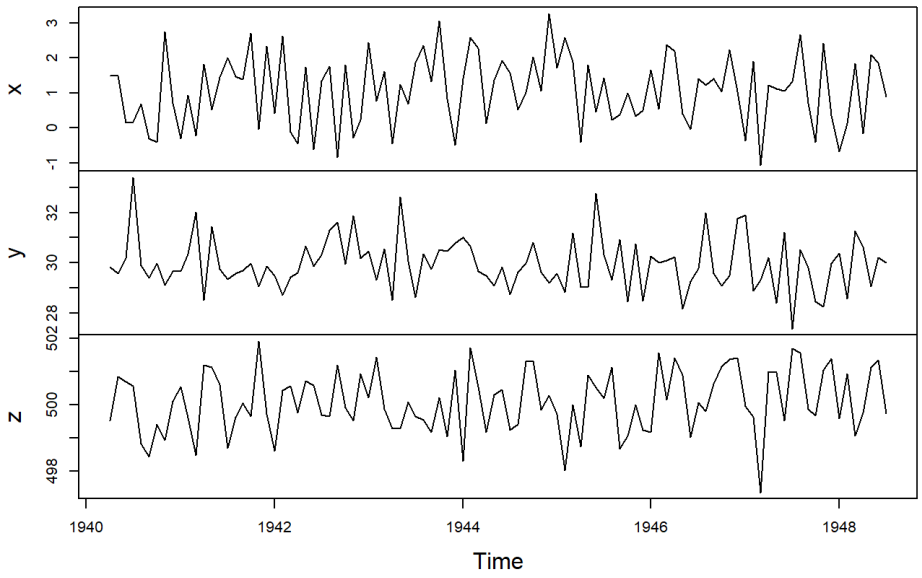


Hình 1.1: Chuỗi thời gian đơn biến

Chuỗi thời gian đa biến

Chuỗi thời gian đa biến (Multivariate Time Series) biểu thị sự thay đổi theo thời gian cho nhiều biến khác nhau cùng một lúc. Trong đó, mỗi biến sẽ bị ảnh hưởng bởi cả giá trị trong quá khứ, thường được gọi là phụ thuộc thời gian (temporal dependencies), và các biến khác dựa trên mối quan hệ tương quan của chúng. Mối quan hệ tương quan giữa các biến khác nhau được gọi là phụ thuộc không gian (spatial dependencies) hoặc phụ thuộc liên biến (intermetric dependencies) [7]. Chúng ta có thể xem xét đến việc ghi lại áp suất không khí, nhiệt độ và độ ẩm mỗi giờ trong ngày như là một ví dụ trong thực tế về một chuỗi thời gian đơn biến. Một ví dụ về chuỗi thời gian đa biến với ba chiều là áp suất không khí, nhiệt độ và độ ẩm qua từng năm được minh họa trong Hình 1.2.

Xét một chuỗi thời gian đa biến được biểu diễn dưới dạng một chuỗi các vector X. Mỗi vector tại thời điểm i được ký hiệu là x_i, có d chiều, được biểu diễn như sau:



Hình 1.2: Chuỗi thời gian đa biến

$$\begin{aligned} \mathrm{X} &= (\mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_\mathrm{t}) \\ &= \Big((x_1^1, x_1^2, ..., x_1^d), (x_2^1, x_2^2, ..., x_2^d), ..., (x_t^1, x_t^2, ..., x_t^d) \Big) \end{aligned}$$

trong đó mỗi x_i^j là một quan sát tại thời điểm i cho chiều thứ j, với $j = 1, 2, \cdots, d$.

1.1.2 Bài toán dự báo trong dữ liệu chuỗi thời gian

Dự báo chuỗi thời gian là một quá trình sử dụng phương pháp thống kê và mô hình hóa để phân tích các chuỗi dữ liệu được ghi dấu thời gian. Quá trình này bao gồm việc xây dựng các mô hình thông qua việc phân tích dữ liệu lịch sử, sau đó sử dụng những mô hình này để thực hiện quan sát và đưa ra quyết định chiến lược cho tương lai. Tuy nhiên, sự phức tạp của dữ liệu và những biến đổi không thể lường trước có thể làm cho kết quả dự báo không chính xác. Thông qua việc dự báo, người ta có thể xác định được những kết quả nào có khả năng xảy ra hơn hoặc ít xảy ra hơn so với các kết quả khác.

Dự báo chuỗi thời gian có nhiều ứng dụng trong nhiều lĩnh vực và ngành

công nghiệp khác nhau, từ dự báo thời tiết, khí hậu, đến kinh tế, y tế, và các dự báo trong nghiên cứu xã hội. Trong một số ngành, mục tiêu chính của việc phân tích chuỗi thời gian chính là để tạo điều kiện cho việc dự báo.

Tuy nhiên, do những mô hình dữ báo này không thể nhận diện được những thay đổi đột ngột trong quá khứ vì nhiệm vụ chính của nó là tập trung học những quy luật (patterns) tổng quát nên có thể dẫn đến dự báo sai lệch trong thực tế. Vì vậy, bài toán xác định điểm bất thường được ra đời nhằm giải quyết vấn đề này.

1.1.3 Bài toán xác định điểm bất thường trong dữ liệu chuỗi thời gian

Việc phát hiện những điểm dữ liệu bất thường cực kì quan trọng, tùy vào từng tình huống cụ thể, các nhà quản lý tài chính hoặc người dùng có thể đưa ra những quyết định phù hợp. Ví dụ, địa chỉ IP của một người dùng đăng nhập vào một mạng xã hội đã thay đổi so với trước đó, điều này có thể ngụ ý rằng tài khoản đang có nguy cơ bị đánh cắp. Nếu tài khoản bất thường có thể được phát hiện chính xác, các biện pháp thích hợp có thể được thực hiện để bảo vệ an toàn tài khoản, do đó rủi ro có thể được tránh hiệu quả.

Có 3 loại bài toán phát hiện bất thường:

- Bất thưởng điểm (Point anomaly): Là một điểm dữ liệu hoặc một chuỗi dữ liệu đột ngột chệch khỏi mức trung bình của dữ liệu. Các hiện tượng như vậy có thể xuất hiện như là nhiễu tạm thời và thường là do các lỗi của cảm biến hoặc hoạt động không bình thường của hệ thống.

- Bất thường ngữ cảnh (Contextual anomaly): Biểu thị một điểm dữ liệu hoặc một chuỗi dữ liệu được quan sát trong một thời gian ngắn nhưng không chệch khỏi phạm vi bình thường của dữ liệu. Tuy nhiên, khi xem xét ở một ngữ cảnh cụ thể, các điểm dữ liệu này không phù hợp với

hình dạng dự kiến. Các bất thường ngữ cảnh khó phát hiện hơn so với bất thường điểm.

- Nhóm bất thường (Collective anomaly): Loại này đề cập đến một tập hợp các điểm dữ liệu mà nên được coi là bất thường vì chúng đang dần dần thể hiện một hình dạng khác biệt so với dữ liệu bình thường. Khi xem xét chúng một cách riêng biệt thì có vẻ như không gây ra rắc rối, nhưng về tổng thể thì chúng có thể gây ra sự nghi ngờ.

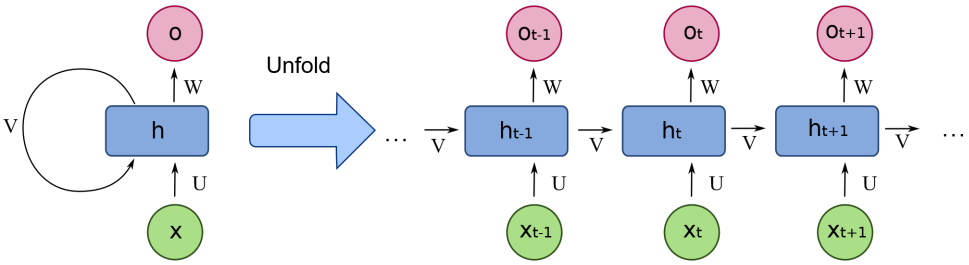
Việc phát hiện điểm thay đổi có ích trong việc mô hình hóa và dự đoán chuỗi thời gian và được tìm thấy trong các lĩnh vực ứng dụng như giám sát tình trạng y tế, phát hiện biến đổi khí hậu, phân tích giọng nói và hình ảnh, và phân tích hoạt động của con người.

Vì bài toán phát hiện bất thường trong ngữ cảnh thường khó phát hiện, đòi hỏi phải kết hợp với các chuyên gia trong lĩnh vực kinh tế. Nên trong phạm vi của khóa luận này, ta chỉ tập trung vào bài toán phát hiện điểm bất thường và nhóm bất thường trong dữ liệu chuỗi thời gian được đưa ra từ việc huấn luyện các mô hình học sâu.

1.2 Một số mô hình học sâu đơn giản cho bài toán chuỗi thời gian

1.2.1 Recurrent nơ-ron networks (RNNs)

Kiến trúc nơ-ron Network đơn giản nhất chỉ bao gồm các lớp Fully-connected thường dùng cho bài toán hồi quy (Regression) giúp học các mối quan hệ phi tuyến giữa các đặc trưng, hay mở rộng hơn là mạng Convolutional nơ-ron Network nhằm trích xuất các thông tin lân cận được áp dụng trong xử lý ảnh. Tuy nhiên cả 2 kiến trúc này đều không hiệu quả đối với dữ liệu chuỗi thời gian hoặc dữ liệu có tính tuần tự Sequence to Sequence (seq2seq). Bằng cách tạo ra một vùng nhớ và trích lại những thông tin trong



Hình 1.3: Kiến trúc tổng quát của RNNs

quá khứ, Recurrent nơ-ron network (RNN) [16] có thể "nhớ" được những gì nó đã từng học cho dạng dữ liệu này.

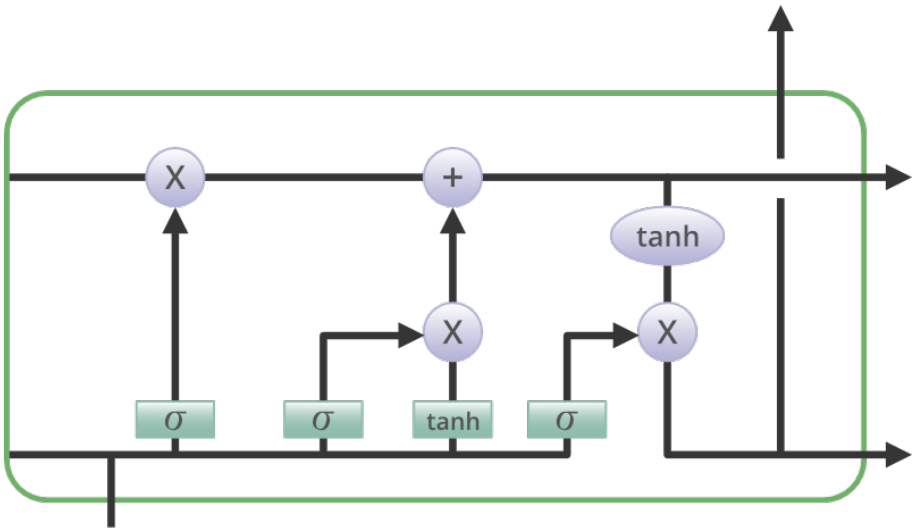
RNN được gọi là "tái diễn" (Recurrent) bởi lẽ chúng thực hiện cùng một tác vụ cho tất cả các phần tử của một chuỗi với đầu ra phụ thuộc vào cả các phép tính trước đó. Kiến trúc của RNN được mô tả trong Hình 1.3.

Điểm tạo nên sự khác biệt giữa RNN và một mạng Fully-connected truyền thống nằm ở việc RNN có thể "chia sẻ trọng số" (sharing weights) điều này giúp nó có thể học được thông tin trước đó, tạo nên một bộ nhớ, tuy nhiên bộ nhớ này chỉ trong ngắn hạn (short-term memory). Việc chia sẻ trọng số này cũng vô tình dễ dẫn đến vanishing/exploding gradient hơn so với nơ-ron Network thông thường. Điều này là động lực để các kiến trúc khác ra đời.

1.2.2 Long short-term memory (LSTM)

Long short-term memory[15] được sinh ra để khắc phục những hạn chế còn tồn đọng của RNN. Điểm mạnh của LSTM là nó có thể học được những thông tin trong dài hạn (long-term), cũng như quyết định thông tin nào quan trọng, chọn lọc để học (selectively learning).

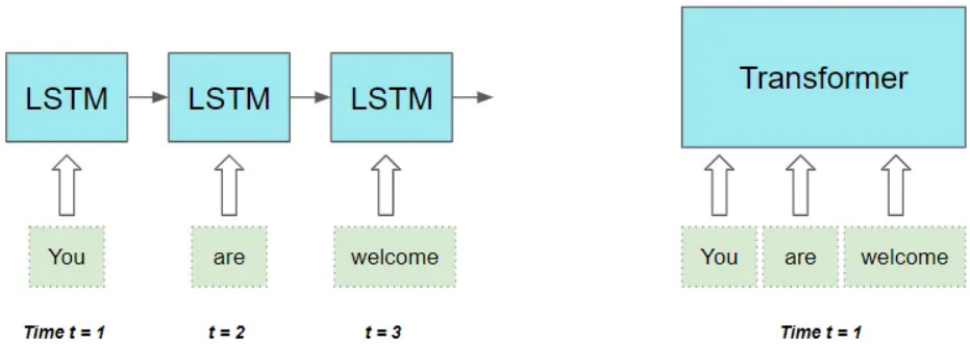
Kiến trúc của LSTM chỉ là mở rộng từ kiến trúc của RNN, nó chỉnh sửa bên trong khối h_t. Thay vì chỉ sử dụng một hàm kích hoạt *Tanh* như RNN với mục đích cho biết lượng thông tin học được (giá trị 1) hay quên đi (giá



Hình 1.4: Kiến trúc tổng quát của LSTM

trị -1), LSTM được thiết kế thêm vào 3 cổng kích hoạt *Sigmoid* nhằm phục vụ cho từng mục đích riêng biệt Hình 1.4.

Nếu RNN chỉ nhận một đầu vào là thông tin tại thời điểm x_t thì LSTM mở rộng lên hai đầu vào (nếu không tính input x_t). Đầu vào Long-term và Short-term. Đầu Long-term chính là khối Cell State C_t chạy xuyên xuốt network để mang thông tin từ quá khứ đến hiện tại. Điều này giúp LSTM tránh xảy ra hiện tượng vanishing do Cell State không phải đi qua một hàm kích hoạt nào (điều này tương tự với skip connection). Đầu vào Short-term chứa 3 cổng là Forget gate, Input gate, Output gate nhằm xác định lượng thông tin cần phải quên đi, lượng thông tin học được trong thời điểm hiện tại và cuối cùng là lượng thông tin cần thiết làm đầu vào cho Short-term trong thời điểm kế tiếp. Nhờ việc tách ra 2 đường đầu vào như trên, LSTM có thể hạn chế vấn đề suy biến gradient hoặc gây bùng nổ gradient (Vanishing/Exploding Gradient) do 2 đường này ảnh hưởng vào hỗ trợ lẫn nhau.



Hình 1.5: Transformer cho phép xử lý song song dữ liệu đầu vào

1.3 Transformers

1.3.1 Động lực phát triển của Transformers

Các mô hình Seq2Seq trước đó như RNNs và LSTM vẫn còn tồn tại một số hạn chế như sau: Đầu tiên, các mô hình này tính hoạt động một cách tuần tự, các nơ-ron phía sau phải đợi thông tin được xuất ra từ nơ-ron phía trước mới có thể bắt đầu hoạt động, điều này dẫn đến hạn chế lớn về mặt thời gian. Thứ hai, RNNs và LSTM vẫn chưa mô hình hóa được thông tin ngữ cảnh của câu văn mà chỉ được ngầm hiểu bằng cơ chế hoạt động tuần tự của chúng, điều này dẫn đến việc chúng khó khăn trong việc xử lý các câu dài hoặc các cấu trúc ngữ pháp phức tạp, nơi mà thông tin quan trọng có thể nằm cách xa nhau trong chuỗi. Hơn nữa, các mô hình này thường gặp phải vấn đề suy biến gradient hoặc gây bùng nổ gradient, làm giảm hiệu quả trong việc học và cập nhật các trọng số qua nhiều bước thời gian. Những hạn chế này đã thúc đẩy sự phát triển của các mô hình mới đơn cử như Transformer [27], cho phép xử lý dữ liệu song song (Hình 1.5) và khả năng nắm bắt ngữ cảnh một cách toàn diện hơn.

Transformer được giới thiệu lần đầu bởi Google trong bài báo "Attention Is All You Need" [27] vào năm 2017. Đây là kiến trúc đặc biệt nổi bật trong lĩnh vực xử lý ngôn ngữ tự nhiên (Natural Language Processing) và dịch máy (Machine Translation). Kiến trúc này đã đem lại những tiến bộ đáng

kể trong việc xử lý dữ liệu dạng chuỗi, vượt qua những hạn chế của các kiến trúc truyền thống như RNNs hoặc LSTM. Chìa khóa cho những thành công của Transformer nằm ở cơ chế tự chú ý (Self-Attention), cơ chế này cho phép Transformer tập trung vào các thành phần quan trọng của chuỗi dữ liệu đầu vào mà nó đang xử lý. Nhờ đó, cho phép Transformer có thể xử lý được các chuỗi dữ liệu có độ dài khác nhau mà không cần phải chia nhỏ hoặc giới hạn độ dài cố định như các mô hình truyền thống, giúp mô hình tạo ra một biểu diễn dữ liệu phong phú và hiệu quả hơn, cải thiện khả năng nắm bắt ngữ cảnh và tạo ra các dự đoán chính xác.

1.3.2 Kiến trúc tổng quát của Transformers

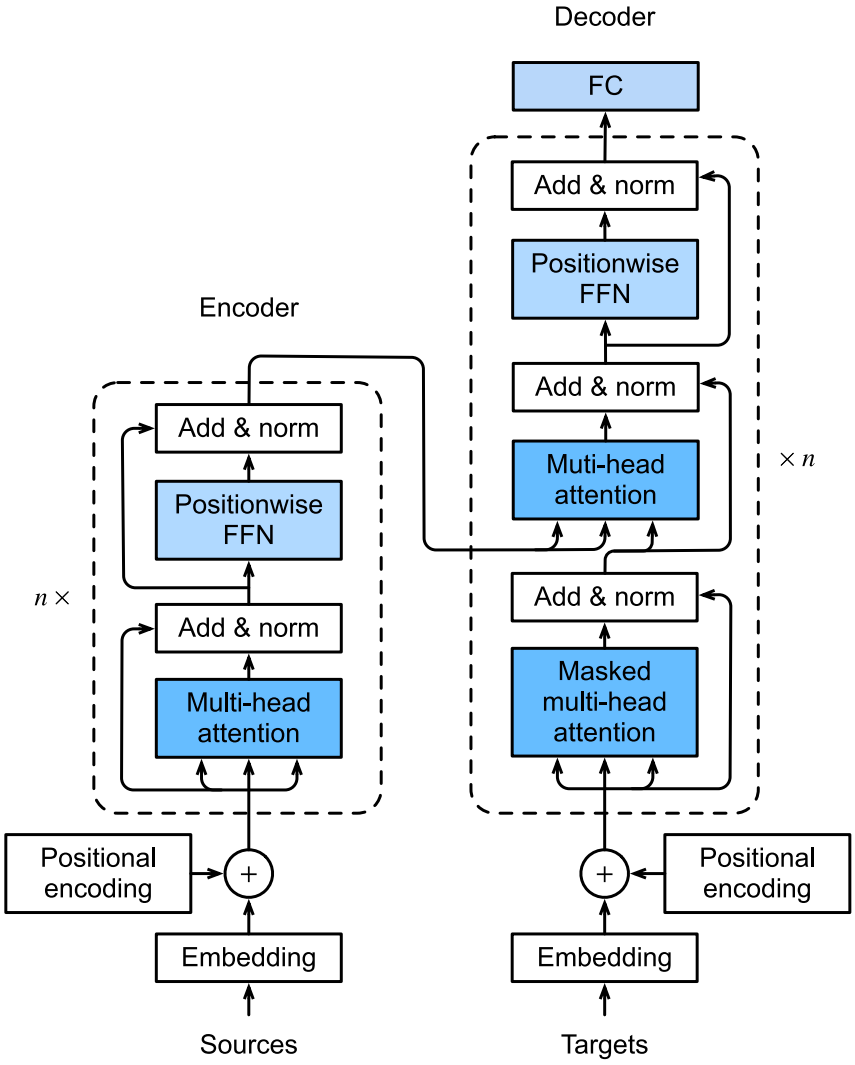
Cấu trúc của Transformer bao gồm 2 thành phần chính: Bộ mã hóa (Encoder) và Bộ giải mã (Decoder). Kiến trúc của Transformer có thể được quan sát thấy ở Hình 1.6, nó bao gồm n bộ mã hóa và n bộ giải mã được xếp chồng lên nhau tạo thành các ngăn xếp tương ứng.

Bộ mã hóa - Encoder

Tất cả các khối Encoder trong cùng một ngăn xếp sẽ có cấu trúc giống hệt nhau, bao gồm:

- Input Embedding: Có nhiệm vụ chuyển đổi các đầu vào dạng chuỗi thành các vector số thực trong không gian đa chiều, giúp mô hình hiểu mối quan hệ ngữ nghĩa và ngữ cảnh giữa các từ mà chưa xét đến vị trí của chúng trong chuỗi.

- Positional Encoding: Vì Transformer hiện tại đang gặp vấn đề là chưa có thông tin về vị trí của các từ trong chuỗi đầu vào, do đó lớp này dùng để cung cấp thông tin về vị trí thứ tự của các từ trong chuỗi đầu vào. Có nhiều phương pháp để thực hiện điều này nhưng trong Transformer



Hình 1.6: Kiến trúc tổng quát của Transformer

người ta sử dụng mã hóa dựa trên Sin-Cos.

- Multi-Head Attention (Encoder Attention): Dùng để tính toán mối quan hệ giữa các từ khác nhau trong chuỗi dữ liệu đầu vào. Transformer gọi mỗi bộ xử lý của cơ chế Self-Attention là Attention Head và lặp lại nó nhiều lần song song với nhau. Điều này được gọi là Multi-Head Attention. Ở cơ chế này, chuỗi đầu vào được chia nhỏ thành nhiều Attention Head có thể xử lý độc lộc với nhau. Nó mang lại khả năng phân biệt được mức độ và mối quan hệ phức tạp hơn giữa các từ trong chuỗi với nhau một cách tốt hơn.

- Feed-Forward Network: Được sử dụng với mục đích kết hợp các Head-Attention lại với nhau.

- Residual connection: Có nhiệm vụ mang thông tin từ trạng thái ở quá khứ đến trạng thái ở hiện tại. Thông tin ở đây chính là các vectơ mã hóa về ngữ nghĩa, ngữ cảnh và vị trí của chuỗi đầu vào. Điều này có thể tránh được việc mất mát thông tin cũng như gặp phải hiện tượng suy biến gradient.

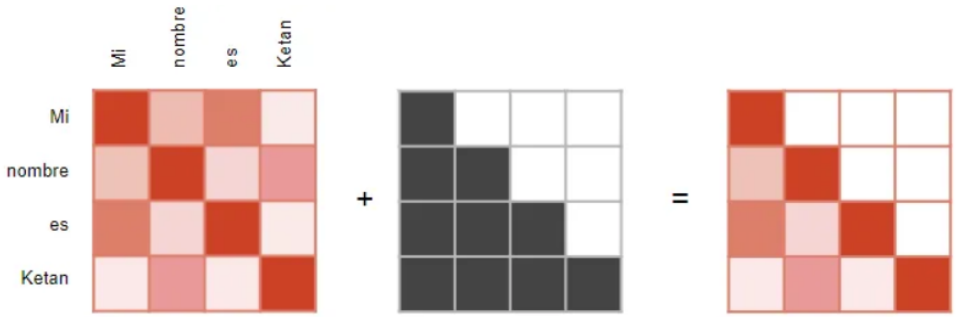
- Norm: Transformer sử dụng kỹ thuật Layer Normalize [4] trong bước chuẩn hóa, đây là kỹ thuật thường được sử dụng trong các mô hình như RNNs, có tác dụng điều chỉnh kích thước đầu ra của lớp trước đó, đảm bảo rằng các giá trị đầu ra từ các lớp không thay đổi quá nhiều, nhằm mục đích tăng cường sự ổn định và hiệu suất cho mô hình.

Bộ giải mã - Decoder

Mỗi khối Decoder cũng có cấu trúc tương tự như Encoder, bao gồm Input Embedding, Multi-Head Attention, Feed-Forward Network, Positional encoding, Feed-Forward Network, Residual connection và Norm. Điểm khác biệt

cần lưu ý ở đây chính là có thêm một lớp Masked Multi-Head Attention đặt trước lớp Multi-Head Attention và cơ một cơ chế được gọi là Cross-Attention.

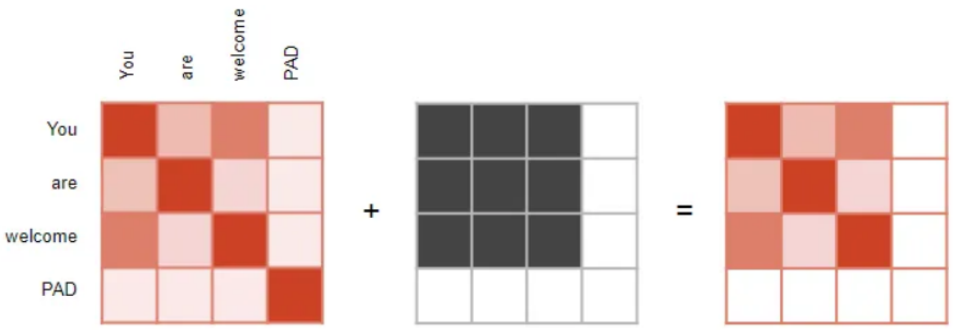
- Masked Multi-Head Attention (Decoder Attention): Điểm đặc biệt nằm ở cơ chế "Masking", mỗi từ trong chuỗi chỉ có thể tập trung vào các từ ở vị trí trước đó hoặc chính nó, không thể tập trung vào các từ ở vị trí sau (Hình 1.7). Điều này ngăn Decoder có thể biết trước được phần còn lại của chuỗi mục tiêu khi thực hiện việc dự đoán từ tiếp theo, đảm bảo tính chính xác và khách quan của mô hình.



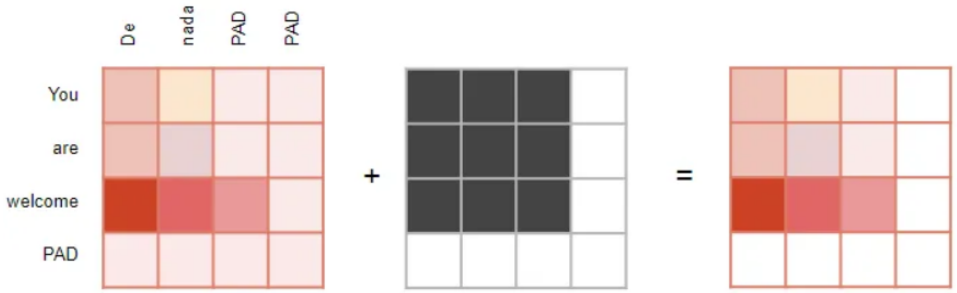
Hình 1.7: Cơ chế Masking trong Decoder Attention

- Cross-Attention (Encoder - Decoder Attention): Tính toán sự tương tác giữa chuỗi được đưa ra từ Encoder và chuỗi mục tiêu đến từ Decoder. Tại đây, thay vì chỉ tính toán sự tương tác nội bộ trong cùng một chuỗi như Self-Attention. Mục đích của điều này vẫn là đem thông tin từ Encoder sang Decoder mà vẫn đảm bảo rằng thông tin của những từ quan trọng không bị mất đi.

Ngoài ra, cơ chế "Masking" nói trên còn xuất hiện ở Encoder Attention (Hình 1.8) và Encoder - Decoder Attention (Hình 1.9), dùng để đặt giá trị Attention của các phầm đệm (Padding - phần thêm vào để làm cho các chuỗi có cùng độ dài với nhau) về 0, mục đích là để đảm bảo rằng các phần đệm này không góp phần vào quá trình tính toán Attention.



Hình 1.8: Cơ chế Masking trong Encoder Attention



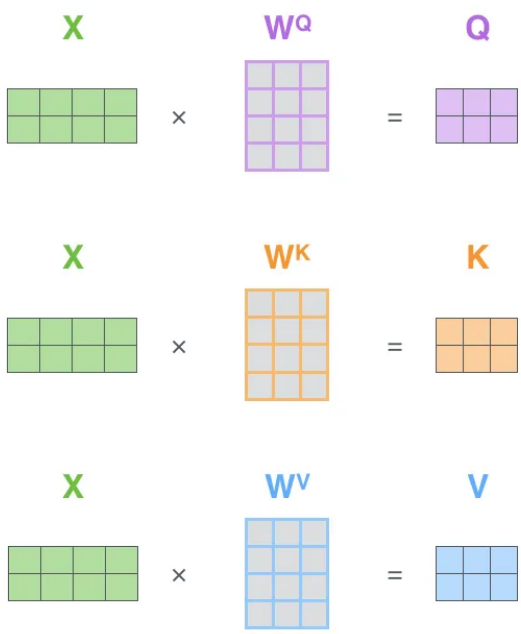
Hình 1.9: Cơ chế Masking trong Encoder-Decoder Attention

1.3.3 Quá trình tính toán Attention

Trong Transformer, cơ chế Attention được sử dụng để tạo ra các của từng từ trong câu. Cụ thể, ta có thể hiểu như sau:

- Đầu vào đầu tiên là các vector embeddings. Nhân mỗi vector embedding đầu vào với 3 ma trận trọng số W_q, W_k, W_v để tạo ra 3 vector q, k, v.
- Vector q và k được dùng để tính trọng số khuếch đại thông tin cho các từ trong câu và vector v là vector biểu diễn của các từ trong câu.

Ví dụ ta có 2 vector embeddings (tương ứng với 2 từ đầu vào "Xin", "chào") là x_1, x_2. Nhân 2 vector này với 3 ma trân Wq, Wk, Wv ta được tập các vector: $\{q_1, q_2\}$, $\{k_1, k_2\}$, $\{v_1, v_2\}$. Để tính toán vector biểu diễn cho từ "Xin". Đầu tiên ta cần tính trọng số khuếch đại thông tin cho mỗi từ(gọi là Attention), Attention cho từ "Xin"(a_1) và từ "chào"(a_2) được tính theo



Hình 1.10: Ma trận trọng số Attention

công thức sau:

$$a_1 = \text{softmax}(q_1 * k_1/\sqrt{(d)})$$
$$a_2 = \text{softmax}(q_2 * k_2/\sqrt{(d)})$$

Trong đó d là số chiều của vector k. Cuối cùng vector biểu diễn cho từ "Xin" được tính theo công thức:

$$z_1 = a_1 * v_1 + a_2 * v_2$$

1.4 Mô hình cơ sở

1.4.1 Giới thiệu

Các mô hình cơ sở (Foundation Models - FMs), là một lớp các mô hình học sâu đã được huấn luyện trước trên một lượng lớn dữ liệu (Pre-train Model), do đó chúng được trang bị một phạm vi rộng lớn các kiến thức tổng quát. Các mô hình này đóng vai trò như một điểm khởi đầu linh hoạt cho nhiều nhiệm vụ khác nhau, trải dài trên nhiều lĩnh vực. Những mô hình này có thể được tinh chỉnh cho các nhiệm vụ cụ thể khác nhau tùy theo từng tình

huống chỉ với một lượng nhỏ dữ liệu. Với cách tiếp cận này, chúng ta không chỉ tiết kiệm thời gian dùng để phát triển một mô hình đặc thù cho từng nhiệm vụ mà còn có thể tận dụng sự hiểu biết rộng lớn về thông tin mà vẫn giúp cho các mô hình này có được sự linh hoạt và hiệu quả đặc biệt [5].

Trong lĩnh vực thị giác máy tính, các mô hình cơ sở như CLIP [24] và SAM [18] đã thúc đẩy sự phát triển trong nhận diện hình ảnh, phát hiện đối tượng và nhiều hơn thế nữa. Trong xử lý ngôn ngữ tự nhiên, các mô hình như BERT [9] và GPT-3 [6] đã tạo nên một cuộc cách mạng hóa cho các nhiệm vụ hiểu và sinh ra văn bản.

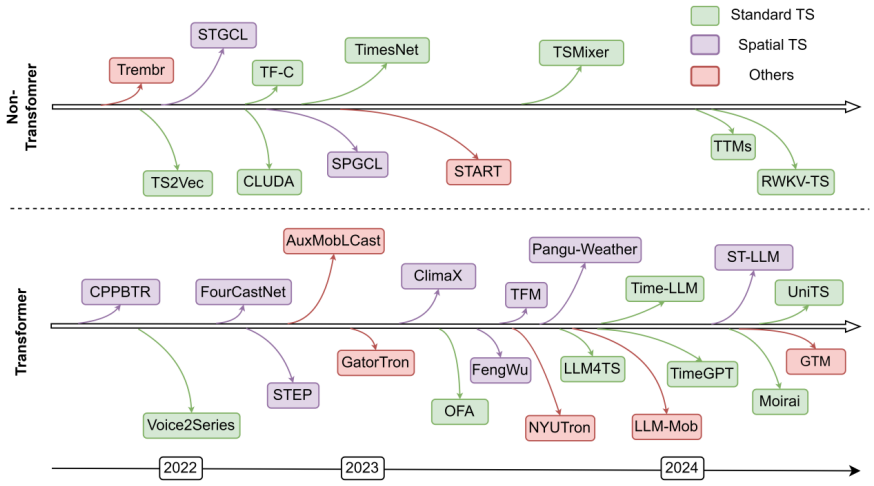
1.4.2 Mô hình cơ sở cho bài toán chuỗi thời gian

Được truyền cảm hứng từ những thành tựu của các mô hình cơ sở trong các lĩnh vực rộng lớn như thị giác máy tính và xử lý ngôn ngữ tự nhiên, khái niệm Time Series Foundation Models (TSFMs) đã thu hút sự chú ý như một hướng đi đầy hứa hẹn cho các nhiệm vụ liên quan đến chuỗi thời gian. Bằng cách khai thác các tập dữ liệu chuỗi thời gian có quy mô lớn, TSFMs hứa hẹn sẽ đạt được hiệu suất vượt trội trên một loạt các nhiệm vụ chuỗi thời gian, cung cấp một khung thống nhất, thúc đẩy các hoạt động nghiên cứu và phát triển ứng dụng trong lĩnh vực này.

Trong một bài khảo sát mới đây về các mô hình cơ sở cho dữ liệu chuỗi thời gian, các tác giả đã cho rằng các thành phần của một mô hình cơ sở bao gồm: Loại dữ liệu, kiến trúc mô hình, cách thức huấn luyện mô hình và các kỹ thuật tinh chỉnh chỉnh [20]. Trong đó:

- Loại dữ liệu: Đề cập đến loại dữ liệu sẽ được sử dụng để huấn luyện mô hình, có thể kể đến như dữ liệu chuỗi thời gian, văn bản, hình ảnh hoặc âm thanh.

- Kiến trúc của mô hình: Đề cập đến việc lựa chọn mạng nơ-ron nào sẽ



Hình 1.11: Sự phát triển hiện tại của các mô hình cơ sở cho chuỗi thời gian [20]

được sử dụng làm xương sống cho mô hình cơ sở. Trong đó, kiến trúc Transformers [28, 27] là một lựa chọn phổ biến nhờ khả năng xử lý dữ liệu tuần tự hiệu quả.

- Cách thức huấn luyện mô hình: Đề cập đến cách huấn luyện mô hình trên các tập dữ liệu lớn và đa dạng để có được sự hiểu biết rộng về dữ liệu, ở đây có thể kể đến như các kỹ thuật như học có giám sát (supervised learning) hoặc học tự giám sát (self-supervised learning).

- Các kỹ thuật tinh chỉnh chỉnh: Chẳng hạn như kỹ thuật fine-tuning hoặc few-shot learning sẽ được sử dụng để điều chỉnh các mô hình cơ sở đã huấn luyện trước đó cho từng nhiệm vụ cụ thể khác nhau.

1.4.3 Kiến trúc Transformer trong các mô hình cơ sở chuỗi thời gian

Như đã đề cập ở trên, kiến trúc Transformers [28] là loại kiến trúc phổ biến nhất cho các mô hình cơ sở chuỗi thời gian vì đặc tính hiệu quả của nó. Tuy nhiên, việc lựa chọn khung mô hình vẫn còn đang được tranh luận khá nhiều. Chúng ta có các công trình nghiên cứu đáng chú ý bao gồm các mô hình chỉ bao gồm bộ mã hóa (Encoder-only) như [11, 23, 29], các mô

hình bao gồm cả bộ mã hóa và bộ giải mã (Encoder-Decoder) như [2, 10, 12] và các các mô hình chỉ bao gồm bộ giải mã (Decoder-only) như [8, 21, 25]. Ansari và các cộng sự của mình [11] đã phân tích tính ứng dụng của bộ khung Encoder-Decoder đối với các mô hình sử dụng bộ khung Decoder-only. Bên cạnh đó, Liu và các cộng sự của mình [21] cũng đã thảo luận rằng trong khi các mô hình Decoder-only được ưa chuộng trong dự báo chuỗi thời gian hơn nhờ tính hiệu quả của nó trên các tập dữ liệu nhỏ, trong khi đó các mô hình mang kiến trúc Decoder-only với khả năng tổng quát hóa mạnh mẽ và dung lượng lớn, có thể được ưu tiên cho các chuỗi thời gian quy mô lớn. Qua đây, chúng ta có thể thấy cũng chính vì sự đa dạng trong các lựa chọn kiến trúc mô hình đã nhấn mạnh được tiềm năng và sự cần thiết phải tiếp tục nghiên cứu trong sự phát triển của các mô hình cơ sở cho chuỗi thời gian.

Gần đây, nhiều mô hình cơ sở dựa trên Transformer cũng đã được giới thiệu. Trong đó, có một số mô hình đáng chú ý như sau:

- TimesFM [8]: Được phát triển bởi Google Research, TimesFM là mô hình sử dụng kiến trúc Decoder-only với 200 triệu tham số. Mô hình này được huấn luyện trên một bộ dữ liệu bao gồm 100 tỷ điểm dữ liệu, bao gồm cả dữ liệu tổng hợp và thực tế từ nhiều nguồn khác nhau như Google Trends và Wikipedia Pageviews. TimesFM có khả năng thực hiện zero-shot forecasting trong nhiều lĩnh vực, bao gồm bán lẻ, tài chính, sản xuất, chăm sóc sức khỏe và khoa học tự nhiên, với các mức độ tần suất thời gian khác nhau.

- Lag-Llama [25]: Được phát triển bởi các nhà nghiên cứu đến từ Université de Montréal, Mila-Québec AI Institute và McGill University, Lag-Llama là một mô hình cơ sở được thiết kế cho việc dự báo chuỗi thời gian đơn biến. Dựa trên cơ sở của Llama, mô hình này sử dụng kiến trúc Decoder-only (dùng các kích thước biến thời gian và tần suất

thời gian khác nhau để dự báo). Mô hình này được huấn luyện trên các bộ dữ liệu chuỗi thời gian đa dạng từ nhiều nguồn khác nhau thuộc 6 lĩnh vực, bao gồm: Năng lượng, giao thông, kinh tế, tự nhiên, chất lượng không khí và vận hành đám mây.

- Chronos [2]: Đây là một mô hình dự được phát triển bởi Amazon, Chronos là một mô hình dự báo xác suất chuỗi thời gian, được xây dựng dựa trên kiến trúc T5 Transformer. Chronos đã được huấn luyện trước trên một loạt các bộ dữ liệu công khai và dữ liệu tổng hợp được tạo ra từ Gaussian Process. Chronos khác với TimesFM [8] ở chỗ nó là một mô hình Encoder-Decoder, nhờ đó có thể dễ dàng trích xuất được các encoder embeddings từ dữ liệu chuỗi thời gian đầu vào.

- MOMENT [14] Được phát triển dựa trên sự hợp tác bởi Đại học Carnegie Mellon và Đại học Pennsylvania, Moment là một mô hình cơ sở dành cho dữ liệu chuỗi thời gian. Các biến thể của MOMENT được xây dựng dựa trên các biến thể của kiến trúc T5. Mô hình này được huấn luyện trước trên bộ dữ liệu "Time-series Pile", là một bộ sưu tập dữ liệu chuỗi thời gian công khai và đa dạng trên nhiều lĩnh vực khác nhau. Khác với các FMs khác, MOMENT được thiết kế để dùng trong nhiều nhiệm vụ chuỗi thời gian khác nhau, giúp nâng cao hiệu quả của nó trong các ứng dụng bao gồm dự báo, phân loại, phát hiện bất thường và điền dữ liệu thiếu.

1.4.4 Mô hình MOIRAI

MOIRAI [29] là một mô hình cơ sở được phát triển bởi Salesforce AI Research, được thiết kế cho việc dự báo tổng quát. Mô hình này được huấn luyện trên bộ dữ liệu có tên "Large-scale Open Time Series Archive (LOTSA)", chứa 27 tỷ quan sát từ chín lĩnh vực khác nhau, tạo nên một bộ dữ liệu mã

nguồn mở chuỗi thời gian lớn nhất tại thời điểm hiện tại. Nhờ bộ dữ liệu đa dạng này, Moirai có thể học được từ nhiều loại dữ liệu chuỗi thời gian khác nhau, giúp nó xử lý các nhiệm vụ dự báo khác nhau, trải dài trên nhiều lĩnh vực.

MOIRAI trình bày một cách tiếp cận trong bài toán dự báo chuỗi thời gian như sau:

Xét một tập dữ liệu gồm N chuỗi thời gian $\mathcal{D} = \{(\boldsymbol{Y}^{(i)}, \boldsymbol{Z}^{(i)})\}_{i=1}^{N}$, trong đó $\boldsymbol{Y}^{(i)} = (\boldsymbol{y}_1^{(i)}, \boldsymbol{y}_2^{(i)}, \ldots, \boldsymbol{y}_{T_i}^{(i)}) \in \mathbb{R}^{d_{y_i} \times \boldsymbol{T_i}}$ là chuỗi thời gian mục tiêu gồm d_{y_i} biến và T_i bước thời gian. Mỗi chuỗi thời gian $\boldsymbol{Y}^{(i)}$ được gắn với một tập hợp các đồng biến (covariates) $\boldsymbol{Z}^{(i)} = (\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}, \ldots, \boldsymbol{z}_{T_i}^{(i)}) \in \mathbb{R}^{d_{z_i} \times T_i}$. Mục tiêu là dự đoán một phân phối $p(\boldsymbol{Y}_{t:t+h}|\phi)$ bằng cách ước lượng các tham số ϕ thông qua mô hình $\boldsymbol{f}_\theta : (\boldsymbol{Y}_{t-l:t}, \boldsymbol{Z}_{t-l:t+h}) \mapsto \hat{\phi}$ sao cho tối đa hóa được Log-Likelihood:

$$\max_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{\substack{(\mathbf{Y},\mathbf{Z})\sim p(\mathcal{D}) \\ (\mathrm{t,l,h})\sim p(\mathcal{T}|\mathcal{D})}} \log p\left(\mathbf{Y}_{\mathrm{t:t+h}} \mid \hat{\boldsymbol{\Phi}}\right)$$

sao cho $\hat{\boldsymbol{\Phi}} = \boldsymbol{f}_\theta(\mathbf{Y}_{t-l:t}, \mathbf{Z}_{t-l:t+h})$, trong đó $p(\mathcal{D})$ được gọi là phân phối dữ liệu (data distribution) với các mẫu $(\boldsymbol{Y}, \boldsymbol{Z})$ của nó là một chuỗi thời gian và $p(\mathcal{T}|\mathcal{D})$ được gọi là phân phối nhiệm vụ (task distribution) dùng để định nghĩa "lookback window" $\boldsymbol{Y}_{t-l:t} = (\boldsymbol{y}_{t-l}, \ldots, \boldsymbol{y}_{t-1})$ với độ dài ngữ cảnh l và khoảng thời gian cần dự báo $\boldsymbol{Y}_{t:t+h} = (\boldsymbol{y}_t, \ldots, \boldsymbol{y}_{t+h-1})$ có độ dài dự đoán h.

Kiến trúc mô hình

MOIRAI có kiến trúc dựa trên Transformer nhưng chỉ bao gồm bộ mã hóa, được minh họa trong Hình 1.12. Một trong những cải tiến đáng chú ý được đề xuất của MOIRAI là "Any-Variate", cho phép mô hình có thể xử lý được chuỗi thời gian đa biến. Đề xuất này sẽ làm phẳng các chuỗi thời gian đa biến và xem xét tất cả các biến này như một chuỗi đầu vào duy nhất.

Hình 1.12: Kiến trúc của MOIRAI

Ngoài ra, MOIRAI còn tiến hành phân tách chuỗi thời gian đầu vào thành các mảnh (Patch) không bị trùng lặp với nhau để mô hình hóa chúng với một kiến trúc mã hóa theo tuân theo cơ chế "Masked Encoder Architecture". Ở đây, "Mask" sẽ thay thế cho các mảnh nằm trong khoảng dự báo và nó là một Embedding có thể đào tạo được trong quá trình huấn luyện mô hình. Tiếp theo, các mảnh dữ liệu đầu vào sẽ được chiếu thành các biểu diễn vector bằng cách đưa chúng qua một lớp với nhiều mảnh có kích thước khác nhau gọi là "Multi Patch Size Projection", cho phép MOIRAI có khả năng xử lý chuỗi thời gian với đa dạng các tần số khác nhau (phút/giờ/ngày/tuần/tháng...). Cuối cùng, các token đầu ra sẽ được giải mã thông qua một lớp "Multi Patch Size Projection" thứ hai để trở thành các tham số của một phân phối hỗn hợp.

Các tác giả đã sử dụng tiền chuẩn hóa (pre-normalization) của Xiong et al., 2020 và thay thế tất cả các LayerNorm trong kiến trúc gốc bằng RMSNorm (Zhang & Sennrich, 2019), đồng thời họ cũng áp dụng chuẩn hóa truy vấn - khóa (query-key normalization) của Henry et al., 2020). Độ phi tuyến trong các lớp FFN đã được thay thế bằng SwiGLU (Shazeer, 2020) và thực hiện điều chỉnh kích thước ẩn để có được số lượng tham số bằng với lớp FFN gốc. Cuối cùng, họ loại bỏ các bias trong tất cả các lớp của module Transformer.

Multi Patch Size Projection

Một mô hình cơ sở nên có khả năng xử lý các chuỗi thời gian trải dài trên nhiều miền tần số khác nhau. Trước đó, các mô hình có kiến trúc dựa trên Patch đều chỉ dựa vào một siêu tham số Patch size duy nhất, đây là một đặc điểm kế thừa từ tính chất "mỗi mô hình được xây dựng và tối ưu nhất cho từng bộ dữ liệu riêng biệt". Thay vào đó, các tác giả của MOIRAI hướng tới một chiến lược linh hoạt hơn: Chọn Patch size lớn hơn để xử lý dữ liệu có tần số cao, từ đó giảm gánh nặng chi phí tính toán bậc hai của Attention, đồng thời duy trì được độ dài của ngữ cảnh. Bên cạnh đó, họ cũng đề nghị một Patch size nhỏ hơn cho dữ liệu có tần số thấp để chuyển sự tính toán sang các lớp Transformer, thay vì chỉ dựa vào các lớp Embedding tuyến tính đơn giản. Để thực hiện ý tưởng này, họ đề xuất cho mô hình học nhiều lớp embedding đầu vào và đầu ra, mỗi lớp tương ứng với các Patch size khác nhau, gọi là "Multi Patch Size Projection", trong đó, mỗi phép chiếu là một lớp Linear đơn giản.

Any-Variate Attention

Kiến trúc Transformer truyền thống nhận vào một chuỗi giá trị mục tiêu duy nhất. Tuy nhiên, các mô hình cơ sở lại được kỳ vọng rằng có thể xử lý nhiều chuỗi giá trị mục tiêu và các biến covariates cùng một lúc, trong trường hợp làm việc với chuỗi thời gian đa biến. Do đó, các tác giả đã giới thiệu "Any-Variate Attention". Sau khi làm phẳng chuỗi thời gian đa biến thành một chuỗi duy nhất thì một cơ chế mang tên "Variate Encoding" sẽ được áp dụng, cho phép mô hình phân biệt được các biến khác nhau trong chuỗi, điều này thực sự quan trọng quá trình tính toán điểm số Attention sau này.

Any-Variate Attention có hai đặc điểm chính:

- Thứ nhất là tính hoán vị đồng nhất đối với thứ tự các biến (Permutation Equivariance): Điều này có nghĩa là nếu thứ tự quan sát trong một biến được hoán đổi, kết quả của mô hình cho biến đó cũng phải phản ánh được sự hoán đổi tương tự, bảo đảm tính nhất quán của động lực học chuỗi thời gian.

- Thứ hai là tính hoán vị bất biến đối với chỉ số các biến (Permutation Invariance): Điều này có nghĩa là kết quả của mô hình không thay đổi khi các biến được hoán vị với nhau.

Để đạt được tính đồng nhất và bất biến, Moirai sử dụng hai phương pháp:

- Rotary Positional Embeddings (RoPE) [26]: Có nhiệm vụ đảm bảo tính hoán vị đồng nhất thông qua việc mã hóa vị trí bằng cách xoay biểu diễn của các token trong không gian nhúng, duy trì khoảng cách tương đối giữa các token.

- Binary Attention Bias [31]: Cho phép mô hình xử lý các biến như thể chúng không có thứ tự, mô hình có thể tự do điều chỉnh sự tập trung của nó bằng cách sử dụng các độ lệch attention khác nhau dựa trên việc các phần tử thuộc cùng một biến hoặc khác biến, giúp cho cơ chế Any-Variate Attention xử lý được số lượng biến và hoán vị tùy ý.

Mixture Distribution

Thế mạnh của Moirai nằm ở chỗ nó là một mô hình dự báo xác suất, nghĩa là MOIRAI dự đoán các tham số của một phân phối thay vì chỉ đưa ra một dự đoán điểm duy nhất, cho phép người ta có thể đánh giá được mức độ không chắc chắn của các kết quả dự đoán, với các khoảng tin cậy rộng hơn thể hiện mức độ không chắc chắn cao hơn từ mô hình. Vì MOIRAI là một mô hình cơ sở, nên nó được thiết kế để dự báo trên nhiều lĩnh vực dữ

liệu khác nhau và do đó không thể giới hạn nó vào một phân phối duy nhất. Để phù hợp với tất cả các kịch bản có thể xảy ra, mô hình học các tham số của một phân phối hỗn hợp, trong đó mỗi phân phối thành phần sẽ thích hợp với các loại dữ liệu khác nhau.

Phân phối hỗn hợp của MOIRAI có hàm mật độ xác suất như sau:

$$p(\mathbf{Y}_{t:t+h}|\hat{\phi}) = \sum_{i=1}^{c} w_i p_i(\mathbf{Y}_{t:t+h}|\hat{\phi}_i),$$

trong đó $\hat{\phi} = \{w_1, \hat{\phi}_i, ..., w_c, \hat{\phi}_c\}$ và p_i là phân phối thành phần thứ i của hàm mật độ xác suất. Các thành phần này được trình bày chi tiết trong phần 3.4.

Pre-Training

Các nghiên cứu hiện tại chủ yếu dựa vào ba nguồn dữ liệu chính: Monash Time Series Forecasting Archive [13], GluonTS library [1], và các bộ dữ liệu phổ biến khác [19, 30]. Tuy nhiên, những nguồn này bị hạn chế về kích thước, chỉ có khoảng 1 tỷ quan sát. Do đó, MORAI được huấn luyện trước trên bộ dữ liệu LOSTA (Large Open Time Series Archive). Đây là bộ dữ liệu được nhóm tổng hợp nhằm giải quyết những hạn chế của các bộ dữ liệu chuỗi thời gian hiện có. LOTSA được tổng hợp từ nhiều nguồn dữ liệu công khai khác nhau, bao gồm 27,646,462,733 quan sát trải dài trên chín lĩnh vực khác nhau.

Khi huấn luyện mô hình, để ước lượng các tham số của phân phối hỗn hợp, MOIRAI thực hiện tối ưu hóa bằng cách giảm thiểu một hàm mất mát, cụ thể là Negative Log-likelihood:

$$\min_{\boldsymbol{\theta}} - \mathop{\mathbb{E}}_{\substack{(\mathbf{Y},\mathbf{Z})\sim p(\mathcal{D}) \\ (\mathrm{t,l,h})\sim p(\mathcal{T}|\mathcal{D})}} \log p\left(\mathbf{Y}_{\mathrm{t:t+h}} \mid \hat{\boldsymbol{\Phi}}\right)$$

Việc pre-training bao gồm hai khía cạnh chính:

- Data Distribution: Dữ liệu được phân phối bằng cách chia nhỏ thành các tập con và mỗi tập con được lấy mẫu theo cách cân bằng để tránh mất cân đối dữ liệu. Đối với K tập con, một tập con được chọn từ $p(\mathcal{D})$, và một chuỗi thời gian được chọn từ tập con đó. Để giải quyết vấn đề mất cân đối dữ liệu, sự đóng góp của mỗi tập con bị giới hạn ở mức $\epsilon = 0.001$ trước khi tái chuẩn hóa.

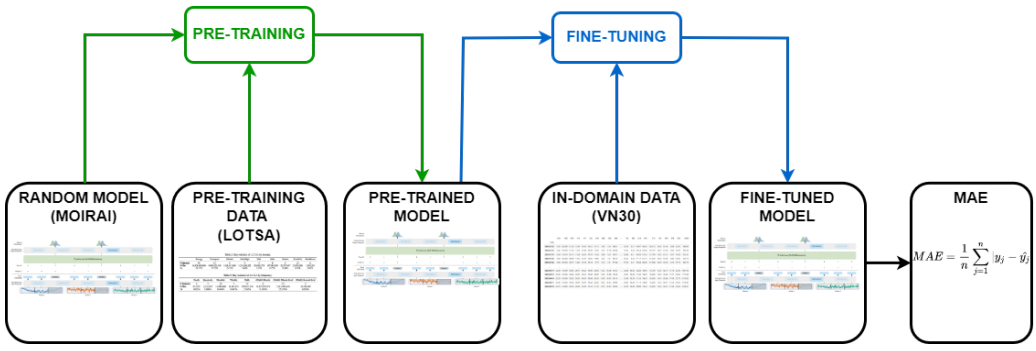
- Task Distribution: Điều này liên quan đến việc lấy mẫu các nhiệm vụ từ một phân phối được thiết kế để phản ánh sự đa dạng của các ứng dụng thực tế mà mô hình có thể gặp phải, tăng cường khả năng tổng quát hóa của mô hình.

CHƯƠNG 2:

PHƯƠNG PHÁP ĐỀ XUẤT

2.1 Xác định điểm bất thường trong chuỗi thời gian dựa trên dự báo

Chúng tôi nhận thấy sự phát triển nhanh chóng của các mô hình cơ sở chuỗi thời gian chỉ trong một khoảng thời gian ngắn. Những mô hình này được trang bị một lượng kiến thức khổng lồ, được thiết kế để tương thích cho nhiều nhiệm vụ khác nhau và có khả năng thực hiện zero-shot với hiệu suất ấn tượng trên các tập dữ liệu mà chúng chưa từng nhìn thấy trước đó. Bên cạnh đó, chúng tôi cũng nhận thấy sự tiềm năng của các mô hình dự báo xác suất với khả năng cung cấp thông tin về sự không chắc chắn trong các giá trị dự báo, hỗ trợ tốt hơn cho việc ra quyết định. Từ đó, trong bài nghiên cứu này, chúng tôi đề xuất một phương pháp phát hiện điểm bất thường trên dữ liệu chuỗi thời gian dựa vào kết quả của khả năng dự báo xác suất từ một mô hình cơ sở, tận dụng được sức mạnh tiềm năng mà mô hình này mang lại. Quy trình của chúng tôi bao gồm hai phần: Phần một là tinh chỉnh mô hình dự báo xác suất và phần hai là tính toán khoảng cách và xác định các điểm bất thường. Quy trình này được minh họa trong Hình 2.1.



Hình 2.1: Tổng quan về phương pháp đề xuất

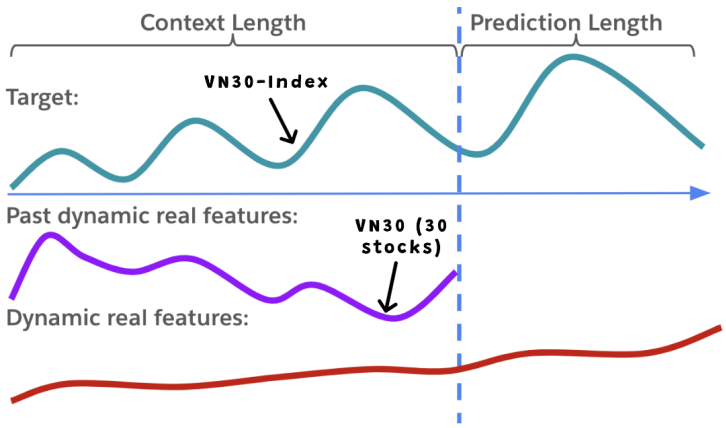
Một mô hình dự báo xác suất được kỳ vọng là sẽ cho ra nhiều kịch bản có thể xảy ra tại mỗi bước thời gian mà nó dự báo, do đó chúng tôi mong muốn đầu ra của mô hình này là một phân phối xác suất. Giả sử một bài toán đang cần phát hiện ra những điểm bất thường trong một khoảng thời gian có độ dài h, trước tiên mô hình dự báo xác suất của chúng tôi sẽ dự báo ra một khoảng thời gian h' tương ứng, trong đó ở mỗi bước thời gian của h' sẽ có một số lượng giá trị dự đoán có thể xảy ra tại thời điểm này, chúng được lấy mẫu thông qua phân phối xác suất đầu ra của mô hình.

Do bản chất của mỗi giá trị dự đoán là xác suất có thể xảy ra tại mỗi điểm nên trong giai đoạn tiếp theo, chúng tôi sẽ thực hiện tính khoảng cách từ mỗi giá trị dự báo đến điểm giá trị tương ứng trong thực tế. Giả sử chúng tôi lấy 100 mẫu ở mỗi bước thời gian dự báo thì khoảng cách sẽ được tính là trung bình cộng khoảng cách của 100 điểm này tới bước thời gian thực tế tương ứng và sẽ có tổng cộng h khoảng cách được tính toán. Cuối cùng, chúng tôi chọn ra một số lượng k điểm có khoảng cách lớn nhất để xem chúng là những điểm bất thường trong khoảng thời gian h.

2.1.1 Giai đoạn một: Tinh chỉnh mô hình dự báo xác suất

Do các mô hình cơ sở thường được huấn luyện trước trên một lượng lớn dữ liệu nên chúng thường có một lượng tham số khổng lồ, điều này trực tiếp dẫn đến việc chúng có khả năng cao sẽ mắc phải hiện tượng quá khớp khi làm việc với một tập dữ liệu nhỏ hơn. Do đó, tại bước này chúng tôi cho rằng cần có một thủ tục để hạn chế việc này xảy ra. Cụ thể, chúng tôi đề xuất sử dụng phương pháp đóng băng phần lớn các lớp của mô hình cơ sở, chỉ giữ lại một lượng tham số thích hợp với mỗi bộ dữ liệu cụ thể.

Tùy theo từng bài toán, mô hình dự báo tại bước này sẽ được tinh chỉnh dưới kịch bản của nhiệm vụ dự báo chuỗi thời gian đa biến hoặc đơn biến.



Hình 2.2: Fine-tuning với các biến Covariate

Dù trong bất kỳ trường hợp nào, chúng ta cũng cần xác định một hoặc một tập hợp (trong trường hợp dự báo đa biến) chuỗi thời gian mục tiêu cần phải làm dự báo và tập hợp các chuỗi thời gian đồng biến (Covariates) nếu có. Trong đó, các chuỗi đồng biến (hay còn gọi là biến đồng thời hoặc biến dự báo ngoài) là các biến độc lập gây ảnh hưởng đến giá trị của chuỗi thời gian đang cần dự báo. Một minh họa về việc sử dụng các đồng biến trong dự báo chuỗi thời gian được chúng tôi cung cấp trong Hình 2.2.

2.1.2 Giai đoạn hai: Phương pháp tính khoảng cách

Như đã nhắc đến trước đó ở phần 2.1, trong giai đoạn này, chúng tôi bắt đầu lựa chọn một chỉ số khoảng cách thích hợp để thực hiện việc tính khoảng cách trung bình từ tập hợp các giá trị dự đoán đến giá trị thực tế ở mỗi bước thời gian. Việc xác định các điểm bất thường cũng được chúng tôi đề xuất như sau: Giả sử có một chuỗi thời gian độ dài t đang cần phải được xác định những điểm bất thường, chúng tôi tiến hành sử dụng mô hình đã được tinh chỉnh trong giai đoạn trước đó và thực hiện quá trình dự báo trên chuỗi thời gian này, cũng như tính toán các khoảng cách. Kết quả chúng tôi sẽ có một tập hợp t' khoảng cách, với $t' < h$. Cuối cùng, chúng tôi lấy một Top-K các khoảng cách lớn nhất trong tập hợp t' và xem những ngày tương ứng với các khoảng cách này là bất thường.

CHƯƠNG 3:

THỰC NGHIỆM

3.1 Thông tin bộ dữ liệu

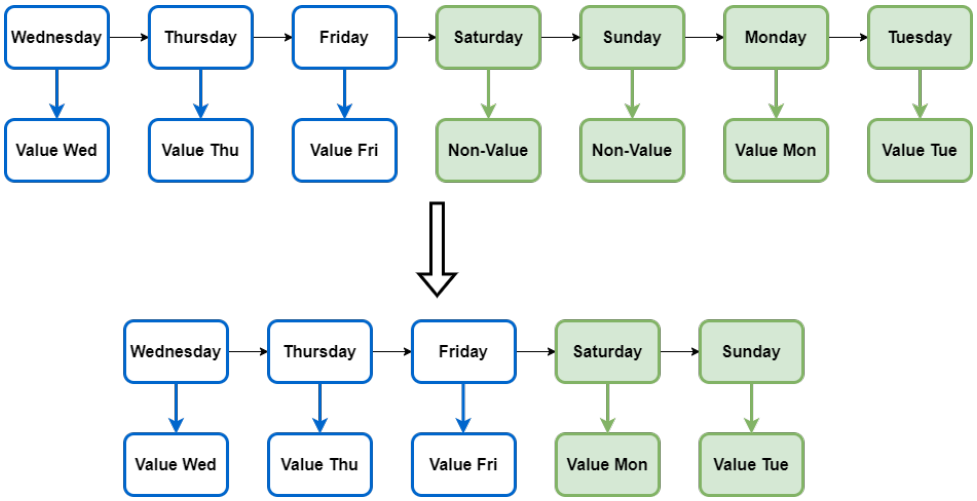
Chúng tôi đã thu thập một bộ dữ liệu về giá của các cổ phiếu trong tập hợp VN30 bao gồm 31 cột giá trị: Trong đó, 30 cột đầu tiên đại diện cho 30 mã cổ phiếu khác nhau và cột cuối cùng đại diện cho chỉ số VN30-Index. Giá trị trong mỗi cột thể hiện cho giá đóng cửa cuối ngày của cổ phiếu đó, tần suất dữ liệu được lấy theo từng ngày giao dịch. Bộ dữ liệu này gồm 991 ngày giao dịch liên tục kể từng ngày 01/01/2020. Hình 3.1 cung cấp một cái nhìn trực quan về bộ dữ liệu của chúng tôi.

3.2 Chi tiết cài đặt

Chúng tôi tiến hành triển khai các thí nghiệm của mình trong môi trường Python và sử dụng thư viện mã nguồn mở Pytorch cho việc xây dựng và huấn luyện mô hình.

time	ACB	BID	BVH	CTG	FPT	GAS	GVR	HDB	HPG	KDH	...	TCB	TPB	VCB	VHM	VIC	VJC	VNM	VPB	VRE	VN30
2020-01-02	9.46	32.369	61.39	14.35	28.95	68.44	10.15	8.91	9.94	18.61	...	23.08	8.37	59.07	62.60	102.210	148.0	81.40	6.88	34.35	886.88
2020-01-03	9.46	32.229	61.04	14.22	28.46	68.59	10.50	8.83	10.00	18.54	...	22.94	8.45	58.48	62.82	102.477	148.0	81.66	6.84	34.60	883.28
2020-01-04	9.26	31.610	60.15	14.25	28.16	70.76	10.32	8.65	9.94	18.61	...	22.35	8.31	56.92	61.50	101.766	146.0	81.04	6.67	33.65	872.34
2020-01-05	9.26	32.229	60.06	14.49	28.70	70.25	10.05	8.56	9.81	18.46	...	22.50	8.33	57.11	61.79	101.943	145.9	81.80	6.82	33.95	876.70
2020-01-06	9.09	32.509	58.72	14.45	28.06	70.18	9.70	8.56	9.71	17.90	...	22.06	8.21	56.59	60.83	101.588	144.8	82.00	6.73	32.70	865.18
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2022-09-14	23.20	41.500	39.50	26.70	96.20	76.00	20.20	19.00	24.36	30.35	...	29.68	16.95	82.20	39.90	43.700	103.0	66.74	17.76	22.55	1097.45
2022-09-15	23.35	41.000	39.05	26.05	94.00	75.50	20.30	18.40	24.18	30.30	...	29.14	16.90	81.50	39.90	43.200	105.0	65.66	17.47	22.60	1111.86
2022-09-16	23.25	41.200	39.20	26.50	94.60	75.00	20.25	18.95	24.73	30.10	...	29.68	17.00	81.20	39.50	43.200	104.3	65.66	17.38	22.70	1115.52
2022-09-17	23.30	41.300	39.35	26.55	94.70	75.20	20.30	18.80	24.73	29.80	...	29.43	16.90	81.10	39.95	43.250	104.3	66.74	17.33	23.00	1115.94
2022-09-18	23.75	41.800	39.35	26.55	95.10	74.70	21.20	18.75	24.73	30.05	...	29.72	16.90	80.90	40.20	43.200	105.3	66.44	17.28	22.95	1128.51

Hình 3.1: Bộ dữ liệu VN30



Hình 3.2: Quy trình xử lý các ngày không có dữ liệu

3.2.1 Xử lý dữ liệu

Xử lý các ngày không có dữ liệu

Đối với các giao dịch chứng khoán tại Việt Nam, thị trường sẽ không có biến động vào hai ngày thứ bảy và chủ nhật hoặc những ngày nghỉ lễ trong năm, do đó dữ liệu của chúng tôi sẽ không có cập nhật vào những ngày này, làm dữ liệu không được liên tục. Trong ngữ cảnh của dữ liệu giao dịch chứng khoán, chúng tôi xem xét loại những ngày không xảy ra giao dịch ra khỏi tập dữ liệu và chỉ giữ lại những ngày xảy ra biến động trong giá cả, mà vẫn đảm bảo ý nghĩa là một tập dữ liệu chứa thông tin giá cả của các mã cổ phiếu trong những ngày giao dịch liên tiếp. Chúng tôi xem những ngày còn lại trong bộ dữ liệu là những ngày liên tiếp. Quy trình này được minh họa trong Hình 3.2

Chuẩn hóa dữ liệu

Trước khi đưa dữ liệu vào mô hình, chúng tôi có một bước chuẩn hóa dữ liêu (Z-score Normalization), trong đó mỗi điểm dữ liệu sẽ được trừ cho trung bình và chia cho độ lệch chuẩn của cột tương ứng, tạo ra một phân phối dữ liệu có trung bình bằng 0 và phương sai bằng 1. Điều này giúp cải

thiện hiệu suất của các mô hình và tối ưu hóa quá trình hội tụ của các thuật toán tối ưu.

$$x_{scaled} = \frac{x - \mu}{\sigma}$$

3.2.2 Giai đoạn một: Tinh chỉnh mô hình MOIRAI

Trong giai đoạn một, chúng tôi lựa chọn sử dụng MOIRAI [29], đã được giới thiệu ở phần 1.4.4 để làm mô hình cơ sở cho phần dự báo xác suất. Chúng tôi sử dụng phiên bản MOIRAI-Base với mong muốn cho hiệu suất dự báo tốt hơn. Mô hình MOIRAI-Base bao gồm 91 triệu tham số với 12 lớp, kích thước đầu ra của mô hình $d_{model} = 768$, kích thước đầu ra của lớp feed forward là $d_{ff} = 3072$, $n_{HeadsAtten} = 12$, kích thước của vector Value $d_v = 64$ và kích thước của vector Key $d_k = 64$.

Như chúng tôi đã trình bày trước đó, MOIRAI-Base cần được fine-tuning trên tập dữ liệu cụ thể để có thể đạt hiệu suất tối ưu. Vì vậy, chúng tôi đã tiến hành tinh chỉnh mô hình này với tập dữ liệu VN30.

Tập dữ luyện huấn luyện và tập dữ liệu kiểm tra

Với bộ dữ liệu VN30, chúng tôi thực hiện chia tập 991 ngày giao dịch về giá của 30 cổ phiểu và chỉ số VN30-Index như sau: Tập dữ liệu huấn luyện (Training) bao gồm 900 ngày đầu tiên và tập dữ liệu xác thực (Validation) sẽ bao gồm 91 ngày giao dịch còn lại.

Đối với tập dữ liệu kiểm tra (Testing), chúng tôi tiến hành thu thập thêm 91 ngày giao dịch kể từ ngày 02/01/2024 đến ngày 20/05/2024. Cấu trúc và các bước xử lý của tập dữ liệu này cũng tương tự như tập VN30.

Freeze Layer

Chúng tôi thực hiện đóng băng phần lớn các lớp của mô hình và chỉ thực hiện tinh chỉnh trên một số thành phần của ba lớp cuối cùng. Cụ thể, với mỗi lớp này, chúng tôi chỉ thực hiện tinh chỉnh trên ba thành phần cuối cùng của chúng với tổng cộng 5,2 triệu tham số được cập nhật qua quá trình huấn luyện, điều này ngăn cho mô hình bị quá khớp trên tập dữ liệu của chúng tôi.

3.2.3 Giai đoạn hai: Tính toán khoảng cách

Chúng tôi sử dụng Mean Absolute Error (MAE) là độ đo khoảng cách giữa các điểm giá trị dự báo và điểm giá trị thực tế. Như đã trình bày ở phần 3.2.2, tập dữ liệu kiểm tra của chúng tôi có 91 ngày giao dịch. Chúng tôi sử dụng mô hình MOIRAI-Base đã được tinh chỉnh để thực hiện việc dự báo trên tập dữ liệu này bắt đầu từ ngày 13/02/2024. Sau quá trình này, chúng tôi thu được 49 ngày giao dịch được dự báo từ mô hình, do đó chúng tôi cũng có được 49 giá trị khoảng cách MAE. Từ đây, chúng tôi thực hiện chọn ra Top-K khoảng cách lớn nhất trong số này và xem chúng là các điểm bất thường trong thực tế. Trong thực nghiệm, chúng tôi chọn $k = 5$ và $k = 10$.

3.3 Kết quả thực nghiệm

3.3.1 Hiệu suất của mô hình MOIRAI-Base trước và sau tinh chỉnh

Bảng dưới đây là kết quả giá trị của các chỉ số trước và sau khi tinh chỉnh mô hình MOIRAI-Base (được lấy trung bình sau 10 lần thực nghiệm). Các siêu tham số được chúng tôi cài đặt như được mô tả ở phần 3.5.2 :

Kết quả tinh chỉnh mô hình MOIRAI-Base		
Chỉ số	**Trước tinh chỉnh**	**Sau tinh chỉnh**
MSE	0.0211	0.0177
MAE	0.1161	0.1048
MASE	3.3849	3.0538
MAPE	0.2888	0.2862
MSIS	25.9817	22.7509
Dynamic Time Warping	4.7920	4.6894

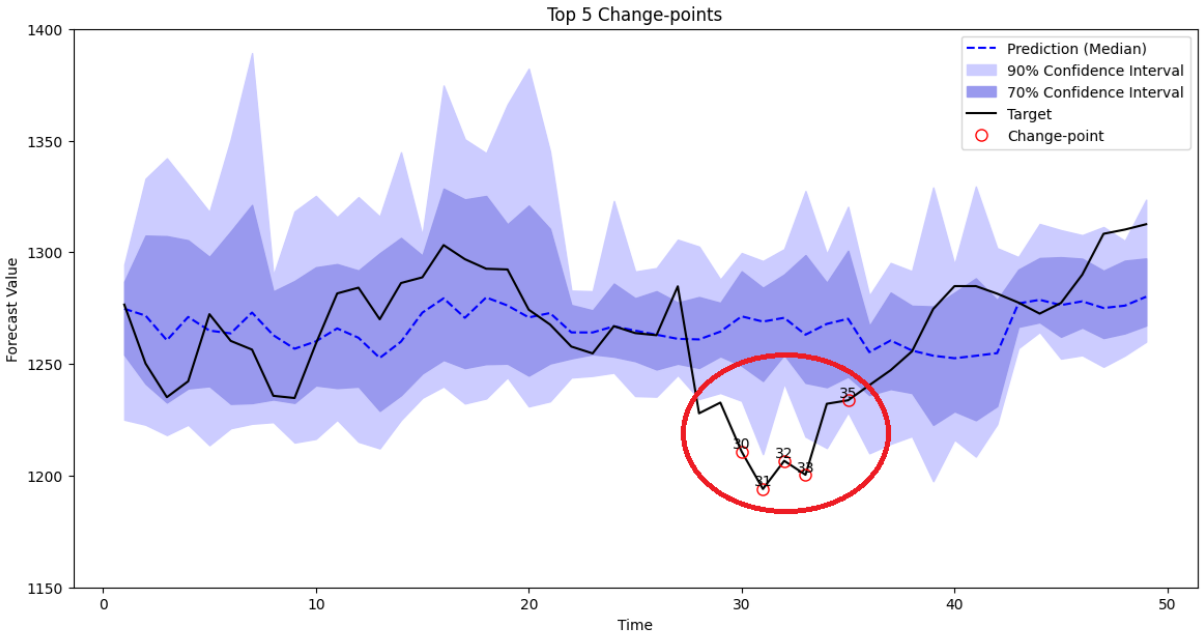
Bảng 3.1: So sánh các chỉ số trước và sau khi tinh chỉnh mô hình.

Đánh giá kết quả của các chỉ số

Dựa vào kết quả được trình bày trong bảng 3.1, mô hình MOIRAI-Base sau khi được tinh chỉnh đã cho thấy sự cải thiện ở tất cả các chỉ số đánh giá mà chúng tôi sử dụng. Điều này cho thấy sai số trong dự báo đã được giảm thiểu, nâng cao độ chính xác của mô hình so với khi sử dụng mô hình pre-train chưa được tinh chỉnh cụ thể với tập dữ liệu VN30. Bên cạnh đó, chỉ số MSIS cũng đã được cải thiện, khẳng định độ tin cậy của các điểm dự báo cũng đã được nâng cao rõ rệt. Nhìn chung, kết quả tinh chỉnh này cho thấy mô hình MOIRAI-Base đã trở nên chính xác và hiệu quả hơn trong việc dự báo, tạo cơ sở vững chắc hơn cho việc xác định các điểm bất thường dựa trên phương pháp dự báo mà chúng tôi đã đề xuất.

3.3.2 Kết quả tính toán khoảng cách và xác định các điểm bất thường

Hình 3.3 và Hình 3.4 thể hiện kết quả xác định Top-5 và Top-10 các điểm bất thường sử dụng phương pháp do chúng tôi đề xuất. Qua đó, phương pháp của chúng tôi có thể dễ dàng phát hiện được những điểm lệch hoàn toàn khỏi hình dạng trung bình của dữ liệu (vùng khoanh tròn màu đỏ), đồng thời cũng có thể tìm ra một số điểm mà chỉ số VN30-Index bắt đầu có xu hướng tăng hoặc giảm (Hình 3.4). Kết quả này đưa ra một gợi ý khá tiềm năng và có thể được dùng để tham khảo khi bất kỳ ai có nhu cầu tìm kiếm một cơ sở



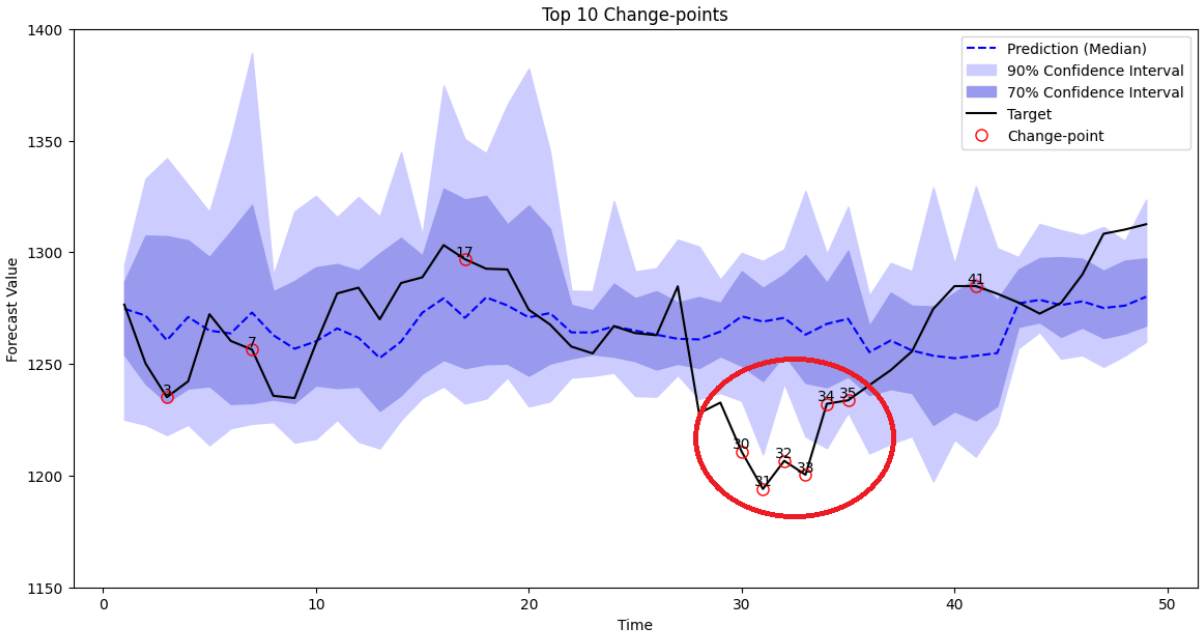
Hình 3.3: Kết quả Top-5 điểm bất thường (Trên tập Test)

để hỗ trợ việc ra quyết định khi thực hiện các giao dịch chứng khoán trong thị trường Việt Nam, đặc biệt là trên đường VN30-Index.

3.3.3 Kết quả triển khai chiến thuật đầu tư dựa vào các điểm bất thường

Chiến thuật 1: Mua ở những điểm bất thường nằm ngoài khoảng tin cậy 90% và bán ra ở ngày hôm sau

Để tận dụng tiềm năng của các điểm bất thường cho quá trình đầu tư, chúng tôi xây dựng một chiến thuật đầu tư như sau: Nhà đầu tư sẽ mua vào ở những điểm bất thường nằm ngoài khoảng tin cậy 90% và bán ngay vào ngày giao dịch hôm sau. Chúng tôi sẽ triển khai đồng thời chiến thuật này với chiến thuật Bollinger Bands trong 250 ngày giao dịch của năm 2019. Kết quả các điểm bất thường được thể hiện trong Hình 3.5, kết quả so sánh giữa hai chiến thuật được thể hiện trong bảng 3.2.



Hình 3.4: Kết quả Top-10 điểm bất thường (Trên tập Test)

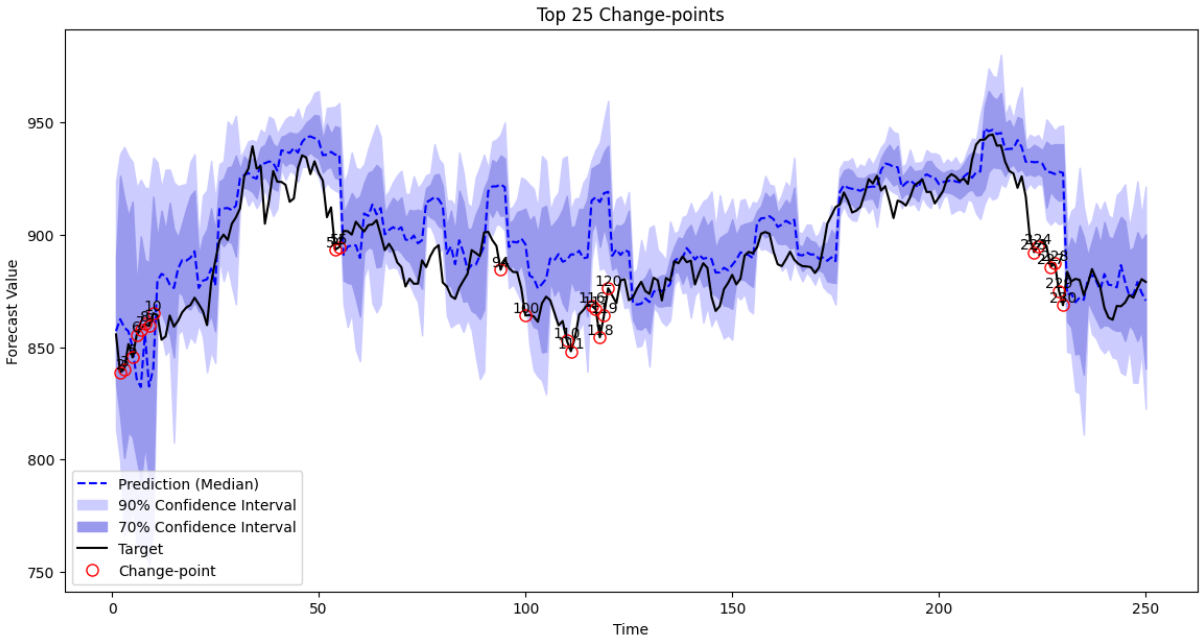
Chiến thuật 2: Sử dụng chiến thuật Moving Average Crossover kết hợp với Change-point

Chiến thuật này sử dụng Moving Average Crossover (MAC) làm cơ sở: Nhà đầu tư thực hiện mua và bán theo MAC và giữ nguyên vị thế của mình cho đến khi gặp một điểm bất thường, họ sẽ rút khỏi thị trường.

Chúng tôi chia tập dữ liệu 250 ngày giao dịch của năm 2019 ra thành 5 bộ dữ liệu con, mỗi bộ bao gồm 50 ngày theo thứ tự. Đối với mỗi tập dữ

Bollinger Bands và Change-point		
Chỉ số / **Phương pháp**	**Sharpe Ratio**	**Cumulative Return**
Bollinger Bands window 20	-1.3008	-0.0225
Bollinger Bands window 40	0.2524	0.017672
Bollinger Bands window 60	0.2752	0.017687
Bollinger Bands window 80	0.3074	0.017821
Bollinger Bands window 80	-0.9358	-0.0082
Change-point	0.6065	0.0311

Bảng 3.2: Bảng so sánh kết quả đầu tư của chiến thuật Bollinger Bands và chiến thuật dựa trên Change-point



Hình 3.5: Kết quả Top-25 điểm bất thường trên đường VN30-Index trong năm 2019

Moving Average Crossover					
Tập dữ liệu / Chỉ số	**Data 1**	**Data 2**	**Data 3**	**Data 4**	**Data 5**
Sharpe Ratio	2.8237	-4.3702	-2.7379	-4.4191	-3.1207
% Lợi nhuận tích lũy	0.02734	-0.0387	-0.0134	-0.0023	-0.01962
Số lần giao dịch	3	4	4	3	3

Bảng 3.3: Bảng so sánh kết quả đầu tư của chiến thuật MAC

liệu con, chúng tôi tiến hành dùng chiến lược đã được trình bày phía trên để thực hiện chạy quá trình đầu tư thử nghiệm trong 30 ngày. Kết quả của hai chiến thuật đầu tư này được trình bày trong bảng 3.3 và bảng 3.4.

Chúng tôi nhận thấy việc mua và bán theo các chiến thuật đầu tư có sử dụng điểm bất thường sẽ đạt kết quả tốt hơn việc đầu tư theo chiến thuật Bollinger Bands hay Moving Average Crossover thông thường, cho thấy sự

Moving Average Crossover (Kết hợp Change-point)					
Tập dữ liệu / Chỉ số	**Data 1**	**Data 2**	**Data 3**	**Data 4**	**Data 5**
Sharpe Ratio	2.8748	-3.1490	3.1425	1.8918	-3.1207
% Lợi nhuận tích lũy	0.0464	-0.0387	0.0142	0.0032	-0.01962
Số lần giao dịch	2	2	4	2	2

Bảng 3.4: Bảng so sánh kết quả đầu tư của chiến thuật kết hợp MAC với Change-point

tiềm năng của việc tận dụng những điểm bất thường trong quá trình đưa ra quyết định của nhà đầu tư.

Các kết quả trên chỉ là thử nghiệm trên những chiến thuật đầu tư đơn giản và phương pháp của chúng tôi cũng phục thuộc rất nhiều vào xác suất dự báo của mô hình, vì vậy không phải lúc nào cũng thu về được kết quả tốt nhất. Tuy nhiên, trong quá trình thực nghiệm, chúng tôi nhận thấy việc dựa vào các điểm bất thường để ra quyết định đầu tư phần lớn là thu về lợi nhuận dù ít hay nhiều. Chúng tôi cho rằng nhà đầu tư nên kết hợp phương pháp này cùng với những phân tích chuyên sâu để xây dựng một chiến thuật phức tạp hơn thì có thể giảm bớt rủi rỏ và đạt được nhiều lợi nhuận.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Thông qua bài nghiên cứu này, chúng tôi đã giới thiệu các mô hình cơ sở dành cho chuỗi thời gian và cho thấy việc ứng dụng chúng trong đa dạng các bài toán khác nhau là rất có tiềm năng, đặc biệt là trong các bài toán dự báo và xác định điểm bất thường. Bên cạnh đó, kết quả từ phương pháp mà chúng tôi đề xuất cũng đã đạt được những ý nghĩa nhất định trong quá trình đầu tư, qua đó thể hiện sự thành công trong phương pháp và tầm quan trọng của các điểm bất thường trong đầu tư chứng khoán.

Tuy nhiên, trong quá trình nghiên cứu và thực nghiệm, chúng tôi cũng nhận thấy rằng phương pháp mà chúng tôi đã đề xuất hiện tại vẫn còn một số những hạn chế. Một trong những số đó là phương pháp này hiện tại vẫn chưa được thiết kế để triển khai việc phát hiện các điểm bất thường theo thời gian thực, các thực nghiệm của chúng tôi chỉ mới được triển khai trên một tập dữ liệu có kích thước cố định. Điều này làm cho nghĩa thực tiễn của đề tài bị hạn chế, vì thị trường tài chính là luôn luôn biến động. Việc khắc phục được hạn chế này sẽ là một bước cải thiện đáng kể trong giá trị thực tiễn của phương pháp. Bên cạnh đó, đối với vấn đề tránh hiện tượng quá khớp của mô hình cơ sở khi thực hiện giai đoạn dự báo, chúng tôi vẫn chưa có nhiều thử nghiệm để áp dụng một phương pháp tối ưu nhất, bên cạnh việc đóng băng phần lớn các lớp của mô hình như hiện tại.

Trong tương lai, chúng tôi dự định sẽ khắc phục những hạn chế nêu trên bằng cách cải tiến giai đoạn dự báo phương pháp này, cũng như áp dụng các kỹ thuật tinh chỉnh mô hình hiện đại hơn, đơn cử như sử dụng LoRA [17] thay vì đóng băng mô hình như hiện tại để để tinh chỉnh một cách có hiệu quả.

PHỤ LỤC

3.4 Các thành phần của phân phối hỗn hợp

Mô hình MOIRAI dự đoán các tham số của một phân phối xác suất, trong trường hợp này là một phân phối hỗn hợp. Các tác giả đã áp dụng một lớp Softmax lên các tham số liên quan đến trọng số của các thành phần hỗn hợp, nhằm ràng buộc chúng vào một xác suất đơn giản. Các thành phần của phân phối hỗn hợp này được mô tả như sau:

Phân phối T Student

Một biến ngẫu nhiên x tuân theo phân phối Student's t có hàm mật độ xác suất như sau:

$$p(x;\nu,\mu,\tau) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{\pi\nu\tau}}\left(1+\frac{1}{\nu}\left(\frac{x-\mu}{\tau}\right)^2\right)^{-\frac{\nu+1}{2}}$$

Trong đó $\nu > 0$, $\mu \in \mathbb{R}$, $\tau > 0$, Γ lần lượt là bậc tự do (df), location, scale và hàm gamma. Mô hình MOIRAI dự đoán cả ba tham số df, location, scale, đồng thời áp dụng hàm softplus để đảm bảo ràng buộc tính dương cho các tham số này. Các tác giả cũng đã cài đặt giới hạn dưới cho tham số df là 2, vì phương sai sẽ không được xác định nếu không thực hiện điều này.

Phân phối Log-normal

Một biến ngẫu nhiên x tuân theo phân phối log-normal có hàm mật độ xác suất như sau:

$$p(x;\mu,\sigma) = \frac{1}{x\sigma\sqrt{2\pi}}\exp\left(-\frac{(\ln x-\mu)^2}{2\sigma^2}\right),$$

với các tham số $\mu \in \mathbb{R}$, $\sigma > 0$. MOIRAI dự đoán cả hai tham số này và áp dụng hàm softplus để đảm bảo ràng buộc tính dương cho chúng.

Phân phối nhị thức âm (Negative binomial)

Dựa theo nghiên cứu của Awasthi và cộng sự [3], các tác giả đã thực hiện một mở rộng của phân phối nhị thức âm. Theo đó, một biến ngẫu nhiên x tuân theo phân phối này có hàm mật độ xác suất như sau:

$$p(x; r, p) \propto \frac{\Gamma(x+r)}{\Gamma(x+1)\Gamma(r)}(1-p)^r p^x$$

với các tham số $r > 0$ và $p \in [0, 1]$, và Γ là hàm gamma. MOIRAI thực hiện dự đoán cả hai tham số này, đồng thời áp dụng hàm softplus để đảm bảo ràng buộc tính dương, và hàm sigmoid để giới hạn chúng trong khoảng xác suất.

Phân phối chuẩn với phương sai thấp (Low variance normal)

Một biến ngẫu nhiên x tuân theo phân phối chuẩn có hàm mật độ xác suất như sau:

$$p(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

với $\mu \in \mathbb{R}$, $\sigma > 0$. Đối với phân phối chuẩn có phương sai thấp, mô hình chỉ dự đoán tham số μ và cố định σ ở một giá trị nhỏ, cụ thể cố định $\sigma = 1 \times 10^{-3}$.

3.5 Tinh chỉnh mô hình MOIRAI-Base

3.5.1 Chi tiết kết quả tinh chỉnh mô hình

Chúng tôi thực hiện việc tinh chỉnh mô hình với các siêu tham số được cài đặt như được trình bày trong phần 3.5.2. Bên cạnh đó, chúng tôi sẽ dừng sớm việc huấn luyện nếu như mô hình trải qua 5 epoch liên tiếp không có

cải thiện về giá trị của hàm mất mát. Qua mỗi epoch, chúng tôi lưu lại mô hình có giá trị hàm mất mát tốt nhất để làm kết quả cuối cùng. Sau 7 epoch, việc huấn luyện kết thúc và mô hình tốt nhất mà chúng tôi tìm được có $Val\_Loss = -0.691$. Kết quả chi tiết của quá trình huấn luyện được trình bày trong bảng 3.5.

Kết quả của quá trình tinh chỉnh		
Epoch number	**Training loss**	**Validation loss**
Epoch 0	-0.366	0.643
Epoch 1	-0.629	0.244
Epoch 2	-0.691	0.146
Epoch 3	-0.305	0.0568
Epoch 4	-0.588	-0.0302
Epoch 5	-0.643	-0.0488
Epoch 6	-0.666	-0.148
Epoch 7	-0.463	-0.141

Bảng 3.5: Kết quả chi tiết của quá trình tinh chỉnh

3.5.2 Giá trị của các siêu tham số

Phần này trình bày các siêu tham số mà chúng tôi đã cài đặt trong quá trình tinh chỉnh MOIRAI-Base trên tập dữ liệu VN30.

Trong đó, một số siêu tham số cần được chú ý như:

- Context length: Số lượng bước thời gian trong quá khứ sẽ được dùng để làm đầu vào cho mô hình (số lượng ngữ nghĩa).
- Prediction length: Số lượng bước thời gian sẽ được dự báo trong tương lai.
- Number sample: Số lần mô hình phải lấy mẫu từ phân phối được dự báo cho mỗi bước thời gian dự báo.
- Patch size: Xác định kích thước của một mảnh khi chia chuỗi thời gian đầu vào trong quá trình đánh giá thành các mảnh đều nhau.

- Windows: Xác định số lượng cửa sổ "Rolling Evaluation" trong quá trình đánh giá mô hình. Tham số này được xác định bằng công thức $windows = \frac{TEST}{Predictionlength}$, trong đó $TEST$ là số bước thời gian của tập dữ liệu validation hoặc testing.

Cài đặt của thuật toán tối ưu

Thuật toán tối ưu trong quá trình huấn luyện được chúng tôi sử dụng là AdamW được triển khai với Pytorch. Chi tiết các siêu tham số của trình tối ưu này được chúng tôi cài đặt như trong bảng 3.6.

AdamW Optimizer	
Siêu tham số	**Giá trị cài đặt**
Learning rate	0.005
Weight decay	0.5
Beta1	0.9
Beta2	0.98

Bảng 3.6: Các giá trị siêu tham số được cài đặt cho trình tối ưu AdamW

Cài đặt của Validation Rolling Evaluation

Chi tiết cài đặt của cơ chế Rolling Evaluation trong quá trình Validation được trình bày trong bảng 3.7.

Validation Rolling Evaluation	
Siêu tham số	**Giá trị cài đặt**
Offset	900
Windows	13
Distance	7
Prediction Length	7
Context length	100
Patch size	16
Number sample	100

Bảng 3.7: Các giá trị siêu tham số được cài đặt cho Rolling Evaluation trong quá trình Validation

Cài đặt của Testing Rolling Evaluation

Chi tiết cài đặt của cơ chế Rolling Evaluation trong quá trình Testing được trình bày trong bảng 3.8.

Testing Rolling Evaluation	
Siêu tham số	**Giá trị cài đặt**
Offset	35
Windows	8
Distance	7
Prediction Length	7
Context length	200
Patch size	32
Number sample	100

Bảng 3.8: Các giá trị siêu tham số được cài đặt cho Rolling Evaluation trong quá trình Testing

Một số siêu tham số khác

Ngoài ra, bảng 3.9 trình bày một số các siêu tham số khác đóng vai trò quan trọng trong kết quả của việc tinh chỉnh mô hình.

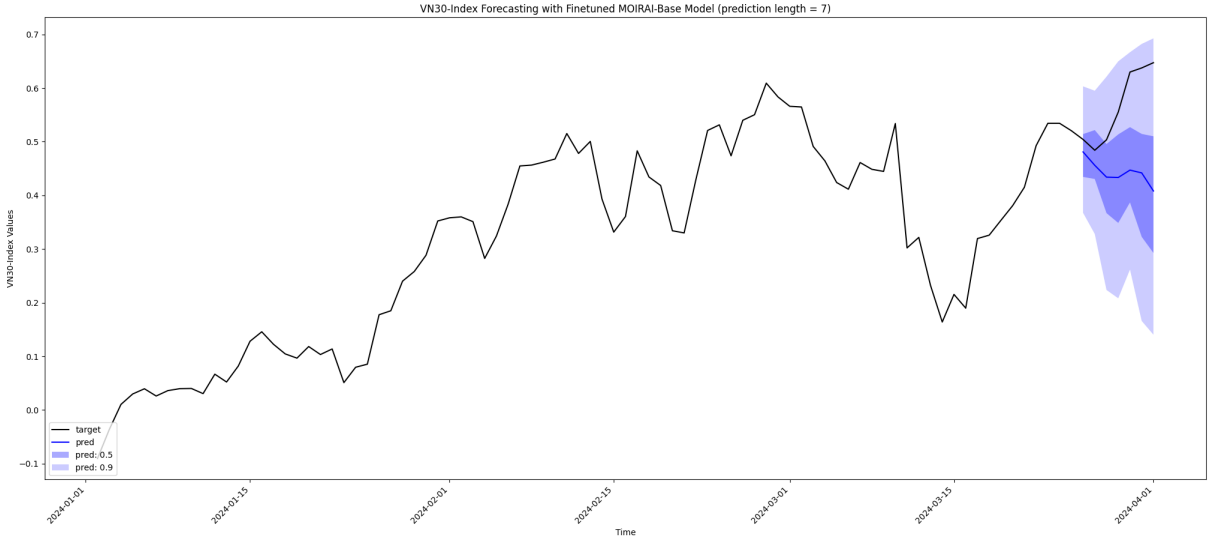
Các siêu tham số khác	
Siêu tham số	**Giá trị cài đặt**
Batch size	8
Max epochs	100
Num batches/Epoch (Training step)	100
Attention dropout	0.1
Dropout (Global)	0.1
Monitor (Early Stopping)	Validation loss
Monitor (ModelCheckpoint)	Validation loss
Patience (Early Stopping)	5

Bảng 3.9: Các giá trị siêu tham số được cài đặt trình huấn luyện

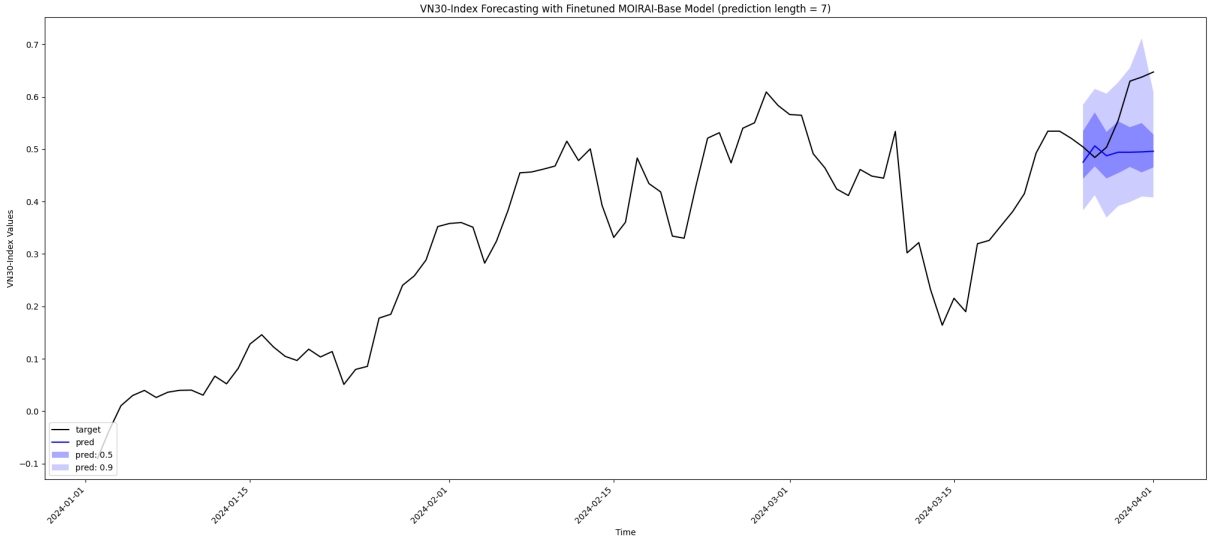
3.5.3 Trực quan kết quả dự báo của MOIRAI-Base trước là sau khi tinh chỉnh

Nhằm mục đích so sánh một cách trực quan hiệu suất dự báo của hai mô hình MOIRAI-Base trước và sau khi tinh chỉnh, chúng tôi thực hiện việc dự

báo một khoảng thời gian bảy ngày liên tiếp kể từ ngày 26/03/2024 đến ngày 01/04/2024. Kết quả có thể được quan sát ở Hình 3.6 và Hình 3.7



Hình 3.6: Trực quan khả năng dự báo của mô hình MOIRAI-Base trước khi được tinh chỉnh



Hình 3.7: Trực quan khả năng dự báo của mô hình MOIRAI-Base sau khi được tinh chỉnh

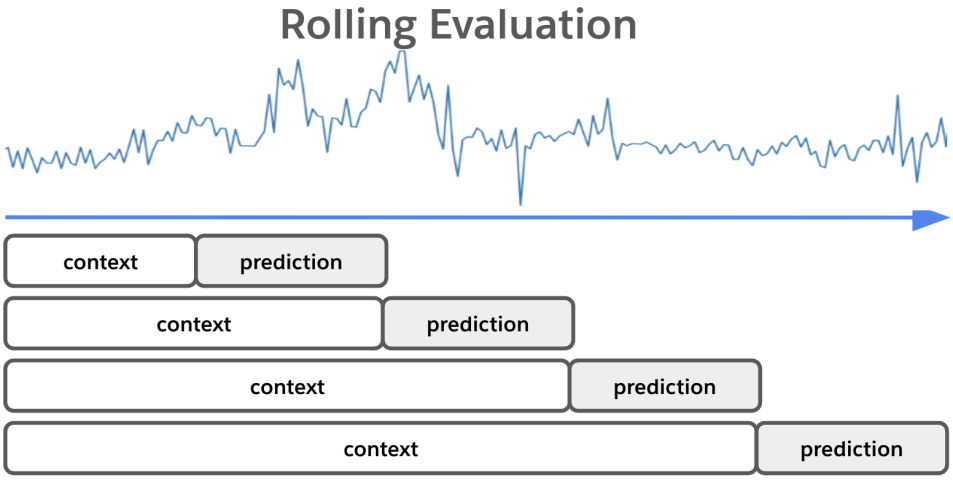
Hình 3.6 thể hiện khả năng dự báo của MOIRAI-Base khi chưa được tinh chỉnh trên tập dữ liệu VN30. Các giá trị này có độ lệch lớn so với giá trị thực tế và khoảng tin cậy rộng cho thấy mô hình này có độ không chắc chắn cao trong dự báo. Ngược lại, Hình 3.7 cho thấy độ sai lệch ít hơn giữa các giá trị dự báo và giá trị thực tế, đồng thời cũng có các khoảng tin cậy hẹp hơn, cho

thấy khả năng dự báo tốt hơn của mô hình sau khi được tinh chỉnh.

3.6 Đánh giá mô hình

3.6.1 Cơ chế evaluation của mô hình MOIRAI

Cơ chế "Rolling Evaluation" được MOIRAI sử dụng trong hai quá trình validation và testing. Cụ thể, để đánh giá hiệu suất của mô hình, cơ chế này định nghĩa một cửa sổ "Evaluation windows" và thực hiện việc trượt cửa sổ này trên chuỗi thời gian mục tiêu, đồng thời cũng tạo ra các dự báo trong mỗi lần trượt, cho đến khi đi hết chuỗi thời gian này. Chọn bước nhảy $stride = prediction\_length$ để thực hiện việc đánh giá không có trùng lặp (non-overlapping) hoặc $stride < prediction\_length$ để thực hiện việc đánh giá có trùng lặp (overlapping) giữa các cửa sổ. Chi tiết được minh họa trong hình Hình 3.8



Hình 3.8: Cơ chế Rolling Evaluation của mô hình MOIRAI

3.6.2 Các chỉ số đánh giá mô hình

MSE

Mean squared error (MSE) là một chỉ số đánh giá hiệu suất của mô hình dự đoán, đặc biệt là trong các bài toán hồi quy. Nó được sử dụng để đo lường

sự khác biệt giữa các giá trị dự đoán của mô hình và các giá trị thực tế. MSE được tính theo công thức sau:

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

Trong đó, y_i và $\hat{y}_i$ lần lượt là điểm dữ liệu thực tế và điểm dữ liệu dự đoán (trung vị) thứ i, n là tổng số điểm dữ liệu.

MAE

Mean Absolute Error (MAE) là một chỉ số đánh giá hiệu suất của mô hình dự đoán, nó đo lường độ lớn trung bình của các sai số giữa các giá trị dự đoán và giá trị thực tế mà không quan tâm đến hướng của sai số. MAE được tính bằng công thức sau:

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

Trong đó, y_i và $\hat{y}_i$ lần lượt là điểm dữ liệu thực tế và điểm dữ liệu dự đoán (trung vị) thứ i, n là tổng số điểm dữ liệu.

MAPE

Mean Absolute Percentage Error (MAPE) là một chỉ số đánh giá hiệu suất của mô hình dự đoán, thường được sử dụng trong các bài toán dự đoán trong lĩnh vực dự báo. MAPE đo lường độ lớn trung bình của sai số phần trăm giữa các giá trị dự đoán và giá trị thực tế.

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

Trong đó, y_i và $\hat{y}_i$ lần lượt là điểm dữ liệu thực tế và điểm dữ liệu dự đoán (trung vị) thứ i, n là tổng số điểm dữ liệu.

MASE

Mean Absolute Scaled Error (MASE) là một phép đo thống kê được sử dụng để đánh giá hiệu suất của mô hình dự đoán trong các bài toán dự báo chuỗi thời gian.

$$MASE = \frac{\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|}{\frac{1}{n-m}\sum_{i=m+1}^{n}|y_i - y_{i-m}|}$$

Trong đó, y_i và $\hat{y}_i$ lần lượt là điểm dữ liệu thực tế và điểm dữ liệu dự đoán (trung vị) thứ i, n là tổng số điểm dữ liệu. MASE được chuẩn hóa bằng cách chia cho $\frac{1}{n-m}\sum_{i=m+1}^{n}|y_i - y_{i-m}|$, trong đó m là yếu tố theo mùa (seasonal).

Mean Scaled Interval Score (MSIS)

Mean Scaled Interval Score là một chỉ số để đánh giá độ không chắc chắn xung quanh các dự báo điểm, được giới thiệu trong cuộc thi M4 [22]. Công thức để tính MSIS như sau:

$$MSIS = \frac{1}{h}\frac{\sum_{t=1}^{h}(U_t - L_t) + \frac{2}{a}(L_t - Y_t)\mathbb{1}_{\{Y_t<L_t\}} + \frac{2}{a}(Y_t - U_t)\mathbb{1}_{\{Y_t>U_t\}}}{\frac{1}{n-m}\sum_{t=m+1}^{n}|Y_t - Y_{t-m}|}$$

Trong đó, U_t và L_t lần lượt là dự báo cận trên và cận dưới, Y_t là giá trị thực tế, h là độ dài của khoảng dự báo và a là mức ý nghĩa, với giá trị $a = 0.05$ cho khoảng dự báo 95%. MSIS được chuẩn hóa bằng cách chia cho $\frac{1}{n-m}\sum_{t=m+1}^{n}|Y_t - Y_{t-m}|$, trong đó m là yếu tố theo mùa (seasonal).

Dynamic Time Warping

Dynamic Time Warping (DTW) là một phương pháp đo lường sự tương đồng giữa hai chuỗi thời gian có chiều dài không cố định bằng cách tìm ra sự căn chỉnh tối ưu giữa các điểm trong hai chuỗi thời gian sao cho tổng khoảng cách giữa các điểm tương ứng là nhỏ nhất.

$$\mathrm{DTW}(i,j) = \text{distance}\,(x_i, y_j) + \min \begin{cases} DTW(i, j-1) & \text{repeat } x_i \\ DTW(i-1, j) & \text{repeat } y_j \\ DTW(i-1, j-1) & \text{repeat neither} \end{cases}$$

Sharpe Ratio

Sharpe Ratio là một chỉ số đo lường hiệu quả của một quỹ đầu tư hoặc một cổ phiếu. Chỉ số này giúp đánh giá tỷ lệ lợi ích so với rủi ro mà một khoản đầu tư mang lại. Sharpe Ratio thường được sử dụng để đánh giá hiệu suất của các quỹ đầu tư, các khoản vay hoặc các cổ phiếu, đặc biệt là trong lĩnh vực tài chính.

$$SharpeRatio = \frac{R_p - R_f}{\sigma_p}$$

Trong đó:

- R_p là lợi nhuận kỳ vọng của khoản đầu tư (trung bình).
- R_f là lãi suất phi rủi ro (thường là lãi suất trái phiếu hoặc lãi suất tiền gửi ngắn hạn).
- σ_p là độ lệch chuẩn của khoản đầu tư, thể hiện mức độ biến động của nó.

Sharpe Ratio càng cao, đồng nghĩa với việc khoản đầu tư đem lại lợi nhuận cao hơn so với rủi ro mà nó mang lại. Ngược lại, Sharpe Ratio thấp cho thấy tỷ lệ lợi ích so với rủi ro không hợp lý.

DANH MỤC TÀI LIỆU TRÍCH DẪN

[1] Alexander Alexandrov, Konstantinos Benidis, Michael Bohlke-Schneider, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Danielle C Maddix, Syama Rangapuram, David Salinas, Jasper Schulz, et al. Gluonts: Probabilistic and neural time series modeling in python. *Journal of Machine Learning Research*, 21(116):1–6, 2020.

[2] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

[3] Pranjal Awasthi, Abhimanyu Das, Rajat Sen, and Ananda Theertha Suresh. On the benefits of maximum likelihood estimation for regression and forecasting. *arXiv preprint arXiv:2106.10370*, 2021.

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[5] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[7] Zahra Zamanzadeh Darban, Geoffrey I Webb, Shirui Pan, Charu C Aggarwal, and Mahsa Salehi. Deep learning for time series anomaly detection: A survey. *arXiv preprint arXiv:2211.05244*, 2022.

[8] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. *arXiv preprint arXiv:2310.10688*, 2023.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[10] Jiaxiang Dong, Haixu Wu, Yuxuan Wang, Yunzhong Qiu, Li Zhang, Jianmin Wang, and Mingsheng Long. Timesiam: A pre-training framework for siamese time-series modeling. *arXiv preprint arXiv:2402.02475*, 2024.

[11] Shanghua Gao, Teddy Koker, Owen Queen, Thomas Hartvigsen, Theodoros Tsiligkaridis, and Marinka Zitnik. Units: Building a unified time series model. *arXiv preprint arXiv:2403.00131*, 2024.

[12] Azul Garza and Max Mergenthaler-Canseco. Timegpt-1. *arXiv preprint arXiv:2310.03589*, 2023.

[13] Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I Webb, Rob J Hyndman, and Pablo Montero-Manso. Monash time series forecasting archive. *arXiv preprint arXiv:2105.06643*, 2021.

[14] Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*, 2024.

[15] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[16] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

[17] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[18] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[19] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks.

In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 95–104, 2018.

[20] Yuxuan Liang, Haomin Wen, Yuqi Nie, Yushan Jiang, Ming Jin, Dongjin Song, Shirui Pan, and Qingsong Wen. Foundation models for time series analysis: A tutorial and survey. *arXiv preprint arXiv:2403.14735*, 2024.

[21] Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Transformers for time series analysis at scale. *arXiv preprint arXiv:2402.02368*, 2024.

[22] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The m4 competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36(1):54–74, 2020.

[23] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[25] Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Hena Ghonia, Rishika Bhagwatkar, Arian Khorasani, Mohammad Javad Darvishi Bayazi, George Adamopoulos, Roland Riachi, Nadhir Hassen, et al. Lag-llama: Towards foundation models for probabilistic time series forecasting. *arXiv preprint arXiv:2310.08278*, 2024.

[26] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[28] Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*, 2022.

[29] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. *arXiv preprint arXiv:2402.02592*, 2024.

[30] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

[31] Jingfeng Yang, Aditya Gupta, Shyam Upadhyay, Luheng He, Rahul Goel, and Shachi Paul. Tableformer: Robust transformer modeling for table-text encoding. *arXiv preprint arXiv:2203.00274*, 2022.