

# Meta-learning in movement prediction problem of aperiodic time-series data

Bao-Long Nguyen,<sup>1,\*</sup> Tom Ichibha,<sup>1,†</sup> Kenta Hongo,<sup>2,‡</sup> and Ryo Maezono<sup>1,§</sup>

<sup>1</sup>*School of Information Science, JAIST, Ishikawa, Japan*

<sup>2</sup>*Research Center for Advanced Computing Infrastructure, JAIST, Ishikawa, Japan*

(Dated: August 25, 2024)

**Abstract.** Dự đoán aperiodic time-series data (e.g. stock price, foreign exchange, Bitcoin price,...) là một tác vụ khó khăn đối với các mô hình học máy bởi vì loại dữ liệu này có phương sai cao và không cố định; không thể hiện chu kỳ rõ ràng nên khó rút trích đặc trưng; không chỉ phụ thuộc vào giá trị quá khứ mà còn phụ thuộc các yếu tố khác, điều khiến chúng bất ổn và không có chu kỳ như tình hình kinh tế, chính trị. Để khắc phục các thách thức nêu trên, chúng tôi sử dụng Meta-learning để huấn luyện kết hợp mạng LSTM và CNN, từ đó rút trích và tổng hợp hiệu quả các đặc trưng ẩn của dữ liệu theo thời gian. Thử nghiệm trên dữ liệu foreign exchange của 15 cặp tiền tệ trong vòng 24 năm (2000-2024) cho thấy phương pháp đề xuất hoạt động tốt và có độ chính xác cao hơn mô hình NHITS - state-of-the-art (SOTA) model của năm 2023 trên dữ liệu time-series, trong bài toán dự đoán xu hướng (upward or downward) của ngày giao dịch tiếp theo.

## I. INTRODUCTION

Aperiodic time-series prediction nói chung hay foreign exchange (FX) prediction nói riêng từ lâu đã là vấn đề đáng quan tâm của nhiều nghiên cứu [11, 15, 18]. Hai kỹ thuật chính được sử dụng trong aperiodic time-series prediction là fundamental analysis and technical analysis [2]. Trong khi fundamental analysis thiên về phân tích các yếu tố tác động từ bên ngoài và khó có thể capture từ các biến thiên giá trị trong quá khứ như chính sách, chiến lược kinh tế của công ty, quốc gia để dự đoán tương lai; technical analysis dựa hoàn toàn vào lịch sử biến động giá trị để phân tích xu hướng tương lai.

Việc dự đoán trên aperiodic time-series data gặp một vài thách thức cố hữu. Trong đó có thể kể tới: (1) - Variance của loại dữ liệu này biến thiên mạnh qua các thời kỳ. Từ đó giả thuyết rằng chúng tuân theo một phân phối để có thể xấp xỉ lỗi là không thể sử dụng được, dẫn đến việc các mô hình học máy khó có thể dự đoán chính xác giá trị cũng như xu hướng trong tương lai; (2) - Aperiodic data không tuân theo bất kỳ quy luật rõ ràng nào nên việc học các đặc trưng ẩn của dữ liệu để tiến hành dự đoán gặp rất nhiều khó khăn; (3) - Aperiodic time-series data (e.g. giá cổ phiếu của một công ty) không hoàn toàn phụ thuộc vào dữ liệu quá khứ mà còn phụ thuộc nhiều yếu tố ngoại lai (e.g. tin tức, tình hình kinh tế, chính trị) [18].

Đối với thách thức đầu tiên, các mô hình ensemble [cite something here] thường được sử dụng để hạn chế ảnh hưởng của sự biến đổi variance. Ensemble model giúp cung cấp một cái nhìn tổng quát, tổng hợp từ nhiều khía cạnh dựa trên các sub-models, từ đó giúp mô hình tổng quát thích ứng được với sự thay đổi mạnh của variance.

Chúng tôi tiếp cận bài toán theo hướng này nhưng ở mức cao hơn bằng cách sử dụng Meta-learning (ML) [9]. Phương pháp này tổng hợp hiệu quả tham số của các mô hình cục bộ, giúp giảm thiểu đáng kể variance loss.

Để khắc phục thách thức thứ hai, các nghiên cứu phần lớn sử dụng các đặc trưng rút trích được từ Long short-term memory neural network (LSTM) [12], Artificial neural network (ANN), và Convolution neural network (CNN) [17]. Cụ thể, trong năm 2022, 20% tổng số các bài báo liên quan đến dự đoán chỉ số tài chính sử dụng LSTM, 20% sử dụng ANN và 6% sử dụng CNN [2]. Để tận dụng hết được các đặc trưng được rút trích bởi các mô hình nêu trên, chúng tôi đề xuất phương pháp kết hợp các đặc trưng này.

Đối với thách thức thứ ba, nghiên cứu [8] đưa ra giả thuyết rằng các time-series datasets khác nhau của cùng một lĩnh vực tại cùng thời điểm phản ánh sự tác động của các yếu tố ngoại lai. Ví dụ, các nghiên cứu [1, 20] cũng chỉ ra sự phụ thuộc giữa chỉ số tài chính của một công ty nhất định và các chỉ số của các công ty khác. Điều này càng làm tăng tính đúng đắn của giả thuyết trong nghiên cứu [8]. Ngoài ra, chúng tôi cho rằng, loại dữ liệu này còn có những phụ thuộc ngầm vào các thời điểm nhất định trong quá khứ (hidden long-term dependency). Đối với cách tiếp cận truyền thống, người ta sử dụng một lượng dữ liệu quá khứ cố định (lookback window) để huấn luyện mô hình. Điều này gây một trở ngại lớn cho quá trình học vì các đặc trưng dài hạn theo thời gian sẽ bị quên. Mặt khác, ML chia nhỏ tập dữ liệu thành nhiều phần để học và tổng hợp hiệu quả các tham số học được nên có thể xử lý tốt thách thức này.

Cuối cùng, chúng tôi chứng minh tính ưu việt của thuật toán đề xuất bằng cách giải bài toán dự đoán xu hướng (tăng hoặc giảm) của tỉ giá ngoại hối và so sánh kết quả với mô hình SOTA hiện tại (NHITS [3]) trên hai loại dữ liệu: (1) - Dữ liệu tỉ giá của cặp tiền tệ USD/JPY; (2) - Dữ liệu tỷ giá của 60 cặp tiền tệ, cấu thành từ 18 quốc gia. Các tập dữ liệu này được công khai trên Internet và có thể tải về dễ dàng. Bản cài đặt chính thức có thể xem

\* mwklng2309@icloud.com

† ichibha@icloud.com

‡ kenta\_hongo@mac.com

§ rmaezono@mac.com

tại [bỏ cái link github vào đây].

Tóm lại, đóng góp chính của chúng tôi như sau:

- Kết hợp đặc trưng: Rút trích đặc trưng bằng cách kết hợp các đặc trưng của LSTM và CNN.
- Hidden long-term dependency: Chứng minh thực nghiệm rằng một aperiodic time-series data nhất định không chỉ phụ thuộc vào các yếu tố ngoại lai mà còn có các phụ thuộc ẩn với chính nó tại nhiều thời điểm khác nhau trong quá khứ.
- Tổng hợp hiệu quả tham số mô hình: Sử dụng ML thay cho các mô hình ensemble truyền thống trong việc tổng hợp kết quả từ các mô hình học máy.
- Experiment: Thực nghiệm trên các bộ dữ liệu về tỉ giá hối đoái và so sánh với mô hình SOTA NHITS để chứng minh tính hiệu quả của phương pháp đề xuất.

## II. RELATED WORK

### A. LSTM & CNN model

Như đã đề cập, LSTM là mạng neural rất phổ biến trong việc handle các bài toán liên quan đến dự đoán trên dữ liệu time-series. LSTM được sử dụng phổ biến như vậy bởi vì nó xử lý tốt vấn đề vanishing gradient (dễ dàng bắt gặp khi sử dụng Recurrent neural network) và có thể khai thác hiệu quả các mối quan hệ phi tuyến trong dữ liệu. Thật vậy, bằng cách duy trì cell-state trong mỗi iteration, LSTM có thể khắc phục vấn đề vanishing gradient, từ đó bảo toàn khả năng capture các phụ thuộc dài hạn [5]. Ngoài ra, LSTM thực hiện rút trích đặc trưng với các hàm kích hoạt phi tuyến, giúp các tham số mô hình có thể capture tính phi tuyến của dữ liệu [10]. Hai yếu tố nêu trên khiến cho LSTM trở thành lựa chọn đầu tiên được nghĩ đến khi giải quyết các bài toán trên dữ liệu time-series.

Mạng CNN được sử dụng rất nhiều trong các tác vụ xử lý hình ảnh [21, 23] bởi khả năng tổng hợp các quan hệ cục bộ. Không chỉ vậy, CNN còn được dùng rất nhiều trong các tác vụ xử lý dữ liệu time-series như speech recognition [6], natural language processing [24]. Điều đó chứng tỏ được khả năng của CNN trong việc khám phá mối quan hệ thời gian cục bộ giữa các mẫu dữ liệu. Mặc dù vậy, CNN lại rất ít được dùng trong các tác vụ dự đoán stock price hay foreign exchange. Trong nghiên cứu này, chúng tôi tận dụng khả năng rút trích đặc trưng cục bộ tuyệt vời của CNN để tích hợp thêm thông tin ẩn vào quá trình huấn luyện của mô hình.

### B. Model-agnostic Meta-learning (MAML)

Các thuật toán Meta-learning (ML), điển hình là MAML [9] được biết đến với khả năng huấn luyện một

mô hình có tính tổng quát cao, thích ứng nhanh trên tập dữ liệu mới thông qua một lượng nhỏ dữ liệu và số bước huấn luyện [13, 25]. Với khả năng này, ML được sử dụng rất nhiều trong các tác vụ đòi hỏi khả năng đáp ứng của mô hình trên dữ liệu (e.g. cá nhân hóa mô hình học [4, 7, 22], domain adaptation trong online learning [14, 16]).

Một thuật toán ML cơ bản sẽ được học trên nhiều tác vụ  $t$  rút ra từ cùng một phân phối tác vụ  $\mathcal{T}$  [13]. Dữ liệu của mỗi tác vụ được chia thành tập support  $\mathcal{D}_t^{support}$  (thường có kích thước nhỏ, khoảng 20%) và tập query  $\mathcal{D}_t^{query}$ . Trong quá trình học, hai bước tối ưu inner và outer optimization được perform đan xen. Inner optimization cố gắng tìm ra một bộ tham số tối ưu  $\theta_t^*$  cho từng mô hình học máy trên tập support của mỗi tác vụ bằng phương trình 1.

$$\theta_t^* = \theta_t(\phi) = \arg \min_{\theta} \mathcal{L}_t^{task}(\phi, \mathcal{D}_t^{support}) \quad (1)$$

Trong đó,  $\phi$  là kết quả của quá trình outer optimization, đóng vai trò là giá trị khởi tạo của  $\theta_t$ .  $\mathcal{L}_t$  là hàm loss của mô hình trên tác vụ  $t$ .

Sau đó, thuật toán sử dụng các bộ tham số tối ưu  $\theta_t^*$  để perform trên tập query tương ứng. Loss của toàn bộ mô hình sau đó được tổng hợp để thực hiện quá trình outer optimization như phương trình 2.

$$\begin{aligned} \phi^* &= \arg \min_{\phi} \sum_t \mathcal{L}_t^{meta}[\theta_t^*, \mathcal{D}_t^{query}] \\ &= \arg \min_{\phi} \sum_t \mathcal{L}_t^{meta}[\theta_t(\phi), \mathcal{D}_t^{query}] \end{aligned} \quad (2)$$

Trong inference phase, giá trị khởi tạo cho tham số của mô hình được gán bằng  $\phi^*$ . Mô hình sau đó được huấn luyện nhanh trên tập support sau đó perform trên tập query. Kết quả trên tập query chính là kết quả của mô hình.

Bằng hình thức huấn luyện trên, mô hình  $\phi^*$  sẽ có mức tổng quát hóa cao trên các tác vụ khác nhau, có thể nhanh chóng đáp ứng một tác vụ mới chỉ sau một vài bước huấn luyện.

Các mô hình hybrid ensemble vốn được sử dụng rất nhiều trong các bài toán xử lý time-series và được chứng minh thực nghiệm là có độ chính xác cao hơn so với các mô hình handle time-series data tiêu chuẩn vì có thể tổng hợp được sức mạnh của nhiều mô hình [2]. Tuy vậy, các hình thức tổng hợp của ensemble model hiện nay vẫn còn rất cứng nhắc vì chỉ có thể tổng hợp dựa trên kết quả cuối (cơ chế voting của bagging models) và kết quả gần cuối (đối với stacking models). Dưới góc nhìn của ensemble model, có thể coi phương trình 2 là một phương pháp tổng hợp hiệu quả các sub-model, giúp tận dụng khả năng rút trích đặc trưng của từng mô hình. Nói cách khác, mô hình sau khi tổng hợp có thể rút trích

đặc trưng ở mức sâu hơn, cải thiện đáng kể khả năng dự đoán so với các mô hình ensemble truyền thống.

### C. Neural Hierarchical Interpolation for Time Series (NHITS)

NHITS được thiết kế để hướng đến việc dự đoán các long-horizon time-series data. Theo nghiên cứu [3], cấu trúc của NHITS bao gồm nhiều stack liên tiếp nhau. Mỗi stack bao gồm nhiều block nối tiếp nhau. Tại mỗi block, dữ liệu lịch sử được sử dụng để dự đoán dữ liệu tương lai và dữ liệu quá khứ. Cụ thể, tại block  $l$ , với  $L$  mẫu dữ liệu quá khứ ( $\mathbf{y}_{t-L:t,l-1}$ ), các đặc trưng sẽ được rút trích như sau (theo [3]):

$$\mathbf{y}_{t-L:t,l}^{(p)} = \text{Pooling}(\mathbf{y}_{t-L:t,l-1}) \quad (3)$$

$$\theta_l^b = \text{FullyConnected}^b(\mathbf{y}_{t-L:t,l}^{(p)}) \quad (4)$$

$$\theta_l^f = \text{FullyConnected}^f(\mathbf{y}_{t-L:t,l}^{(p)}) \quad (5)$$

$$\hat{\mathbf{y}}_{t-L:t,l} = g(\theta_l^b) \quad (6)$$

$$\hat{\mathbf{y}}_{t+1:t+H,l} = g(\theta_l^f) \quad (7)$$

Trong đó, **FullyConnected** là các lớp multi-layer perception (MLP) xếp chồng với hàm kích hoạt phi tuyến.  $\theta_l^f, \theta_l^b$  là các hệ số nội suy forecast và backcast, được dùng để tổng hợp các giá trị đầu ra của block  $l$  bằng hàm nội suy  $g(\cdot)$ . Đầu ra của block  $l$  là giá trị forecast  $\hat{\mathbf{y}}_{t+1:t+H,l}$  và giá trị backcast  $\hat{\mathbf{y}}_{t-L:t,l}$ . Input của block  $l+1$  được tính theo phương trình 8.

$$\mathbf{y}_{t-L:t,l+1} = \mathbf{y}_{t-L:t,l-1} - \hat{\mathbf{y}}_{t-L:t,l} \quad (8)$$

Giả sử mô hình gồm có  $S$  stacks, mỗi stack có  $B$  blocks. Tổng hợp các giá trị forecast của các block như phương trình 9, ta được giá trị forecast của một stack. Backcast của block cuối cùng của một stack chính là đầu vào cho stack tiếp theo. Cuối cùng, tổng hợp giá trị forecast của các stack như phương trình 10, ta được giá trị forecast dự đoán của toàn mạng.

$$\hat{\mathbf{y}}_{t+1:t+H}^s = \sum_{l=1}^B \hat{\mathbf{y}}_{t+1:t+H,l} \quad (9)$$

$$\hat{\mathbf{y}}_{t+1:t+H} = \sum_{s=1}^S \hat{\mathbf{y}}_{t+1:t+H}^s \quad (10)$$

Bằng cách xếp chồng các stack, stack sau nhận vào phần dư của stack trước, kiến trúc trên được kỳ vọng là sẽ phân rã dữ liệu thành các frequency bands khác nhau (weekly, daily, even hourly). Trên thực tế, NHITS perform rất tốt đối với các bộ dữ liệu có tính chu kỳ cao như mức tiêu thụ điện, thời tiết, giao thông. Tuy nhiên, chúng

tôi đang hướng đến aperiodic time-series dataset, vốn có tính chu kỳ rất thấp, thậm chí không có (see figure 1). Điều này gây ra khó khăn rất lớn cho NHITS.

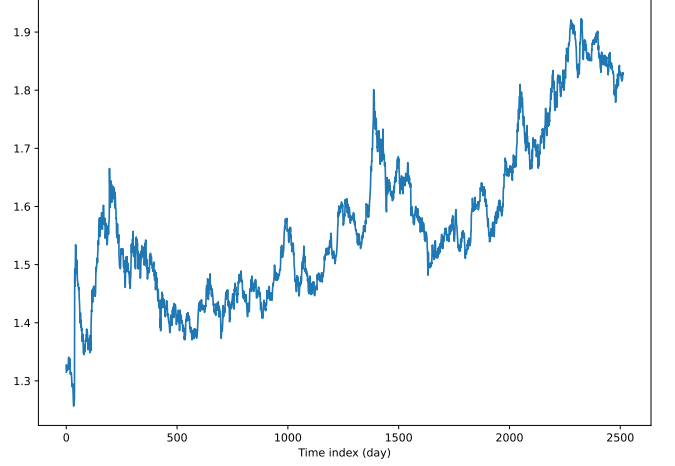


FIG. 1. Exchange rate (close price) between Swiss franc and New Zealand dollar by day (2014-2024).

## III. METHODOLOGY

Phương pháp của chúng tôi bao gồm hai phần chính hoạt động song song nhau: (1) - Feature extraction; (2) - Parameter synthesis. Tổng quan phương pháp được minh họa trong hình 2. Trong phần feature extraction, chúng tôi kết hợp hai loại đặc trưng từ mạng CNN và LSTM. Trong phần parameter synthesis, chúng tôi sử dụng MAML để tổng hợp tham số của các mô hình. Với sự góp mặt của các đặc trưng LSTM và CNN, chúng tôi kỳ vọng sẽ rút trích được các đặc trưng ẩn trong dữ liệu aperiodic. Bằng việc sử dụng MAML trong quá trình tổng hợp trọng số, phương pháp đề xuất được kỳ vọng là một giải pháp thay thế hợp lý và hiệu quả cho các mô hình ensemble truyền thống trong việc giảm thiểu tác động của sự biến thiên variance, tổng hợp hiệu quả các yếu tố ngoại lai, cũng như giữ được các hidden long-term dependency ẩn trong quá khứ.

### A. Data preparation

Phương pháp đề xuất sử dụng các thuật toán ML để huấn luyện mô hình. Do đó, dữ liệu cần được tổ chức lại để các thuật toán ML có thể hoạt động được. Trong trường hợp dữ liệu bao gồm nhiều datasets khác nhau thuộc cùng một lĩnh vực, mỗi dataset sẽ được coi là một task của MAML. Trong trường hợp dữ liệu bao gồm một dataset duy nhất, cần chia nhỏ dataset này thành các tập con ứng với các task riêng biệt. Tóm lại, tập dữ liệu sau khi chuẩn bị bao gồm  $n$  tasks:  $\mathcal{D} = \{\mathcal{D}_t\}_{t=1}^n$ . Dữ liệu tại mỗi task được chia thành tập support và query:

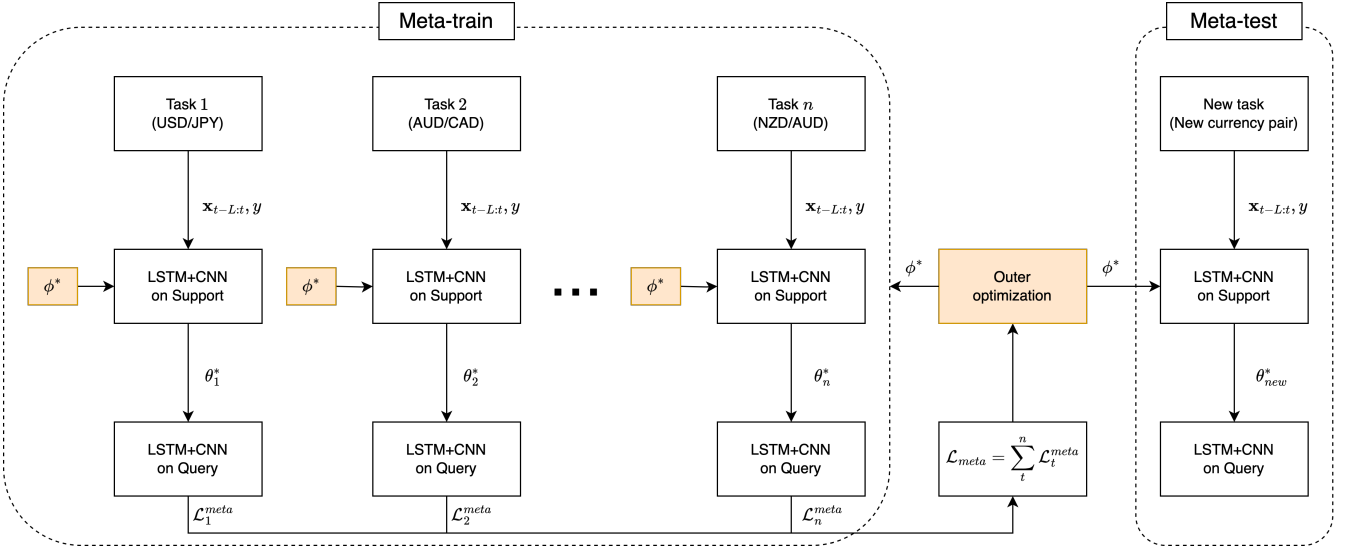


FIG. 2. The full-flow of meta-training and meta-testing on multi-fx data. Each currency pair is regarded as a task.

$$\mathcal{D}_t = \{\mathcal{D}_t^{support}, \mathcal{D}_t^{query}\}.$$

Một sample dữ liệu bao gồm các cặp giá trị  $(\mathbf{x}_{t-L:t}, y)$ . Trong đó,  $\mathbf{x}_{t-L:t}$  bao gồm  $L$  giá trị lịch sử tính từ thời điểm  $t$  trở về trước;  $y \in \{0, 1\}$  là nhãn dữ liệu, thể hiện xu hướng giảm, hoặc tăng của mẫu dữ liệu  $x_{t+1}$  so với  $x_t$ . Tùy vào từng bài toán và cách cài đặt mà các phần tử trong  $\mathbf{x}_{t-L:t}$  có thể là các vector hoặc các scalar number. Ví dụ, đối với dữ liệu chứng khoán,  $\mathbf{x}_{t-L:t}$  có thể chứa các vector dữ liệu  $\tilde{x}_i = (\text{open}, \text{low}, \text{high}, \text{close})$  hoặc chỉ một giá trị close price duy nhất.

### B. Feature extraction

Lấy cảm hứng từ nghiên cứu [26], chúng tôi đề xuất kết hợp các đặc trưng rút trích được từ mạng LSTM và CNN. Cụ thể, chúng tôi đưa từng phần tử trong vector  $\mathbf{x}_{t-L:t}$  qua một lớp MLP có đầu ra lớn hơn số chiều của  $\tilde{x}_i, i \in [t-L, t]$  để phân giải thành các đặc trưng nhỏ  $\mathbf{x}'_i$ . Các đặc trưng này sau đó được truyền qua mạng LSTM và CNN để lần lượt rút trích các phụ thuộc thời gian dài hạn ( $\mathbf{h}_{LSTM}$ ) và các đặc trưng thời gian cục bộ ( $\mathbf{h}_{CNN}$ ). Để có thể khai thác tối đa các ràng buộc thời gian dài hạn, chúng tôi sử dụng BidirectionalLSTM để rút trích từ hai phía của của  $\mathbf{x}_{t-L:t}$ . Toàn bộ quy trình rút trích đặc trưng được tóm tắt như sau:

$$\mathbf{x}'_{t-L:t} = \text{FullyConnected}(\mathbf{x}_{t-L:t}) \quad (11)$$

$$\mathbf{h}_{LSTM} = \text{BidirectionalLSTM}(\mathbf{x}'_{t-L:t}) \quad (12)$$

$$\mathbf{h}_{CNN} = \text{Convolution1D}(\mathbf{x}'_{t-L:t}) \quad (13)$$

Mạng LSTM duy trì giá trị cell-state nhằm lưu trữ có chọn lọc các phụ thuộc dài hạn. Điều này rất thích hợp trong việc giải quyết các bài toán time-series data. Mặt

khác, giá trị tương lai thường phụ thuộc rất lớn vào các giá trị lịch sử gần nhất. Chúng tôi đề xuất sử dụng mạng CNN để nhấn mạnh các đặc trưng cục bộ, từ đó hướng một phần sự chú ý của mô hình vào các thời điểm nhất định. Do đó, phương pháp đề xuất không chỉ nhớ được các đặc trưng long-term mà còn highlight được các đặc trưng short-term.

Tiếp đến,  $\mathbf{h}_{LSTM}$  và  $\mathbf{h}_{CNN}$  được nối với nhau (phương trình 14) sau đó chuyển đến phần phân lớp của mạng NN (phương trình 15).

$$\mathbf{h}_{t-L:t} = \text{Concatenate}(\mathbf{h}_{LSTM}, \mathbf{h}_{CNN}) \quad (14)$$

$$\hat{y} = \text{FullyConnected}(\mathbf{h}_{t-L:t}) \quad (15)$$

### C. Effective synthesis of models' parameters

Chúng tôi sử dụng MAML để huấn luyện và tổng hợp trọng số của các mô hình tại các task. Như đã đề cập trong phần II, tối ưu tham số theo cách tiếp cận của ML chính là đi giải hai phương trình 1 và 2 bằng các phương pháp tối ưu trên dữ liệu support và query. Cụ thể, quá trình tối ưu bao gồm nhiều bước toàn cục (outer optimization), thực hiện trên tất cả các tasks tham gia huấn luyện. Mỗi bước toàn cục bao gồm nhiều bước cục bộ (inner optimization) thực hiện trên từng task riêng lẻ. Tại bước toàn cục  $r$ , quá trình tối ưu cục bộ lần thứ  $e$  tại tập support của task  $t$  diễn ra như sau:

$$\begin{cases} \theta_t^{(0)} &= \phi_{r-1} \\ \theta_t^{(e)} &= \theta_t^{(e-1)} - \alpha \nabla_{\theta} \mathcal{L}_t^{task}(\theta_t^{(e-1)}, \mathcal{D}_t^{support}) \end{cases} \quad (16)$$

Trong đó,  $\phi_{r-1}$  là kết quả của quá trình tối ưu toàn cục lần thứ  $r-1$ ,  $\alpha$  là inner learning rate.

Tiếp đó, quá trình outer optimization tại bước toàn cục được thực hiện bằng cách tổng hợp độ lỗi trên tập query của các task và tối ưu trên đó (phương trình 17).

$$\begin{cases} \phi_0 = \text{Random Initialization} \\ \phi_r = \phi_{r-1} - \beta \nabla_{\phi} \sum_{t=1}^n \mathcal{L}_t^{\text{meta}}(\theta_t^*(\phi), \mathcal{D}_t^{\text{query}}) \end{cases} \quad (17)$$

Trong đó,  $\beta$  là outer learning rate. Giả sử thuật toán chạy  $E$  steps trong inner optimization, lượng đạo hàm tại phương trình 17 được viết lại như sau (the notations of dataset are removed):

$$\begin{aligned} & \beta \nabla_{\phi} \sum_{t=1}^n \mathcal{L}_t^{\text{meta}}(\theta_t - \alpha \nabla_{\theta} \mathcal{L}_t^{\text{task}}(\theta_t)) \\ &= \beta \sum_{t=1}^n \frac{\partial \mathcal{L}_t^{\text{meta}}(\theta_t^{(E)})}{\partial \theta_t^{(E)}} \frac{\partial \theta_t^{(E)}}{\partial \phi} \\ &= \beta \sum_{t=1}^n \nabla_{\theta} \mathcal{L}_t^{\text{meta}}(\theta_t^{(E)}) \prod_{j=0}^{E-1} \left[ \mathbb{I} - \alpha \nabla_{\theta}^2 \mathcal{L}_t^{\text{task}}(\theta_t^{(j)}) \right] \end{aligned} \quad (18)$$

Sự xuất hiện của tích các đạo hàm bậc hai trong phương trình 18 khiến quá trình đạo hàm trở nên phức tạp vì phải tốn rất nhiều chi phí để duy trì các ma trận Hessian. Do đó, số bước tính toán để tìm ra  $\theta^*$  cần phải hạn chế. Trên thực tế, các phương pháp sử dụng ML [4, 7, 9, 19, 22] thường sẽ chọn  $E \in [1, 5]$ .

## IV. NUMERICAL EXPERIMENT

### A. Dataset & Metric

FX nói riêng và các chỉ số tài chính nói chung là dạng dữ liệu điển hình cho aperiodic time-series data. Do đó, chúng tôi chọn loại dữ liệu này để kiểm thử mô hình. Cụ thể, chúng tôi cấu hình hai bộ dữ liệu sử dụng dữ liệu FX. Bộ dữ liệu USD/JPY chỉ gồm dữ liệu của cặp tiền tệ USD/JPY, được chia thành 60 tập dữ liệu con theo trình tự thời gian với kích thước bằng nhau. Dữ liệu được sample theo giờ từ năm 2000 đến năm 2024, bao gồm các thuộc tính open, low, high, và close price. Bộ dữ liệu multi-fx bao gồm 60 cặp tiền tệ giữa 18 quốc gia Australia, Canada, Switzerland, Denmark, EU, United Kingdom, Hong Kong, Iceland, Japan, Norway, New Zealand, Singapore, Sweden, Turkey, United States, Mexico, China, South Africa. Dữ liệu có thuộc tính tương tự như USD/JPY và được sample theo ngày từ năm 2014 đến 2024. Số lượng mẫu dữ liệu của hai tập dữ liệu là tương đương nhau và xấp xỉ 156000 mẫu.

Bộ dữ liệu multi-fx được sử dụng để rút trích và tổng hợp thông tin về các yếu tố ngoại lai (i.e. thông tin thị trường, kinh tế, chính trị,...), vốn được cho là có ảnh hưởng đến kết quả của một chỉ số tài chính nhất định [1, 8, 20]. Bộ dữ liệu USD/JPY được sử dụng để kiểm chứng giả thuyết của chúng tôi về việc dữ liệu tương lai ngầm phụ thuộc vào các thời điểm nhất định trong quá khứ và cần phải tổng hợp đặc trưng quá khứ một cách hiệu quả để làm rõ được các phụ thuộc này.

Nghiên cứu sử dụng macro accuracy, macro precision, macro recall, và macro F1 để đánh giá các mô hình. Theo đó, trong quá trình inference, mô hình sẽ chạy trên từng task để tính metrics của mỗi task. Sau đó tính trung bình cộng các metrics của các task để thu được kết quả cuối.

### B. Experiment

Chúng tôi tiến hành so sánh phương pháp đề xuất với mô hình NHITS thông qua các metrics nêu trên. Dữ liệu tổng quan được cấu trúc thành training sets, validation sets, và testing sets với tỷ lệ 6:2:2. Tập train dùng để huấn luyện mô hình, tập validation dùng để tìm kiếm siêu tham số và tập test để đánh giá mô hình. Đối với NHITS, chúng tôi chia dữ liệu như bình thường theo tỷ lệ định sẵn ở trên. Đối với thuật toán đề xuất, vì quá trình meta-training và meta-testing đòi hỏi phải chia dữ liệu thành các task nhỏ, và cho phép mô hình adapt trên tập support của mỗi task, chúng tôi chia dữ liệu thành 60 tasks. Trong mỗi task, tập support chiếm 20% với mục đích để mô hình học cách thích ứng với dữ liệu, tập query chiếm 80% để kiểm tra khả năng tương thích của mô hình. Chúng tôi sử dụng 30 tasks để meta-train, 15 tasks để meta-validate, 15 tasks để meta-test. Với cách chia này, chúng tôi đảm bảo được tính công bằng khi mô hình ML được huấn luyện với lượng dữ liệu tương đương mô hình NHITS.

## V. RESULTS AND DISCUSSION

## VI. CONCLUSION & FUTURE DIRECTION

## VII. ACKNOWLEDGEMENTS

## VIII. AUTHOR CONTRIBUTIONS

## IX. DATA AVAILABILITY STATEMENT

[1] Andrew W. Lo, A.C.M., 1990. When are contrarian profits due to stock market overreaction? The review of

financial studies 3, 175–205.

TABLE I. meo meo

	USD/JPY				multi-fx			
	<i>accuracy</i>	<i>precision</i>	<i>recall</i>	$F_1 - score$	<i>accuracy</i>	<i>precision</i>	<i>recall</i>	$F_1 - score$
NHITS								
Ours					$62.39 \pm 0.06$	$71.11 \pm 9.34$	$70.14 \pm 9.6$	$69.21 \pm 11.07$

- [2] Ayitey Junior, M., Appiahene, P., Appiah, O., Bombie, C.N., 2023. Forex market forecasting using machine learning: Systematic literature review and meta-analysis. *Journal of Big Data* 10, 9.
- [3] Challu, C., Olivares, K.G., Oreshkin, B.N., Ramirez, F.G., Canseco, M.M., Dubrawski, A., 2023. Nhits: Neural hierarchical interpolation for time series forecasting, in: *Proceedings of the AAAI conference on artificial intelligence*, pp. 6989–6997.
- [4] Chen, F., Luo, M., Dong, Z., Li, Z., He, X., 2018. Federated meta-learning with fast convergence and efficient communication. *arXiv preprint arXiv:1802.07876*.
- [5] Cheng, L.C., Huang, Y.H., Wu, M.E., 2018. Applied attention-based lstm neural networks in stock prediction, in: *2018 IEEE International Conference on Big Data (Big Data)*, IEEE. pp. 4716–4718.
- [6] Dua, S., Kumar, S.S., Albagory, Y., Ramalingam, R., Dumka, A., Singh, R., Rashid, M., Gehlot, A., Alshamrani, S.S., AlGhamdi, A.S., 2022. Developing a speech recognition system for recognizing tonal speech signals using a convolutional neural network. *Applied Sciences* 12, 6223.
- [7] Fallah, A., Mokhtari, A., Ozdaglar, A., 2020. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*.
- [8] Fama, E.F., 1970. Efficient capital markets: A review of theory and empirical work. *Journal of finance* 25, 383–417.
- [9] Finn, C., Abbeel, P., Levine, S., 2017. Model-agnostic meta-learning for fast adaptation of deep networks, in: *International conference on machine learning*, PMLR. pp. 1126–1135.
- [10] He, T., Droppo, J., 2016. Exploiting lstm structure in deep neural networks for speech recognition, in: *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE. pp. 5445–5449.
- [11] Heryadi, Y., Wibowo, A., et al., 2021. Foreign exchange prediction using machine learning approach: A pilot study, in: *2021 4th International Conference on Information and Communications Technology (ICOIACT)*, IEEE. pp. 239–242.
- [12] Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. *Neural computation* 9, 1735–1780.
- [13] Hospedales, T., Antoniou, A., Micaelli, P., Storkey, A., 2021. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence* 44, 5149–5169.
- [14] Hu, N., Mitchell, E., Manning, C.D., Finn, C., 2023. Meta-learning online adaptation of language models. *arXiv preprint arXiv:2305.15076*.
- [15] Islam, M.S., Hossain, E., 2021. Foreign exchange currency rate prediction using a gru-lstm hybrid network. *Soft Computing Letters* 3, 100009.
- [16] Khoei, A.G., Yu, Y., Feldt, R., 2024. Domain generalization through meta-learning: A survey. *arXiv preprint arXiv:2404.02785*.
- [17] LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., Jackel, L., 1989. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems* 2.
- [18] Li, C., Song, D., Tao, D., 2019. Multi-task recurrent neural networks and higher-order markov random fields for stock price movement prediction: Multi-task rnn and higher-order mrfs for stock price classification, in: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1141–1151.
- [19] Li, Z., Zhou, F., Chen, F., Li, H., 2017. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*.
- [20] Mech, T.S., 1993. Portfolio return autocorrelation. *Journal of Financial Economics* 34, 307–344.
- [21] Naranjo-Torres, J., Mora, M., Hernández-García, R., Barrientos, R.J., Fredes, C., Valenzuela, A., 2020. A review of convolutional neural network applied to fruit image processing. *Applied Sciences* 10, 3443.
- [22] Nguyen, B.L., Cao, T.C., Le, B., 2022. Meta-learning and personalization layer in federated learning, in: *Asian Conference on Intelligent Information and Database Systems*, Springer. pp. 209–221.
- [23] Sharma, N., Jain, V., Mishra, A., 2018. An analysis of convolutional neural networks for image classification. *Procedia computer science* 132, 377–384.
- [24] Varshitha, K.S., Kumari, C.G., Hasvitha, M., Fiza, S., Amarendra, K., Rachapudi, V., 2023. Natural language processing using convolutional neural network, in: *2023 7th International Conference on Computing Methodologies and Communication (ICCMC)*, IEEE. pp. 362–367.
- [25] Vettoruzzo, A., Bouguelia, M.R., Vanschoren, J., Rognvaldsson, T., Santosh, K., 2024. Advances and challenges in meta-learning: A technical review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- [26] Vo, Q.H., Nguyen, H.T., Le, B., Nguyen, M.L., 2017. Multi-channel lstm-cnn model for vietnamese sentiment analysis, in: *2017 9th international conference on knowledge and systems engineering (KSE)*, IEEE. pp. 24–29.