

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



ĐỒ ÁN TỐT NGHIỆP

Đề tài:

**Mô hình dự đoán chuỗi thời gian mờ đa chiều
sử dụng LSTM và kỹ thuật phân tích
tương quan giữa các chiều dữ liệu
áp dụng cho tài nguyên đám mây**

Sinh viên thực hiện: **Nguyễn Đức Thắng**

Giảng viên hướng dẫn: **TS. Nguyễn Bình Minh**

Hà Nội, 12/2018

Tóm tắt

Hiện nay, hầu hết các hệ thống đám mây cung cấp các phương thức tự động mở rộng tài nguyên dựa trên giá trị ngưỡng mà người dùng định nghĩa. Mặc dù vậy, ứng dụng các phương thức mở rộng thông qua dự đoán tài nguyên vẫn gặp phải vấn đề với độ chính xác trong hoạt động dự đoán mặc dù các phương pháp này mới chỉ giải quyết vấn đề dữ liệu một chiều. Trong những năm gần đây, đã có một vài nỗ lực cho việc sử dụng đồng thời dữ liệu nhiều chiều để tiến hành dự đoán tài nguyên sử dụng và ra quyết định mở rộng hệ thống. Động lực để thực hiện các phương thức đa chiều này là có thể có sự tương quan giữa các chiều dữ liệu và chúng phải được kiểm tra để cải thiện ứng dụng mô hình trong thực tế. Trong nghiên cứu này, chúng tôi đã xây dựng một mô hình dự đoán về mặt lý thuyết cho các hệ thống đám mây tự động mở rộng với sự kết hợp của một vài phương pháp. Trong bước tiền xử lý dữ liệu, để giảm thiểu sự biến động dữ liệu, chúng tôi sử dụng kỹ thuật mờ hóa dữ liệu. Chúng tôi tính toán sự tương quan giữa các chiều dữ liệu khác nhau để chọn ra các chiều dữ liệu phù hợp làm đầu vào cho mô hình dự đoán. Ngoài ra, mô hình mạng bộ nhớ ngắn hạn dài (LSTM) cũng được xây dựng để tiến hành dự đoán tài nguyên sử dụng với dữ liệu chuỗi thời gian đa chiều tại cùng thời điểm. Mô hình của chúng tôi vì thế được gọi là mô hình MF- LSTM. Mô hình trên được thử nghiệm với dữ liệu thực tế ghi lại lịch sử tài nguyên sử dụng của các cụm máy chủ Google để đánh giá sự hiệu quả và tính khả thi của mô hình khi áp dụng và các mô hình đám mây.

Từ khóa: Bộ dữ liệu lịch sử tài nguyên sử dụng các cụm máy chủ Google, chuỗi thời gian đa chiều, Điện toán đám mây, sự tương quan dữ liệu.

Lời cảm ơn

Đầu tiên, em xin được gửi lời cảm ơn chân thành đến các thầy giáo, cô giáo thuộc trường đại học Bách Khoa Hà Nội, đặc biệt là các thầy giáo, cô giáo thuộc Viện Công nghệ Thông tin và Truyền Thông đã tận tình dạy dỗ và trang bị cho em những kiến thức bổ ích trong năm năm vừa qua.

Đồng thời em cũng xin được gửi lời cảm ơn đặc biệt đến TS Nguyễn Bình Minh. Thầy là người đã chỉ dẫn tận tình, cho em những kinh nghiệm quý báu để có thể hoàn thành đồ án này.

Em xin gửi lời cảm ơn tới gia đình và bạn bè đã sát cánh, động viên và hỗ trợ em trong suốt những năm qua.

Lời cam đoan

Họ và tên sinh viên: Nguyễn Đức Thắng

Điện thoại liên lạc: 0974120553 Email: thangbk2209@gmail.com

Lớp: CNTT2-2 K59

Hệ: Kỹ sư chính quy

Tôi - Nguyễn Đức Thắng - cam kết Đồ án Tốt nghiệp (DATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của TS. Nguyễn Bình Minh. Các kết quả nêu trong DATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong DATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

Hà Nội, ngày 28 tháng 12 năm 2018

Chữ ký của giảng viên hướng dẫn

Tác giả đồ án tốt nghiệp

Nguyễn Bình Minh

Nguyễn Đức Thắng

Mục lục

Abstract	2
Acknowledgements	3
1 Giới thiệu đề tài	10
1.1 Đặt vấn đề bài toán	10
1.2 Bố cục đồ án	12
2 Tổng quan các nghiên cứu và các kiến thức cơ sở	13
2.1 Điện toán đám mây	13
2.2 Bài toán tự động mở rộng	17
2.3 Các phương pháp dự đoán chuỗi thời gian	21
2.3.1 Tổng quan	21
2.3.2 Các phương pháp tuyến tính	23
2.3.3 Mô hình mạng nơ-ron nhân tạo	26
2.3.4 Phương pháp học sâu	28
2.3.5 Phương pháp chuỗi thời gian mờ	31

2.4	Phương pháp phân tích mối tương quan nhiều chiều	33
3	Mô hình đề xuất	36
3.1	Tổng quan mô hình	36
3.2	Tiền xử lý dữ liệu	37
3.3	Lựa chọn đặc trưng	38
3.4	Huấn luyện mô hình	39
3.5	Dự đoán	40
4	Thực nghiệm và đánh giá	42
4.1	Cài đặt thực nghiệm	42
4.2	Mở hóa dữ liệu	43
4.3	Lựa chọn đặc trưng	44
4.4	Dự đoán	46
5	Kết luận và hướng phát triển	50

Danh sách bảng

4.1	Kết quả thực nghiệm lựa chọn đặc trưng	45
4.2	Dự đoán CPU	47
4.3	Dự đoán Memory	47

Danh sách hình vẽ

2.1	Các mô hình dịch vụ trong Điện toán đám mây. Nguồn [11]	15
2.2	Sự cung cấp tài nguyên theo yêu cầu. Nguồn [11]	18
2.3	Phương pháp chu kỳ. Nguồn [4]	19
2.4	Phương pháp sử dụng ngưỡng.	20
2.5	Phương pháp dự đoán trước lượng tài nguyên sử dụng. Nguồn [11]	21
2.6	Mô hình mạng nơ-ron truyền thẳng với một tầng ẩn	25
2.7	Mạng nơ-ron hồi quy, Nguồn [3]	28
2.8	Đồ thị tính toán để tính ra lỗi huấn luyện của mạng hồi quy, Nguồn [4]	29
2.9	Mô hình mạng LSTM. Nguồn [4]	30
3.1	Kiến trúc tổng quan của hệ thống	37
3.2	Mức độ tiêu thụ CPU theo thời gian của một job trong bộ dữ liệu Google trace	38
3.3	Mô hình LSTM	40
4.1	Chuỗi thời gian thực tế và chuỗi thời gian mờ tương ứng	44
4.2	Nonlinearity correlation	45
4.3	Kết quả dự đoán với $sw=2$ và $\delta = 0.8$	47

4.4	Kết quả dự đoán với $sw=2$ và $\delta = 0.45$	47
4.5	Kết quả dự đoán với $sw=2$ và $\delta = 0.3$	48
4.6	Kết quả dự đoán với $sw=2$ và đầu vào đơn biến	48

Chương 1

Giới thiệu đề tài

1.1 Đặt vấn đề bài toán

Sự phát triển của các mô hình điện toán đám mây trong những năm qua được cho là nhờ lợi ích mà các mô hình trả phí theo nhu cầu sử dụng (pay-as-you-go model) mang lại. Mô hình pay-as-you-go là mô hình mà những người dùng điện toán đám mây trả phí cho các nhà cung cấp dịch vụ dựa trên lượng tài nguyên tính toán mà họ đã sử dụng. Lợi ích chính của mô hình điện toán đám mây là tránh lãng phí tài nguyên. Người dùng sẽ chỉ cần trả phí cho những tài nguyên mà họ sử dụng thay vì cung cấp một lượng tài nguyên cố định mà có thể không dùng đến. Với mô hình pay-as-you-go, các nhà cung cấp dịch vụ đám mây cần bảo đảm chất lượng cung cấp dịch vụ đám mây của mình với người dùng dựa trên các độ đo chất lượng dịch vụ khác nhau (Quality of Service). Nhờ đó, người dùng dịch vụ điện toán đám mây có thể tập trung vào việc cung cấp các dịch vụ của mình tới khách hàng mà không cần phải có kiến thức, kinh nghiệm hay các cơ sở hạ tầng phục vụ các hệ thống đám mây. Các mô hình điện toán đám mây khắc phục được 2 yếu điểm quan trọng của mô hình truyền thống về “khả năng mở rộng (scalability)” và “độ linh hoạt (flexibility)”. Theo đó, các hệ thống điện toán có khả năng co giãn (mở rộng hoặc thu hồi) tài nguyên tính toán một cách tự động dựa vào nhu cầu sử dụng của người dùng mà không cần họ phải có các kiến thức cho quá trình này. Với khả năng mở rộng và tính linh hoạt, các hệ thống điện toán đám mây có thể ngăn ngừa tình trạng

vượt quá khả năng cung cấp vào giờ cao điểm hoặc lãng phí tài nguyên vào giờ thấp điểm.

Hiện nay, hầu hết các hệ thống điện toán đám mây đều cung cấp khả năng tự động mở rộng tài nguyên sử dụng (auto scaling). Tuy nhiên, hầu hết các cách tiếp cận mở rộng tài nguyên đều dựa trên ngưỡng mà người dùng định nghĩa. Ví dụ, một người dùng định nghĩa rằng ngưỡng tự động mở rộng là 80% tài nguyên của một máy ảo. Điều đó có nghĩa là, khi lượng tài nguyên sử dụng của hệ thống ở thời điểm t vượt quá ngưỡng 80% tài nguyên của máy ảo đang sử dụng, hệ thống sẽ tự động gửi yêu cầu mở rộng để bật thêm một máy ảo nữa. Tuy nhiên, việc bật thêm một máy ảo mới sẽ mất một khoảng thời gian. Và trong khoảng thời gian này, lượng tài nguyên sử dụng của ứng dụng đã thay đổi, có thể đã vượt quá khả năng cung cấp của máy ảo hoặc đã giảm xuống mức không cần bật thêm máy ảo nữa. Ngoài ra, việc xác định được một giá trị ngưỡng phù hợp để đáp ứng chất lượng dịch vụ với người dùng mà không gây lãng phí tài nguyên sử dụng là một công việc hết sức khó khăn. Tuy gặp phải những vấn đề như vậy, nhưng trên thực tế, các mô hình tự động mở rộng tài nguyên dựa trên ngưỡng vẫn đang phổ biến hơn các hệ thống tự động mở rộng dựa trên việc dự đoán tài nguyên sử dụng. Lí do là vì hầu hết các nhà cung cấp dịch vụ điện toán đám mây vẫn còn đang nghi ngờ về độ chính xác của các mô hình dự đoán. Vì các mô hình dự đoán có độ chính xác thấp có thể ảnh hưởng đến chất lượng dịch vụ hoặc gây lãng phí tài nguyên. Hơn nữa, các mô hình dự đoán hiện nay hầu hết chỉ tập trung vào các thông số tài nguyên đơn lẻ trong khi để ra quyết định mở rộng, cần quan tâm tới tất cả các thông số tài nguyên của hệ thống, như: CPU, Memory, Disk io time, Disk space,...

Trong đề án này, em đề xuất một mô hình dự đoán tài nguyên sử dụng trong tương lai cho dịch vụ điện toán đám mây áp dụng cho mô hình hạ tầng được cung cấp như một dịch vụ (IaaS) dựa trên mô hình học máy. Mô hình của em sử dụng mạng LSTM - mô hình mạng đã đạt được những kết quả rất tốt trong những năm gần đây với dữ liệu dạng chuỗi thời gian để tiến hành phân tích chuỗi thời gian tài nguyên sử dụng. Bên cạnh đó, việc dự đoán chuỗi thời gian tài nguyên sử dụng trong môi trường đám mây yêu cầu phân tích dữ liệu nhiều chiều trên các thông số tài nguyên của hệ thống như CPU, memory, disk io time, disk local space,... nên chúng tôi sử dụng đầu vào đa chiều để dự đoán tài nguyên sử dụng trong tương lai. Chúng tôi

cũng sử dụng kĩ thuật mờ hóa dữ liệu để làm giảm đi độ biến động của dữ liệu. Trên một khía cạnh khác, việc sử dụng các thông số có độ tương quan thấp để đưa vào dữ liệu nhiều chiều đưa vào mô hình mạng cũng có thể làm bóp méo dữ liệu, khiến việc sử dụng dữ liệu nhiều chiều không đem lại kết quả như mong muốn. Vì vậy, chúng tôi đề xuất một thuật toán sử dụng độ đo tương quan phi tuyến tính nhằm xác định độ tương quan của các chiều dữ liệu để lựa chọn ra các cặp thông số có độ tương quan cao để đưa vào mô hình dự đoán. Mô hình được đề xuất sử dụng bộ dữ liệu ghi lại lịch sử sử dụng tài nguyên của các cụm máy chủ Google (Google cluster trace data) trong thực tế. Kết quả thực nghiệm cho thấy, mô hình được đề xuất có độ chính xác tốt hơn khi so sánh với các mô hình khác.

1.2 Bố cục đề án

Phần còn lại của báo cáo đề án tốt nghiệp này được tổ chức như sau:

Trong **Chương 2** sẽ giới thiệu về các khái niệm cơ bản trong điện toán đám mây và vấn đề cung cấp tài nguyên mà điện toán đám mây gặp phải dẫn đến nhu cầu thực tiễn của bài toán. Tiếp theo là những nghiên cứu đã được công bố nhằm giải quyết bài toán trên và các kiến thức về các phương pháp cơ bản và các phương pháp được áp dụng cho mô hình mà chúng tôi đề xuất.

Chương 3 mô tả hệ thống mà chúng tôi đề xuất bao gồm chi tiết các mô đun trong hệ thống và các giải thuật mà chúng tôi đề xuất.

Trong **Chương 4** thực hiện phần thử nghiệm mô hình hệ thống được đề xuất. Quá trình cài đặt thực nghiệm với các tham số được thực nghiệm, bộ dữ liệu Google cluster trace và các độ đo được sử dụng để đánh giá độ hiệu quả của hệ thống được trình bày. Sau đó, các kết quả thực nghiệm được đánh giá nhằm xác định độ hiệu quả của các phương pháp, các thuật toán đề xuất và của tổng thể hệ thống.

Chương 5 đưa ra kết luận những kết quả đạt được trong đề án và định hướng phát triển cho bài toán.

Chương 2

Tổng quan các nghiên cứu và các kiến thức cơ sở

2.1 Điện toán đám mây

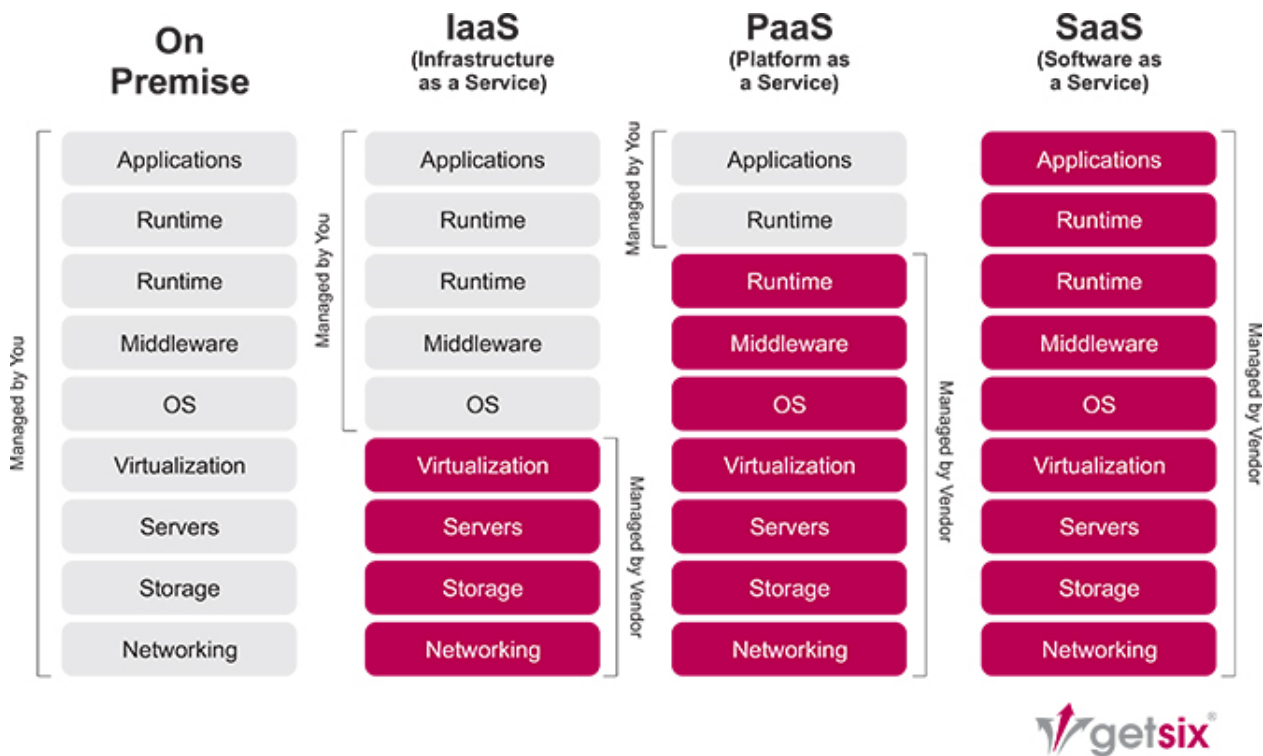
Thuật ngữ **Điện toán đám mây** ra đời từ những năm 2007 với mục đích khái quát lại các hướng đi của cơ sở hạ tầng thông tin vốn đã và đang diễn ra từ mấy năm qua. Trước đây để có thể triển khai một ứng dụng (ví dụ một trang Web), bạn phải đi mua/thuê một hay nhiều máy chủ (server), sau đó đặt máy chủ tại các trung tâm dữ liệu (data center) thì nay điện toán đám mây cho phép bạn giản lược quá trình mua/thuê đi. Bạn chỉ cần nêu ra yêu cầu của mình, hệ thống sẽ tự động gom nhặt các tài nguyên rồi (free) để đáp ứng yêu cầu của bạn. Điện toán đám mây có các lợi ích như cung cấp những gì cần thiết một cách tức thời các tài nguyên, tốc độ xử lý nhanh, cắt giảm chi phí, mở rộng linh hoạt, tiếp cận từ xa, bảo mật tích hợp, cài đặt và bảo trì tài nguyên, thay vào đó bạn chỉ cần xác định nhu cầu của mình sau đó các nhà cung cấp dịch vụ sẽ cung cấp.

Theo Viện tiêu chuẩn và công nghệ quốc gia Hoa Kỳ (NICS), Điện toán đám mây là một mô hình cho phép truy cập thông qua mạng ở mọi nơi, thuận tiện và dựa trên yêu cầu vào một nhóm tài nguyên tính toán có thể cấu hình được chia sẻ (ví dụ: Mạng, máy chủ, các ứng dụng, kho lưu trữ và các dịch vụ). Các tài nguyên này có thể được cung cấp và phát hành

nhANH chóng với sự quản lý và sự tương tác với các nhà cung cấp dịch vụ ít nhất có thể. Mô hình đám mây hiện nay bao gồm năm đặc điểm cơ bản, ba mô hình dịch vụ và bốn mô hình triển khai.

Năm đặc điểm cơ bản bao gồm:

- Tự phục vụ dựa trên yêu cầu (On-demand self-service): Một khách hàng có thể tự mình cung cấp khả năng tính toán, như thời gian máy chủ và lưu trữ mạng khi cần một cách tự động mà không cần tương tác với nhà cung cấp dịch vụ đám mây.
- Truy cập mạng diện rộng (Broad network access): Khả năng cung cấp dịch vụ luôn sẵn sàng trong toàn mạng và được truy cập thông qua các cơ chế chuẩn.
- Nhóm tài nguyên (Resource pooling): Các tài nguyên tính toán được lưu trữ để phục vụ nhiều khách hàng bằng mô hình multi-tenant với việc chỉ định và thu hồi các tài nguyên vật lý và tài nguyên ảo khác nhau một cách linh hoạt dựa vào yêu cầu khách hàng. Khách hàng cũng không cần biết vị trí chính xác của nơi tài nguyên được cung cấp cho mình được lưu trữ. Các tài nguyên được nhắc đến ở đây có thể bao gồm lưu trữ, xử lý, bộ nhớ và băng thông mạng.
- Khả năng cung cấp một cách mềm dẻo (Rapid elasticity). Tài nguyên tính toán có thể được cung cấp và phát hành một cách mềm dẻo, tự động trong một vài trường hợp, co dãn tài nguyên sử dụng một cách nhanh chóng tương ứng với yêu cầu. Đối với khách hàng, các tài nguyên tính toán luôn sẵn sàng cho việc cung cấp tài nguyên gần như không giới hạn và có thể được sử dụng bất kỳ lúc nào với khối lượng bất kỳ.
- Dịch vụ đo lường (Measured service): Các hệ thống đám mây điều khiển và tối ưu tài nguyên một cách tự động nhờ khả năng đo đếm ở một mức độ trừu tượng nào đó phù hợp với loại dịch vụ (lưu trữ, xử lý, băng thông và tài khoản người dùng đang hoạt động). Tài nguyên sử dụng được theo dõi, điều khiển và báo cáo, cung cấp sự minh bạch cho cả nhà cung cấp dịch vụ và khách hàng đối với dịch vụ được sử dụng.



Hình 2.1: Các mô hình dịch vụ trong Điện toán đám mây. Nguồn [11]

Các mô hình dịch vụ:

- Phần mềm được cung cấp như một dịch vụ (Software as a Service - SaaS): Đây là mức độ cao nhất đối với một mô hình dịch vụ điện toán đám mây. Khách hàng có khả năng sử dụng các ứng dụng mà nhà cung cấp đã phát triển trên nền tảng điện toán đám mây. Các ứng dụng này có thể được truy cập thông qua một giao diện khách nhỏ như trình duyệt web hoặc một giao diện chương trình. Các dịch vụ như Outlook, Gmail, ... là ví dụ điển hình cho SaaS. Khách hàng không quản lý hay điều khiển các hạ tầng đám mây cơ bản mà ứng dụng đang sử dụng.
- Nền tảng được cung cấp như một dịch vụ (Platform as a Service - PaaS). Đây là mô hình dịch vụ có mức độ thấp hơn SaaS. Theo đó, các nhà cung cấp dịch vụ đám mây cung cấp các hạ tầng đám mây cơ bản như mạng, máy chủ, hệ điều hành và lưu trữ. Đồng thời, người dùng dịch vụ điện toán đám mây cũng có thể triển khai các ứng dụng, giải pháp phần mềm sử dụng ngôn ngữ lập trình, các thư viện, dịch vụ hay các công cụ được hỗ trợ bởi nhà cung cấp. Người dùng có khả năng điều khiển toàn bộ ứng dụng được triển khai và có khả năng cài đặt cấu hình cho các môi trường lưu trữ ứng dụng.

- Hạ tầng được cung cấp như một dịch vụ (Infrastructure as a Service - IaaS): Đây là mô hình dịch vụ cơ bản nhất của điện toán đám mây. Các nhà cung cấp điện toán đám mây cung cấp tới khách hàng các dịch vụ về khả năng xử lý, lưu trữ, mạng hay các tài nguyên tính toán cơ bản khác mà khách hàng có thể triển khai và chạy các phần mềm tùy ý có thể bao gồm hệ điều hành hoặc các ứng dụng. Khách hàng không quản lý hay điều khiển các hạ tầng đám mây cơ bản nhưng quản lý toàn bộ hệ điều hành, lưu trữ và các ứng dụng đã được triển khai và có khả năng điều khiển lựa chọn các thành phần mạng một cách giới hạn. Khi sử dụng dịch vụ IaaS, người dùng cần phải cài đặt nhiều thứ bao gồm máy chủ, hệ điều hành, cơ sở dữ liệu, ... thì mới có thể triển khai ứng dụng của mình. Vì vậy, mô hình này thường hướng tới người dùng cuối là những người muốn có một môi trường để triển khai các ứng dụng của mình như lập trình viên, công ty, ...

Bốn mô hình triển khai:

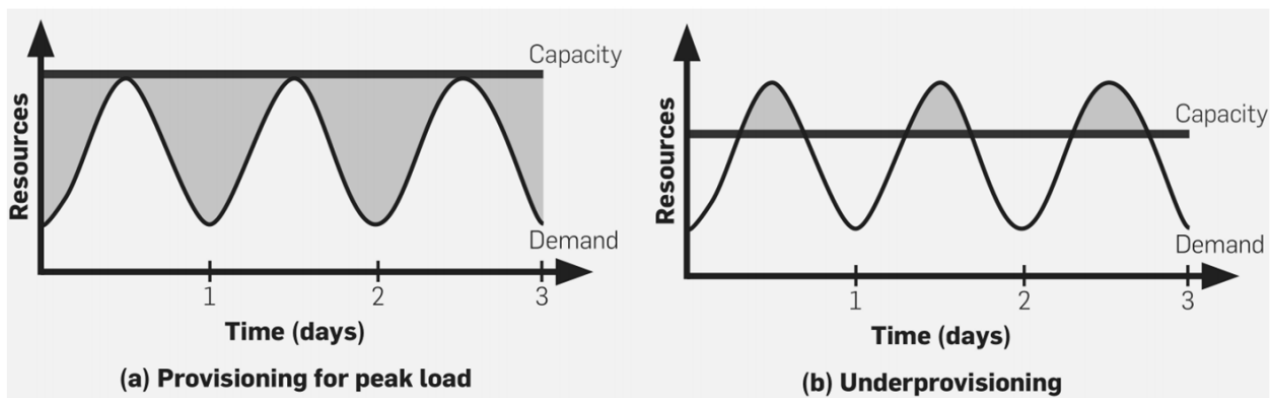
- Đám mây nội bộ (Private cloud): Các hạ tầng đám mây được cung cấp độc quyền cho một tổ chức duy nhất bao gồm nhiều người dùng. Nó có thể được sở hữu, quản lý và hoạt động bởi tổ chức, một bên thứ ba hoặc một vài đối tác của họ và có thể tồn tại bên trên hoặc bên ngoài cơ sở.
- Đám mây cộng đồng (Community cloud): Các hạ tầng đám mây được cung cấp độc quyền cho một cộng đồng khách hàng cụ thể từ các tổ chức có chung mối quan tâm (Ví dụ: Nhiệm vụ, yêu cầu bảo mật, chính sách và cân nhắc tuân thủ). Nó được sở hữu, quản lý và điều hành bởi một hoặc nhiều tổ chức trong cộng đồng, một bên thứ ba hoặc một số đối tác của họ và có thể tồn tại trong hoặc ngoài cơ sở.
- Đám mây công khai (Public cloud): Hạ tầng đám mây được cung cấp để sử dụng rộng rãi bởi công chúng. Nó có thể được sở hữu, quản lý và hoạt động bởi một tổ chức kinh doanh, tổ chức giáo dục, tổ chức chính phủ hoặc một vài đối tác của họ. Nó tồn tại trên cơ sở của các nhà cung cấp đám mây.

- Đám mây lai (Hybrid cloud): Hạ tầng đám mây là một sự kết hợp của hai hay nhiều hạ tầng đám mây riêng lẻ (public, private hoặc community) duy trì các thực thể duy nhất nhưng được gắn kết với nhau bởi công nghệ được chuẩn hóa hoặc độc quyền cho phép khả năng di chuyển dữ liệu và ứng dụng.

Các mô hình điện toán đám mây đem lại cho người dùng rất nhiều tiện ích mà nổi bật nhất là tiết kiệm chi phí. Theo đó, người dùng không cần tốn chi phí để đầu tư cho các thiết bị phần cứng, mạng, vận hành, bảo dưỡng,... để triển khai ứng dụng của mình. Bạn cũng có thể truy cập ứng dụng ở mọi nơi một cách thuận tiện thông qua mạng mà không cần quan tâm về các thiết bị phần cứng mình đang sử dụng. Hơn nữa, các ứng dụng điện toán đám mây có khả năng co giãn. Theo đó, hệ thống có thể mở rộng tài nguyên vào giờ cao điểm nhằm đảm bảo không bị vượt mất các khách hàng tiềm năng cũng như tiết kiệm tài nguyên vào những giờ thấp điểm. Một tiện ích nữa của điện toán đám mây là tính an toàn và liên tục. Các dữ liệu được người dùng đưa lên các hệ thống điện toán đám mây sẽ được các nhà cung cấp lưu trữ và bảo vệ. Các nhà cung cấp cũng có các cơ chế để phục hồi dữ liệu nhằm đảm bảo dữ liệu của bạn không bị mất dù xảy ra các vấn đề liên quan đến phần cứng,... Tuy nhiên, vấn đề mà các hệ thống điện toán đám mây gặp phải đó là bảo mật thông tin. Thông thường, các ứng dụng của bạn sẽ sử dụng chung tài nguyên với các ứng dụng khác (việc này giúp giảm chi phí), tuy nhiên lại khiến ứng dụng của bạn có thể bị truy cập bất hợp pháp từ một đối tượng nào đó. Điều này khiến các nhà cung cấp dịch vụ đám mây phải luôn đảm bảo các hệ thống bảo mật luôn được cập nhật và có độ an toàn cao.

2.2 Bài toán tự động mở rộng

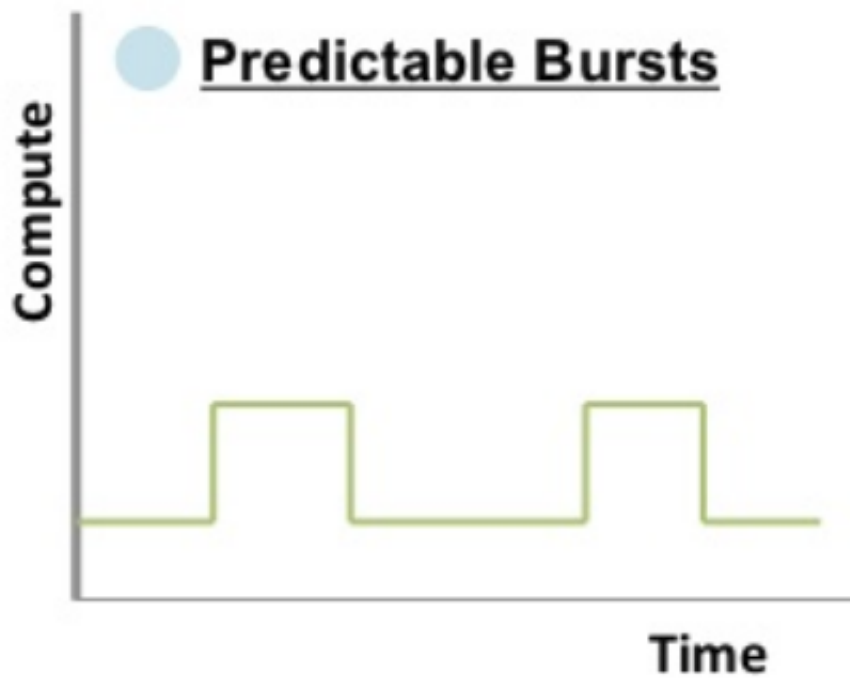
Với tính linh hoạt và mềm dẻo của các mô hình điện toán đám mây, người dùng các dịch vụ đám mây có thể tập trung vào việc cung cấp dịch vụ của họ tới khách hàng trong khi vẫn được đáp ứng một cách đầy đủ nhu cầu tính toán. Ngoài ra, mô hình Pay-as-you-go còn giúp tránh tình trạng cung cấp thừa, gây lãng phí tài nguyên trong những giờ thấp điểm cũng như cấp không đủ lượng tài nguyên trong những giờ cao điểm làm mất đi những khách hàng



Hình 2.2: Sự cung cấp tài nguyên theo yêu cầu. Nguồn [11]

tiềm năng. Hình 2.2 minh họa quá trình cung cấp tài nguyên một cách cố định tới người dùng các dịch vụ điện toán đám mây với sự biến thiên của người dùng hệ thống theo thời gian. Theo đó, hình bên trái thể hiện việc cung cấp tài nguyên quá hào phóng có thể dẫn đến tình trạng lãng phí tài nguyên (phần màu xám) mặc dù đảm bảo rằng tất cả người dùng sẽ được đáp ứng yêu cầu của họ. Trong khi với hình bên phải, việc cung cấp lượng tài nguyên thấp hơn sẽ giúp tiết kiệm tài nguyên hơn, tuy nhiên, phần yêu cầu màu xám là phần không đáp ứng đủ của hệ thống đối với yêu cầu người dùng. Vì vậy, giải pháp đặt ra là sẽ không cố định lượng tài nguyên cung cấp cho người dùng mà thay vào đó cung cấp một cách linh hoạt cho họ, nhằm đảm bảo việc tránh cung cấp thừa gây lãng phí tài nguyên và đảm bảo chất lượng dịch vụ (QoS) tới khách hàng.

Có hai cách tiếp cận cho việc mở rộng tài nguyên đối với các hệ thống điện toán đám mây. Đó là mở rộng tài nguyên theo chiều dọc và mở rộng tài nguyên theo chiều ngang. Đối với mở rộng tài nguyên theo chiều dọc, việc mở rộng tài nguyên sẽ tiếp cận theo hướng bổ sung tài nguyên vào máy ảo đã được cung cấp cho người dùng. Ví dụ khi người dùng đang được cung cấp một máy ảo có 4GB bộ nhớ và 2 lõi CPU. Khi người dùng muốn mở rộng tài nguyên, hệ thống sẽ bổ sung thêm 4GB bộ nhớ và thêm 2 lõi CPU vào máy ảo hiện tại của người dùng. Tuy nhiên, cách tiếp cận này ít phổ biến hơn vì việc tăng tài nguyên như thế sẽ gặp giới hạn nếu máy ảo cần lượng tài nguyên lớn hơn tài nguyên của máy chủ vật lý. Cách tiếp cận mở rộng theo chiều ngang được sử dụng phổ biến hơn. Với cách tiếp cận này, khi người dùng cần thêm tài nguyên, hệ thống sẽ bổ sung một máy ảo mới để phục vụ nhu cầu của người dùng.

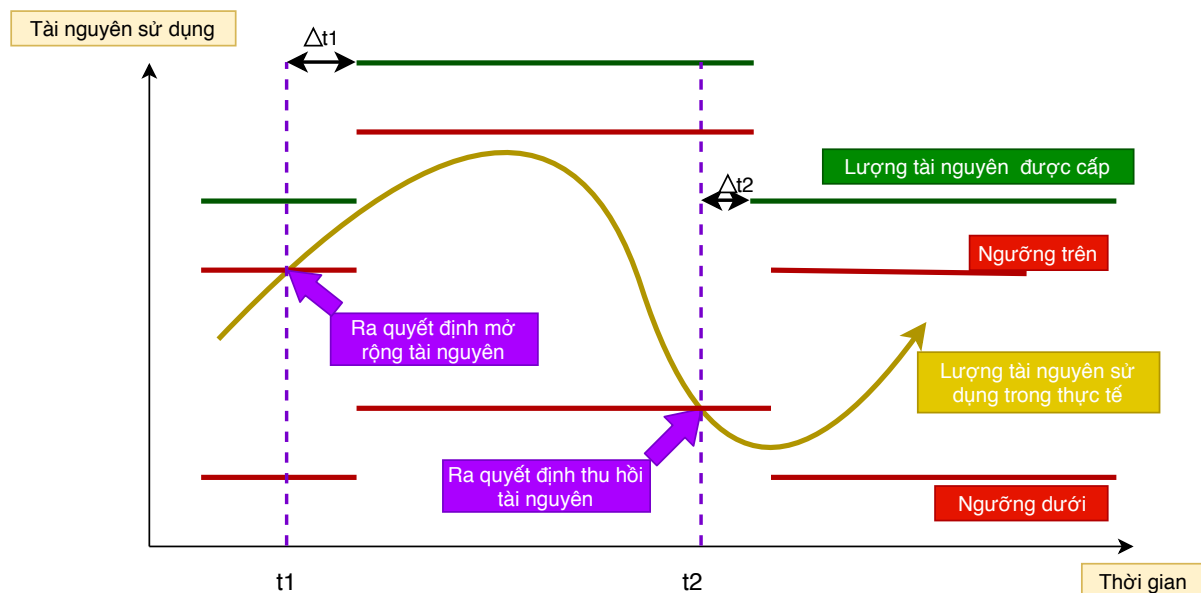


Hình 2.3: Phương pháp chu kỳ. Nguồn [4]

Đối với bài toán tự động mở rộng tài nguyên sử dụng đối với các dịch vụ đám mây, có 3 phương pháp chính bao gồm: Phương pháp chu kỳ (Periodicity), Phương pháp mở rộng dựa trên ngưỡng (Threshold) và phương pháp dự đoán trước tài nguyên sử dụng.

Trước tiên, ta xét tới phương pháp chu kỳ. Với phương pháp chu kỳ này, trong suốt quá trình sử dụng mô hình SaaS có chu kỳ tiêu dùng tài nguyên theo thời gian, minh họa ở hình 2.3 (ví dụ phút, giờ, ngày hay tháng). Sử dụng những đặc điểm trên, nhà cung cấp có thể căn cứ vào các thời điểm phù hợp để có thể ra quyết định tăng giảm cho tài nguyên cung cấp, cũng chính cách làm trên dẫn đến nhược điểm lớn nhất của phương pháp này là hệ thống không thể đáp ứng các yêu cầu cần tài nguyên ngay tức thì từ ứng dụng.

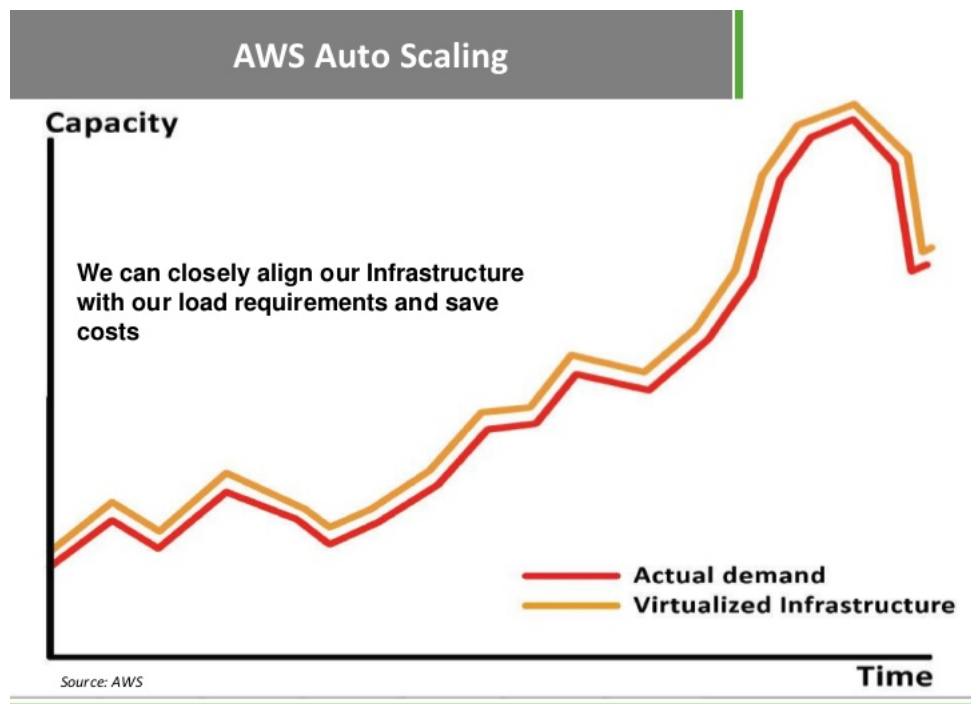
Đối với phương pháp sử dụng ngưỡng, người dùng sẽ định nghĩa một giá trị ngưỡng đối với các thông số sử dụng của hệ thống (CPU, Memory, ...) để quyết định xem khi nào hệ thống sẽ tăng hay giảm lượng tài nguyên cung cấp cho ứng dụng. Hình 2.4 minh họa việc sử dụng ngưỡng để mở rộng tài nguyên cho hệ thống. Tại thời điểm t , hệ thống được cung cấp một lượng tài nguyên sử dụng. Người dùng cũng sẽ định nghĩa các giá trị ngưỡng trên và các giá trị ngưỡng dưới. Khi lượng tài nguyên tiêu thụ của hệ thống chạm tới giá trị ngưỡng trên,



Hình 2.4: Phương pháp sử dụng ngưỡng.

hệ thống sẽ tự động mở rộng thêm máy ảo để đáp ứng nhu cầu của khách hàng và làm ngược lại nếu lượng tài nguyên tiêu thụ của hệ thống chạm tới giá trị ngưỡng dưới nhằm tiết kiệm tài nguyên của hệ thống. Tuy nhiên, cách tiếp cận này có một vài nhược điểm. Đầu tiên, đó là việc rất khó để xác định được giá trị ngưỡng phù hợp để đảm bảo việc tránh lãng phí tài nguyên cũng như đảm bảo chất lượng người dùng. Hơn nữa, việc ra quyết định mở rộng thêm máy ảo sẽ mất một khoảng thời gian (Δ) để tiến hành. Sau khoảng thời gian này, lượng tài nguyên tiêu thụ của hệ thống đã thay đổi, có thể đã vượt khả năng cung cấp của hệ thống (ảnh hưởng tới chất lượng dịch vụ) hoặc đã giảm xuống tới mức không cần mở rộng nữa (lãng phí tài nguyên).

Một cách tiếp cận khác cho bài toán tự động mở rộng đó là dự đoán trước lượng tài nguyên sử dụng của hệ thống trong tương lai. Kỹ thuật cốt lõi được sử dụng trong cách tiếp cận này là phân tích chuỗi thời gian. Theo đó, các giá trị chuỗi thời gian thường có các tính chất: Tính chu kỳ, tính xu hướng, tính ngẫu nhiên và tính có thể dự đoán được. Công việc của cách tiếp cận này là thu thập lượng tài nguyên sử dụng trong quá khứ, sau đó xây dựng các mô hình nhằm học được các tính chất của chuỗi thời gian đó và mô hình hóa chuỗi thời gian nhằm đưa ra dự đoán về giá trị tài nguyên sử dụng trong tương lai. Hình 2.5 mô tả tài nguyên sử dụng trong thực tế và tài nguyên sử dụng được dự đoán. Từ đó, việc mở rộng hệ thống sẽ được quyết định dựa vào lượng tài nguyên dự đoán. Hiện nay cũng đã có rất nhiều các nghiên



Hình 2.5: Phương pháp dự đoán trước lượng tài nguyên sử dụng. Nguồn [11]

cứu được công bố nhằm giải quyết bài toán này. Tuy nhiên, cách tiếp cận này vẫn chưa được áp dụng phổ biến ở thời điểm hiện tại. Lí do là các nhà cung cấp dịch vụ đám mây vẫn còn nghi ngờ về độ chính xác của các mô hình dự đoán. Ngoài ra, các nghiên cứu này mới chỉ đánh giá các độ đo đơn lẻ trong khi việc ra quyết định mở rộng hệ thống lại phụ thuộc vào nhiều độ đo khác nhau về tài nguyên sử dụng của hệ thống.

2.3 Các phương pháp dự đoán chuỗi thời gian

2.3.1 Tổng quan

Mục đích chính của **kỹ thuật phân phối tài nguyên chủ động** là dự đoán một cách chính xác lượng tài nguyên sử dụng. Đã có rất nhiều nghiên cứu về các mô hình dự đoán trong điện toán đám mây được công bố trong những năm qua. Trong [11], các phương pháp bao gồm phương pháp tự hồi quy (AR), phương pháp trung bình trượt (MA), lỗi theo xu hướng và chu kỳ (ETS), tự hồi quy kết hợp trung bình trượt (ARIMA) và mô hình mạng nơ-ron lan

truyền ngược (BPNN) được đánh giá và so sánh độ chính xác trong việc dự đoán tải của hệ thống đám mây. Trong [8], Tran và các cộng sự đã sử dụng thuật toán PD- GABP để kết hợp phát hiện chu kỳ dữ liệu và mạng nơ-ron để cải thiện độ chính xác dự đoán. Gần đây, các mô hình học sâu xuất hiện như một giải pháp hiệu quả cho các vấn đề dự đoán. Trong số đó, mạng nơ-ron hồi quy (RNN) được cho là hữu ích đối với việc phân tích dữ liệu chuỗi thời gian vì có sự liên quan giữa các điểm dữ liệu trong loại dữ liệu này và RNNs được thiết kế để mô hình phụ thuộc thời gian và đào tạo bằng cách lan truyền ngược thông qua thời gian. Mạng RNN khác biệt với mạng nơ-ron truyền thẳng ở chính sự hoạt động của nó khi mà luồng thông tin được truyền ngược lại thông qua các vòng lặp phản hồi. Để cải thiện khả năng ghi nhớ của RNN, Hochreiter và các cộng sự trong [5] đã đề xuất mạng bộ nhớ ngắn hạn dài (Long short term memory network - LSTM), là mô hình phổ biến nhất trong các tầng mạng RNN. Với sự cải thiện này, RNN có khả năng nhận diện và học các phụ thuộc trong khoảng thời gian dài. Trong các nghiên cứu áp dụng vào trong hệ thống đám mây, Jitendra và các cộng sự [6] đã đề xuất mô hình LSTM-RNN cho việc dự đoán tải trên các trung tâm dữ liệu đám mây sử dụng bộ dữ liệu NASA. Trong [9], các tác giả đã xây dựng một hệ thống đám mây tự động mở rộng khai thác các tài nguyên đa biến, giải thuật di truyền, giải thuật lan truyền ngược và mạng nơ-ron lan truyền ngược để dự đoán tải sử dụng. Mặc dù các nỗ lực học sâu đã được đề xuất để sử dụng cho các mô hình đám mây nhưng những mô hình này mới chỉ tập trung vào các chuỗi thời gian một chiều.

Như đã đề cập ở phần trên về **vấn đề phân tích dữ liệu đa chiều**, kết quả dự đoán sẽ tốt hơn nếu sự tương quan giữa các thông số cũng được xem xét [1]. Mặc dù trong lý thuyết, bộ dữ liệu đa chiều có thể cho độ chính xác cao hơn, nhưng trong thực tế, các thuộc tính không có sự tương quan tốt có thể dẫn đến sự bóp méo dữ liệu. Một vài giải pháp đã được đề xuất để xác định sự tương quan giữa các chiều dữ liệu. Sabine trong [10] đã xây dựng một mô hình nhóm các biến thông qua đặc trưng tương quan sử dụng phương pháp deverse. Các tác giả của [12] đã tạo ra một mô hình lựa chọn đặc trưng hiệu quả thông qua việc phân tích sự tương quan và không tương quan sử dụng phương pháp xác định tương quan phi tuyến. Vergara trong [?] đã phát triển một giải pháp thống nhất để lựa chọn các đặc trưng lý thuyết với các thuật toán cho sự lựa chọn chuyển tiếp tuần tự và loại bỏ lùi tuần tự. Tuy nhiên, không

cơ chế nào cho phép hệ thống dự đoán xác định xấp xỉ số lượng thông số để đưa chúng vào mô hình dự đoán trong cùng một thời điểm.

Lý thuyết về chuỗi thời gian mờ đã được đưa ra lần đầu tiên trong [7] chỉ giải quyết mô hình một chiều. Để giải quyết vấn đề đa chiều, Erol đã đề xuất một mạng nơ ron có thể xác định mối quan hệ mờ giữa hai hệ số có các chuỗi thời gian mờ cao [2]. Trần trong [9] đã giới thiệu các chuỗi thời gian mờ cao k hệ số sử dụng giải thuật di truyền kết hợp mạng nơ ron nhân tạo để khám phá mối quan hệ mờ ẩn cũng như sử dụng kỹ thuật làm mờ để giảm thiểu sự biến động trong dữ liệu thực tế. Trong quá trình học, chúng tôi cũng đề xuất một mô hình các chuỗi thời gian mờ với k hệ số và vấn đề nhiễu của dữ liệu được thu thập từ các tài nguyên đám mây. Tuy nhiên, trong quá trình tiền xử lý dữ liệu, quá trình học, chúng tôi đánh giá và lựa chọn các thông số có độ tương quan cao và sử dụng mạng LSTM để cải thiện độ chính xác dự đoán. Dưới đây tôi sẽ trình bày các phương pháp cơ bản cũng như các phương pháp được sử dụng trong đề án này.

2.3.2 Các phương pháp tuyến tính

Giả sử chuỗi thời gian có dạng $\{X = X_1, X_2, \dots, X_n\}$. Giả sử X_t là giá trị tại thời điểm t của chuỗi thời gian X . Các mô hình tuyến tính để dự đoán giá trị X_t được trình bày lần lượt dưới đây.

2.3.2.1 Mô hình tự hồi quy (Auto Regressive)

Mô hình tự hồi quy là một mô hình tuyến tính xây dựng giá trị tại thời điểm t dựa vào các giá trị trong quá khứ của chuỗi thời gian. Theo đó, mô hình tự hồi quy bậc nhất được trình bày theo công thức sau:

$$X_t = c + \theta X_{t-1} + \epsilon_t, -1 < \theta < 1 \quad (2.1)$$

$$\text{Trong đó, } \begin{cases} c \text{ là hằng số} \\ \theta \text{ là tham số mô hình} \\ \epsilon_t \text{ là nhiễu trắng - là yếu tố ngẫu nhiên thuần túy.} \end{cases}$$

Giá trị của X tại thời điểm t phụ thuộc vào giá trị của nó tại thời điểm trước đó và một yếu tố ngẫu nhiên. Xét một cách tổng quát, mô hình tự hồi quy bậc p thỏa mãn công thức sau:

$$X_t = c + \theta_{t-1}X_{t-1} + \theta_{t-2}X_{t-2} + \dots + \theta_{t-p}X_{t-p} + \epsilon_t, -1 < \theta_{t-1}, \theta_{t-2}, \dots, \theta_{t-p} < 1 \quad (2.2)$$

2.3.2.2 Mô hình trung bình trượt (Moving average)

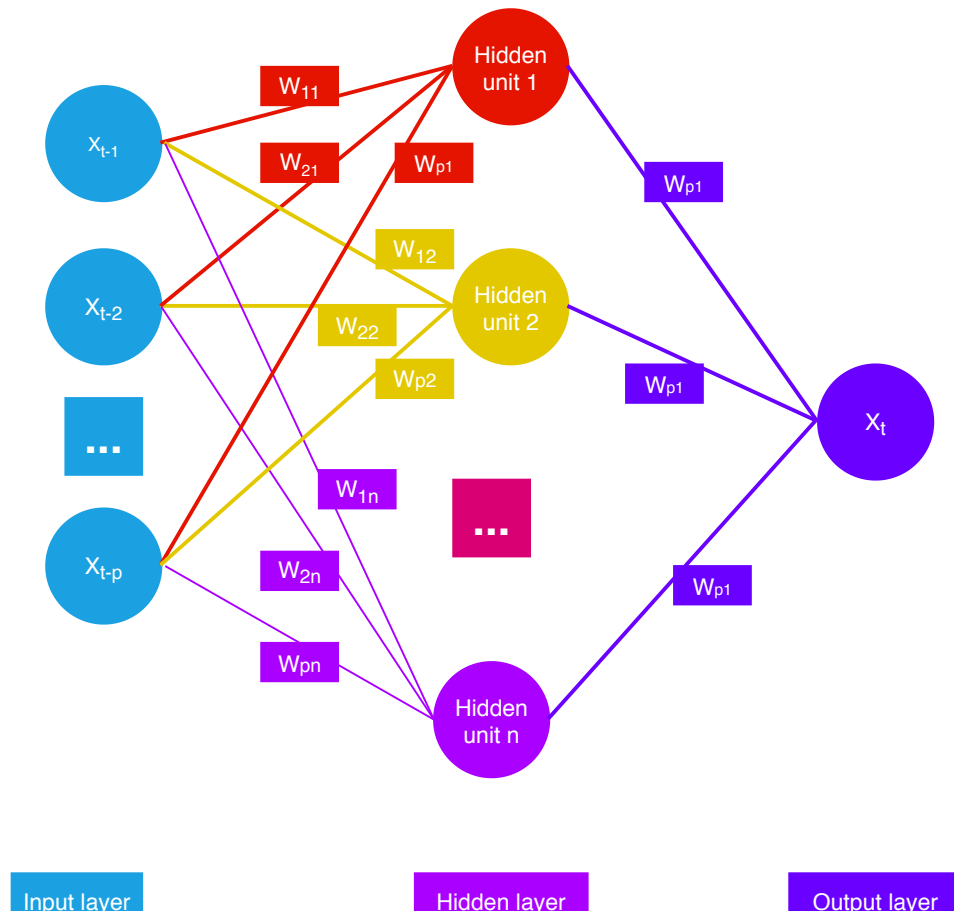
Mô hình trung bình trượt là một mô hình tuyến tính xây dựng giá trị tại thời điểm t của chuỗi thời gian dựa vào giá trị trung bình của chuỗi và yếu tố ngẫu nhiên thuần túy tại các thời điểm trước đó. Mô hình trung bình trượt bậc nhất được trình bày như sau:

$$X_t = \mu + \theta\epsilon_{t-1} + \epsilon_t, -1 < \theta < 1 \quad (2.3)$$

$$\text{Trong đó, } \begin{cases} \mu \text{ là giá trị trung bình của chuỗi thời gian} \\ \theta \text{ là tham số mô hình} \\ \epsilon_t \text{ là nhiễu trắng - là yếu tố ngẫu nhiên thuần túy.} \end{cases}$$

Xét một cách tổng quát, ta có mô hình trung bình trượt bậc q :

$$X_t = \mu + \theta_{t-1}\epsilon_{t-1} + \theta_{t-2}\epsilon_{t-2} + \dots + \theta_{t-q}\epsilon_{t-q} + \epsilon_t, -1 < \theta_{t-1}, \theta_{t-2}, \dots, \theta_{t-q} < 1 \quad (2.4)$$



Hình 2.6: Mô hình mạng nơ-ron truyền thẳng với một tầng ẩn

2.3.2.3 Mô hình làm mịn theo hàm số mũ

$$X_t = \alpha X_{t-1} + (1 - \alpha) \widehat{X}_{t-1} \quad (2.5)$$

Trong đó, α là hằng số làm mịn

Giá trị dự đoán tại thời điểm t là phép cộng giá trị của chuỗi thời gian tại thời điểm $t-1$ và giá trị dự đoán tại thời điểm $t-1$. Phương pháp này thường được sử dụng cho các dự đoán ngắn hạn. Giá trị α càng lớn thì mức độ làm mịn sẽ càng ít đi vì khi đó giá trị dự đoán sẽ phụ thuộc nhiều hơn vào giá trị của chuỗi thời gian trong quá khứ.

2.3.3 Mô hình mạng nơ-ron nhân tạo

Mô hình mạng nơ-ron nhân tạo là mô hình được sử dụng phổ biến trong những năm gần đây bởi đặc tính: Thích nghi nhanh, phi tuyến và khả năng xấp xỉ hàm tốt. Mô hình mạng nơ-ron nhân tạo là một mô hình toán học hay các mô hình tính toán được xây dựng dựa trên các mô hình mạng nơ-ron sinh học. Mạng nơ-ron truyền thẳng hay mạng nơ-ron nhiều lớp (Multilayer perceptron) có là mô hình mạng có sức ảnh hưởng lớn nhất. Mô hình này đã được áp dụng khá rộng rãi và đạt được nhiều kết quả khả quan trong thực tiễn để mô hình hóa các dạng dữ liệu thống kê phi tuyến tính. Mô hình mạng nơ-ron truyền thẳng cũng là nền tảng của rất nhiều mô hình mạng phức tạp như: Mạng nơ-ron hồi quy (Recurrent neural networks), mạng nơ-ron tích chập (Convolution neural network),... Mô hình mạng nơ-ron truyền thẳng được minh họa trong hình 2.6. Theo đó, mạng này bao gồm một tầng đầu vào (Input layer), một hoặc nhiều tầng ẩn (Hidden layers) và một tầng đầu ra (Output layer). Mỗi tầng sẽ bao gồm một hoặc nhiều đơn vị nơ-ron. Các nơ-ron ở tầng trước sẽ được kết nối đầy đủ (fully connected) với các nơ-ron ở tầng tiếp theo. Mỗi liên kết giữa hai nút trong mạng mang một giá trị trọng số liên kết w . Mỗi nút sẽ nhận thông tin từ các nút liên kết với nó ở tầng trước đó, xử lý thông tin một cách độc lập dựa vào giá trị trọng số liên kết và sử dụng một hàm kích hoạt biến đổi thông tin đó để tạo thông tin đầu vào cho các nút tầng tiếp theo. Áp dụng cho bài toán dự đoán chuỗi thời gian, đầu ra sẽ là giá trị của chuỗi thời gian tại thời điểm t và đầu vào là giá trị của chuỗi thời gian tại những điểm trước đó. Mục tiêu của mô hình mạng nơ-ron truyền thẳng là xấp xỉ quan hệ sau:

$$X_t = f(X_{t-1}, X_{t-2}, \dots, X_{t-p}) \quad (2.6)$$

Quá trình huấn luyện mạng nơ-ron truyền thẳng về cơ bản là đi tìm bộ trọng số phù hợp để tạo ra một mô hình mang đầy đủ thông tin từ bộ dữ liệu huấn luyện. Mô hình này cũng cần mang tính tổng quát để có thể được áp dụng với dữ liệu mới trong thực tế. Đối với bài toán dự đoán chuỗi thời gian, đây là bài toán hồi quy trong phương pháp học có giám sát, ở đó mỗi một giá trị đầu vào sẽ có một giá trị đầu ra tương ứng. Để huấn luyện mô hình mạng nơ-ron, trước hết chúng ta sẽ cần một bộ dữ liệu phục vụ cho bài toán của mình, ở đây sẽ là một chuỗi thời gian.

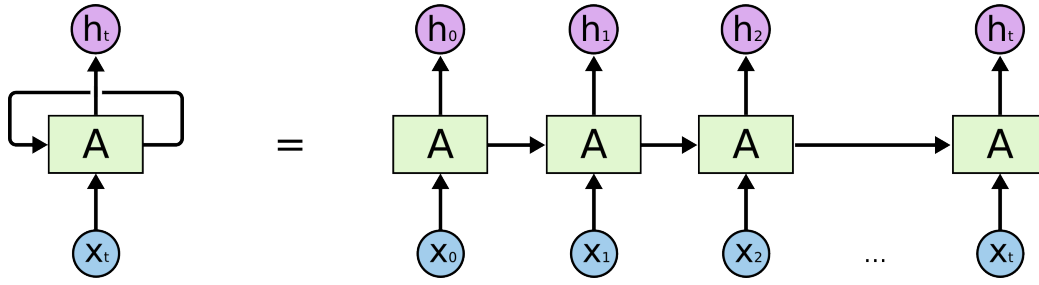
Toàn bộ tập dữ liệu này sẽ được chia thành 2 phần: Tập huấn luyện và tập kiểm thử. Thông thường, tập huấn luyện sẽ lớn hơn tập kiểm thử và được sử dụng để ước lượng bộ trọng số của mô hình mạng (w) và các tham số tùy chỉnh khác nhau (số tầng ẩn, số nút, hàm kích hoạt, tốc độ học,...). Tập kiểm thử được sử dụng để đánh giá độ chính xác mô hình được huấn luyện. Thuật toán được sử dụng nhiều nhất trong quá trình huấn luyện mạng nơ-ron truyền thẳng là thuật toán lan truyền ngược (Back propagation) sử dụng kĩ thuật giảm đạo hàm (Gradient descent) để tiến tới giá trị tối ưu địa phương. Giải thuật ?? mô tả quá trình huấn luyện mạng nơ-ron truyền thẳng sử dụng giải thuật lan truyền ngược.

Algorithm 1: Giải thuật lan truyền ngược áp dụng cho mạng nơ-ron truyền thẳng một tầng ẩn

```

1 Khởi tạo ngẫu nhiên một bộ tham số đầu vào  $w_p$ 
2 repeat
3   //Dựa vào giá trị đầu vào  $x$ , thông qua tầng ẩn tính toán giá trị đầu ra của mô hình
4   for  $j = 1$  to  $h$  do
5      $H_j = \phi(\sum_i^n x_i * w_{ij}^{[1]} + b_{ij}^{[1]})$ 
6   end
7    $\hat{y}_j = \phi(\sum_i^h H_i * w_j^{[2]} + b_j^{[2]})$ 
8   //Xác định sai số
9    $L(w) = loss(\hat{y}_j, y_j)$ 
10  // Truyền ngược sai số tới tất cả các tham số
11   $\Delta(w_{ij}^2) = \frac{\partial(L(w))}{\partial(w_{ij}^2)}$ 
12   $\Delta(w_{ij}^1) = \frac{\partial(L(w))}{\partial(w_{ij}^1)}$ 
13  //Cập nhật tham số
14   $w_{ij}^2 = w_{ij}^2 - \eta * \Delta(w_{ij}^2)$ 
15   $w_{ij}^1 = w_{ij}^1 - \eta * \Delta(w_{ij}^1)$ 
16 until Khi số vòng lặp kết thúc hoặc giá trị lỗi đã hội tụ;
```

Trong đó, $\left\{ \begin{array}{l} n, h : \text{Số nút tầng đầu vào và tầng ẩn} \\ w^{[l]} : \text{trọng số tầng } l \\ b^{[l]} : \text{Trọng số bổ sung (bias) tầng } l \\ \phi : \text{Hàm kích hoạt (activation)} \\ loss : \text{Hàm lỗi. Có thể là MSE, RMSE, ...} \end{array} \right.$



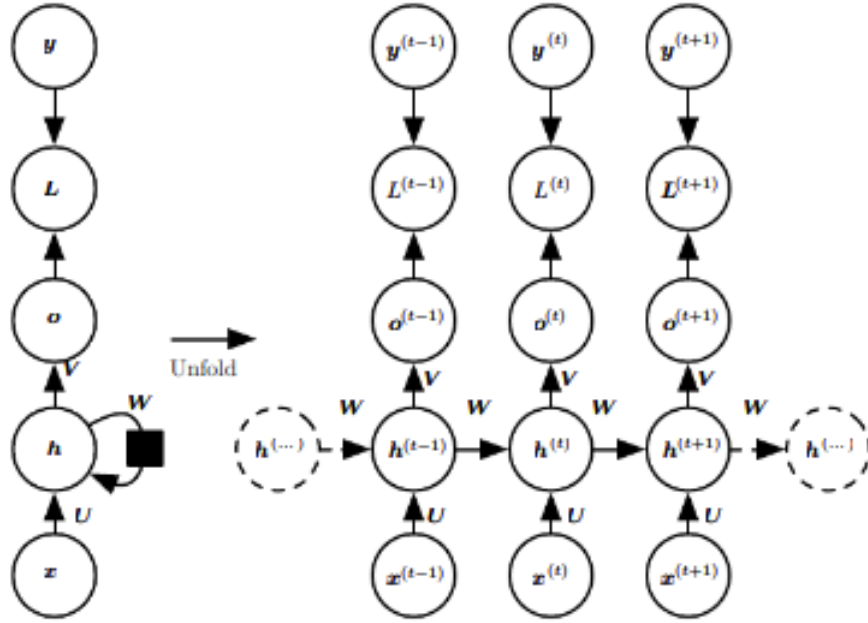
Hình 2.7: Mạng nơ-ron hồi quy, Nguồn [3]

2.3.4 Phương pháp học sâu

2.3.4.1 Phương pháp mạng nơ-ron hồi quy

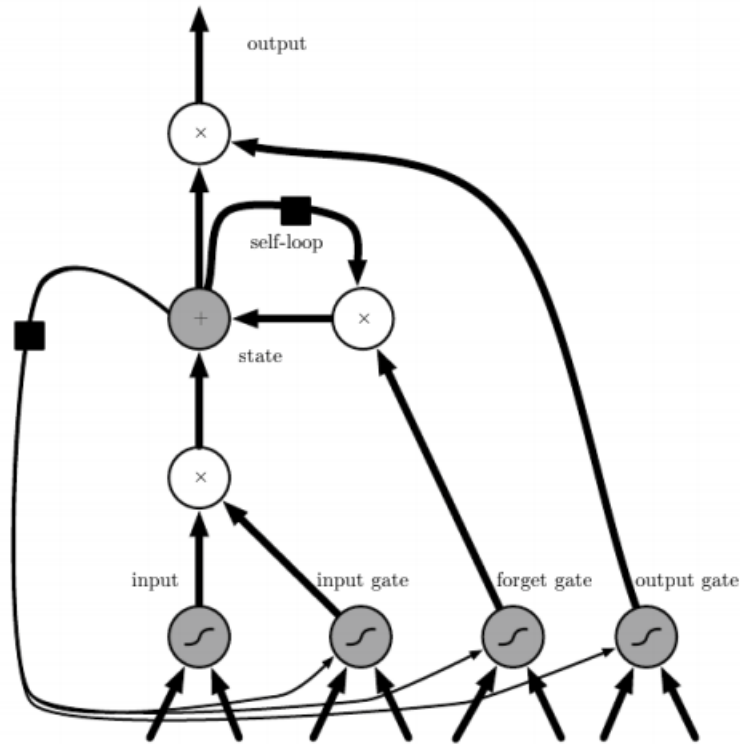
Mặc dù mạng nơ-ron nhân tạo đã đạt được những kết quả khả quan, nhưng mô hình mạng này dựa vào các giải thuật tối ưu địa phương, dễ dàng rơi vào giá trị cực tiểu cục bộ và bị tác động lớn bởi bộ trọng số được thiết lập ban đầu. Gần đây, các phương pháp học sâu (deep learning) đã đạt được những kết quả ấn tượng trong nhiều lĩnh vực khác nhau như xử lý ảnh, bài toán chuỗi thời gian, Trong số đó, mạng nơ-ron hồi quy (Recurrent Neural Networks - RNNs) là một loại của mô hình mạng nơ-ron áp dụng cho các bài toán liên quan đến xử lý dữ liệu dạng tuần tự: $x_{(1)}, \dots, x_{(t)}$, dạng dữ liệu mà giá trị tại điểm t có sự phụ thuộc vào giá trị tại các điểm trước đó: $x_{(t)} = f(x_{(t-1)}, x_{(t-2)}, \dots, x_{(1)})$. Dữ liệu tuần tự có thể là dữ liệu chuỗi thời gian (giá chứng khoán theo từng ngày, nhiệt độ, độ ẩm theo thời gian,...), dữ liệu ngôn ngữ tự nhiên,... Mô hình mạng nơ-ron hồi quy ghi nhớ lại thông tin của chuỗi dựa vào các điểm trước đó và kết hợp những thông tin này với thông tin dữ liệu đầu vào tại điểm t để dự đoán đầu ra tại thời điểm $t+1$ trong khi mạng nơ-ron thông thường bỏ qua các thông tin của dữ liệu trong quá khứ. Điểm khác biệt của mô hình RNN với các mô hình khác là module bộ nhớ. Toàn bộ thông tin của chuỗi thời gian được lưu tại trạng thái ẩn h . Trạng thái ẩn này sẽ được tính toán và bổ sung thông tin khi mỗi giá trị mới của chuỗi thời gian được đưa vào mô hình. Trạng thái ẩn được tính toán theo công thức:

$$h_{(t)} = \phi(W h_{(t-1)} + U x_{(t)}) \quad (2.7)$$



Hình 2.8: Đồ thị tính toán để tính ra lỗi huấn luyện của mạng hồi quy, Nguồn [4]

Trong đó, $h_{(t)}, h_{(t-1)}$ là trạng thái ẩn tại thời điểm t và $t-1$. W là ma trận chuyển trạng thái, U là ma trận trọng số đầu vào, $x_{(t)}$ là giá trị đầu vào tại thời điểm t . ϕ là hàm kích hoạt. Quá trình truyền thẳng của mô hình mạng RNN được minh họa trong hình 2.8. Đầu vào của mạng là một chuỗi tuần tự các giá trị x và đầu ra là chuỗi outputs o tương ứng. Độ đo lỗi L được sử dụng để xác định sai số giữa giá trị dự đoán o và giá trị mục tiêu y . U là ma trận trọng số đầu vào, W là ma trận chuyển trạng thái. V là ma trận trọng số chuyển trạng thái hidden-outputs. Việc huấn luyện mạng RNN cũng tương tự như các mô hình mạng nơ-ron truyền thống, nhưng giải thuật lan truyền ngược (Backpropagation) được thay đổi một chút. Đó là đạo hàm hàm lỗi tại thời điểm t cũng sẽ phụ thuộc vào các thời điểm trước đó nữa. Để tính đạo hàm tại thời điểm t , ta phải lan truyền ngược đạo hàm về các bước phía trước rồi cộng tổng đạo hàm của chúng lại với nhau. Việc tính đạo hàm kiểu này được gọi là lan truyền ngược liên hồi (BPTT - Backpropagation Through Time). Việc huấn luyện mạng RNN dừng lại khi kết thúc số vòng lặp hoặc hàm lỗi hội tụ.



Hình 2.9: Mô hình mạng LSTM. Nguồn [4]

2.3.4.2 Mạng Long Short Term Memory Neural Network

Mặc dù có thể nhận thấy hiệu quả của mô hình mạng nơ-ron hồi quy khi kết hợp các thông tin ẩn của chuỗi thời gian trong quá khứ nhưng mô hình RNNs cũng gặp phải rất nhiều vấn đề gây ảnh hưởng lớn tới hiệu năng sử dụng mạng. Đó là việc đạo hàm được lan truyền ngược ở quá nhiều tầng, dẫn đến các vấn đề triệt tiêu đạo hàm (vanish gradients) hoặc bùng nổ đạo hàm (explode gradients). Triệt tiêu đạo hàm là khi giá trị đạo hàm không đủ lớn để thay đổi trọng số dẫn đến việc huấn luyện trở nên vô ích hoặc khiến hàm kém hiệu quả. Trong một số trường hợp khác, bùng nổ đạo hàm là khi giá trị đạo hàm quá lớn dẫn đến việc các trọng số phải cập nhật một giá trị quá lớn khiến việc chúng bị phân rã và mô hình trở nên kém ổn định. LSTM được đề xuất bởi Hochreiter và Schmidhuber đã giải quyết được các vấn đề này. LSTM có khả năng loại bỏ hoặc bổ sung các thông tin cần thiết. Những thông tin này được điều chỉnh bởi các nhóm được gọi là cổng. Về mặt cấu trúc, LSTM có một đơn vị bộ nhớ và ba cổng: Cổng đầu vào (Input Gate), cổng quên (Forget Gate) và cổng đầu ra (Output Gate).

Cổng đầu vào quyết định có bao nhiêu thông tin mới sẽ được cập nhật trong bộ nhớ, cổng quên cho phép bao nhiêu thông tin nên được ghi nhớ lại hay là quên đi trong tế bào và cổng đầu ra quyết định số lượng trạng thái tế bào ảnh hưởng tới đầu ra. Mô hình LSTM được định nghĩa bởi công thức:

$$f_t^{(t)} = \sigma(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)}) \quad (2.8)$$

$$g_t^{(t)} = \sigma(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)}) \quad (2.9)$$

$$o_t^{(t)} = \sigma(b_i^o + \sum_j U_{i,j}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{(t-1)}) \quad (2.10)$$

$$s_t^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_t^{(t)} \tanh(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} h_j^{(t-1)}) \quad (2.11)$$

$$s_t^{(t)} = \tanh(s_i^{(t)} q_i^{(t)}) \quad (2.12)$$

Trong đó, $x_j(t), h_t$ biểu diễn vector đầu vào hiện tại và trạng thái ẩn. W's, U's, b's là ma trận trọng số đầu vào, ma trận trọng số hồi quy và giá trị bổ sung tương ứng (bias). Các cổng đầu vào, cổng quên và cổng đầu ra được kí hiệu là g_t, f_t, o_t . \tanh, σ là các hàm kích hoạt tanh và sigmoid.

2.3.5 Phương pháp chuỗi thời gian mờ

Chuỗi thời gian mờ được giới thiệu lần đầu tiên bởi Song và Chissom dựa trên khái niệm tập mờ được giới thiệu bởi Zadeh năm 1965. Trái ngược với các phương pháp phân tích chuỗi thời gian thông thường, các giả định lý thuyết khác nhau không cần phải được kiểm tra theo các cách tiếp cận chuỗi thời gian mờ. Lợi ích lớn nhất của cách tiếp cận chuỗi thời gian mờ là có thể làm việc với tập dữ liệu nhỏ hơn.

Chuỗi thời gian mờ được định nghĩa như sau: Cho U là tập vũ trụ, $U = u_1, u_2, \dots, u_b$. Tập mờ A của U được định nghĩa là $A_i = f_{A_1}(u_1)/u_1 + f_{A_2}(u_2)/u_2 + \dots + f_{A_b}(u_b)/u_b$. Trong đó, f_{A_i} là hàm phụ thuộc của tập mờ A_i , $f_{A_i} : U \rightarrow [0, 1]$. u_a là giá trị chung của tập mờ A_i . $f_{A_i}(u_a)$ là mức độ mờ của u_a với tập mờ A_i . $f_{A_i}(u_a) \in [0, 1]$ và $1 \leq a \leq b$.

Định nghĩa 1: Giả thiết rằng tồn tại mối quan hệ mờ $F(t)$ và $F(t-1)$. Khi đó, mối quan hệ này có thể được biểu diễn: $F(t) = F(t-1) * R(t, t-1)$, với $*$ là toán tử mờ. Giả sử rằng, $F(t-1) = A_i, F(t) = A_j$. Mối quan hệ mờ giữa $F(t-1)$ và $F(t)$ có thể được ghi chú bằng $A_i \rightarrow A_j$ trong đó, A_i ám chỉ bên trái và A_j ám chỉ bên phải của quan hệ mờ. Các mối quan hệ mờ này cũng có thể được nhóm lại để thiết lập các quan hệ mờ khác nhau.

Định nghĩa 2: Cho $F(t)$ là một chuỗi thời gian mờ. Nếu $F(t)$ phụ thuộc bởi $F(t-1), F(t-2), \dots, F(t-n)$ thì mối quan hệ mờ có thể được biểu diễn thành:

$$F(t-n), \dots, F(t-2), F(t-1)F(t) \quad (2.13)$$

Đây được gọi là mô hình dự đoán chuỗi thời gian mờ bậc n .

Định nghĩa 3: Cho F và G là hai chuỗi thời gian mờ. Giả sử rằng, $F(t-1) = A_i, G(t-1) = B_k$ và $F(t) = A_j$. Một mối quan hệ mờ hai chiều được định nghĩa: $A_i, B_k \rightarrow A_j$ trong đó A_i, B_k ám chỉ về trái và A_j ám chỉ về phải của phép suy diễn. Khi đó, mô hình dự đoán chuỗi thời gian mờ bậc 1:

$$F(t-1), G(t-1) \rightarrow F(t) \quad (2.14)$$

Định nghĩa 4: Cho F và G là hai chuỗi thời gian mờ. Nếu $F(t)$ phụ thuộc bởi $(F(t-1), G(t-1)), (F(t-2), G(t-2)), \dots, (F(t-n), G(t-n))$ thì mối quan hệ mờ này sẽ được biểu diễn như sau:

$$(F(t-1), G(t-1)), (F(t-2), G(t-2)), \dots, (F(t-n), G(t-n)) \rightarrow F(t) \quad (2.15)$$

Đây là mô hình dự đoán chuỗi thời gian mờ bậc n hai chiều, $F(t)$ là yếu tố chính và $G(t)$ là yếu tố phụ tới t tương ứng $= (\dots, 0, 1, 2, \dots)$

Định nghĩa 5: Cho F và G_1, G_2, \dots, G_{k-1} là k chuỗi thời gian mờ. Nếu $F(t)$ phụ thuộc bởi $(F(t-1), G_1(t-n), \dots, G_{k-1}(t-n)), \dots, (F(t-n), G_1(t-n), \dots, G_{k-1}(t-n))$ thì mối quan hệ mờ

này sẽ được biểu diễn như sau:

$$(F_{(t-1)}, G_1(t-1), \dots, G_{k-1}(t-1)), \dots, (F_{(t-n)}, G_1(t-n), \dots, G_{k-1}(t-n)) \rightarrow F(t) \quad (2.16)$$

Đây được gọi là mô hình dự đoán chuỗi thời gian mờ k chiều bậc n. Trong đó, $F_{(t)}$ là chuỗi thời gian mờ chính, $G_i(t)$ là chuỗi thời gian mờ phụ, với $(t = \dots, 0, 1, 2, \dots, i = 1, 2, \dots, k-1)$.

2.4 Phương pháp phân tích mối tương quan nhiều chiều

Phương pháp phân tích dữ liệu chuỗi thời gian nhiều chiều cũng là một phương pháp rất hiệu quả nhằm cải thiện độ chính xác của mô hình dự đoán. Một trong những tính chất quan trọng với dữ liệu nhiều chiều đó là sự tương quan giữa các chiều dữ liệu. Đối với bài toán dự đoán, việc đưa đầu vào là chuỗi dữ liệu nhiều chiều mà các chiều dữ liệu không có sự tương quan phù hợp với nhau có thể gây ra tác dụng ngược, đó là gây nhiễu và dẫn đến bóp méo dữ liệu đơn chiều.

Có hai loại độ đo để xác định độ tương quan giữa hai biến ngẫu nhiên: Độ đo tương quan tuyến tính và độ đo tương quan phi tuyến tính. Đối với độ tương quan tuyến tính, Pearson correlation coefficient đang là độ đo phổ biến nhất. Độ đo tương quan pearson đôi với hai biến ngẫu nhiên (X,Y) được tính toán theo công thức:

$$\rho = \frac{\sum_i (x_i - \bar{x}_i)(y_i - \bar{y}_i)}{\sqrt{\sum_i (x_i - \bar{x}_i)^2} \sqrt{\sum_i (y_i - \bar{y}_i)^2}} \quad (2.17)$$

Trong đó, \bar{x}_i, \bar{y}_i lần lượt là giá trị trung bình của hai biến ngẫu nhiên X,Y. Giá trị của ρ nằm trong khoảng $[-1,1]$. Theo đó, nếu $\rho = -1$ hoặc $\rho = 1$, thì chứng tỏ hai biến ngẫu nhiên X,Y là có mối tương quan tuyệt đối với nhau và X,Y là hoàn toàn độc lập với nhau nếu $\rho = 0$. Tuy nhiên, dữ liệu thực tế thường là phi tuyến nên độ đo tương quan tuyến tính thường không phù hợp. Trong số rất nhiều độ đo tương quan phi tuyến, giá trị entropy được sử dụng khá rộng rãi. Entropy là một khái niệm trong lý thuyết thông tin, là độ đo không chắc chắn của biến

ngẫu nhiên X , được tính toán như sau:

$$H(X) = - \sum_i P(x_i) \log_2 P(x_i) \quad (2.18)$$

Giá trị entropy của X sau khi chia các giá trị của X theo các giá trị của Y , được tính theo công thức sau:

$$H(X|Y) = - \sum_j P(y_j) \sum_i P(x_i|y_j) \log_2 P(x_i|y_j) \quad (2.19)$$

Trong đó, $P(x_i)$, $P(y_i)$ là xác suất tiên nghiệm của các giá trị trong X, Y , $P(x_i|y_j)$ là xác suất hậu nghiệm khi các giá trị X quan sát các giá trị của Y .

Độ giảm sự không chắc chắn của X khi chia các giá trị của X theo các giá trị của Y được gọi là thông tin bổ sung (information gain):

$$IG(X|Y) = H(X) - H(X|Y) \quad (2.20)$$

Đối với độ đo này, biến Z được coi là có độ tương quan với X vào hơn Y nếu như $IG(Z|X) > IG(Z|Y)$. IG là một độ đo đối xứng, tức là $IG(X|Y) = IG(Y|X)$. Ngoài ra, IG thường có giá trị cao hơn đối với các biến có nhiều giá trị hơn. Để giải quyết vấn đề này, độ đo đối xứng không chắc chắn của X và Y [16] được đề xuất theo công thức:

$$SU(X, Y) = 2 \left[\frac{IG(X|Y)}{H(X) + H(Y)} \right] \quad (2.21)$$

SU luôn có giá trị nằm trong khoảng $[0,1]$. Theo đó, nếu SU có giá trị bằng 1, tức là X và Y có sự tương quan tuyệt đối còn nếu SU có giá trị bằng 0, X và Y hoàn toàn độc lập với nhau. Và SU vẫn đảm bảo tính đối xứng, tức là $SU(X|Y) = SU(Y|X)$. Thông qua bước lựa chọn đặc trưng, chúng tôi chọn ra được một số lượng $k (k \leq n)$ thông số có độ tương quan cao cho giai đoạn huấn luyện mô hình.

Trong chương này, em đã trình bày các kiến thức cơ bản về điện toán đám mây và bài toán tự động mở rộng tài nguyên sử dụng đối với dịch vụ điện toán đám mây. Em cũng

đã tiến hành khảo sát một cách tổng quan các nghiên cứu hiện tại cho bài toán dự đoán chuỗi thời gian và đặc biệt là dự đoán tài nguyên sử dụng trong môi trường đám mây đã được công bố thời gian qua. Các nghiên cứu này đã đề xuất các mô hình hệ thống, các thuật toán học, ... khác nhau và cũng đã đạt được những kết quả đáng khích lệ. Tuy nhiên, hầu hết các nghiên cứu đều mới chỉ phân tích đơn lẻ các thông số tài nguyên sử dụng mà bỏ qua đi việc phân tích dữ liệu nhiều chiều. Ngoài ra, kĩ thuật mờ hóa dữ liệu cũng được sử dụng nhằm làm giảm độ biến động dữ liệu và nhóm chúng thành chuỗi thời gian mờ đa chiều để đưa vào mô hình học. Trên một khía cạnh khác, độ tương quan giữa các chiều dữ liệu cũng là một yếu tố quan trọng cần được xét đến trong việc phân tích chuỗi thời gian đa chiều khi mà các thuộc tính không có độ tương quan cao có thể gây nhiễu và dẫn đến sự bóp méo dữ liệu. Rất nhiều nghiên cứu đã đề xuất các phương pháp nhằm phát hiện tương quan giữa các chiều dữ liệu, đặc biệt là các đại lượng trong môi trường điện toán đám mây. Tuy nhiên, chưa có một phương pháp nào tích hợp việc lựa chọn đặc trưng giữa các chiều dữ liệu để cải thiện độ chính xác dự đoán.

Dựa trên các thảo luận và đánh giá đó, các đóng góp được đề cập trong nội dung của đề án này bao gồm:

- Đề xuất một hệ thống dự đoán lý thuyết cho dịch vụ đám mây tự động mở rộng tài nguyên bằng cách phối hợp một vài kĩ thuật cho tiền xử lý dữ liệu và cải thiện độ chính xác dự đoán.
- Đề xuất thuật toán sử dụng độ đo đối xứng không chắc chắn (symmetrical uncertainty) để đánh giá tương quan giữa các chiều dữ liệu. Cơ chế này giúp xác định và lựa chọn ra các thông số có độ tương quan cao để đưa vào mô hình huấn luyện.
- Đề xuất một cơ chế nhằm làm giảm độ biến động của dữ liệu sử dụng kĩ thuật làm mờ và phân tích chuỗi thời gian đa chiều sử dụng mạng nơ-ron ngắn hạn dài (Long short term memory neural network - LSTM) để cải thiện độ chính xác dự đoán. Một cách tiếp cận mới được gọi là mô hình **MF-LSTM**.

Chương 3

Mô hình đề xuất

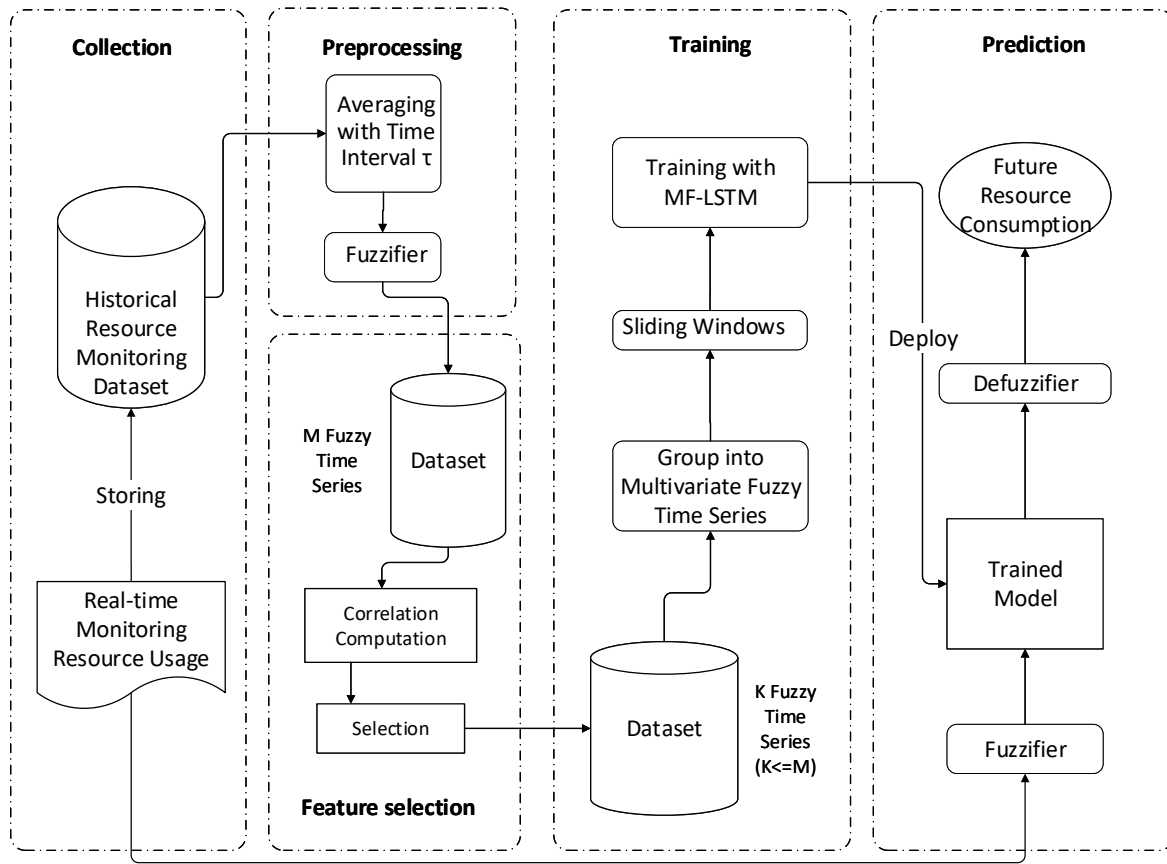
3.1 Tổng quan mô hình

Tổng quan của hệ thống được chỉ ra trong hình 3.1. Trong đó, có 5 mô đun đó là: Thu thập dữ liệu, tiền xử lý dữ liệu, lựa chọn đặc trưng, huấn luyện mô hình và dự đoán.

Trong giai đoạn **thu thập dữ liệu**, các thông số kỹ thuật về tài nguyên sử dụng của hệ thống bao gồm: CPU, Memory, Disk I/O time, Local disk space,... sẽ được theo dõi và ghi lại. Có hai loại dữ liệu được mô tả ở đây đó là dữ liệu lịch sử và dữ liệu thời gian thực. Dữ liệu lịch sử được thu thập trong quá khứ và được sử dụng xuyên suốt trong các giai đoạn tiền xử lý, lựa chọn đặc trưng và huấn luyện mô hình. Với loại thứ hai, dữ liệu thời gian thực sẽ ngay lập tức được làm mờ làm đầu vào cho mô hình đã được huấn luyện để tiến hành dự đoán tài nguyên sử dụng trong tương lai. Dữ liệu này cũng sẽ được lưu trữ vào cơ sở dữ liệu để tiến hành huấn luyện lại mô hình trong tương lai.

Sau đó, ở bước **tiền xử lý dữ liệu**, bộ dữ liệu thô ở bước một được làm sạch và biến đổi trở thành dữ liệu dạng chuỗi thời gian. Các điểm dữ liệu sẽ được biến đổi theo phép mờ hóa. Chuỗi thời gian mờ tạo được sẽ trở thành đầu vào cho giai đoạn lựa chọn đặc trưng.

Mục đích của bước **lựa chọn đặc trưng** là nhằm lựa chọn ra các thông số có độ tương quan cao để đưa vào quá trình học. Ở bước này, độ tương quan phi tuyến giữa các cặp



Hình 3.1: Kiến trúc tổng quan của hệ thống

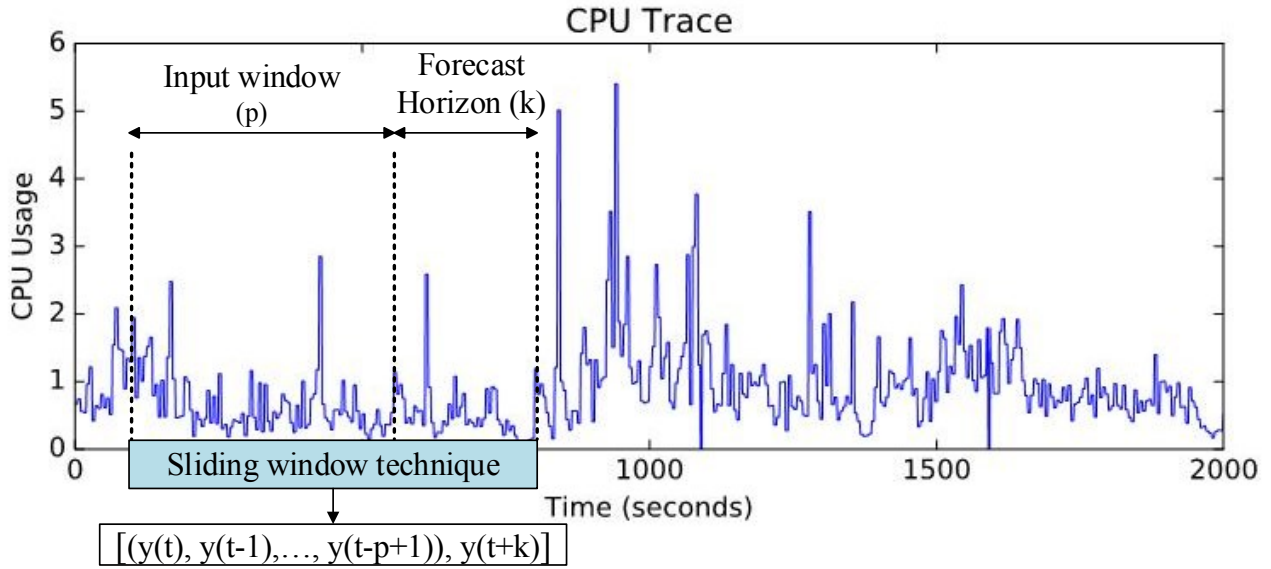
thông số sẽ được tính toán và thuật toán mà em đề xuất được thực thi để lựa chọn ra các thông số dựa vào một giá trị ngưỡng tương quan δ .

Trong giai đoạn **huấn luyện**, đầu tiên, các chuỗi thời gian mờ sẽ được nhóm lại với nhau tạo thành các chuỗi thời gian mờ đa chiều. Sau đó, kỹ thuật cửa sổ trượt được sử dụng để tạo thành các cặp ví dụ học. Chúng tôi đã lựa chọn mạng LSTM để tiến hành quá trình huấn luyện.

Cuối cùng, trong bước **dự đoán**, dữ liệu thực tế được làm mờ và đưa vào mô hình đã được huấn luyện để tiến hành dự đoán tài nguyên sử dụng trong tương lai.

3.2 Tiền xử lý dữ liệu

Thông qua quá trình tiền xử lý dữ liệu, một bộ dữ liệu phù hợp cho bước học sẽ được cấu trúc. Trong bước đầu tiên, các thông số đo đạc sẽ được biến đổi vào trong các chuỗi thời



Hình 3.2: Mức độ tiêu thụ CPU theo thời gian của một job trong bộ dữ liệu Google trace

gian tương ứng $r_i(t) = (i = 1, 2, \dots, M)$ với khoảng thời gian τ .

Mỗi điểm thuộc chuỗi thời gian được tính toán bằng trung bình cộng các giá trị đo được của một đại lượng theo công thức sau:

$$r_i(n) = \frac{\sum_{(n-1)\tau < t < n\tau} D_i(t)}{n_\tau} \quad (3.1)$$

Trong đó, $D_i(t)$ là giá trị đo được của thông số i tại thời điểm t và n là số quan sát $D_i(t)$ trong bước nhảy. Bước tiếp theo đó là mờ hóa dữ liệu để biến đổi các chuỗi thời gian ban đầu thành các chuỗi thời gian mờ tương ứng. Định nghĩa về chuỗi thời gian mờ được trình bày trong mục 2.3.5. Tiến trình làm mờ cũng được trình bày trong [3] với 3 bước chính, đó là: Khai báo không gian giá trị, định nghĩa tập mờ và biến đổi các giá trị vào tập mờ. Chúng tôi sử dụng các bước này cho quá trình mờ hóa dữ liệu.

3.3 Lựa chọn đặc trưng

Trong giai đoạn lựa chọn đặc trưng, chúng tôi triển khai thuật toán 2 để lựa chọn ra các đặc trưng có độ tương quan phi tuyến tốt nhất để huấn luyện mô hình. Theo đó, đầu tiên,

độ tương quan giữa từng cặp thông số được tính toán thông qua độ đo đối xứng không chắc chắn (symmetrical uncertainty - SU). Quá trình tính toán này là bước tính toán độ tương quan (Correlation computation) trong hình 3.1. Sau đó, ở bước tiếp theo, em chọn ra cặp thông số có độ tương quan với nhau cao nhất. Cuối cùng, em kiểm tra với các thông số còn lại, thông số nào có độ tương quan đối với cặp thông số đã được lựa chọn ở bước trước đó vượt quá ngưỡng, em sẽ lựa chọn bổ sung thông số đó. Cuối cùng, chúng tôi sẽ thu được một bộ đầu vào bao gồm các thông số có độ tương quan thỏa mãn thuật toán 2 để đưa vào làm đầu vào cho quá trình huấn luyện.

Algorithm 2: Lựa chọn các thông số có độ tương quan cao

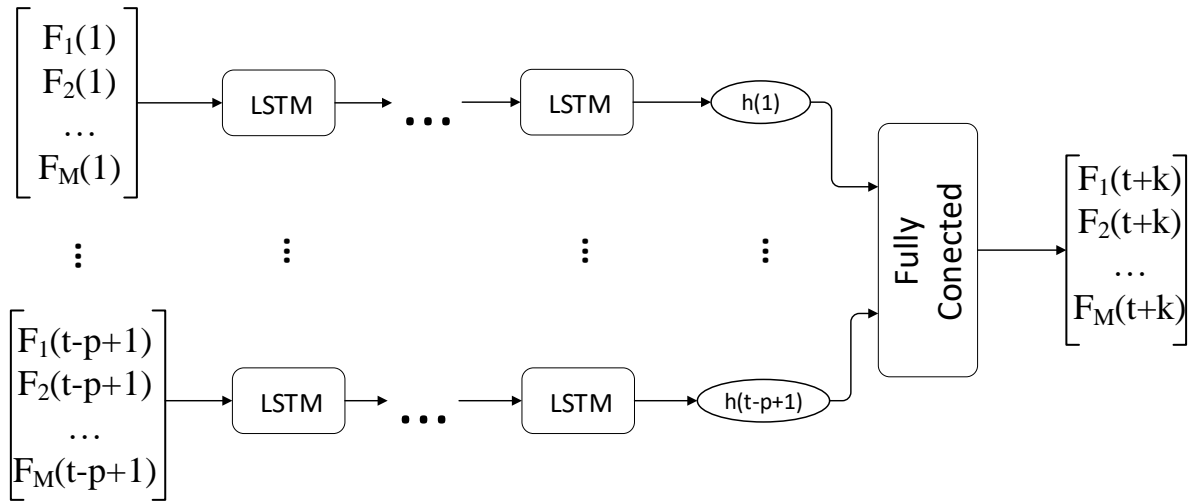
```

1 Đầu vào Ngưỡng  $\delta$  và bộ dữ liệu mờ đa chiều  $F = \{F_1, \dots, F_n\}$ 
2 Đầu ra Tập con SS chứa các thông số có độ tương quan cao
3 //Tính toán SU giữa tất cả các thông số
4 Khởi tạo: S là mảng hai chiều với các giá trị mặc định bằng 0 và có chiều dài =  $n$ 
5 for  $i = 1$  to  $n - 1$  do
6   for  $j = i + 1$  to  $n$  do
7      $S(i, j) = \text{SU}(F_i, F_j)$ 
8   end
9 end
10 // Lựa chọn hai thông số có giá trị lớn nhất trong S
11  $(F_i, F_j) = \text{argmax}(S(...))$ 
12  $SS = \{F_i, F_j\}$ 
13 for  $k = 1$  to  $n$  do
14   if ( $k \neq i$  and  $k \neq j$  and  $\text{SU}(F_i, F_k) \geq \delta$  and  $\text{SU}(F_j, F_k) \geq \delta$ ) then
15      $SS = SS \cup F_k$ 
16   end
17 end
18 Return SS

```

3.4 Huấn luyện mô hình

Giai đoạn huấn luyện mô hình gồm ba bước, đó là: Nhóm các thông số đã được lựa chọn ở bước hai vào chuỗi thời gian đa biến, sử dụng thanh cửa sổ trượt để tạo các cặp đầu vào - đầu ra và huấn luyện mô hình với mạng nơ ron LSTM. Các giá trị tương lai của các chuỗi thời gian mờ có thể được dự đoán khi mối quan hệ logic mờ đã được xác định. Mối quan hệ có



Hình 3.3: Mô hình LSTM

thể được biểu diễn thông qua các chuỗi thời gian mờ trong lịch sử. Trong cách tiếp cận của em, mạng LSTM cũng được sử dụng để xác định các mối quan hệ mờ trong quá trình học từ dữ liệu lịch sử. Mô hình LSTM sử dụng trong quá trình huấn luyện được mô tả trong hình 3.3. Theo đó, các ví dụ học được đưa qua các tầng, mỗi tầng này chứa các đơn vị LSTM. Đầu ra của các đơn vị LSTM này sẽ được đưa qua một tầng kết nối đầy đủ (fully connected) để tính toán đầu ra của mô hình. Sau khi nhóm các giá trị và các chuỗi thời gian mờ đa chiều, quá trình học của mạng LSTM sẽ được chỉ ra. Các cặp đầu vào, đầu ra cũng được cấu trúc bởi kỹ thuật của sổ trượt trước khi đưa vào mạng. Như đã trình bày ở trên, kỹ thuật của sổ trượt được áp dụng để tạo các cặp đầu vào, đầu ra. Kỹ thuật của sổ trượt trong dự đoán chuỗi thời gian cũng được thể hiện trong hình 3.2, với p là kích cỡ của sổ trượt và k là vị trí dự đoán.

Trong các chuỗi thời gian mờ đa biến, chúng tôi cũng áp dụng tương tự kỹ thuật của sổ trượt trong các tương tự như đã mô tả ở trên. Khi đó, đầu vào là $F(t), F(t-1), \dots, F(t-p+1)$ và đầu ra là $F(t+k)$. Một điều kiện chung trong dự đoán các chuỗi thời gian là $p \leq k$. Sự hoạt động của mô hình MF-LSTM được miêu tả bởi thuật toán 3.

3.5 Dự đoán

Dựa vào kiến trúc tổng quan 3.1, bộ thu thập dữ liệu thời gian thực ghi nhận dữ liệu tiêu thụ tài nguyên ở hệ thống. Giá trị tài nguyên sử dụng này sẽ ngay lập tức được và trở

Algorithm 3: MF-LSTM

- 1 Chuẩn hóa tất cả dữ liệu tài nguyên lịch sử của M thông số: $r_1(t), r_2(t), r_3(t), \dots, r_M(t)$
- 2 Mờ hóa chuỗi thời gian của M loại tài nguyên vào trong M chuỗi thời gian M tương ứng: $F_1(t), F_2(t), \dots, F_i(t), \dots, F_M(t)$
- 3 Lựa chọn k thông số có độ tương quan cao nhất: $SS = F_1(t), F_2(t), \dots, F_k(t)$
- 4 Nhóm các chuỗi thời gian mờ vào các đơn vị đa chiều F(t).
- 5 Xây dựng các tầng LSTM, khởi tạo một cửa sổ trượt với p điểm liên tục $F(t), F(t-1), F(t-2), \dots, F(t-p+1)$ làm đầu vào và k điểm tiếp theo ($t = 1, 2, \dots, n$) làm đầu ra.
- 6 Huấn luyện mô hình với thuật toán lan truyền ngược theo thời gian. Mô hình được huấn luyện sẽ biểu diễn mối quan hệ logic mờ: $F(t), F(t-1), F(t-2), \dots, F(t-p+1) \rightarrow F(t+k)$
- 7 Triển khai mô hình vào bước dự đoán
- 8 Thực hiện lại bước 1 - 6 với các chu kỳ thời gian T khác nhau.

thành đầu vào cho mô hình MF-LSTM đã được huấn luyện ở bước 2. Giá trị đầu ra của bước này sẽ được giải mờ theo phương pháp giải mờ trọng tâm và cũng là kết quả cuối cùng của quá trình dự đoán. Chi tiết các bước trong quá trình dự đoán được mô tả chi tiết trong thuật toán 4.

Algorithm 4: Mô hình dự đoán

- 1 Tại mỗi điểm thời gian t, thu thập dữ liệu tài nguyên sử dụng mới $r_1(t), r_2(t), \dots, r_i(t)$, làm mờ chúng vào chuỗi thời gian mờ $F_1(t), F_2(t), \dots, F_i(t), \dots, F_M(t)$. Tính toán các giá trị $F_1(t+k), F_2(t+k), \dots, F_i(t+k), \dots, F_M(t+k)$ dựa vào mô hình huấn luyện.
- 2 Với mỗi giá trị dự đoán, $F_1(t+k), F_2(t+k), \dots, F_i(t+k), \dots, F_M(t+k)$, áp dụng phương pháp giải mờ trọng tâm để lấy ra kết quả cuối cùng, chính là giá trị tài nguyên giả định ở thời điểm tiếp theo: $r_i^{pred}(t+1), i = 1, 2, \dots, M$
- 3 Thực hiện lại bước 1 \rightarrow 2 ở các điểm thời gian tiếp theo.

Chương 4

Thực nghiệm và đánh giá

4.1 Cài đặt thực nghiệm

Bộ dữ liệu Google cluster trace được sử dụng để tiến hành đánh giá mức độ hiệu quả của hệ thống. Bộ dữ liệu là sự thu thập của nhiều job và lượng tài nguyên sử dụng của chúng được báo cáo bởi Google năm 2011. Mỗi job chứa nhiều task khác nhau và chạy đồng thời trên nhiều máy khác nhau. Bộ dữ liệu này bao gồm có 20 thông số khác nhau được đo trên một vài thông số chính đó là CPU, bộ nhớ, không gian lưu trữ cục bộ,... Dữ liệu được đo trên 12,5 nghìn máy khác nhau trong vòng 29 ngày với 1232799308 bản ghi dữ liệu về tài nguyên sử dụng. Để tiến hành đánh giá sự hiệu quả của mô hình với bộ dữ liệu các dấu vết của cụm máy chủ Google, tôi lựa chọn job có ID là 6176858948. Job này bao gồm 25954362 bản ghi dữ liệu chạy trong suốt 29 ngày. Dữ liệu được extract bao gồm 15 trường: timestamp, taskIndex, machineId, mean CPU Usage, Canonical Memory Usage, assigned memory, unmapped cache usage, page cache usage, max mem usage, mean diskIO time, mean local disk space, max CPU usage, max disk io time, CPI, sampled CPU usage từ bảng Task usage của bộ dữ liệu Google Cluster Trace. Tôi sử dụng dữ liệu trong 20 ngày đầu để xây dựng mô hình học và 9 ngày cuối cùng để đánh giá hiệu năng của mô hình. Ngoài ra, giả sử rằng để bật tắt một máy ảo cần mất 10 phút, tức là để ra quyết định khả mở kịp thời, việc dự đoán phải cho ra kết quả trước thời điểm cần dùng máy 10 phút và của số dự đoán không được thiết lập bằng 1. Độ chính xác của hệ

thống cũng được đánh giá bằng cách ước lượng giá trị lỗi trung bình tuyệt đối (MAE) được định nghĩa như sau:

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (4.1)$$

Trong đó, \hat{y}_i là giá trị dự đoán, y_i là giá trị thực tế.

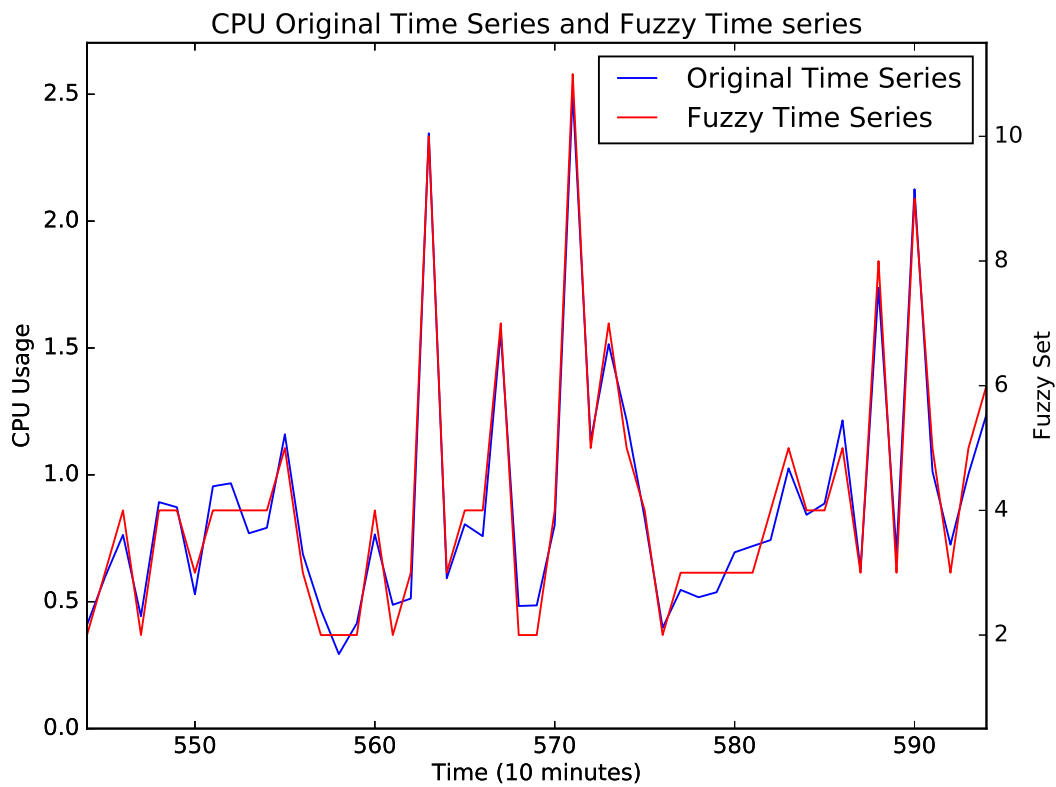
Các tập mờ được lựa chọn dựa theo các phương pháp trong [2]:

$$A_i = \frac{0}{u_1} + \frac{0}{u_2} + \dots + \frac{0.5}{u_{i-1}} + \frac{1.0}{u_i} + \frac{0.5}{u_{i+1}} + \dots + \frac{0}{u_n}.$$

4.2 Mờ hóa dữ liệu

Thí nghiệm mờ hóa dữ liệu được thực hiện để kiểm tra độ hiệu quả của cơ chế làm mờ trong quá trình tiền xử lý dữ liệu nhằm làm giảm độ biến động dữ liệu. Hình 4.1 chỉ ra một ví dụ so sánh chuỗi thời gian của CPU (khi chưa được mờ hóa) và chuỗi thời gian mờ tương ứng. Trong các đoạn (551, 554), (577, 581) ,... độ biến động dữ liệu được giảm thiểu do các giá trị này được biến đổi vào một tập mờ duy nhất.

Do có hai bước bao gồm mờ hóa và giải mờ dữ liệu nên sẽ cần thêm chi phí cho quá trình biến đổi và độ chính xác dự đoán sẽ có thể được cải thiện. Tuy nhiên, cần chú ý rằng, đối với việc ra quyết định mở rộng, chi phí tính toán bổ sung cho việc biến đổi không ảnh hưởng nhiều đến hiệu năng của hệ thống. Ví dụ, trong chuỗi thời gian CPU, cho một tập mờ $A = 0.0/u_1 + 1.0/u_2 + 0.0/u_3$, $u_1 = (0, 100\%)$, $u_2 = (100\%, 200\%)$, $u_3 = (200\%, 300\%)$. Nếu giá trị dự đoán của chuỗi thời gian mờ là A_2 và giá trị thực tế cũng thuộc về tập mờ A_2 , như vậy, cả giá trị giải mờ và giá trị thực tế sẽ trong một khoảng tương tự nhau (100%, 200%). Như vậy, các tiếp cận mờ có quyết định mở rộng tương tự với số lượng lõi CPU là 2. Hiện tượng này cũng giống như các độ đo tài nguyên khác như memory, disk I/O,...

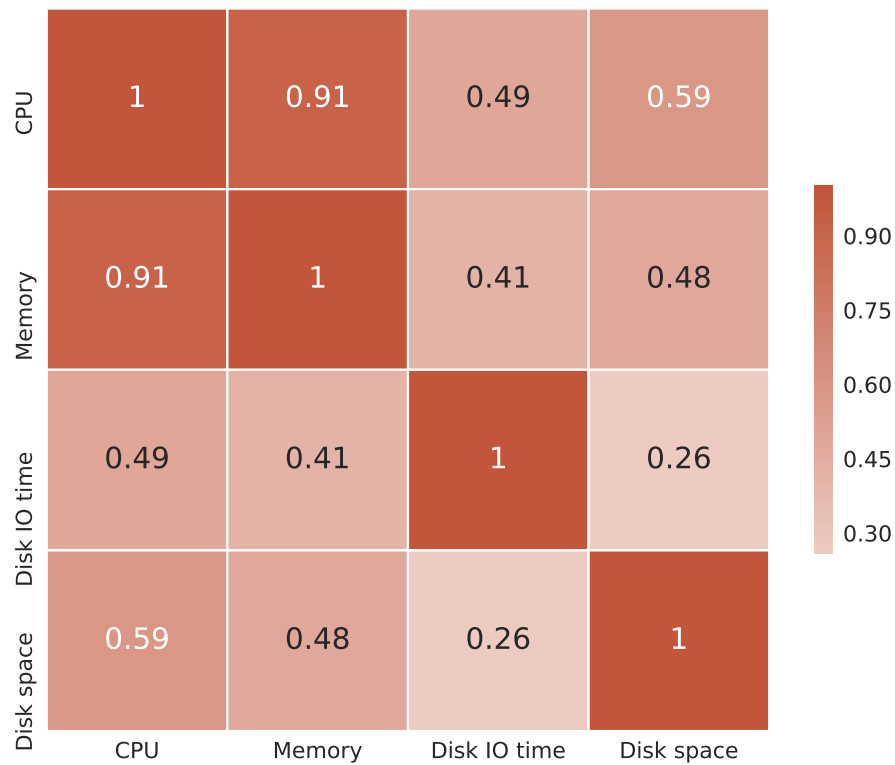


Hình 4.1: Chuỗi thời gian thực tế và chuỗi thời gian mờ tương ứng

4.3 Lựa chọn đặc trưng

Mục đích của thí nghiệm này là để chứng minh tính hiệu quả của thuật toán lựa chọn đặc trưng để chọn ra các thông số có độ tương quan cao cho bước huấn luyện mô hình. Trong bước này, sự tương quan của bốn thông số chính của các cụm máy chủ Google, đó là trung bình CPU sử dụng (mean CPU usage), bộ nhớ sử dụng (memory), trung bình thời gian truy xuất ổ cứng (mean disk I/O time) và trung bình dung lượng lưu trữ cục bộ (mean local disk space) được tính toán thông qua thuật toán 2. Sự tương quan của các thuật toán này được chỉ ra trong hình 4.2.

Có thể thấy rằng CPU và Memory có sự tương quan với nhau tốt nhất (giá trị rơi vào khoảng 0.91). Trong bước tiếp theo, ở dòng 11 của thuật toán 1, tôi chọn ra CPU - Memory là cặp thông số có độ tương quan cao nhất. Sau đó, tôi tiếp tục lựa chọn các thông số còn lại thông qua việc kiểm tra các độ tương quan của chúng với CPU và memory so với giá trị ngưỡng tương quan. Trong thí nghiệm này, tôi lựa chọn ra 3 ngưỡng là 0.8, 0.45, 0.3 để tiến hành thực



Hình 4.2: Nonlinearity correlation

	$\delta = 0.8$	$\delta = 0.45$	$\delta = 0.3$
CPU	Disk io time	Disk io time	Disk io time
	Local disk space	Local disk space	Local disk space
Memory	Disk io time	Disk io time	Disk io time
	Local disk space	Local disk space	Local disk space

Bảng 4.1: Kết quả thực nghiệm lựa chọn đặc trưng

nghiệm cho quá trình lựa chọn đặc trưng. Kết quả của quá trình lựa chọn đặc trưng được chỉ ra trong bảng 4.1. Với mỗi giá trị ngưỡng, ngoài các thông số CPU và Memory đã được lựa chọn đầu tiên, những thông số được lựa chọn sẽ được đánh dấu bằng màu xám. Ví dụ với giá trị ngưỡng bằng 0.45, tôi lựa chọn ra mean local disk space có giá trị ngưỡng thỏa mãn điều kiện ở dòng 14 của thuật toán 2. Tương tự như vậy với giá trị ngưỡng bằng 0.3, ta chọn được tất cả các thông số vào pha tiếp theo.

4.4 Dự đoán

Thí nghiệm này của chúng tôi tập trung vào việc so sánh độ chính xác của mô hình MF-LSTM giữa các ngưỡng tương quan khác nhau và với mô hình học đơn biến. Trong từng trường hợp, chúng tôi cũng tiến hành thực nghiệm trên nhiều giá trị của số trượt (sw) khác nhau. Kết quả của thí nghiệm này được chỉ ra trong bảng 4.2, 4.3.

Kết quả cho thấy, không có một mô hình nào có độ chính xác hơn hẳn so với các mô hình còn lại. Tuy nhiên có một vài điều cần thảo luận ở đây. Đó là với giá trị $\delta = 0.8$, chúng tôi chọn ra được CPU và Memory. Với cặp thông số này, chúng tôi nhận thấy giá trị lỗi MAE của cả CPU và bộ nhớ sử dụng đều có thể đạt cực tiểu. Đó là với CPU, MAE đạt 0.3221 và với giá trị bộ nhớ sử dụng, MAE đạt 0.0303. Kết quả dự đoán của mô hình MF-LSTM với các giá trị ngưỡng tương quan khác nhau và mô hình đơn biến được minh họa ở hình 4.3, 4.4, 4.5, 4.6. Dựa vào hình vẽ, có thể nhận thấy rằng các mô hình đều bắt khá tốt xu hướng của dữ liệu, đặc biệt đối với trường hợp dự đoán memory. Tuy nhiên, mô hình đa chiều với ngưỡng $\delta = 0.8$ bắt được các điểm đột biến trên và dưới tốt hơn. Ngoài ra, cũng có thể nhận thấy rằng việc lựa chọn các thông số để tạo ra đầu vào đa chiều đưa vào mô hình huấn luyện cũng là con dao hai lưỡi. Quá trình thực nghiệm cho thấy, việc lựa chọn giá trị ngưỡng tương quan không tốt dẫn đến việc các tham số đầu vào có giá trị tương quan với nhau không đủ lớn có thể khiến mô hình không tốt và có sai số lớn. Đặc biệt, kết quả dự đoán cho thấy mô hình có khả năng dự đoán tốt xu hướng và tính chu kỳ của chuỗi thời gian cùng với bắt được các điểm đột biến lên và xuống. Hơn nữa, đối với hình 4.3, mô hình dự đoán cho kết quả tốt khi mà các điểm dữ liệu của CPU và Memory được dự đoán chính xác với đầu vào là chuỗi thời gian đa chiều với sự kết hợp của CPU và Memory - được lựa chọn với giá trị ngưỡng bằng 0.8 thông qua thuật toán 2.

Tổng kết

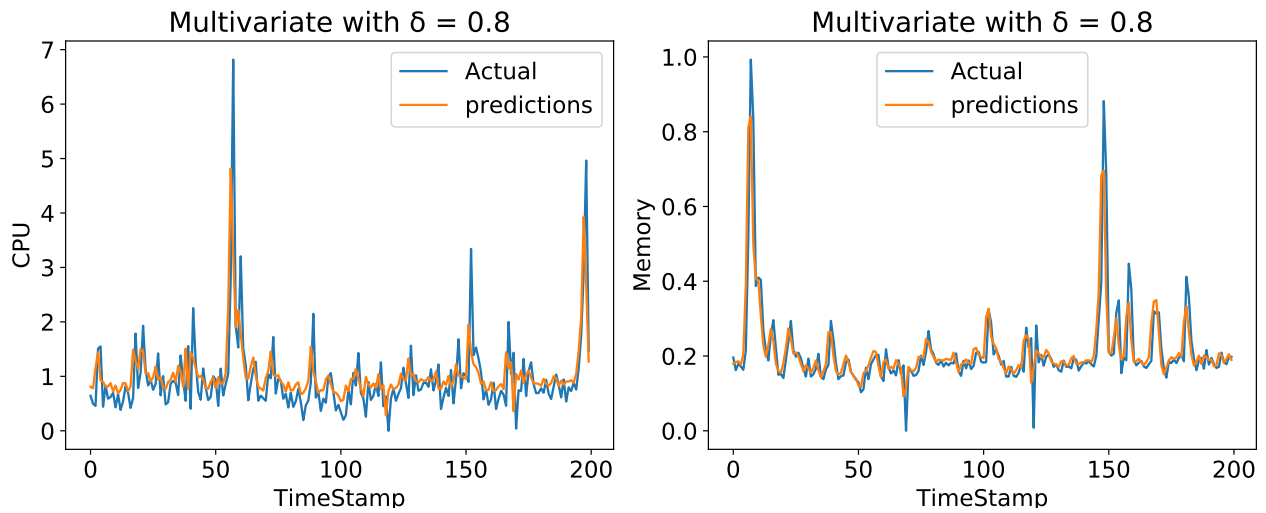
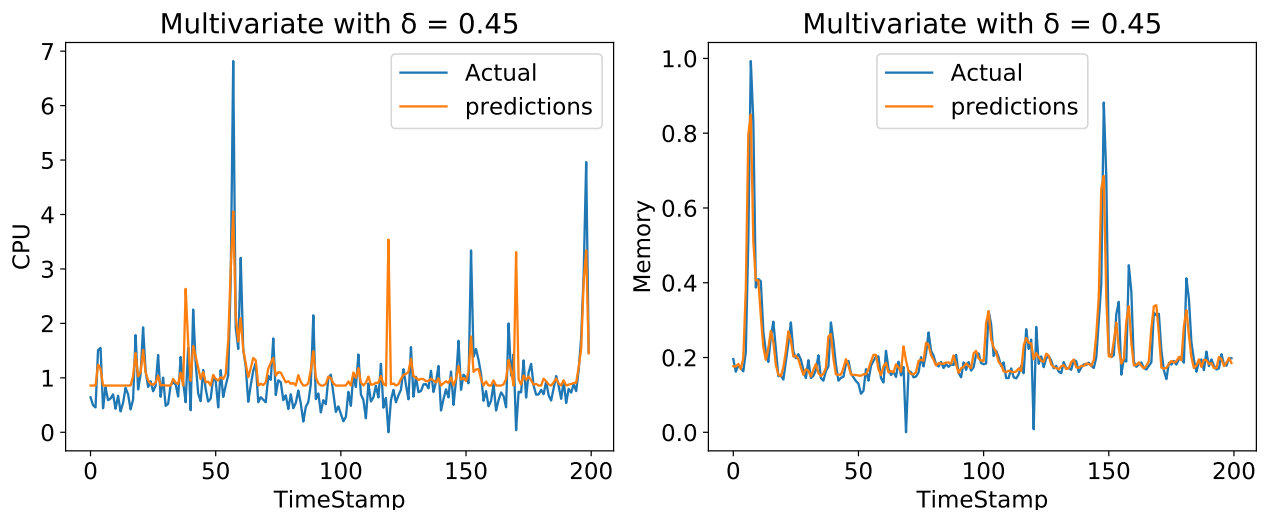
Trong chương này, em đã tiến hành thực nghiệm và đánh giá hệ thống mà mình đề xuất. Quá trình thực nghiệm sử dụng bộ dữ liệu Google cluster trace để dự đoán và đánh giá.

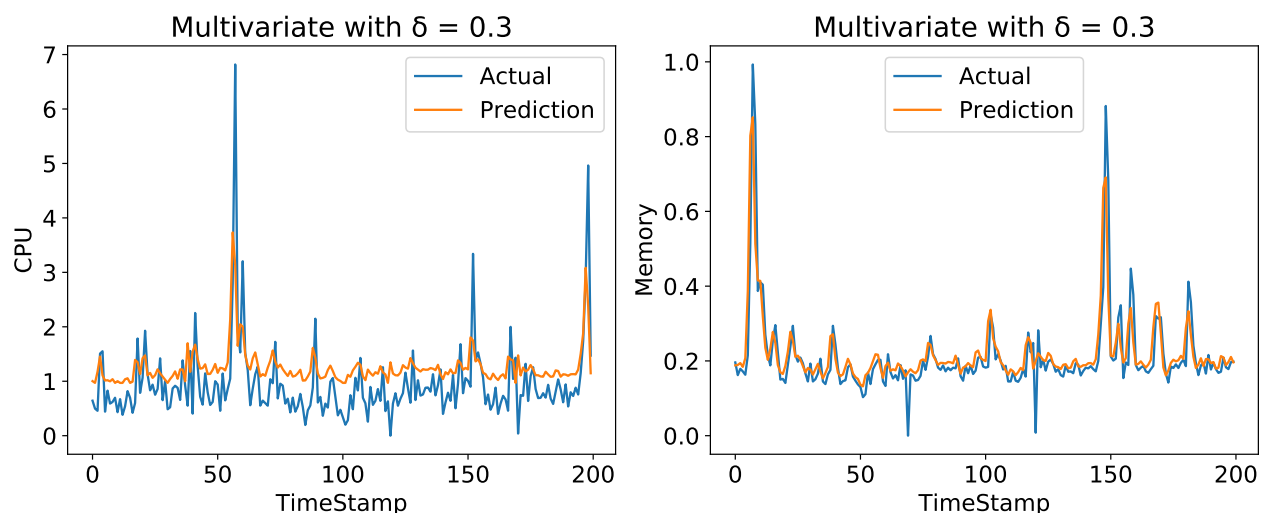
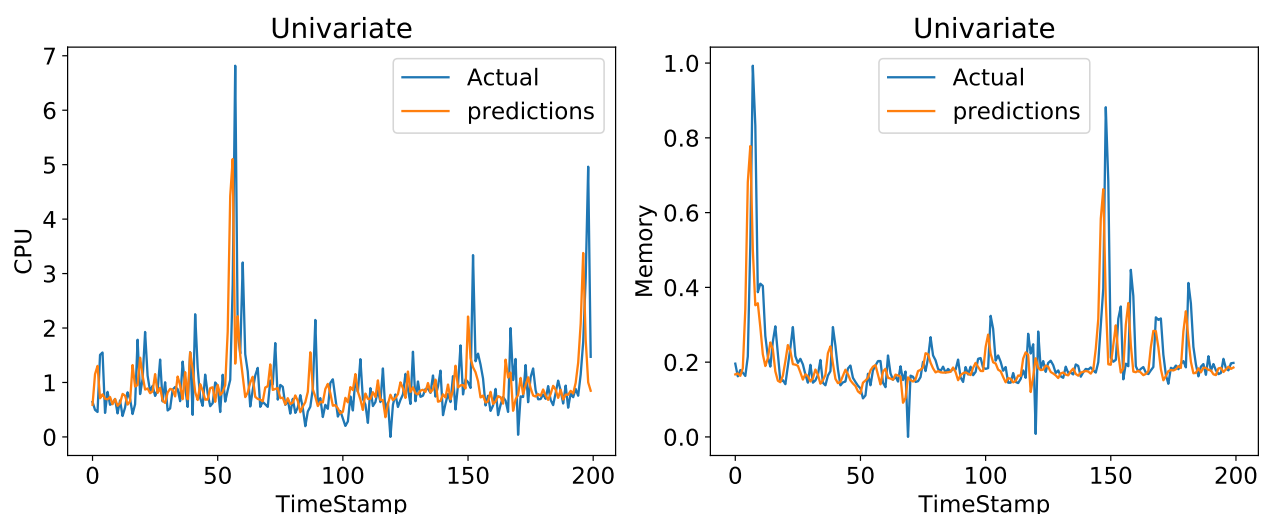
	sw = 2	sw = 3	sw = 4	sw = 5
Đa biến với $\delta = 0.8$	0.3221	0.3318	0.3383	0.3259
Đa biến với $\delta = 0.45$	0.3546	0.3745	0.3673	0.3866
Đa biến với $\delta = 0.3$	0.4244	0.4075	0.4058	0.4335
Đơn biến	0.3510	0.3316	0.3528	0.3278

Bảng 4.2: Dự đoán CPU

	sw = 2	sw = 3	sw = 4	sw = 5
Đa biến với $\delta = 0.8$	0.0303	0.0305	0.0309	0.0307
Đa biến với $\delta = 0.45$	0.0341	0.0339	0.0328	0.0361
Đa biến với $\delta = 0.3$	0.0377	0.0368	0.0423	0.0419
Đơn biến	0.0357	0.0346	0.0406	0.0362

Bảng 4.3: Dự đoán Memory

Hình 4.3: Kết quả dự đoán với $sw=2$ và $\delta = 0.8$ Hình 4.4: Kết quả dự đoán với $sw=2$ và $\delta = 0.45$

Hình 4.5: Kết quả dự đoán với $sw=2$ và $\delta = 0.3$ Hình 4.6: Kết quả dự đoán với $sw=2$ và đầu vào đơn biến

Các thí nghiệm được triển khai bao gồm:

- Đánh giá mức độ hiệu quả của quá trình mờ hóa trong việc làm giảm độ biến động dữ liệu để cải thiện độ chính xác dự đoán.
- Đánh giá độ tương quan giữa các chiều dữ liệu sử dụng độ đo tương quan phi tuyến (symmetrical uncertainty) và sử dụng thuật toán 2 được đề xuất trong mục 3.3 nhằm lựa chọn các chiều dữ liệu có độ tương quan cao để đưa vào quá trình huấn luyện.
- Tiến hành huấn luyện mô hình LSTM sử dụng chuỗi thời gian mờ đa chiều (MF-LSTM) để tiến hành dự đoán chuỗi thời gian tài nguyên sử dụng và đánh giá độ chính xác dự đoán dựa vào độ đo lỗi trung bình tuyệt đối (MAE).

Kết quả thực nghiệm cho thấy, các mô hình đề xuất đã đạt được những kết quả khích lệ. Việc sử dụng phương pháp mờ hóa đã giúp làm giảm độ biến động dữ liệu trên nhiều đoạn của chuỗi thời gian. Điều này cho thấy việc lựa chọn đặc trưng với giá trị ngưỡng = 0.8 đã đem lại kết quả tích cực. Trên hết, các thực nghiệm đã chỉ ra các kết quả chỉ ra đã cho thấy mô hình đề xuất đã đem lại hiệu quả và giải quyết được các mục tiêu mà tôi đề xuất ở chương 3.

Chương 5

Kết luận và hướng phát triển

Trong nghiên cứu này, chúng tôi đã đề xuất ra một hệ thống dự đoán mới cho các dịch vụ chủ động mở rộng tài nguyên đám mây. Điều này được áp dụng thực tế trong các hệ thống đám mây hiện nay. Trong nghiên cứu này, chúng tôi cũng đã đề xuất một vài cơ chế để cải thiện độ chính xác cũng như tính hiệu quả của mô hình dự đoán. Để giảm thiểu độ biến động của dữ liệu, chúng tôi sử dụng kỹ thuật mờ hóa dữ liệu để tạo ra các chuỗi thời gian mờ. Chúng tôi cũng trình bày sự cần thiết của việc phân tích sự tương quan giữa các chiều dữ liệu dựa trên độ đo đối xứng không chắc chắn SU. Theo đó, chúng tôi đã tạo ra một bộ dữ liệu phù hợp cho mô hình dự đoán sử dụng mạng LSTM. Mô hình này đã được chứng minh là một mô hình học hiệu quả cho phân tích dữ liệu chuỗi thời gian. Chúng tôi đã thử nghiệm hệ thống với dữ liệu thực tế dấu vết của các cụm máy chủ Google và phân tích tính ứng dụng của hệ thống của chúng tôi trong các hệ thống đám mây thực tế. Trong tương lai, chúng tôi muốn đánh giá tính chính xác của các kỹ thuật học khác cho mô hình dự đoán của chúng tôi. Ngoài ra, các phương pháp dự đoán dựa trên cách tiếp cận Bayesian cũng đã đem lại các kết quả tốt trong việc đánh giá mức độ không chắc chắn của giá trị dự đoán. Chúng tôi cũng kì vọng sẽ đánh giá hiệu quả của các phương pháp này và áp dụng giá trị không chắc chắn dự đoán đối với bài toán mở rộng tài nguyên trong môi trường điện toán đám mây. Bên cạnh đó, chúng tôi cũng muốn xây dựng mô đun mở rộng tài nguyên cho hệ thống đã được đề xuất.

Tài liệu tham khảo

- [1] Kanad Chakraborty, Kishan Mehrotra, Chilukuri K. Mohan, and Sanjay Ranka. Forecasting the behavior of multivariate time series using neural networks. *Neural Networks*, 5(6):961 – 970, 1992.
- [2] Erol Egrioglu, Cagdas Hakan Aladag, Ufuk Yolcu, Vedide R. Uslu, and Murat A. Basaran. A new approach based on artificial neural networks for high order multivariate fuzzy time series. *Expert Systems with Applications*, 36(7):10589 – 10594, 2009.
- [3] Christopher Olah et al. Understanding lstm networks. <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>.
- [4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning Book*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [5] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.
- [6] Jitendra Kumar, Rimsha Goomer, and Ashutosh Kumar Singh. Long short term memory recurrent neural network (lstm-rnn) based workload forecasting model for cloud datacenters. *Procedia Computer Science*, 125:676 – 682, 2018. The 6th International Conference on Smart Computing and Communications.
- [7] Qiang Song and Brad S. Chissom. Fuzzy time series and its model. 54:269–277, 03 1993.
- [8] D. Tran, N. Tran, B. M. Nguyen, and H. Le. Pd-gabp 2014; a novel prediction model applying for elastic applications in distributed environment. In *2016 3rd National Founda-*

tion for Science and Technology Development Conference on Information and Computer Science (NICS), pages 240–245, Sept 2016.

- [9] Dang Tran, Nhuan Tran, Giang Nguyen, and Binh Minh Nguyen. A proactive cloud scaling model based on fuzzy time series and sla awareness. *Procedia Computer Science*, 108:365 – 374, 2017. International Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland.
- [10] A. Tucker, S. Swift, and X. Liu. Variable grouping in multivariate time series via correlation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 31(2):235–245, Apr 2001.
- [11] Carlos Vazquez, Ram Krishnan, and Eugene John. Time series forecasting of cloud data center workloads for dynamic resource provisioning. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, 6(3):87–110, 2015.
- [12] Lei Yu and Huan Liu. Efficient feature selection via analysis of relevance and redundancy. *J. Mach. Learn. Res.*, 5:1205–1224, December 2004.