PD-GABP - Một mô hình dự đoán tiêu dùng tài nguyên cho các ứng dụng trong môi trường phân tán

Trần Văn Đặng, Trần Đức Nhuận

Tóm tắt--**Trong số các kỹ thuật co giãn tài nguyên (scaling techniques) thì phương pháp dự đoán trước sự tiêu dùng tài nguyên và tải công việc (workload) của ứng dụng mang lại hiệu quả đáng kể cho hoạt động của mô hình phần mềm như một dịch vụ (Software-as-a-Service – SaaS) trong môi trường đám mây. Nguyên nhân là bởi hệ thống sẽ biết trước và chính xác lượng tài nguyên cần cung cấp trong tương lai gần cho ứng dụng là bao nhiêu, từ đó cho phép tăng giảm tài nguyên trước khi nảy sinh các vấn đề về hoạt động do thừa hoặc thiếu tài nguyên cung cấp. Mặc dù nhiều mô hình dự đoán khác nhau đã được đề xuất, độ chính xác của các mô hình này vẫn cần phải nâng cao hơn nữa. Trong bài báo này, chúng tôi đề xuất một mô hình dự đoán mới được xây dựng bằng sự kết hợp giữa giải thuật phát hiện chu kỳ và mạng nơ-ron nhân tạo. Bên cạnh đó, giải thuật di truyền - lan truyền ngược sai số (genetic-backpropagation algorithm) được sử dụng để huấn luyện mạng nhằm nâng cao độ chính xác. Các kết quả thực nghiệm với tập dữ liệu của một ứng dụng web thực tế đã chứng minh hiệu quả đáng kể của mô hình dự đoán này, do đó tăng hiệu quả hoạt động của ứng dụng chạy trên môi trường đám mây và phân tán.**

Từ khóa--**mô hình dự đoán, phát hiện chu kỳ, mạng nơ-ron nhân tạo, ứng dụng linh hoạt, quản lý tài nguyên tự động, giải thuật di chuyền, điện toán đám mây.**

1. GIỚI THIỆU

Phần mềm như một dịch vụ (Software-as-a-Service - SaaS) là một khái niệm ám chỉ tới mô hình triển khai phần mềm trong đó ứng dụng IT được cung cấp cho khách hàng như môt dịch vụ theo nhu cầu (on-demand service), thường thông qua mạng Internet. Đặc điểm của mô hình SaaS là người sử dụng trả tiền theo sự tiêu dùng tài nguyên trong thực tế (pay-per-use). Việc giảm chi phí bản quyền phần mềm, giảm thời gian cấu hình ứng dụng, và sự gia tăng của nhu cầu sử dụng công nghệ thông tin trong doanh nghiệp đã thúc đẩy sự phát triển thị trường SaaS từ khi mô hình này ra đời vào cuối năm 2000. Mặc dù SaaS đã trở thành một giải pháp công nghệ thông tin không thể thiếu cho cá nhân và doanh nghiệp trong mọi lĩnh vực, một vài lo lắng vẫn còn tồn tại liên quan tới người lập trình ứng dụng SaaS. Thứ nhất, việc triển khai và cấu hình ứng dụng cần phải được tự động hóa. Thứ hai, tài nguyên cung cấp cho ứng dụng cần có khả năng linh hoạt thay đổi tùy nhu cầu thực tế của ứng dụng theo thời gian.

Cùng thời gian này, công nghệ ảo hóa có những bước cải tiến đáng kể và trở thành một chuẩn không chính thức hứa hẹn giải quyết các vấn đề của SaaS. Công nghệ ảo hóa đã cho phép giảm chi phí (phần cứng và năng lượng), phân phối tài nguyên tính toán tốt hơn và quản lý tập trung [13]. Người sử dụng nhận được các lợi ích to lớn trên từ các nhà cung cấp dịch vụ ảo hóa mà không cần đầu tư chi phí cho việc mua sắm phần cứng. Các nhà cung cấp này thường được gọi là các nhà cung cấp dịch vụ điện toán đám mây.

Tuy nhiên, vẫn có nhiều nghi ngại trong việc di trú các ứng dụng lên đám mây tính toán. Một trong số đó là làm thế nào khai thác triệt để tính linh hoạt tài nguyên ảo hóa cho ứng dụng dạng SaaS theo cách thức cho phép tự động sự tăng, giảm tài nguyên dựa vào yêu cầu người dùng hay phầm mềm trong quá trình sử dụng. Đặc biệt là khi mà các yêu cầu này thường xuyên thay đổi theo thời gian. Hiện nay, phần lớn các nhà cung cấp đám mây đưa ra các chức năng hoặc dịch vụ để đo lường sự tiêu dùng tài nguyên, một số khác cung cấp giải pháp cân bằng tải (load balance) [12]. Giải pháp này nhằm mục đích cho phép triển khai tài nguyên linh hoạt trong môi trường đám mây nhưng đáng tiếc rằng với một vài trường hợp đặc biệt các giải pháp này là chưa đủ để chắc chắn rằng hệ thông đáp ứng hoàn toàn các yêu cầu khi ứng dụng triển khai trong thực tế. Theo hướng tiếp cận này, kĩ thuật co giãn tài nguyên (auto-scaling technique) trong môi trường phân tán và đám mây có thể chia thành 3 loại sau:

- Periodicity (theo chu kỳ tiêu dùng tài nguyên): Thông thường trong suốt quá trình vận hành, mô hình SaaS có chu kỳ tiêu dùng tài nguyên theo thời gian (ví dụ giờ, ngày hay tháng). Dựa trên đặc điểm này, người quản trị có thể quyết định thời điểm phù hợp để co giãn tài nguyên cho ứng dụng. Nhược điểm của phương pháp này là khi xem xét trên cả chu kỳ để quyết định co giãn, hệ thống không thể đáp ứng các yêu cầu nảy sinh tức thời từ ứng dụng. Phương pháp phát hiện và đánh giá chu kỳ được đề xuất trong một số công trình nghiên cứu như [3] và [18].

- Thresholds (dựa vào ngưỡng): Các giá trị ngưỡng được thiết lập để xác định khi nào tăng hoặc giảm tài nguyên. Phương pháp này hoạt động dựa trên số liệu tiêu dùng tài nguyên như tỉ lệ phân trăm sử dụng của CPU, bộ nhớ trong hay số kết nối tới hệ thống. Tuy nhiên nhược điểm của phương pháp này là rất khó để xác định chính xác giá trị ngưỡng thỏa mãn yêu cầu của ứng dụng mà vẫn tránh được việc lãng phí hay thiếu tài nguyên (do co giãn sớm hơn hoặc muộn hơn). Ngoài ra, tại thời điểm thực hiện co giãn tài nguyên, trạng thái và yêu cầu của hệ thống có thể đã được thay đổi, do đó việc co giãn này thường chưa đạt được

Công trình này được thực hiện dưới sự hướng dẫn của TS Nguyễn Bình Minh.

Trần Văn Đặng, sinh viên lớp KSTN - CNTT, khóa 57, Viện Công nghệ thông tin và Truyền thông, trường Đại học Bách Khoa Hà Nội (điện thoại: 0964351894, e-mail: 20121515@student.hut.edu.vn).

Trần Đức Nhuận, sinh viên lớp CNTT-TT 2.04, khóa 57, Viện Công nghệ thông tin và Truyền thông, trường Đại học Bách Khoa Hà Nội (điện thoại: 01296415766, e-mail: 20122206@student.hut.edu.vn).

hiệu quả như mong đợi.

- Prediction (dự đoán tiêu dùng tài nguyên): Phương pháp này sử dụng dữ liệu thu thập được trong khoảng thời gian trước đó để dự đoán lượng tài nguyên tiêu dùng hoặc tải công việc (workload) trong tương lai, từ đó xác định trước và chính xác lượng tài nguyên mà hệ thống sẽ phải cung cấp cho ứng dụng. Nhiều công trình nghiên cứu đã đưa ra các mô hình dự đoán khác nhau như [9], [19], [10], đặc biệt áp dụng trong điện toán đám mây như [8], [14]. Tuy nhiên độ chính xác của các mô hình dự đoán này vẫn cần cải tiến hơn nữa.

Trong bài báo này, chúng tôi đề xuất một mô hình dự đoán tiêu thụ tài nguyên mới áp dụng cho mô hình SaaS. Dựa trên việc phân tích dữ liệu tiêu dùng theo thời gian của ứng dụng, chúng tôi nhận thấy hầu hết các ứng dụng dạng này đều có chu kỳ tiêu thụ tài nguyên khá rõ ràng. Do đó mô hình dự đoán trong nghiên cứu này được xây dựng bằng sự kết hợp giữa giải thuật phát hiện chu kỳ và mạng nơ-ron nhân tạo. Bên cạnh đó, giải thuật di truyền kết hợp với lan truyền ngược sai số (genetic-backpropagation algorithm) được sử dụng để huấn luyện mạng nhằm nâng cao độ chính xác trong dự đoán. Để đánh giá hiệu quả của mô hình này, chúng tôi đã tiến hành thử nghiệm trên bộ dữ liệu thực tế của ứng dụng FIFA World Cup 98 [2]. Các thông số được ghi lại trong tập dữ liệu này là một trong các nhân tố sử dụng cho việc quyết định co giãn tài nguyên cung cấp cho ứng dụng SaaS. Các kết quả thu được đã chứng minh mô hình của chúng tôi có độ chính xác cao hơn các phương pháp dự đoán trước đó.

Phần tiếp theo của bài báo được tổ chức như sau. Phần 2 trình bày về các công trình nghiên cứu liên quan. Mô hình mạng nơ-ron nhân tạo sử dụng giải thuật di truyền kết hợp lan truyền ngược sai số (GA-BPNN) được trình bày chi tiết ở phần 3. Phần 4 trình bày mô hình đề xuất dựa trên sự kết hợp giữa giải thuật phát hiện chu kỳ và mô hình GA-BPNN (PD-GABP). Phần 5 trình bày kết quả thực nghiệm của mô hình đề xuất và so sánh với các phương pháp hiện tại. Phần cuối cùng kết luận và trình bày hướng nghiên cứu trong tương lai.

2. CÁC CÔNG TRÌNH LIÊN QUAN

Bài toán dự đoán chuỗi thời gian (time series forecasting) đang nhận được sự quan tâm của rất nhiều các công trình nghiên cứu gần đây với những cố gắng nhằm nâng cao độ chính xác. Trong suốt ba thập kỷ qua, các mô hình thống kê truyền thống như tự hồi quy (Autoregressive – AR), trung bình trượt (Moving Average – MA), tự hồi quy và trung bình trượt (ARMA), hay mô hình trung bình trượt kết hợp tự hồi quy (ARIMA) được sử dụng phổ biến và rộng rãi cho dự đoán chuỗi thời gian. Tuy nhiên, các mô hình tuyến tính có những hạn chế khi không thể áp dụng đối với dữ liệu có tính phi tuyến.

Để khắc phục nhược điểm trên, một hướng tiếp cận khác là sử dụng mạng nơ-ron nhân tạo (Artificial neural networks – ANNs) bởi đặc tính: thích nghi nhanh, phi tuyến và khả năng xấp xỉ hàm tùy ý [19]. Mạng nơ-ron truyền thẳng (feed-forward neural network) sử dụng giải thuật lan truyền ngược sai số, back-propagation neural network (BPNN), là một kiểu mạng phổ biến nhất sử dụng để dự đoán chuỗi thời gian [19]. Tuy nhiên, BPNN có nhược điểm là dễ tìm phải điểm cực trị địa phương, thời gian hội tụ lớn và phụ thuộc nhiều vào điểm khởi tạo ban đầu. Nhược điểm này được khắc phục bằng cách kết hợp giải thuật di chuyền và lan truyền ngược sai số để huấn luyện mạng, được đề xuất trong các công trình [17], [4]. Giải thuật này được gọi là giải thuật GA-BP.

Khi áp dụng cho bài toán dự đoán chuỗi thời gian, các đầu vào của mạng nơ-ron thường là p giá trị gần nhất của chuỗi thời gian (y_t, y_{t-1}… y_{t-p}) và đầu ra là giá trị trong tương lai y_{t+k}. Tuy nhiên trong chuỗi thời gian có chu kỳ, bởi tính lặp lại của dữ liệu, giá trị trong tương lai có mối quan hệ khá rõ với các giá trị tương ứng tại các chu kỳ trước đó. Các tác giả trong nghiên cứu [9], [10] đề xuất kiến trúc mạng nơ-ron áp dụng cho chuỗi thời gian theo mùa (seasonal time series), với đầu vào bao gồm các giá trị gần nhất y_t, y_{t-1},… và các giá trị tương ứng trong các mùa trước đó y_{t-s}, y_{t-2s},… (s là thời gian theo mùa). Mặt khác, mô hình mạng nơ-ron theo mùa (Seasonal Artificial Neural Network - SANN) được đề xuất trong nghiên cưu [5], sử dụng các giá trị trong mùa thứ i là đầu vào của mạng, và các giá trị trong mùa thứ (i+1) tiếp theo là đầu ra của mạng. Tuy nhiên các mô hình trên được sử dụng với giả thiết chu kỳ của dữ liệu đã biết, trong khi thông số này thường là ẩn trong các chuỗi thời gian thực tế.

Tính chu kỳ của chuỗi thời gian đã được nghiên cứu trong nhiều công trình khoa học với các giải thuật phát hiện và đánh giá chu kỳ khác nhau. Có thể chia làm 2 nhóm như sau: phương pháp trên miền thời gian dựa vào tự tương quan (autocorrelation) và phương pháp trên miền tần số dựa vào chu kỳ đồ (periodogram) [3]. Tuy nhiên mỗi hướng tiếp cận đều có các nhược điểm riêng: sử dụng tự tương quan khó tìm ra chu kỳ hơn, trong khi các chu kỳ đánh giá bởi chu kỳ đồ có thể không đúng. Phương pháp AUTOPERIOD đề xuất trong [18] đã giải quyết được nhược điểm trên bằng cách sử dụng thông tin từ cả tự tương quan và chu kỳ đồ của chuỗi thời gian để đưa ra đánh giá chu kỳ chính xác. Từ các phân tích trên, có thể thấy sự kết hợp giữa AUTOPERIOD và ANN có nhiều tiềm năng để nâng cao độ chính xác và hình thành mô hình dự đoán mới.

Trong lĩnh vực điện toán đám mây, phương pháp dự đoán chuỗi thời gian được sử dụng nhiều trong việc dự đoán tải công việc hay tiêu dùng tài nguyên. Các tác giả trong công trình [7] sử dụng mô hình dự đoán tài nguyên dựa trên phương pháp liên tiến lũy thừa kép (Double Exponential Smoothing). Mặt khác, [8] sử dụng phương pháp mạng nơ-ron với giải thuật lan truyền ngược sai số và phương pháp hồi quy tuyến tính. Tác giả của [14] sử dụng phương pháp trung bình trượt tự hồi quy bậc hai (second order autoregressive moving average – ARMA) để dự đoán tải công việc. Trong công trình [16], một vài phương pháp bao gồm R, MA, exponential smoothing, ETS, Automated ARIMA và BPNN được sử dụng để dự đoán cho cùng một tải công việc trong môi trường đám mây thực tế. Các kết quả thu được cho thấy mạng nơ-ron nhân tạo dự đoán với độ chính xác cao hơn hầu hết các phương pháp khác.

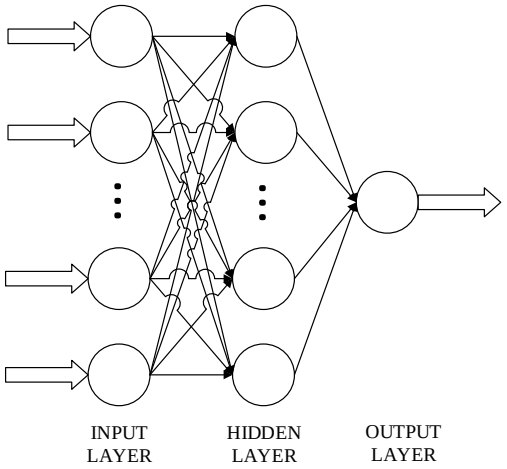
Tuy vậy, tất cả các phương pháp sử dụng để dự đoán tiêu dùng tài nguyên của ứng dụng trong môi trường đám mây đều tồn tại nhược điểm như đã trình bày phần trước, bao gồm hạn

chế của mô hình tuyến tính, vấn đề cực trị địa phương của BPNN. Hơn nữa, rất nhiều các dữ liệu tải công việc (workload) của ứng dụng trên Internet đã được thu thập và phân tích nhằm chỉ rõ tính chu kì của chúng [1]. Vì vậy, đóng góp chính trong nghiên cứu của chúng tôi là mô hình kết hợp giữa giải thuật đánh giá chu kỳ và mạng nơ-ron nhân tạo được huấn luyện bằng giải thuật GA-BP để dự đoán chuỗi thời gian. Chúng tôi trước tiên sử dụng phương pháp AUTOPERIOD để đánh giá chu kỳ của dữ liệu. Sau đó mạng nơ-ron và giải thuật GA-BP với đầu vào xác định bởi các chu kỳ này được dùng để dự đoán và nâng cao độ chính xác cũng như tốc độ hội tụ. Mô hình của chúng tôi được thử nghiệm trên bộ dữ liệu thực tế của một mô hình SaaS điển hình. Kết quả thu được đã chứng minh hiệu quả của mô hình này trong bài toán dự đoán.

3. MẠNG NƠ-RON NHÂN TẠO SỬ DỤNG GIẢI THUẬT DI CHUYỀN – LAN TRUYỀN NGƯỢC SAI SÔ (GA-BPNN)

Mô hình của chúng tôi bao gồm 2 phần: phần thứ nhất là phương pháp GA-BPNN sử dụng mạng nơ-ron để dự đoán không xem xét đến chu kỳ của chuỗi thời gian; phần thứ hai là phương pháp PD-GABP dựa trên sự kết hợp giữa giải thuật đánh giá chu kỳ với GA-BPNN đê tăng hiệu quả của phương pháp này. Tuy nhiên, cả hai phương pháp GA-BPNN và PD-GABP là hoàn toàn giống nhau nếu cùng áp dụng chúng cho chuỗi thời gian không có tính chu kỳ.

Phương pháp GA-BPNN sẽ được trình bày trong phần này. Trong phương pháp GA-BPNN, mạng nơ-ron trước tiên được huấn luyện bởi giải thuật GA-BP để tìm ra mối quan hệ giữa các giá trị đã biết trong quá khứ với giá trị trong tương lai. Mạng nơ-ron thu được sẽ được sử dụng để dự báo trong các thời điểm tiếp theo của chuỗi thời gian.



Hình 1: Mạng nơ-ron truyền thẳng nhiều lớp - multi-layer perceptron (MLP)

3.1. Artificial neural network (ANN)

Mạng nơ-ron truyền thẳng nhiêu lớp (multi-layer perceptron - MLP) với 3 lớp (hình 1) là một mạng nơ-ron điên hình sử dụng trong các bài toán dự báo. Ba lớp của mạng bao gồm: một lớp đầu vào (I), một lớp ẩn (H) và một lớp đầu ra (O). Các nơ-ron trong một lớp được liên kết với toàn bộ các nơ-ron trong lớp kế tiếp. Mỗi liên kết đều được gắn với một trọng số, các trọng số này sẽ được điều chỉnh trong suốt quá trình huấn luyện mạng. học. Hàm chuyển đổi phi tuyến được sử dụng trong bài báo này là hàm sigmoid $f(x) = \left(1 + e^{-x}\right)^{-1}$

Đối với dự đoán chuỗi thời gian k bước (k-step-ahead forecasting), đầu vào của mạng nơ-ron là *p* giá trị liên tiếp gần nhất của chuỗi thời gian *y(t), y(t − 1),..., y(t − p)* được gọi là *cửa sổ trượt,* và đầu ra là giá trị *y(t+k).* Vì vậy, mạng nơ-ron tương đương với phép ánh xạ phi tuyến:

$$y(t+k) = f\left(y(t), y(t-1), \ldots, y(t-p)\right)$$

3.2. Giải thuật GA-BP – một kết hợp của giải thuật di chuyền và lan truyền ngược sai số

Giải thuật lan truyền ngược sai số, được đề xuất trong bài báo [15], là phương pháp được sử dụng phổ biến nhất để huấn luyện mạng nơ-ron. Giải thuật này thực chất là một phương pháp xuống đồi theo hướng đạo hàm (gradient descent), trong đó các trọng số được điều chỉnh để giảm tối thiểu sai số ở đầu ra. Hàm lỗi được định nghĩa như sau:

$$E = \frac{1}{2}\sum_{k=1}^{m}\left(d_k - y_k\right)^2$$

Trong đó d_k và y_k lần lượt là giá trị đầu ra của mạng và giá trị mong muốn, M là số lượng nơ-ron ở lớp đầu ra (output layer).

Giải thuật di chuyền (Genetic algorithm - GA) là một phương pháp áp dụng cho bài toán tìm kiếm và tối ưu dựa trên quá trình tiến hóa tự nhiên. Montana và Davis trong nghiên cứu [11] đã áp dụng giải thuật này cho quá trình huấn luyện mạng nơ-ron với không gian tìm kiếm được hình thành bằng tập các trọng số của mạng.

Như đã trình bày trong phần trước, cả giải thuật di chuyền và lan truyền ngược sai số đều có những nhược điểm riêng. Khi độ phức tạp của bài toán tăng lên, hiệu quả của giải thuật lan truyền ngược sai số giảm đáng kể vì lời giải tìm được có xu hướng rơi vào cực địa phương. Mặt khác, giải thuật di chuyền có thể tìm kiếm được các lời giải xung quanh vùng cực trị toàn thể nhưng khá chậm để tiến đến lời giải chính xác. Do đó, giải thuật kết hợp GA-BP sẽ có hiệu quả hơn vì nó kế thừa ưu điểm từ cả hai giải thuật trên. Giải thuật GA-BP bao gồm hai giai đoạn. Trước tiên, giải thuật di chuyền được sử dụng để tìm ra bộ trọng số của mạng nơ-ron gần với điểm cực trị toàn thể. Sau đó, bộ trọng số này được sử dụng làm điểm khởi tạo cho giải thuật lan truyền ngược sai số để tìm ra lời giải tối ưu cuối cùng. Theo hướng tiếp cận này, cả hai vấn đề về cực trị địa phương và tốc độ hội tụ sẽ được giải quyết. Giải thuật GA-BP có thể được trình bày ngắn gọn qua các bước như sau:

- Bước 1: Tạo ngẫu nhiên quần thể ban đầu của các chuỗi nhiễm sắc thể
- Bước 2: Tính các trọng số tương ứng cho mạng nơ-ron từ mỗi cá thể.
- Bước 3: Xác định giá trị hàm mục tiêu của từng chuỗi nhiễm sắc thể
- Bước 4: Tái tạo quần thể mới từ quần thể hiện tại sử dụng các toán tử: chọn lọc (selection), lai ghép (Crossover) và đột biến (Mutation)
- Bước 5: Lặp lại bước 3 đến bước 5 cho đến khi thỏa mãn điều kiện dừng

- Bước 6: Lựa chọn cá thể có độ thích nghi lớn nhất trong quần thể.
- Bước 7: Sử dụng giải thuật lan truyền ngược sai số để tìm lời giải tối ưu cuối cùng.

Các thành phần của giải thuật di chuyền được lựa chọn như sau:

- Mã hóa nhiễm sắc thể: Bộ trọng số của mạng nơ-ron được mã hóa thành chuỗi số thực
- Hàm thích nghi: $F = \frac{1}{RMSE} = \frac{1}{\sqrt{\frac{1}{n}\sum_{k=1}^{n}(d_k - y_k)^2}}$.

  Trong đó, d_k và y_k lần lượt là giá trị đầu ra của mạng và giá trị mong muốn, n là kích thước của tập học
- Khởi tạo: Quần thể ban đầu được khởi tạo bằng cách lựa chọn ngẫu nhiên các trọng số theo phân phối đều trong khoảng $\left(\sqrt{6}(n_{layer\ A} + n_{layer\ B}); -\sqrt{6}(n_{layer\ A} + n_{layer\ B})\right)$, trong đó trong đó $n_{layer\ A}$, $n_{layer\ B}$ là số nơ-ron trong 2 lớp mà trọng số đó kết nôi
- Chọn lọc: Các cá thể được chọn lọc bằng cách quay bánh xe roulette (Roulette wheel selection). Xác suất lựa chọn của một cá thể được cho bởi công thức:

  $p_i = f_i / \sum_{i=1}^{s} f_i$, trong đó s là kích thước quần thể, f_i là độ thích nghi của cá thể
- Toán tử di chuyền: lai ghép trọng số (crossover - weights**)** và đột biến gaussian (Gaussian mutation).

4. KẾT HỢP GA-BPNN VỚI GIẢI THUẬT ĐÁNH GIÁ CHU KỲ

Trong phần này, chúng tôi trình bày phương pháp thứ hai PD-GABP: kết hợp GA-BPNN với giải thuật đánh giá chu kỳ. Hướng tiếp cận này khai thác mối quan hệ giữa giá trị trong tương lai và các giá trị tương ứng trong các chu kỳ trước khi dữ liệu chuỗi thời gian có tính chu kỳ, để nâng cao độ chính xác cho dự đoán. Có thể coi phương pháp này là một biến thể của GA-BPNN với đầu vào của mạng bao gồm các giá trị trong chu kỳ trước (y_{t-T} , y_{t-2T}, ...) ngoài *p* giá trị gần nhất (y_t, y_{t-1}… y_{t-p}) của chuỗi thời gian.

Giải thuật PD-GABP bao gồm 2 bước: bước thứ nhất là phát hiện và đánh giá chu kỳ của chuỗi thời gian; bước thứ hai sử dụng chu kỳ này để xác định các đầu vào cho mạng nơ-ron (bao gồm các giá trị gần nhất và các giá trị trong chu kỳ trước), sau đó mạng được huấn luyện bằng giải thuật GA-BP.

4.1 Giải thuật phát hiện và đánh giá chu kỳ chuỗi thời gian

Phương pháp AUTOPERIOD được đề xuất trong [18] được sử dụng để đánh giá chu kỳ lặp của dữ liệu. Phương pháp này bao gồm 2 bước. Trước tiên, chu kỳ đồ (periodogram) được sử dụng để ước lượng các giá trị có khả năng là chu kỳ (được gọi là "hint"). Chu kỳ đồ có thể được tính thông qua DFT (biến đổi fourier rời rạc) của một chuỗi như sau:

$$P(f_{k/N}) = \left\| X(f_{k/N}) \right\|^2, k = 0,1,\ldots,\left\lceil (N-1)/2 \right\rceil$$

Trong đó, *P* là chu kỳ đồ, X là biến đổi Fourier rời rạc của chuỗi x(n), n = 0, 1,… N − 1.

Sau đó, vì các dự đoán "hint" có thể không chính xác, các giá trị này sẽ được kiểm chứng sử dụng hàm tự tương quan vòng (circular auto-correlation function - ACF). Các giá trị nằm trên "đồi" (hill) của ACF được xác thực là hợp lệ, và được loại bỏ nếu thuộc các trường hợp khác. Các "hint" hợp lệ sẽ tiếp tục được hiệu chỉnh để tiến tới điểm cực trị gần nhất. Hàm tự tương quan vòng ACF của chuỗi x(n) được cho bởi công thức:

$$ACF(\tau) = \frac{1}{N}\sum_{n=0}^{N-1} x(\tau).x(n+\tau), \tau = 0,1,\ldots,N-1$$

Độ phức tạp tính toán của chu kỳ đồ và ACF sử dụng DFT là *O(NlogN)*. Kết quả của bước đánh giá chu kỳ này là toàn bộ các chu kỳ của chuỗi thời gian. Nếu không tồn tại giá trị hợp lệ, chuỗi thời gian là không có chu kỳ

4.2 Giải thuật PD-GABP

Trong trường hợp tập dữ liệu có chu kỳ, giả sử T_1, T_2,…, T_r là tập giá trị xác định trong bước xác định chu kỳ của dữ liệu, ta định nghĩa véc-tơ đầu vào gồm p phần tử liên tục trong quá khứ *y(t), y(t−1),..., y(t−p)* – được gọi là cửa sổ trượt *(sliding window)*, và các giá trị trong *m* chu kỳ trước *y(t+k-T_1), y(t+k-2T_1),…,y(t+k-mT_1), y(t+k-T_2), y(t+k-2T_2),…, y(t+k-mT_2),…, y(t+k-T_r), y(t+k-2T_r),..., y(t+k-mT_r)* – được gọi là các đầu vào theo chu kỳ (*periodic inputs*). Vì vậy, mạng nơ-ron tương đương với phép ánh xạ phi tuyến sau:

$$\begin{aligned} y(t+k) = f(&y(t), y(t-1),\ldots, y(t-p), (t+k-T_1),\\ &y(t+k-2T_1),\ldots y(t+k-mT_1),\ y(t+k-T_2),\\ &y(t+k-2T_2),\ldots, y(t+k-mT_2),\ldots,\\ &y(t+k-T_r),\ y(t+k-2T_r),\ldots y(t+k-mT_r)) \end{aligned}$$

Sau đây là giải thuật PD-GABP:

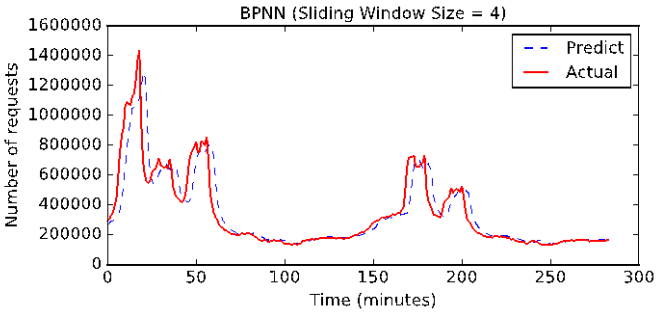
Thuật toán 1. PD-GABP

1: Đánh giá các chu kỳ $T_1, T_2, \ldots, T_r$ của chuỗi thời gian bằng phương pháp AUTOPERIOD
2: **if** $r > 0$ (chuỗi thời gian có chu kỳ) **then**
4: Thiết lập các đầu vào cho mạng nơ-ron là: y(t), y(t−1),…, y(t−p), y(t+k-T_1), y(t+k-2T_1),… y(t+k-mT_1), y(t+k-T_2), y(t+k-2T_2),… y(t+k-mT_2),.., y(t+k-T_r), y(t+k-2T_r),… y(t+k-mT_r)
5: **else**
6: Thiết lập các đầu vào cho mạng nơ-ron là: y(t), y(t−1),…, y(t−p)
7: Thiết lập đầu ra cho mạng nơ-ron: y(t+k)
8: Huấn luyện mạng bằng giải thuật GA-BP

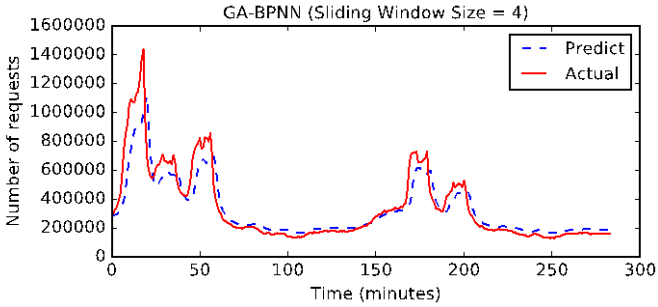
5. THỬ NGHIỆM VÀ ĐÁNH GIÁ

5.1 Thiết lập môi trường thử nghiệm

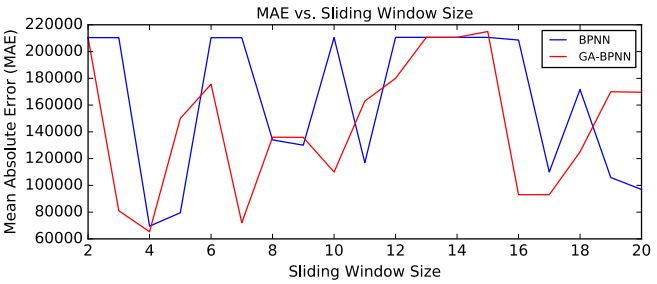
Mô hình dự đoán sử dụng tập dữ liệu workload của trang web WorldCup 1998 [2]. Tập dữ liệu thời gian được thu thập dưới dạng số lượng các yêu cầu gửi tới trong từng khoảng thời gian 10 phút. Chúng tôi sử dụng dữ liệu huấn luyện từ ngày 40 tới 46. Mô hình sẽ học và dự đoán số lượng workload của 10 phút tiếp theo, dựa trên các dữ liệu quá khứ. Mạng nơ ron được cấu tạo 3 lớp: lớp đầu vào, lớp ẩn với 15 nút và lớp đầu ra chỉ có 1 nút. Số lượng nút đầu vào bằng với kích thước của vector đầu vào. Tham số sử dụng cho mạng nơ ron và thuật toán di truyền: số lượng quần thể Psize=225, tỉ lệ lai ghép PC=0.9, tỉ lệ đột biến PM=0.01, hệ số học η=0.000001.



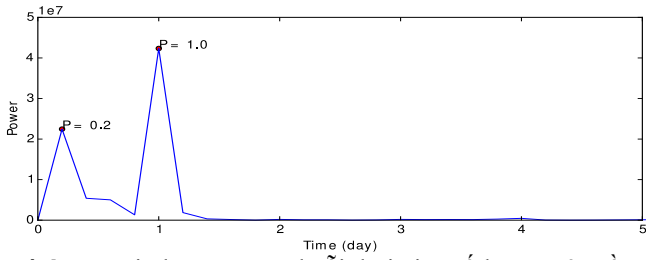
Hình 2. Kết quả dự đoán của mô hình BPNN với p =4



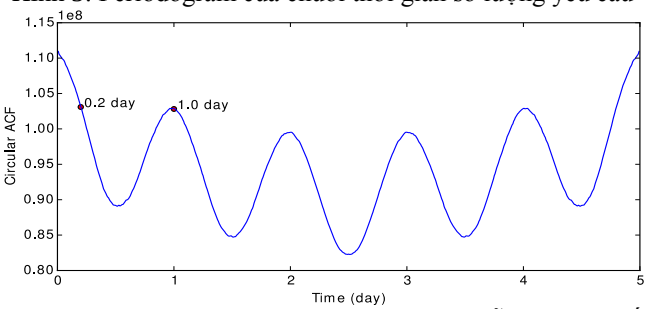
Hình 3. Kết quả dự đoán của mô hình GA-BPNN với p =4



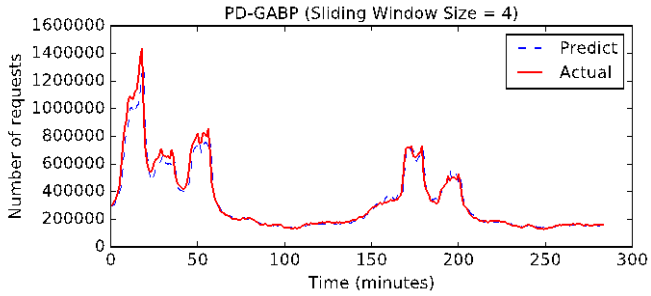
Hình 4: So sánh MAE giữa GA-BPNN and BPNN theo kích thước cửa sổ trượt



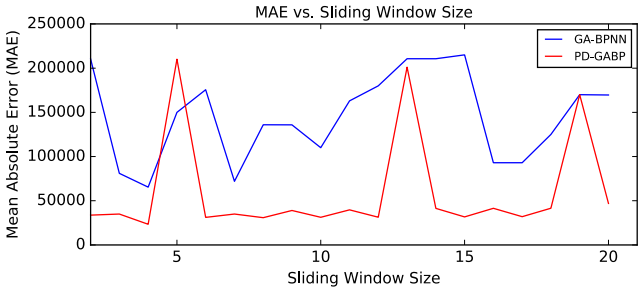
Hình 5: Periodogram của chuỗi thời gian số lượng yêu cầu



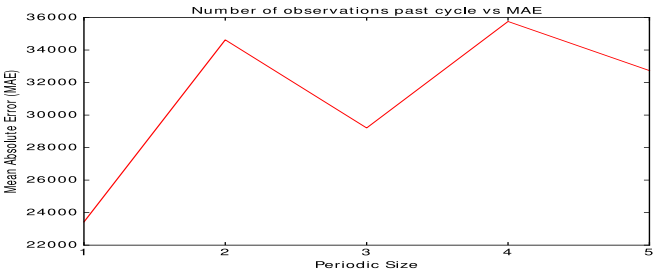
Hình 6: Hàm tương quan vòng (ACF) của chuỗi thời gian số lượng yêu cầu



Hình 7. Kết quả dự đoán của mô hình PD-GABP với p = 4, m = 1



Hình 8: So sánh MAE giữa GA-BPNN và PD-GABP theo kích thước cửa sổ trượt



Hình 9: Ảnh hưởng của số chu kỳ sử dụng tới MAE trong mô hình PD-GABP

Dữ liệu đầu vào được chuẩn hóa trước khi cho vào mô hình học theo công thức sau:

$$x_i = \frac{x_i - x_{\min}}{x_{\max} - x_{min}}$$

Trong đó x_i và x_i tương ứng dữ liệu gốc và dữ liệu chuẩn hóa.

Độ chính xác của mô hình được đánh giá bởi các độ đo sau:

- Root mean square error (RMSE) [18,21]:

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

- Mean absolute percentage error (MAPE) [18]:

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

- Mean absolute error (MAE) [21]:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - y_i\right)$$

5.2 Phương pháp GA-BPNN

Với thử nghiệm đầu tiên, chúng tôi tiến hành để kiểm chứng tính hiệu quả của mô hình GA-BPNN trong việc dự đoán. Mạng nơ ron được sử dụng để dự đoán số lượng yêu cầu tới trong 10 phút tiếp theo. Chúng tôi đánh giá sự chính xác của mô hình GA-BPNN với BPNN dựa vào các độ đo RMSE, MAE và MAPE. Bảng so sánh được biểu diễn ở Bảng I:

	BPNN		
	p = 2	***p = 4***	***p = 6***
RMSE	328554.33	120275.67	328554.33
MAE	210424.84	69521.13	210425.17
MAPE	1.65	0.13	1.63
	GA BPNN		
	p = 2	***p = 4***	***p = 6***
RMSE	328552.29	104314.704	297306.04
MAE	210383.97	65649.54	175616.01
MAPE	1.63	0.12	0.98
	PD-GABP		
	p = 2	***p = 4***	***p = 6***
RMSE	70656.17	47109.41	60548.61
MAE	33742.82	23425.64	31186.69
MAPE	0.07	0.06	0.07

Bảng I: So sánh độ chính xác dự đoán giữa các mô hình BPNN, GA-BPNN và PD-GABP với kích thước cửa sổ trượt khác nhau

Hình 2 và Hình 3 thể hiện kết quả dự đoán của 2 mô hình BPNN và GA-BPNN. Kết quả thực nghiệm cho thấy dự đoán của GA-BPNN sát với xu hướng dữ liệu thực tế hơn. Cụ thể, từ bảng I, sai số RMSE của GA-BPNN nhỏ hơn so với BPNN với các cửa sộ trượt khác nhau, sai số MAE của GA-BPNN nhỏ hơn BPNN và sai số MAPE của mô hình GA-BPNN thấp hơn BPNN. Điều này chứng tỏ mô hình GA-BPNN thể hiện được sự hiệu quả hơn so với mô hình BPNN.

Các kích thước cửa số trượt khác nhau cũng sẽ ảnh hưởng tới kết quả dự đoán của mô hình. Hình 4 so sánh độ chính xác với các kích thước cửa sổ trượt khác nhau dựa trên sai số MAE. Nhìn một cách tổng quát, số liệu của GA-BPNN nhỏ hơn so với BPNN. Trong mô hình GA-BPNN, tại kích thước cửa sổ bằng 4, giá trị lỗi thấp nhất bằng 65649.54. Trong khi đó BPNN chỉ đạt giá trị 69521.13 với cùng kích thước cửa sổ.

5.3 Phương pháp PD-GABPNN

Trong thử nghiệm này, mô hình đề xuất PD-GABP được sử dụng trong bài toán dự đoán và so sánh với mô hình GA-BPNN.

Hình 5 và hình 6 thể hiện kết quả của phương pháp phát hiện chu kì. Hai giá trị có thể của chu kì được phát hiện trong hình 5 là $P_1=0.2$ và $P_2=1.0$. Các giá trị có thể này sẽ được kiểm chứng lại sử dụng hàm ACF vòng. Chỉ có giá trị P_2 nằm trên "đồi" của ACF là hợp lệ, trong khi đó P_1 bị loại bỏ. Kết quả cuối cùng của thuật toán phát hiện chu kì cho thấy chuỗi thời gian số lượng yêu cầu chỉ có chu kì $P = 1$ ngày.

Với chu kì 1 ngày, vectơ đầu vào cho mạng GA-BPNN được xác định như sau: (y(t), y(t-1), …, y(t-p), (t+k-T), (t+k-2T), …, (t+k-mT)), ở đó T = 144 là số các quan sát trong 1 chu kì (1 ngày), p là kích thước cửa sổ trượt và m là số lượng các đầu vào theo chu kì. Hình 7 thể hiện kết quả dự đoán của phương pháp PD-GABPNN trong trường hợp p = 4 và m = 1. Dễ dàng nhận thấy kết quả dự đoán gần giống với đường thực tế. Độ chính xác của mô hình được tổng hợp ở bảng I, khi mà các số liệu tốt nhất đều thuộc về mô hình PD-GABPNN. RMSE của PD-GABP thấp hơn tới 60% so với GA-BPNN. Đặc biệt tại kích thước cửa sổ bằng 6, sai số RMSE giảm 79%. Sai số MAE của PD-GABPNN nhỏ hơn 60% so với mô hình GA-BPNN. Tương tự, sai số MAPE của PD-GABPNN cũng nhỏ hơn so với mô hình còn lại, thể hiện sự hiệu quả của mô hình trong bài toán dự đoán.

Hơn nữa, kết quả dự đoán bị ảnh hưởng bởi các kích thước cửa sổ khác nhau. Hình 8 minh họa giá trị MAE của hai mô hình với các cửa sổ trượt khác nhau. Khi kích thước cửa sổ tăng từ 6 lên tới 12, sai số MAE của mô hình PD-GABPNN hoàn toàn nhỏ hơn mô hình GA-BPNN.

Hình 9 thể hiện sự thay đổi sai số MAE khi mà p = 4 và m tăng từ 1 tới 5. Rõ ràng rằng sai số MAE đều thấp hơn 36000 ngoại trừ trường hợp m = 3. Giá trị MAE tối ưu đạt được tại giá trị m = 1. Hiển nhiên, việc chọn 1 giá trị từ chu kì trong quá khứ kết hợp các giá trị gần nhất làm giá trị đầu vào đem lại kết quả dự đoán tốt nhất.

6. KẾT LUẬN

Trong nghiên cứu này, chúng tôi đã đề xuất một mô hình mới trong vấn đề phân tích chuỗi thời gian. Mô hình là sự kết hợp giữa phát hiện chu kì và mạng nơ ron huấn luyện bởi thuật toán di truyền – lan truyền ngược. Mô hình được kiểm chứng trong việc dự đoán số lượng yêu cầu gửi tới hệ thống World Cup 1998. Các kết quả thu được đều được cải thiện so với các mô hình đã có. Kết quả dự đoán sẽ giúp cho hệ thống chủ động mở rộng trước thay vì phải theo dõi tài nguyên và mở rộng theo ngưỡng như các giải pháp co giãn tài nguyên hiện nay của đám mây. Trong tương lai, chúng tôi có kế hoạch tích hợp mô hình dự đoán vào trong một thành phần của một hệ thống điện toán đám mây như OpenStack để tối ưu vấn đề phân phối tài nguyên. Thêm vào đó, chúng tôi cũng sẽ áp dụng mô hình này với nhiều tham số khác của máy ảo như CPU, RAM, tốc độ đọc/ghi ổ đĩa, băng thông,...Dựa vào đó hệ thống có thể tự động ra quyết định co giãn tài nguyên một cách chính xác hơn.

7. LỜI TRI ÂN

Trong công trình nghiên cứu này, chúng em xin cảm ơn sự hướng dẫn tận tình của TS. Nguyễn Bình Minh, cũng như sự giúp đỡ từ tập thể các thầy cô giáo ở Viện Công nghệ thông tin và Truyền thông. Chúng em cũng xin cảm ơn ban tổ chức cuộc thi SVNCKH đã giúp chúng em có một môi trường để làm quen với công việc nghiên cứu khoa học.

8. TÀI LIỆU THAM KHẢO

[1] Ali-Eldin, A., Tordsson, J., Elmroth, E., Kihl, M.: Workload classification for efficient auto-scaling of cloud resources. Tech. rep.(2013)

[2] Arlitt, M., Jin, T.: 1998 World Cup Web Site Access Logs, http://ita.ee.lbl.gov/html/contrib/WorldCup.html, accessed: 2016-2-19
[3] Berberidis, C., Aref, W.G., Atallah, M., Vlahavas, I., Elmagarmid, A.K., et al.: Multiple and partial periodicity mining in time series databases. In: ECAI. vol. 2, pp. 370–374 (2002)
[4] Ding, S., Su, C., Yu, J.: An optimizing BP neural network algorithm based on genetic algorithm. Artificial Intelligence Review 36(2), 153–162 (2011)
[5] Hamza¸cebi, C.: Improving artificial neural networks performance in seasonal time series forecasting. Information Sciences 178(23), 4550–4559 (2008)
[6] Hipel, K.W., McLeod, A.I.: Time series modelling of water resources and environmental systems, vol. 45. Elsevier (1994)
[7] Huang, J., Li, C., Yu, J.: Resource prediction based on double exponential smoothing in cloud computing. In: Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on. pp. 2056–2060. IEEE (2012)
[8] Islam, S., Keung, J., Lee, K., Liu, A.: Empirical prediction models for adaptive resource provisioning in the cloud. Future Generation Computer Systems 28(1), 155–162 (2012)
[9] Julian Faraway, C.C.: Time series forecasting with neural networks: A comparative study using the airline data. Journal of the Royal Statistical Society. Series C (Applied Statistics) 47(2), 231–250 (1998)
[10] Kihoro, J., Otieno, R., Wafula, C.: Seasonal time series forecasting: A comparative study of ARIMA and ANN models. AJST 5(2) (2004)
[11] Montana, D.J., Davis, L.: Training feedforward neural networks using genetic algorithms. In: IJCAI. vol. 89, pp. 762–767 (1989)
[12] Nguyen, B.M., Tran, D., Nguyen, Q.: A Strategy for Server Management to Improve Cloud Service QoS. In: Distributed Simulation and Real Time Applications (DS-RT), 2015 IEEE/ACM 19th International Symposium on. pp. 120–127 (Oct 2015)
[13] Nguyen, M.B., Tran, V., Hluchy, L.: A generic development and deployment framework for cloud computing and distributed applications. Computing and Informatics 32(3), 461–485 (2013)
[14] Roy, N., Dubey, A., Gokhale, A.: Efficient autoscaling in the cloud using predictive models for workload forecasting. In: Cloud Computing (CLOUD), 2011 IEEE International Conference on. pp. 500–507. IEEE (2011)
[15] Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error-propagation. In: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, pp. 318–362. MIT Press, Cambridge, MA (1986)
[16] Vazquez, C., Krishnan, R., John, E.: 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)
[17] Venkatesan, D., Kannan, K., Saravanan, R.: A genetic algorithm-based artificial neural network model for the optimization of machining processes. Neural Computing and Applications 18(2), 135–140 (2009)
[18] Vlachos, M., Philip, S.Y., Castelli, V.: On periodicity detection and structural periodic similarity. In: SDM. vol. 5, pp. 449–460. SIAM (2005)
[19] Zhang, G., Patuwo, B.E., Hu, M.Y.: Forecasting with artificial neural networks: The state of the art. International journal of forecasting 14(1), 35–62 (1998).