

THUẬT TOÁN ĐỘNG ĐỂ LỰA CHỌN TÁC VỤ TRONG HỆ THỐNG IOTS DYNAMIC ALGORIZATION FOR SELECTING TASKS IN IOTS SYSTEM

Trần Công Thịnh, Võ Minh Huân

Trường Đại học Sư phạm Kỹ thuật TP.Hồ Chí Minh

Ngày toà soạn nhận bài 28/8/2020, ngày phản biện đánh giá 5/9/2020, ngày chấp nhận đăng 26/1/2021

TÓM TẮT

Kéo dài thời lượng pin và tăng cường chất lượng thông tin thu thập của mạng cảm biến không dây sử dụng nguồn năng lượng tái tạo là một trong những vấn đề của hệ thống Internet of Things (IoTs). Để cải thiện vấn đề trên, chúng tôi đề xuất một thuật toán sử dụng mạng Neural với mục đích dự đoán và chọn lựa tác vụ của nút cảm biến. Thuật toán có thể ứng dụng trên vi điều khiển với khả năng xử lý và lưu trữ hạn chế. Khác với các phương pháp trước đây chủ yếu tập trung vào chế độ ngủ sâu hay lập ra lịch trình cho các tác vụ, thuật toán chúng tôi đề xuất giúp nút cảm biến có tính thích nghi cao hơn với môi trường dựa trên các số liệu thời gian thực tại mỗi nút. Khi có sự chênh lệch số liệu lớn, lượng thông tin thu thập về sẽ tăng lên và khi không có sự thay đổi đáng kể, nút cảm biến được đưa về tác vụ tiêu tốn năng lượng thấp để đảm bảo được thời lượng pin. Các kết quả từ nghiên cứu cho thấy thuật toán thu thập được số lượng thông tin có ích nhiều hơn, thời lượng và tuổi thọ của pin cũng được cải thiện.

Từ khóa: Mạng neural; IoTs; Mạng cảm biến không dây; Tuổi thọ pin; Chất lượng thông tin.

ABSTRACT

Extending the lifespan and improving the quality of information collected by wireless sensor networks using renewable energy is one of the problems of the Internet of Things (IoT) system. To improve the above problem, we propose an algorithm using a Neural network for the purpose of predicting and selecting sensor nodes and can be applied on microcontrollers with low processing and storage capacity. Unlike previous methods that mainly focus on scheduled or deep sleep tasks, the algorithm we recommend makes the sensor node more adaptable to the environment based on the metrics. Real-time at each node. When there is a large data discrepancy, the amount of collected information is increased and when there is no significant change, the sensor node is put into low power duty to ensure battery life. Results from the study show that the algorithm gathers more useful information and battery life is also improved.

Keywords: Neural Network; IoTs; Wireless Sensor Network; battery-life; quality of service.

1. GIỚI THIỆU

Trong các ứng dụng của Internet of Things (IoTs), việc cung cấp các thiết bị thu thập dữ liệu sử dụng mạng cảm biến không dây đóng vai trò quan trọng để phục vụ cho quan sát, phân tích và xử lý thông tin. Đa phần các thiết bị của mạng cảm biến không dây ứng dụng trên vi điều khiển có khả năng tính toán, lưu trữ hạn chế và năng lượng sử dụng của các thiết bị này phần lớn từ nguồn năng lượng tái tạo mặt trời [1], [2]. Hoạt

động của một nút cảm biến trong mạng được chia làm 3 phần: cảm biến, xử lý và truyền dữ liệu. Trong ba nhiệm vụ trên, truyền dữ liệu là hoạt động tiêu tốn nhiều năng lượng nhất (khoảng 80% năng lượng của nút) [3]. Việc kéo dài tuổi thọ hoạt động của mạng cảm biến nhưng vẫn nâng cao chất lượng của thông tin mà nút cảm biến thu thập được là vấn đề quan trọng cần được tập trung nâng cao. Trong các nghiên cứu gần đây, một số thuật toán đã đề xuất nhằm tối ưu năng lượng sử dụng thuật toán lập lịch trình, quản lý

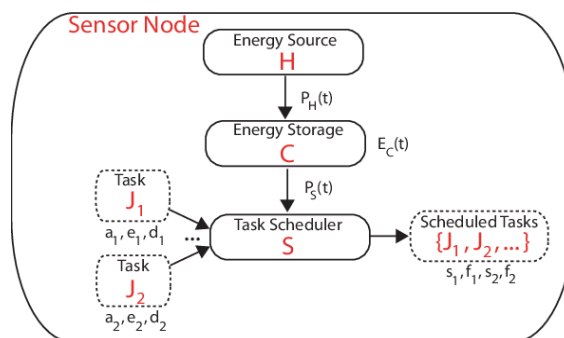
năng lượng tiêu thụ hay tối ưu định tuyến của mạng cảm biến [4]. Trong bài báo này, chúng tôi đề xuất một thuật toán sử dụng mạng neural có thể thực hiện trên vi điều khiển để phân loại và dự đoán nhằm nâng cao chất lượng của thông tin nhưng vẫn kéo dài được thời gian sống của nút.

Trong nghiên cứu của tác giả A. Sinha và Chandrakasan [5], Wang và Xiao [6], tác giả đề xuất phương pháp quản lý điện năng nhưng hầu hết các cơ chế trên đều tận dụng chế độ ngủ sâu để tiết kiệm năng lượng. Tuy nhiên, khi nút đang trong chế độ ngủ sâu thì một lượng thông tin sẽ bị mất và gây ra độ trễ cho mạng. Trong bài báo [6], tác giả L. Wang đề xuất thuật toán quản lý công suất động cho mạng cảm biến với năm chế độ. Các chế độ này được thay đổi lặp lại dựa trên mức năng lượng của pin. Trên thực tế, việc chuyển đổi các trạng thái từ ngủ sâu sang thức hay từ các chế độ nguồn này gây ra hao phí về mặt năng lượng lớn hơn và có độ trễ. Trong nghiên cứu của tác giả X. Fan [7], thuật toán tác giả đề xuất nâng cao việc tính toán các ngưỡng để quyết định nút có ngủ sâu hay không được sử dụng trong bài báo [5] chính xác hơn bằng cách xem xét giữa năng lượng tiêu thụ và năng lượng bổ sung khi đánh thức nút cảm biến trở lại trạng thái hoạt động. Trong bài báo [8], tác giả Sujesha Sudevalayam tối ưu thời gian sống của pin trên nút cảm biến dựa trên nguồn năng lượng nạp được cho pin. Tác giả thay đổi các thông số của nút dựa trên dự đoán mức năng lượng của pin và mức năng lượng hiện tại trong pin nhằm tối ưu thời gian của pin được sạc.

Một số nghiên cứu khác được công bố nhằm tối ưu thời lượng của pin tại nút điển hình là phương pháp giảm tần số hoạt động của vi điều khiển [9] và giảm công suất trên cảm biến. Các cảm biến có độ chính xác cao được giảm hoặc thay thế với cảm biến có độ phân giải, độ chính xác thấp hơn. Phương pháp trên có nhược điểm làm ảnh hưởng không tốt đến chất lượng của thông tin thu thập. Ngoài ra hạ thấp công suất truyền tín hiệu của mạng cảm biến được tác giả G. Amato đề cập trong [10] và thay đổi chu kỳ lấy số liệu của X.Chen trong nghiên cứu [11].

Với những mạng cảm biến có khoảng cách ngắn và vật cản thấp thì phương pháp này tối ưu. Tuy nhiên, đối với những địa điểm có mạng cảm biến rộng và nhiều vật cản việc mất thông tin cũng như chất lượng bị giảm một cách rõ rệt. Các vấn đề nêu trên được nhắc đến trong bài báo nhằm khảo sát về kỹ thuật quản lý điện năng của mạng cảm biến thu thập năng lượng[12].

Ngoài các phương pháp trên, thuật toán lên lịch trình cho các tác vụ cũng được đề cập trong [13]–[18]. Trong bài báo [13], tác giả Soledad Escolar đề xuất thuật toán lập trình nhằm tối ưu mức tổng điện năng tiêu thụ dựa trên tính có ích cũng như năng lượng tiêu thụ trên từng tác vụ. Thuật toán này lên lịch cho các tác vụ, nó sử dụng thông tin dự báo thời tiết có sẵn vào đầu mỗi khoảng thời gian lập lịch (thường là một ngày) và mức pin hiện có lúc lập lịch, từ đó xác định một lịch trình tối ưu. Mục tiêu chính của phương pháp [13] là tìm một lịch trình trung bình về năng lượng, trong một khoảng thời gian dài. Trong bài báo [14], M.Severini cũng trình bày một thuật toán thực hiện thu thập các số liệu một cách trình tự, lập lịch trình như bài báo [13] nhưng sử dụng mô hình LSA (Lazy Scheduling Algorithm). Mô hình LSA (Lazy Scheduling Algorithm) là một thuật toán lập trình để lập lịch trình hoạt động cho nút cảm biến có thể sạc lại và giới hạn nhiệm vụ của một nút cảm biến được nhắc đến trong bài báo [14] và [15]. LSA tạo ra một lịch trình chạy cho các tác vụ chạy song song và sẽ chạy cố định theo lịch trình của tác giả Moser được thực hiện trong bài báo [16].



Hình 1. Thuật toán LSA

Hình 1 mô tả chi tiết về thuật toán LSA lập lịch trình thực hiện tác vụ, khối Task

scheduler sẽ lập ra lịch trình thực hiện của các tác vụ J1, J2 dựa trên thông số energy Storage (C). Ngõ ra của thuật toán là một danh sách các tác vụ sẽ thực hiện tuần tự và sẽ không thay đổi được trong quá trình nút cảm biến chạy. Đây là một điểm chưa tối ưu của thuật toán này. Tuy nhiên, vì LSA giảm thiểu các yếu tố ngõ vào để tăng hiệu suất của thuật toán và là một thuật toán tĩnh nên trong quá trình chạy tác vụ đã được lên lịch trình thì nó không thêm các trạng thái bổ sung ở ngõ vào từ đó không thể hủy ngang lịch trình đã lên sẵn cũng như thêm tác vụ trong lúc thực hiện được đề cập trong bài báo [14], [15], [17]. Trong bài báo [18], tác giả Caruso đề xuất một thuật toán động nhằm lập trình lên lịch trình có thể thay đổi dựa trên thông số ngõ vào là pin và dòng sạc của pin. Vì thuật toán động này có thể thay đổi lịch trình nên nó có tính thích nghi cao hơn so với LSA. Tuy nhiên với một thuật toán động việc quản lý vùng nhớ của thuật toán động rất quan trọng. Nhằm giảm thiểu việc tính toán lặp lại khi sử dụng đệ quy thì các giá trị tính toán trước đó của thuật toán động sẽ được đưa vào một mảng. Khi các nút cảm biến thực hiện tính toán nhiều, bộ nhớ càng tăng lên từ đó không quản lý được trong quá trình thuật toán tính toán. Từ đó có thể phát sinh hiện tượng treo hoặc khởi động lại khiến vi điều khiển phải tính toán lại từ đầu.

Các phương pháp đã nêu ở trên chưa xét đến độ biến thiên của dữ liệu. Đây là một trong những yếu tố quan trọng cho việc đưa ra dự đoán, quyết định chính xác. Trong một số ứng dụng thực tiễn, điển hình như việc nuôi trồng thủy hải sản, các số liệu thay đổi đột ngột có nguy cơ gây hại rất lớn đến kinh tế. Ví dụ như trong mô hình nuôi tôm thẻ [19], nền nhiệt độ và pH của nước cần đạt sự ổn định. Khi dữ liệu có sự thay đổi đột biến của các giá trị trên, người chăn nuôi cần nhiều thông tin hơn nhằm dự đoán và đưa ra quyết định nhanh chóng và chính xác. Trong ứng dụng thu thập dữ liệu trong rừng, chúng ta xét đến sự biến thiên của nhiệt độ có thể dự đoán các vấn đề về cháy rừng và lũ lụt. Vì vậy, độ biến thiên của dữ liệu là thành phần đầu vào quan trọng trong mô hình thu thập dữ liệu.

Để cải thiện những bất cập ở trên, chúng tôi đề xuất một thuật toán sử dụng mạng neural phân loại và dự đoán từ các dữ liệu đầu vào bao gồm lượng pin, dòng sạc và độ biến thiên của dữ liệu, từ đó lựa chọn các tác vụ có mức độ suy hao về mặt năng lượng khác nhau. Mô hình này phù hợp với những đề bài lập trình với thời gian thực theo chu kỳ hoạt động và có thể thực hiện trên vi điều khiển. Chúng tôi thiết kế thuật toán hoạt động linh hoạt hơn với các tác vụ thay đổi. Thay vì lấy mẫu liên tục và gửi theo một chu kỳ, mô hình này phụ thuộc vào dữ liệu ngõ vào của nút để xác định tác vụ từ đó có tính thích nghi cao hơn so với nghiên cứu [13]. Để quyết định những điều trên, chúng tôi dựa trên một mô hình của mạng neural [20], [21]. Thuật toán có thể thực hiện việc tính toán lan truyền thuật trên phần cứng có tốc độ xử lý và bộ nhớ hạn chế như các dòng vi điều khiển 32 bit arm-cortex, mà không cần thực hiện trên phần mềm của máy tính có tốc độ xử lý cao, phương án tính toán không bị lặp lại như khi sử dụng đệ quy và quản lý được vùng nhớ khi sử dụng các bài toán động [18]. Bài báo nghiên cứu xem xét nhiều hơn trong mô hình và tập trung vào vấn đề lập trình lựa chọn tác vụ tối ưu ở từng nút cảm biến tại thời điểm đó. Ngoài ra, thuật toán có thể được thực hiện hiệu quả trong các nút công suất thấp tại điều kiện thực tế.

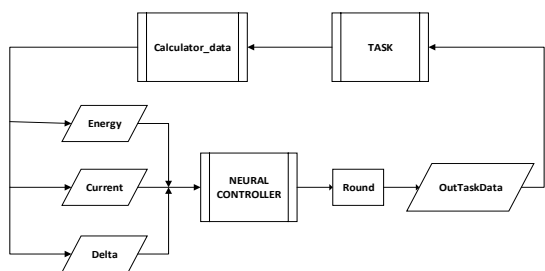
Bài báo nghiên cứu được tổ chức theo 4 mục. Mục 1 giới thiệu và tổng quan về các nghiên cứu trước đây. Mục 2 mô tả quá trình thiết kế các bộ điều khiển neural network trên mô hình hệ thống Simulink. Mục 3 trình bày và thảo luận kết quả đạt được khi sử dụng bộ neural network. Cuối cùng, kết luận và định hướng trong tương lai sẽ được đưa ra trong mục 4.

2. THIẾT KẾ HỆ THỐNG

2.1 Mô hình hệ thống

Mô hình thuật toán của nút cảm biến sử dụng được thể hiện trên hình 2, các thông số ngõ vào cho bộ neural network bao gồm dung lượng pin (energy), dòng sạc tấm pin năng lượng mặt trời (current) và độ biến thiên của dữ liệu thu thập (delta). Trong mô

hình, dung lượng pin là phần trăm năng lượng còn lại trong pin, độ biến thiên của dữ liệu là độ lệch của dữ liệu trong một khoảng thời gian. Khi số liệu thay đổi đột ngột thì độ lệch sẽ cao và ngược lại. Cuối cùng là dòng sạc của pin năng lượng mặt trời. Các dữ liệu này sau khi qua bộ neural, giá trị dự đoán ngõ ra được làm tròn. Ngõ ra cuối cùng sẽ cho khả năng dự đoán và cân đối tác vụ thích hợp nhất từ đó hệ thống sẽ thực hiện tác vụ tiếp theo đã được chọn từ bộ neural trên.



Hình 2. Thuật toán hệ thống

Trong các thông số ngõ vào, năng lượng pin là khoảng giá trị của dung lượng pin được giới hạn trong $[0, 100]$, được đọc thông qua bộ ADC của vi điều khiển. Với mức pin tối đa khi đầy là $V_{bat_full} = 4,3V$ và khi cạn là $V_{bat_low} = 3,7V$. Từ đó có thể tính giá trị của pin dựa trên công thức sau đây:

$$E = \frac{V_{now} - V_{Bat_low}}{V_{Bat_full} - V_{Bat_low}} * 100\% \quad (1)$$

Dòng sạc của tấm pin $[0, 250]$ mA được lấy từ cảm biến INA-219 dựa trên điện áp rơi trên điện trở shunt $R_{sensing}$ có giá trị 0.01Ω sai số thấp. V_1 và V_2 là điện áp trên 2 đầu điện trở shunt, G là hệ số khuếch đại (module INA-219 thực nghiệm giá trị này =1).

$$I_{charge} = G * \frac{(V_2 - V_1)}{R_{sensing}} \quad (2)$$

Sai lệch dữ liệu Delta được xác định là độ lệch chuẩn của 1 chu kỳ lấy dữ liệu. Trong một chu kỳ có N mẫu được lấy trong thời gian T. Độ lệch chuẩn của dữ liệu trong 1 chu kỳ được biểu diễn như sau:

$$Delta = \mu \sqrt{\frac{\sum_i^N (x_i - X)^2}{N-1}} \quad (3)$$

Hệ số μ là hệ số lựa chọn và hệ số này được hiệu chỉnh với từng loại môi trường

khác nhau (nơi có độ biến thiên nhiệt độ khác nhau).

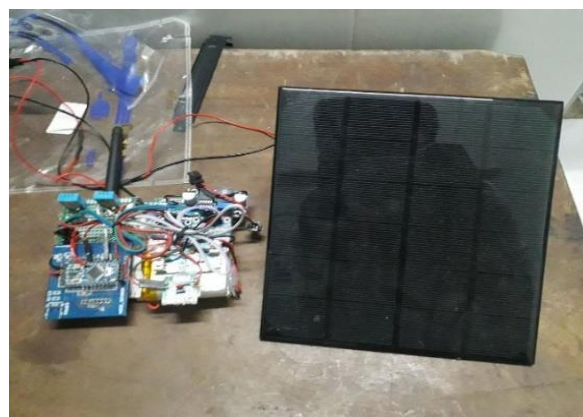
Kết quả mong muốn ở ngõ ra gồm 4 tác vụ tương ứng với 4 giá trị $[1, 2, 3, 4]$ được thiết kế phù hợp với thông số ngõ vào. Chức năng các tác vụ, công suất tiêu hao và giá trị ngõ ra của các tác vụ được biểu diễn thông qua bảng sau:

Bảng 1. Các tác vụ đề xuất

Số TT tác vụ	Chức Năng Task	Công Suất Trung Bình
1	Không gửi dữ liệu	Rất Thấp
2	Kết thúc chu kỳ ngày gửi dữ liệu	Thấp
3	Gửi dữ liệu theo chu kỳ với N mẫu	Trung Bình
4	Gửi dữ liệu liên tục theo 2N mẫu	Cao

2.2 Phương án lấy số liệu

Các tập số liệu được lấy trực tiếp ở quận Thủ Đức, TP. Hồ Chí Minh. Chúng tôi thiết kế một thiết bị như hình 3 sử dụng Arduino Pro Mini, pin Lithium 3V7 2000mAh, pin mặt trời 4.5W và mạch sạc dòng tối đa 250mA. Các số liệu thu thập về được tính toán theo công thức (1), (2) và (3).



Hình 3. Bộ thu thập dữ liệu

2.3 Năng lượng tiêu hao thực tế

Các tác vụ được nêu trên tiêu hao một lượng năng lượng khác nhau với 4 mức khác nhau từ rất thấp đến cao. Các tác vụ này được tính toán và đo lường bằng phương pháp lấy trung bình. Với nguồn pin đạt 100%, chúng tôi tiến hành cho chạy duy nhất 1 tác vụ với

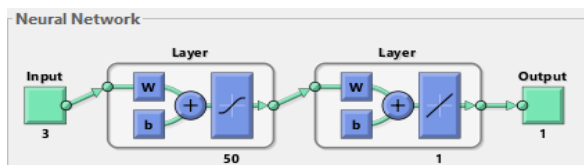
đồng sạc bằng 0 cho đến khi pin đạt còn 80%. Với nguồn pin sử dụng cho hệ thống là 2000mAh thì khi còn 80% pin đã tiêu thụ khoảng 400mAh. Sau đó lấy 400mAh chia cho số chu kỳ mà tác vụ chạy được. Các con số được thống kê ở bảng sau:

Bảng 2. Năng lượng tiêu hao

Tác Vụ	Mức tiêu hao	Số lượng mẫu	Năng lượng tiêu hao 1 chu kỳ (mAh)	Năng lượng tiêu hao 1 giờ (mAh)
1	Rất Thấp	>5000	~0	~0
2	Thấp	435	0.9195402	3.678160
3	Trung Bình	105	3.8095238	15.23809
4	Cao	188	2.1276595	42.55319

2.4 Bộ Neural Network đề xuất

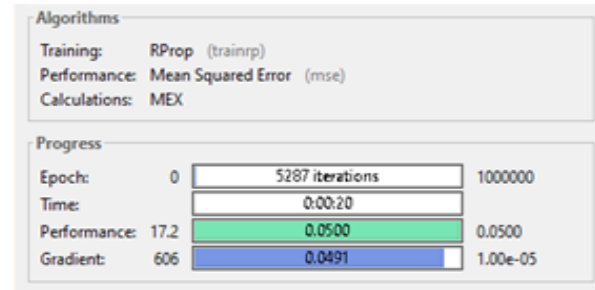
Khác với thuật toán lập trình lịch trình tác vụ như thuật toán LSA đã đề cập ở mục 2, bộ phân loại sử dụng neural network đáp ứng được khả năng thay đổi tác vụ làm việc của nó từ số liệu thay đổi từ ngõ vào. LSA sẽ chạy liên tục theo lịch trình đã được lập sẵn mà không thay đổi được tác vụ mong muốn. Với các ứng dụng đơn giản nhưng yêu cầu sự chính xác cao mà không cần tính toán quá nhiều thì mạng neural với số lớp ẩn và số lượng neural nhỏ thể hiện được sự ưu điểm của nó về tính toán hay bộ nhớ.



Hình 4. Mô hình mạng neural đề xuất

Mô hình mạng đề xuất ở hình 4. Chúng tôi lựa chọn mạng có 3 lớp chính bao gồm 1 lớp ngõ vào, 1 lớp ẩn và 1 lớp ngõ ra. Các neural ở lớp ngõ vào lần lượt là dung lượng của pin, hiệu số giữa đồng sạc, dòng tải trung bình của task đang chạy và cuối cùng là giá trị độ lệch chuẩn của dữ liệu mới nhất với trung bình của các dữ liệu. Lớp ngõ ra có 1 neural tương ứng với giá trị mà task tiếp theo sẽ thực hiện cho Node Sensor. Các giá trị ngõ ra cho việc huấn luyện tương ứng với các tác

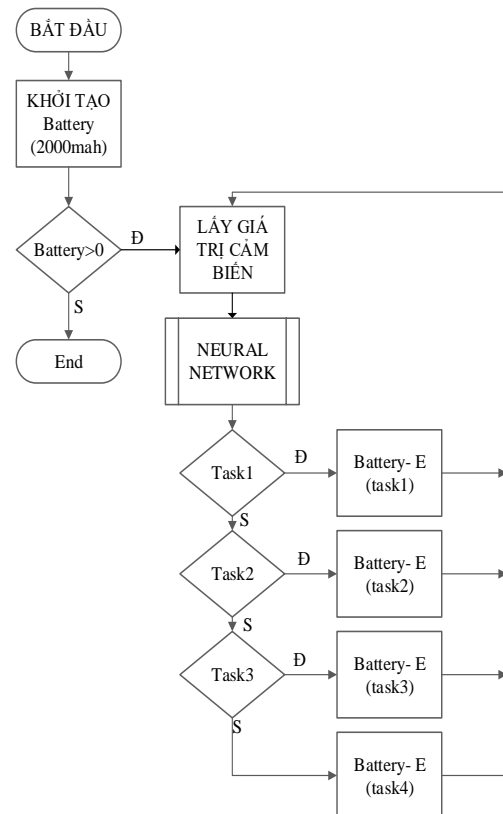
vụ 1, tác vụ 2, tác vụ 3 và tác vụ 4 sẽ là [1 2 3 4]. Sau khi thu thập số liệu, chúng tôi tiến hành để huấn luyện cho mô hình bằng Matlab với phương án học có giám sát.



Hình 5. Quá trình huấn luyện

Trong hình 5, thuật toán sử dụng để huấn luyện mạng là lan truyền ngược đàn hồi (resilient backpropagation). Thuật toán này nhằm mục đích chính là loại bỏ những tác động có hại của độ lớn các đạo hàm riêng. Chỉ dấu của đạo hàm được sử dụng để xác định hướng cập nhật trọng số và độ lớn của đạo hàm không ảnh hưởng đến việc cập nhật trọng số.

2.5 Đánh giá kết quả về mặt năng lượng

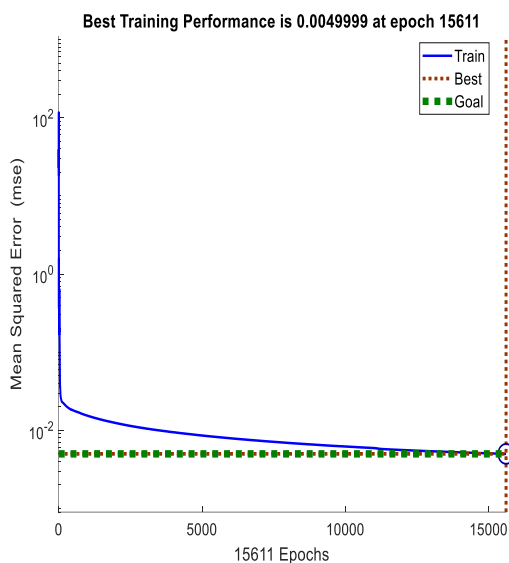


Lưu đồ 1. Thuật toán kiểm chứng kết quả về mặt năng lượng

Lưu đồ 1 bao gồm các giá trị khởi tạo pin 100%, giá trị nhiệt độ lấy thực tế ở phần 3.2. Tiến trình sẽ chạy liên tục khi nguồn pin vẫn còn. Khi nguồn pin hết, các giá trị nhận được sẽ được thống kê bao gồm số lượng tác vụ và khả năng tiết kiệm pin cũng như chất lượng thông tin thu được.

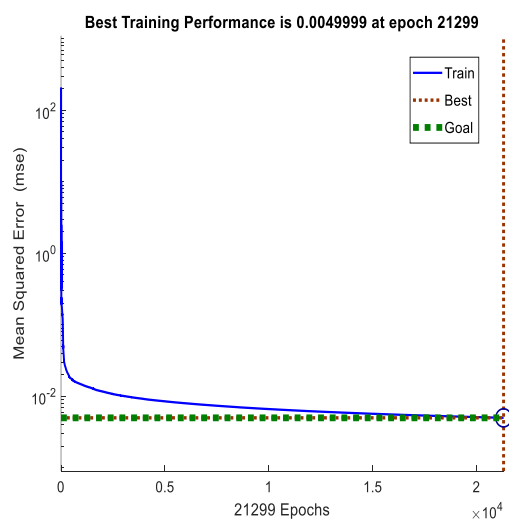
3. KẾT QUẢ

3.1 Đánh giá chất lượng của mạng neural

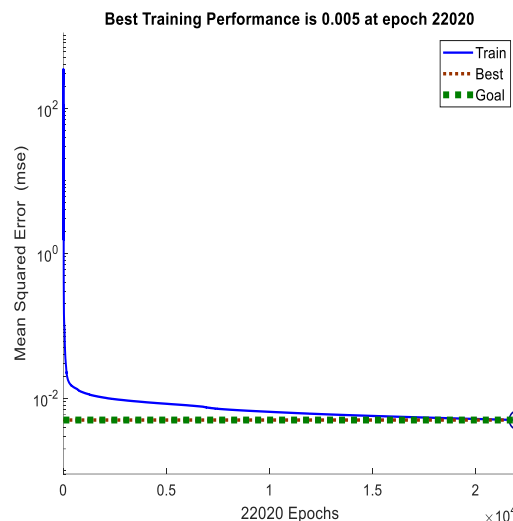


Hình 6. Sai số với 1 lớp ẩn – 20 neural

Với một mạng neural cơ bản 1 lớp ẩn có 20 neural như hình 6 thì kết quả cho chúng ta thấy rằng giá trị sai số hội tụ về giá trị 0.05 sau 15611 vòng lặp. Giá trị hội tụ theo một đường tuyến tính.



Hình 7. Sai số với 1 lớp ẩn – 25 neural

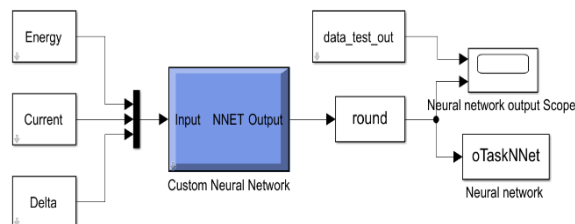


Hình 8. Sai số với 1 lớp ẩn – 50 neural

Trong hình 7 và 8, các giá trị hội tụ về 0.005 với trên 20.000 vòng lặp. Như vậy, ta có thể xác định 20 neural ở lớp ẩn thứ 1 là một lựa chọn tốt cho việc huấn luyện mạng do đó với số lượng neural và số lớp ẩn càng ít thì ở nút cảm biến sẽ càng giảm thiểu được sự công kênh về tính toán cũng như bộ nhớ. Từ đó, ở các nút cảm biến có thể tối ưu về mặt năng lượng hơn.

3.2 Độ chính xác của mạng neural

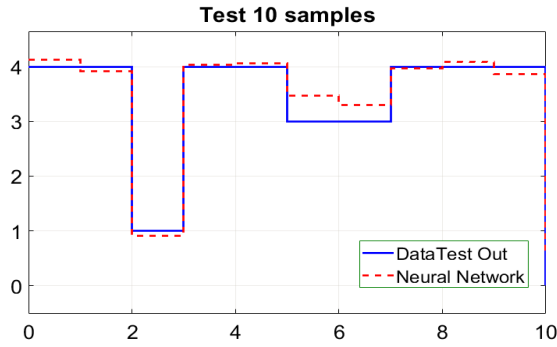
Sau khi đã thực hiện các công việc đưa ra như trên, chúng tôi đã ra một model sử dụng một mạng neural tối ưu (1 lớp ẩn chứa 20 neural) trong việc lựa chọn tác vụ mà chúng tôi đã đề xuất được thể hiện ở hình 9.



Hình 9. Mô hình Simulink trên Matlab

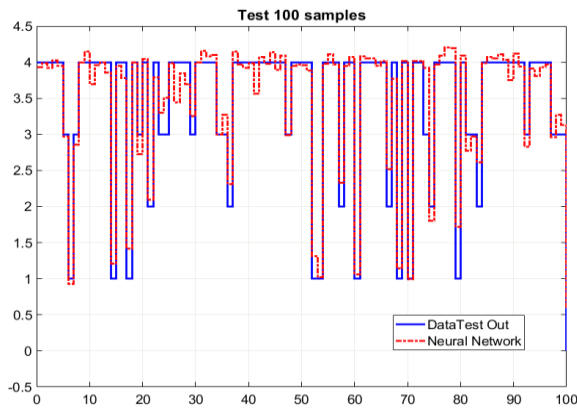
Để có góc nhìn tổng quan nhất về việc model lựa chọn chính xác các tác vụ, chúng tôi tiếp tục thu thập thêm 10 và 100 mẫu dữ liệu ngẫu nhiên để đưa vào model Simulink nhằm xác nhận lại khả năng chính xác nhất của mô hình.

Từ mô hình hệ thống trên, khi thực hiện chúng tôi thu được kết quả:



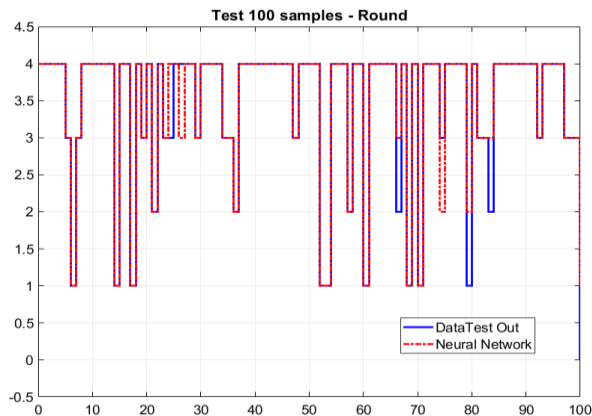
Hình 10. Kết quả của bộ phân loại với 10 mẫu ngẫu nhiên

Kết quả thu được đạt 10/10 so với kết quả mong muốn được thể hiện ở hình 10. Với những kết quả đạt được ở trên, chúng tôi tiếp tục tăng số lượng dữ liệu ngõ vào (100 mẫu nhằm đánh giá xác suất xảy ra lỗi).



Hình 11. Ngõ ra bộ neural với 100 mẫu

Với 2 kết quả thu được (hình 10 và hình 11), ta thấy rằng với bộ neural network có độ chính xác khá cao và dữ liệu dự đoán ở ngõ ra bám tương đối sát với giá trị mong muốn khi huấn luyện mạng.



Hình 12. Ngõ ra bộ neural với 100 mẫu sau khi được làm tròn

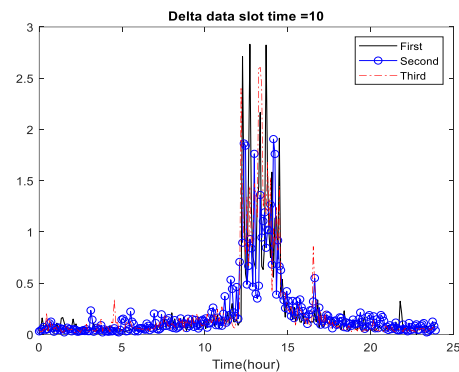
Từ hình 12, các giá trị sau khi làm tròn từ ngõ ra của mạng neural có 4/100 tác vụ sai nhưng giá trị vẫn chấp nhận được. Các số liệu đánh giá đưa lựa chọn từ bảng sau:

Bảng 3. Tỷ lệ dự đoán của mô hình

Số neural ở lớp ẩn	Tỷ lệ lựa chọn chính xác từ mẫu ngẫu nhiên
20	96%
25	93%
50	97%

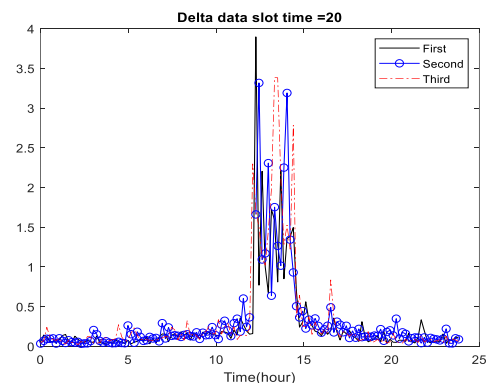
3.3 Giá trị độ lệch của số liệu

Giá trị delta thu được sau đây được lấy số liệu từ thực tế trong 3 ngày ngẫu nhiên tại một vị trí. Sau đó, giá trị này được tính toán dựa theo công thức (3) với N slot lần lượt là 10 và 20. Từ số liệu trên ta thay đổi hệ số μ để tương ứng với nhiều môi trường khác nhau (ở đây chúng tôi thực hiện với giá trị $\mu = 1$).



Hình 13. Giá trị của delta với $N = 10$

Với giá trị của N slot bằng 10. Giá trị delta rơi từ 0 đến gần 3. Giá trị dao động mạnh vào từ 10h đến 13h.



Hình 14. Giá trị của delta với $N = 20$

Các giá trị delta dao động nhiều trong khoảng giữa trưa từ 11h đến 14h. Các thuật toán thường chỉ tính hệ số delta không xác định được rằng trong khoảng thời gian này sẽ thường xuyên có biến động đột biến nên giá trị thường không được lấy nhiều hơn các thời điểm khác. Đối với các khoảng thời gian từ 7h đến 9h các giá trị không cần lấy nhiều nguyên nhân các giá trị độ lệch delta không sai lệch nhiều. Các dữ liệu được đối chiếu với phương pháp quan trắc theo thông tư của bộ tài nguyên môi trường ngày 26/12/2018 quy định kỹ thuật về quan trắc và cung cấp thông tin, dữ liệu khí tượng thủy văn đối với trạm khí tượng thủy văn chuyên dùng [22] và dựa trên nghiên cứu [11]. Các số liệu của môi trường này được lấy theo chu kỳ nhỏ nhất 1 đến 10 phút. Trung bình một mẫu được thu thập với chu kỳ 5 phút, trong một giờ ta có được 20 mẫu được thu thập dựa theo thông tư trên. Các thông số được thể hiện ở bảng 4 sau đây:

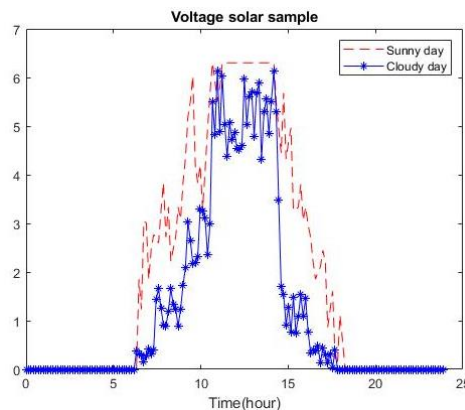
Bảng 4. Số lượng mẫu thu thập được

Thời gian	Chu kỳ 1 phút[11]	Chu kỳ 5 phút[22]	Thuật toán đề xuất
7h – 9h	120	60	31
11h – 13h	120	60	92
17h – 19h	120	60	57
21h – 3h	360	180	120

Từ các thông số ở bảng 4, với chu kỳ lấy mẫu là 1 phút, số lượng thông tin thu thập nhiều hơn rất nhiều, nhưng trong đó có rất nhiều thông tin trùng lặp, không có ích và năng lượng hao phí sẽ nhiều. Đối với phương pháp lấy mẫu theo chu kỳ liên tục trong với 5 phút, số lượng mẫu thu về đạt mức khá, nhưng khoảng thời gian có nền nhiệt biến thiên không có sự thích nghi khi mà trong khoảng thời gian đó chúng ta cần thêm nhiều thông tin hơn. Còn với thuật toán đề xuất, trong khoảng thời gian có biến động lớn về số liệu thì các giá trị thu thập được nhiều hơn. Thời gian trong ngày nền nhiệt độ thay đổi ít thì số liệu được lấy ít hơn nhằm đảm bảo được tính năng lượng của nút cảm biến.

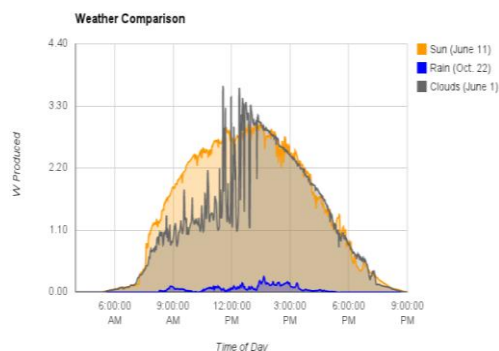
3.4 Năng lượng thu được của pin mặt trời

Điện áp của pin được thu thập thông qua bộ ADC nội của vi điều khiển và dòng điện từ pin qua một bộ lọc thông thấp (LPF) thụ động RC để giảm thiểu nhiễu của nguồn, giữ lại thành phần chính DC.



Hình 15. Điện áp trung bình của pin giữa ngày râm và nắng

Điện áp của pin dao động không đều được thể hiện như hình 15 vì nhiều lý do khách quan ví dụ như mây che phủ mặt trời. Công suất của pin đạt được tối đa trong khoảng thời gian giữa trưa. Những ngày mưa, pin sẽ không đạt được điện áp 6V.



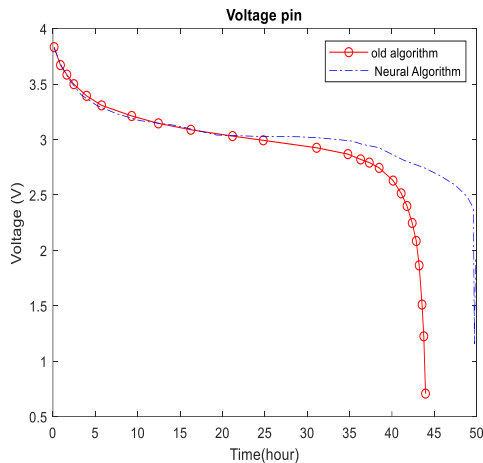
Hình 16. Năng lượng thu thập được của pin giữa ngày râm và nắng (Wh)

Với lượng pin đề xuất là 2000 mAh, khả năng sạc đầy pin trong một ngày với điều kiện năng lý tưởng đạt 100%. Với các ngày độ phủ nắng thấp (2h- 4h/ngày) thì pin được nạp tối thiểu 80%.

3.5 Đánh giá kết quả về mặt năng lượng

Tiến hành thực hiện thuật toán đề xuất với điều kiện dòng sạc vào của pin bằng 0, điện áp của pin được thu thập đồng thời để

chứng minh khả năng đáp ứng về mặt năng lượng của thuật toán. Ngoài ra, chúng tôi đối chiếu với phương pháp lấy mẫu theo chu kỳ 5 phút [22].



Hình 17. So sánh điện áp của pin

Trong khoảng thời gian đầu, lượng pin đang nhiều nên khi sử dụng thuật toán đề xuất thì lượng pin hao hơn vì lúc này nút cảm biến đang sử dụng với tác vụ số 4 đạt với mong muốn ban đầu trong hình 17. Lượng pin hao hụt nhiều hơn nhưng số lượng mẫu thu về là tối ưu. Về lâu dài, khi lượng pin càng thấp ta có thể tối ưu 1 phần về mặt năng lượng của nút vì khi đó tác vụ bị thay đổi xuống tác vụ ít tốn năng lượng hơn và kéo dài được thời lượng và tuổi thọ của pin.



Hình 18. Bộ thu thập số liệu

Hình 18 là phần cứng để được thực hiện 2 phương pháp trên bao gồm một vi điều khiển STM32F103, lora SX1278 433Mhz, pin 2000mah, MCU quản lý sạc CN3065, pin mặt trời 6V 4.5W, cảm biến nhiệt độ DS18B20. Sau khi biên dịch chương trình, thuật toán đề xuất chiếm 12% dung lượng của bộ nhớ flash.

4. KẾT LUẬN

Bài báo cáo trình bày tổng quan phương pháp phân loại sử dụng mạng neural và thiết kế bộ điều khiển neural cho việc phân loại. Kết quả mô phỏng được trình bày và phân tích thông qua Matlab bao gồm:

Mạng neural với 1 mạng nhỏ bao gồm 1 lớp ngõ vào, 1 lớp ẩn và 1 lớp ngõ ra có kết quả ra rất khả quan và tương đối chính xác. Tuy nhiên, với lượng neural ở lớp ẩn càng nhiều thì hệ thống tính toán càng nặng hơn, kéo theo đó năng lượng tiêu tốn trên nút nhiều hơn. Với 20 neural như đã chọn rút gọn được nhiều tài nguyên từ bộ nhớ đến năng lượng để tính toán ở các nút cảm biến. Khi tiến hành huấn luyện, việc lựa chọn hệ số tốc độ học (Learning rate) khá quan trọng cho mô hình được huấn luyện đến điểm hội tụ thấp và nhanh nhất.

Mặt khác, thuật toán có tính cân đối tối ưu hơn giữa 2 yếu tố quan trọng của hệ thống thu thập dữ liệu năng lượng thấp là chất lượng của thông tin và tuổi thọ pin. Lượng pin mất đi tương đương với thuật toán cũ nhưng bù lại các số liệu ở những thời điểm có sự thay đổi lớn được lấy nhiều mẫu hơn và ít hơn khi các số liệu thay đổi không đáng kể. Ngoài ra, thuật toán có tính thích nghi tốt với từng môi trường vì các hệ số của mạng neural có thể được thay đổi dựa trên các thông tin thu thập được.

TÀI LIỆU THAM KHẢO

- [1] S. Escolar, S. Chessa, and J. Carretero, Optimization of Quality of Service in Wireless Sensor Networks Powered by Solar Cells. 2012.
- [2] A. A. Babayo, M. H. Anisi, and I. Ali, "A Review on energy management schemes in energy harvesting wireless sensor networks," Renew. Sustain. Energy Rev., vol. 76, pp. 1176–1184, 2017, doi: <https://doi.org/10.1016/j.rser.2017.03.124>.

- [3] N. Kimura, V. Jolly, and S. Latifi, “Energy restrained data dissemination in wireless sensor networks,” *Int. J. Distrib. Sens. Networks*, vol. 2, no. 3, pp. 251–265, 2006, doi: 10.1080/15501320600642692.
- [4] S. Basagni, M. Y. Naderi, C. Petrioli, and D. Spenza, “Wireless Sensor Networks with Energy Harvesting,” in *Mobile Ad Hoc Networking*, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2013, pp. 701–736.
- [5] A. Sinha and A. Chandrakasan, “Dynamic power management in wireless sensor networks,” *IEEE Des. Test Comput.*, vol. 18, no. 2, pp. 62–74, 2001, doi: 10.1109/54.914626.
- [6] L. Wang and Y. Xiao, “A Survey of Energy-Efficient Scheduling Mechanisms in Sensor Networks,” *MONET*, vol. 11, pp. 723–740, Oct. 2006, doi: 10.1007/s11036-006-7798-5.
- [7] X. Fan, “Sensors Dynamic Energy Management in WSN,” *Wirel. Sens. Netw.*, vol. 02, pp. 698–702, Jan. 2010, doi: 10.4236/wsn.2010.29084.
- [8] S. Sudevalayam and P. Kulkarni, “Energy harvesting sensor nodes: Survey and implications,” *IEEE Commun. Surv. Tutorials*, vol. 13, no. 3, pp. 443–461, Sep. 2011, doi: 10.1109/SURV.2011.060710.00094.
- [9] P. Pillai and K. Shin, “Real-Time Dynamic Voltage Scaling for Low-Power Embedded Operating Systems,” *ACM SIGOPS Oper. Syst. Rev.*, vol. 35, Sep. 2001, doi: 10.1145/502034.502044.
- [10] G. Amato, A. Caruso, and S. Chessa, “Application-driven, energy-efficient communication in wireless sensor networks,” *Comput. Commun.*, vol. 32, pp. 896–906, Mar. 2009, doi: 10.1016/j.comcom.2008.12.022.
- [11] X. Chen, H.-B. Chen, W. Ma, X. Li, and S. X.-. Tan, “Energy-efficient wireless temperature sensing for smart building applications,” in *2016 13th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, 2016, pp. 680–683, doi: 10.1109/ICSICT.2016.7999010.
- [12] J. Khan, H. Qureshi, and A. Iqbal, “Energy management in Wireless Sensor Networks: A survey,” *Comput. Electr. Eng.*, vol. 41, Jul. 2014, doi: 10.1016/j.compeleceng.2014.06.009.
- [13] S. Escolar, A. Caruso, S. Chessa, X. Del Toro, F. J. Villanueva, and J. C. Lopez, “Statistical Energy Neutrality in IoT Hybrid Energy-Harvesting Networks,” *Proc. - IEEE Symp. Comput. Commun.*, vol. 2018-June, pp. 444–449, 2018, doi: 10.1109/ISCC.2018.8538532.
- [14] M. Severini, S. Squartini, and F. Piazza, “Energy Aware Lazy Scheduling Algorithm for Energy-Harvesting Sensor Nodes,” *Neural Comput. Appl.*, vol. 23, Dec. 2013, doi: 10.1007/s00521-012-1088-x.
- [15] C. Moser, J. Chen, and L. Thiele, “Dynamic power management in environmentally powered systems,” in *2010 15th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2010, pp. 81–88, doi: 10.1109/ASPDAC.2010.5419916.
- [16] C. Moser, D. Brunelli, L. Thiele, and L. Benini, *Lazy Scheduling for Energy Harvesting Sensor Nodes*. 2006.
- [17] C. Moser, L. Thiele, D. Brunelli, and L. Benini, “Adaptive Power Management in Energy Harvesting Systems,” in *Proceedings of the Conference on Design, Automation and Test in Europe*, 2007, pp. 773–778.
- [18] A. Caruso, S. Chessa, S. Escolar, X. Del Toro, and J. C. López, “A dynamic programming algorithm for high-level task scheduling in energy harvesting IoT,” *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2234–2248, 2018, doi: 10.1109/JIOT.2018.2828943.
- [19] P. Zhang, X. Zhang, J. Li, and G. Huang, “The effects of body weight, temperature, salinity, pH, light intensity and feeding condition on lethal DO levels of whiteleg shrimp,

- Litopenaeus vannamei (Boone, 1931),” *Aquaculture*, vol. 256, no. 1, pp. 579–587, 2006, doi: <https://doi.org/10.1016/j.aquaculture.2006.02.020>.
- [20] R. E. Uhrig, “Introduction to artificial neural networks,” in *Proceedings of IECON '95 - 21st Annual Conference on IEEE Industrial Electronics*, vol. 1, pp. 33–37, doi: 10.1109/IECON.1995.483329.
- [21] A. K. Jain, J. Mao, and K. M. Mohiuddin, “Artificial neural networks: A tutorial,” *Computer*, vol. 29, no. 3, pp. 31–44, Mar. 1996, doi: 10.1109/2.485891.
- [22] Lê Công Thành, “Quy định kỹ thuật về quan trắc và cung cấp thông tin, dữ liệu khí tượng thủy văn đối với trạm khí tượng thủy văn chuyên dùng,” 2018.

Tác giả chịu trách nhiệm bài viết:

Võ Minh Huân

Trường Đại học Sư phạm Kỹ thuật TP.Hồ Chí Minh

Email: huanvm@hcmute.edu.vn