

ỨNG DỤNG MÔ HÌNH HỌC SÂU TỐI ƯU HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG MÁY ẤP TRỨNG

Lê Thị Minh Tâm*, Bùi Thanh Tùng*, Vũ Đình Công*,
Nguyễn Duy Nguyễn*, Nguyễn Văn An*, Phạm Tiến Đạt*

TÓM TẮT

Ứng dụng học sâu trong máy sản xuất giúp tự động hóa, nâng cao hiệu suất, giảm sai sót, tối ưu quy trình, và tối ưu trong việc xử lý dữ liệu các công nghệ. Mạng nơron là công cụ chính của học sâu, giúp mô phỏng tư duy như con người, xử lý dữ liệu phức tạp và tối ưu hóa mô hình. Bài báo này đề xuất sử dụng mạng nơron đối kháng mô hình hóa quá trình gia nhiệt và tạo độ ẩm của máy ấp trứng dựa vào đặc điểm của chu kỳ ấp trứng gà Đông Tảo. Mô hình toán học của máy ấp trứng được đưa vào hệ thống điều khiển giúp cho việc huấn luyện sử dụng mạng nơron LSTM điều khiển nhiệt độ và độ ẩm cho cả chu kỳ ấp là 21 ngày dùng mạng 10 lớp và sử dụng mạng 3 lớp đối với giai đoạn năm ngày đầu của chu kỳ. Hệ thống có nhiệm vụ tính toán và điều chỉnh hoạt động của hệ thống gia nhiệt, làm mát và tạo ẩm nhằm duy trì môi trường ấp trứng tối ưu và được hiện thực bằng ngôn ngữ Python, với TensorFlow/Keras để xây dựng mô hình học sâu, NumPy/Pandas hỗ trợ xử lý dữ liệu, và GPIO dùng để giao tiếp với phần cứng. Kết quả nghiên cứu cho thấy hệ thống có khả năng nhận diện và điều chỉnh chính xác nhiệt độ, độ ẩm ấp, đáp ứng nhanh với sự thay đổi môi trường và nâng cao tỷ lệ nở so với phương pháp điều khiển truyền thống.

Từ khóa: Học sâu, máy ấp trứng gia cầm, mạng nơron đối kháng, mạng LSTM, Python

Ngày nhận bài: 25/04/2025; Ngày phản biện: 15/05/2025; Ngày duyệt đăng: 25/05/2025

ABSTRACT

Deep learning applications in manufacturing machines help automate, enhance performance, reduce errors, optimize processes, and improve data processing for various technologies. Neural networks are the primary tools for deep learning, enabling simulations of human-like thinking, handling complex data, and optimizing models. This paper proposes using adversarial neural networks to identify the heating and humidity processes of an incubator based on the characteristics of the Dong Tao chicken egg incubation cycle. The recognized data is then fed into a control system, facilitating the training of a LSTM neural network to control temperature and humidity throughout the 21 days incubation cycle using a 10-layer network, and a 3 layer networks for the first five days of the cycle. The system's role is to compute and adjust the heating, cooling, and humidifying processes to maintain the optimal incubation environment. The implementation uses Python, with TensorFlow/Keras to build the deep learning model, NumPy/Pandas for data processing, and GPIO for hardware communication. The research results demonstrate that the system can accurately recognize and adjust incubation temperature and humidity, respond quickly to environmental changes, and improve hatch rates compared to traditional control methods.

Keywords: Deep learning, poultry egg incubator, GANs network, LSTM neural network, Python.

Email: leminhtamutehy@gmail.com

1. Đặt vấn đề

Học sâu (Deep Learning) đã chứng minh nhiều ưu điểm vượt trội trong xử lý dữ liệu và trí tuệ nhân tạo. Nhờ tính linh hoạt và khả năng mở rộng, học sâu ngày càng được ứng dụng rộng rãi trong nhiều lĩnh vực, bao gồm cả nông nghiệp và chăn nuôi gia cầm (Nielsen, 2017). Việc áp dụng học sâu trong điều khiển máy ấp trứng gia cầm, đặc biệt là giống gà Đông Tảo có giá trị

kinh tế cao, là một đề xuất có ý nghĩa thực tiễn lớn trong lĩnh vực sản xuất thực phẩm toàn cầu.

Hiện nay, các phương pháp điều khiển truyền thống như PID vẫn được sử dụng rộng rãi để điều chỉnh nhiệt độ và độ ẩm trong máy ấp trứng. Tuy nhiên, các phương pháp này bộc lộ nhiều hạn chế khi áp dụng vào các hệ thống có tính phi tuyến cao, dễ bị ảnh hưởng bởi điều kiện môi trường và đòi hỏi độ chính xác cao trong điều

*Trường Đại học Sư phạm Kỹ thuật Hưng Yên

hiển (Massey & Hutchings, 2021).

Việc điều chỉnh chính xác các thông số nhiệt độ và độ ẩm theo từng giai đoạn ấp là yếu tố then chốt nhằm tối ưu hóa tỷ lệ nở và đảm bảo sức khỏe cho con giống (Boleli et al., 2016). Học sâu cung cấp khả năng tự động học từ dữ liệu, phát hiện các mẫu phức tạp và xử lý tập dữ liệu lớn mà không cần lập trình thủ công. Điều này giúp nâng cao độ chính xác trong các bài toán nhận dạng hình ảnh, xử lý ngôn ngữ tự nhiên, cũng như dự đoán hành vi (Sutskever et al., 2011).

Mạng noron nhân tạo là một lựa chọn phù hợp nhờ khả năng học hỏi và thích nghi tốt với dữ liệu thực tế. Không giống các bộ điều khiển cổ điển phải xây dựng quy tắc điều khiển cố định, mạng noron có thể tự động xây dựng mô hình từ dữ liệu huấn luyện, từ đó đưa ra quyết định điều khiển tối ưu (Rosenblatt, 1957). Đặc biệt, khả năng xử lý các đặc tính phi tuyến và môi trường thay đổi liên tục khiến mạng noron trở thành công cụ mạnh mẽ trong việc cải thiện độ chính xác và độ ổn định cho hệ thống điều khiển nhiệt độ và độ ẩm trong máy ấp trứng gà Đông Tảo (Mahale & Sonavane, 2016).

Ở bài báo này, nhóm tác giả đề xuất phương pháp điều khiển bằng mạng noron điều khiển tự động nhiệt độ và độ ẩm cho máy ấp trứng gà Đông Tảo. Bài báo được chia làm 5 mục: Mục I là giới thiệu chung. Mục II trình bày về cơ sở lý thuyết bao gồm đặc điểm chu kỳ ấp trứng gà Đông Tảo, giới thiệu về mạng noron đối kháng và mạng noron hồi quy. Mục III ứng dụng lý thuyết ở Mục II thực nhận dạng máy ấp trứng gà Đông Tảo dựa trên dữ liệu về chu kỳ ấp. Và xây dựng mạng noron bộ nhớ ngắn hạn dài (LSTM) gồm 10 lớp để dự đoán và điều chỉnh nhiệt độ, độ ẩm theo yêu cầu ở từng giai đoạn ấp. Mục IV triển khai việc mô phỏng kết quả nghiên cứu sử dụng ngôn ngữ Python và thảo luận. Mục V kết luận. Kết quả nghiên cứu có ý nghĩa thực tế nhằm tối ưu hóa quá trình ấp, góp phần nâng cao chất lượng con giống, đồng thời giảm thiểu rủi ro và tiết kiệm chi phí cho người chăn nuôi.

2. Cơ sở lý thuyết

2.1 Quy trình ấp trứng gà Đông Tảo

Quá trình ấp trứng gà Đông Tảo đòi hỏi điều khiển nhiệt độ của máy ấp trứng khá đặc biệt, trong đó nhiệt độ và độ ẩm cần được duy trì ở mức tối ưu để đảm bảo tỷ lệ nở cao. Gà Đông Tảo là giống gà quý, có giá trị kinh tế cao, nhưng trứng của chúng có vỏ dày hơn các loại trứng gà thông thường, khiến quá trình trao đổi nhiệt và độ ẩm diễn ra chậm hơn. Điều này đòi hỏi hệ thống điều khiển phải có khả năng điều chỉnh linh hoạt để thích ứng với sự

thay đổi của môi trường cũng như đặc tính sinh học của trứng trong từng giai đoạn phát triển của phôi.

Chu kỳ nhiệt độ, độ ẩm đối với máy ấp đơn kỳ cho trứng gà Đông Tảo được trình bày như trong Bảng 1:

Bảng 1. Chu kỳ nhiệt độ, độ ẩm đối với máy ấp đơn kỳ cho trứng gà Đông Tảo

Ngày ấp	Nhiệt độ	Độ ẩm
Ngày 1-5	37.8 °C	60÷61 %
Ngày 6-11	37.6 °C	55÷57 %
Ngày 12-18	37.6 °C	50÷53 %
Ngày 19	37.2 °C	60 %
Ngày 20-21	37.2 °C	70÷75 %

2.2 Mạng noron đối kháng[12][14]

Mạng noron đối kháng GANs là một mô hình học sâu bao gồm hai thành phần chính: bộ sinh (Generator) và bộ phân biệt (Discriminator). Bộ sinh có nhiệm vụ tạo ra dữ liệu giả từ ngẫu nhiên ngẫu nhiên, trong khi bộ phân biệt cố gắng phân tách giữa dữ liệu thật và dữ liệu giả do bộ sinh tạo ra. Quá trình huấn luyện GAN diễn ra theo cơ chế đối số: bộ phân biệt không ngừng cải tiến khả năng nhận dạng dữ liệu giao diện, nhưng bộ sinh liên tục tối ưu để tạo ra sao dữ liệu cho bộ phân biệt không thể phát hiện. Nhờ sự cạnh tranh này, GAN có thể tạo ra dữ liệu giả có chất lượng ngày càng cao, gần giống với dữ liệu thật hơn.

GAN tối ưu hóa một hàm đối kháng:

$$\min_G \max_D V(D, G) = E_{x \sim p_{\text{real}}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log (1 - D(G(z)))] \quad (1)$$

Trong đó:

x là mẫu dữ liệu thật, được lấy từ phân phối dữ liệu thật $p_{\text{real}}(x)$.

z là nhiễu ngẫu nhiên, được lấy từ phân phối $p_z(z)$ (thường là phân phối chuẩn).

$G(z)$ là đầu ra của Generator, tức là dữ liệu giả.

$D(x)$ là xác suất mà Discriminator gán cho dữ liệu thật.

$D(G(z))$ là xác suất mà Discriminator gán cho dữ liệu giả.

Phương trình của Generator

Generator nhận một vector nhiễu z có kích thước d , qua các tầng noron để tạo ra dữ liệu x_{fake} :

$$G(z) = W_2 \cdot \sigma(W_1 \cdot z + b_1) + b_2 \quad (2)$$

Trong đó:

W_1, W_2 là ma trận trọng số của các lớp ẩn.

b_1, b_2 là bias của các lớp.

$\sigma(x)$ là các hàm kích hoạt (ở đây là ReLU cho tầng ẩn, và linear cho đầu ra).

Phương trình của Discriminator

Discriminator nhận đầu vào là dữ liệu thật hoặc giả (x) và đưa ra một xác suất $D(x)$ (thật hay giả):

$$D(x) = \sigma(W_4 \cdot \sigma(W_3 \cdot x + b_3) + b_4) \quad (3)$$

Trong đó:

W_3, W_4 là trọng số của các tầng.

b_3, b_4 là bias.

Hàm kích hoạt $\sigma(x)$ là ReLU cho lớp ẩn và sigmoid cho đầu ra.

Hàm mất mát (Loss Function)

Hàm mất mát của Discriminator:

$$L_D = -E_{x \sim p_{real}(x)} [\log D(x)] - E_{z \sim p_z(z)} [\log (1 - D(G(z)))] \quad (4)$$

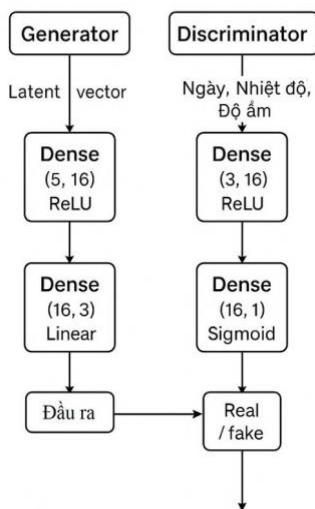
Cực đại hóa L_D giúp $D(x)$ phân biệt dữ liệu thật và giả.

Hàm mất mát của Generator:

$$L_G = -E_{z \sim p_z(z)} [\log D(G(z))] \quad (5)$$

Cực tiểu hóa L_G giúp $G(z)$ tạo ra dữ liệu có xác suất được nhận diện là lớn nhất.

Kiến trúc của Generator và Discriminator được minh họa ở Hình 1.



Hình 1. Kiến trúc của Generator và Discriminator

2.3 Mạng bộ nhớ ngắn hạn dài (Long short term memory - LSTM)[11][13]

LSTM (Long Short-Term Memory) là một kiến trúc mạng nơron được thiết kế để ghi nhớ thông tin trong khoảng thời gian dài. Nó được giới thiệu lần đầu bởi Sepp Hochreiter và Jürgen Schmidhuber vào năm 1997 [13].

LSTM đã được sửa đổi theo một số cách quan trọng cho phép chúng giải thích dữ liệu trong quá khứ bằng các phương pháp ưu việt. Các thay đổi được thực hiện đối với LSTM giải quyết vấn đề độ dốc biến mất và cho phép LSTM xem xét các chuỗi đầu vào dài hơn nhiều.

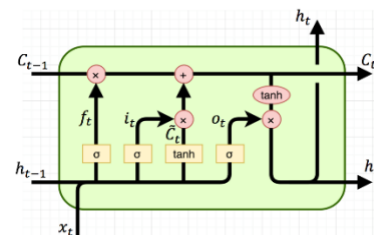
Mỗi tế bào LSTM (LSTM Cell) được cấu trúc đặc biệt với ba cổng điều khiển thông tin chính, bao gồm:

Cổng quên quyết định phần nào của thông tin cũ sẽ bị loại bỏ khỏi bộ nhớ. Nó sử dụng hàm sigmoid để đánh giá mức độ cần thiết của từng thông tin. Giá trị đầu ra gần 0 nghĩa là thông tin đó sẽ bị quên, gần 1 nghĩa là giữ lại.

Cổng đầu vào xác định thông tin mới nào từ đầu vào hiện tại sẽ được lưu vào bộ nhớ. Nó bao gồm hai thành phần: Một hàm sigmoid quyết định phần nào nên được ghi nhớ. Một hàm tanh tạo ra ứng viên bộ nhớ mới, chứa thông tin cần thêm vào. Kết quả từ hai hàm này sẽ kết hợp để cập nhật bộ nhớ hiện tại.

Cổng đầu ra quyết định thông tin nào sẽ được đưa ra làm đầu ra của tế bào LSTM. Nó dùng hàm sigmoid để xác định phần quan trọng, sau đó nhân với hàm tanh của bộ nhớ hiện tại để tạo ra đầu ra.

Cấu trúc LSTM được mô tả ở Hình 2



Hình 2. Cấu trúc LSTM

3. Thiết kế hệ thống

3.1. Nhận dạng máy áp trùng dùng mạng nơron đối kháng

Bài toán đặt ra là xây dựng mô hình kiểm tra và nhận dạng máy áp trùng dựa trên dữ liệu về chu kỳ áp. Với dữ liệu đầu vào của mô hình bao gồm ba thông số: Ngày trong chu kỳ áp, nhiệt độ máy áp (°C) và độ ẩm máy áp (%). Dữ liệu đưa vào nhận dạng là Bảng 1. Từ đó, đưa ra mục tiêu là phân loại, kiểm tra máy áp có tuân thủ đúng

chu kỳ ấp trứng gà Đông Tảo hay không.

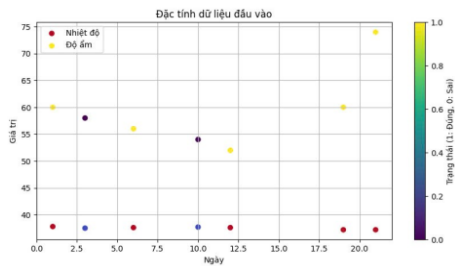
Từ phương trình (2) đầu ra $G(z)$ là vector 3 chiều tương ứng với [Ngày, Nhiệt độ, Độ ẩm]. Đầu ra $G(z)$ là vector 3 chiều tương ứng với [Ngày, Nhiệt độ, Độ ẩm].

Dữ liệu đầu vào bao gồm các cặp giá trị ngày, nhiệt độ, độ ẩm và nhãn (Đúng/ Sai) như Bảng 2:

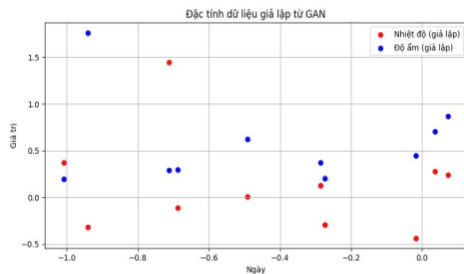
Bảng 2. Dữ liệu mẫu

Ngày	Nhiệt độ	Độ ẩm	Nhãn
1	37.8	60	Đúng
3	37.5	58	Sai
6	37.6	56	Đúng
12	37.6	52	Đúng
19	37.2	60	Đúng
21	37.2	74	Đúng

Để dự đoán máy ấp trứng có hoạt động đúng chu kỳ hay không, nhóm tác giả đã áp dụng mạng nơron đối kháng để tạo dữ liệu giả lập và kiểm tra các điều kiện của máy ấp. Nếu muốn mở rộng dữ liệu mẫu hoặc kiểm tra hiệu năng trên các tình huống khác nhau, có thể sử dụng mô hình GAN để tạo dữ liệu giả, sau đó Discriminator kiểm tra tính hợp lệ. Đồng thời, sử dụng mạng phân loại để huấn luyện mạng nơron, từ đó nhận dạng xem máy ấp có hoạt động đúng chu kỳ hay không như ở Bảng 2. Sau khi chạy code mô phỏng dùng ngôn ngữ Python, ngoài kết quả dự đoán đã trình bày ở Bảng 2 thì đặc tính dữ liệu đầu vào và dữ liệu giả lập từ GAN thu được ở Hình 3 và Hình 4.



Hình 3. Đặc tính dữ liệu đầu vào



Hình 4. Đặc tính dữ liệu giả lập từ GAN

3.2. Xây dựng hệ thống điều khiển

Mô hình mạng LSTM 10 lớp gồm:

3 lớp đầu vào, 3 lớp LSTM: LSTM 128, LSTM 64, LSTM 32 để học các mẫu thời gian. 3 lớp Dense để xử lý dữ liệu phi tuyến. và 1 lớp đầu ra dự đoán điều chỉnh nhiệt độ và độ ẩm.

Mỗi nút LSTM có ba thành phần chính:

Cổng quên: Xác định thông tin nào cần bỏ:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (6)$$

Cổng đầu vào: Quyết định thông tin mới cần lưu

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (7)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (8)$$

Cập nhật trạng thái bộ nhớ:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (9)$$

Cổng đầu ra: quyết định đầu ra của LSTM:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (10)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (11)$$

Trong đó: W_f, W_i, W_c, W_o là các trọng số

b_f, b_i, b_c, b_o là các bias.

Các lớp Dense

Sau LSTM, các lớp Dense áp dụng quy tắc:

$$y = \text{ReLU}(W \cdot x + b) \quad (12)$$

Hàm mất mát và tối ưu hóa

Hàm mất mát Mean Squared Error (MSE):

$$L = \frac{1}{N} \sum_{i=1}^N (y_i - \tilde{y}_i)^2 \quad (13)$$

Mô hình sử dụng Adam optimizer để cập nhật trọng số.

Sau khi dùng GAN để mô hình toán máy ấp trứng, bài toán điều khiển hệ thống cần các yếu tố sau: duy trì nhiệt độ và độ ẩm theo yêu cầu của từng giai đoạn ấp. Nhận các thông tin từ các cảm biến (nhiệt độ, độ ẩm) để tính toán tín hiệu điều khiển. Sử dụng mạng nơron hồi quy để dự đoán và điều chỉnh. Như vậy, để điều khiển hệ thống ta cần các tín hiệu đầu vào bao gồm: Ngày trong chu kỳ ấp, nhiệt độ thực tế và độ ẩm hiện tại. Tín hiệu đầu ra là: mức điều chỉnh quạt/ nhiệt độ và mức điều chỉnh phun sương để đạt yêu cầu. Để xây dựng hệ thống điều khiển nhiệt độ và độ ẩm cho máy ấp trứng gia cầm bằng mạng nơron hồi quy sâu (10 lớp) ta xác định được

các thành phần sau:

Input Layer: Nhận dữ liệu đầu vào

Recurrent Layers: Bao gồm các lớp LSTM/ GRU để học mối quan hệ theo chuỗi thời gian

Hidden Dense Layers: Dùng các lớp Fully connected để tinh chỉnh đầu ra

Output Layer: Đưa ra tín hiệu điều khiển (giá trị điều chỉnh nhiệt độ và độ ẩm)

4. Mô phỏng hệ thống

Triển khai bằng ngôn ngữ Python, sử dụng TensorFlow/ Keras để xây dựng hệ thống điều khiển. Dữ liệu đầu vào của hệ thống bao gồm ngày, nhiệt độ hiện tại và độ ẩm hiện tại, được thu thập từ cảm biến môi trường.

Mạng nơron LSTM đóng vai trò quan trọng trong quá trình ra quyết định của hệ thống. Thông qua quá trình huấn luyện trên tập dữ liệu lịch sử, mô hình có khả năng dự đoán các tín hiệu điều khiển phù hợp dựa trên các điều kiện môi trường hiện tại. Cụ thể, hệ thống sẽ đưa ra các tín hiệu điều chỉnh như bật/tắt bộ gia nhiệt hoặc quạt làm mát để điều chỉnh nhiệt độ, cũng như bật/tắt hệ thống phun sương để điều chỉnh độ ẩm. Việc điều chỉnh hệ thống giúp đảm bảo môi trường trong máy ấp luôn duy trì ở mức tối ưu, phù hợp với điều kiện cần thiết để trứng phát triển một cách tốt nhất[11][13].

Nhờ vào khả năng học sâu của mạng nơron LSTM, hệ thống có thể thích ứng với sự thay đổi của môi trường đưa ra quyết định điều khiển chính xác theo thời gian thực. Điều này giúp duy trì sự ổn định của nhiệt độ và độ ẩm trong máy ấp trứng mà không cần sự can thiệp liên tục của con người. Từ đó, hệ thống góp phần nâng cao tỷ lệ nở của trứng, tối ưu hóa quá trình ấp và giảm thiểu rủi ro do biến động môi trường.

Quá trình kiểm tra và tinh chỉnh bao gồm việc hiệu chỉnh mô hình để giảm sai số dự đoán, sử dụng các chỉ số đánh giá như MSE (Mean Squared Error) hoặc MAE (Mean Absolute Error). Ngoài ra, hệ thống cần được theo dõi và cập nhật dữ liệu thực tế liên tục nhằm cải thiện độ chính xác và khả năng thích ứng với điều kiện môi trường thay đổi.

4.1. Quy trình huấn luyện và kết quả mô phỏng điều khiển nhiệt độ và độ ẩm máy ấp trứng gà Đông Tảo giai đoạn từ ngày 1 đến ngày 5

A. Input Layer (Lớp đầu vào)

Dữ liệu đầu vào có dạng (samples, time_steps, features), trong đó:

samples: số lượng mẫu dữ liệu.

time_steps = 1: mỗi mẫu chỉ có một bước thời gian.
features = 3: gồm Ngày, Nhiệt độ hiện tại, Độ ẩm hiện tại.

B. Lớp LSTM1 – (128 đơn vị nơron, return_sequences=True) 128 đơn vị LSTM giúp học các đặc trưng quan trọng từ dữ liệu đầu vào.

return_sequences=True giữ nguyên chuỗi đầu ra để truyền sang lớp LSTM tiếp theo.

Lớp Dropout (0.2) giúp giảm overfitting bằng cách loại bỏ ngẫu nhiên 20% nơron trong quá trình huấn luyện.

C. Lớp LSTM2 – (64 đơn vị nơron, return_sequences=True)

Giúp trích xuất đặc trưng sâu hơn từ dữ liệu tuần tự.
return_sequences=True để tiếp tục truyền dữ liệu sang lớp LSTM tiếp theo.

Lớp Dropout (0.2) tiếp tục giảm overfitting.

D. Lớp LSTM3 – (32 đơn vị nơron, return_sequences=False)

return_sequences=False vì đây là lớp cuối cùng, chỉ lấy đầu ra cuối cùng của chuỗi.

Mô hình không cần giữ lại toàn bộ chuỗi đầu ra.

E. Các Lớp Dense (Lớp ẩn)

Sau các lớp LSTM, chúng ta sử dụng các lớp Dense để học các đặc trưng phi tuyến.

Lớp 64 nơron + ReLU giúp tiếp tục học các đặc trưng quan trọng.

Lớp Dropout (0.2) giảm overfitting.

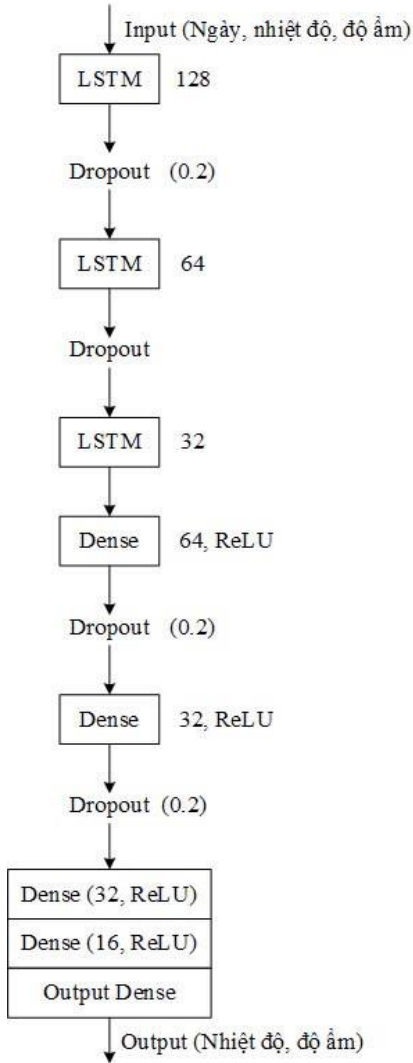
Lớp 32 nơron + ReLU và Lớp 16 nơron + ReLU tăng tính phi tuyến

F. Lớp Đầu Ra

Lớp Dense cuối cùng có 2 nơron tương ứng với Điều chỉnh nhiệt độ và độ ẩm.

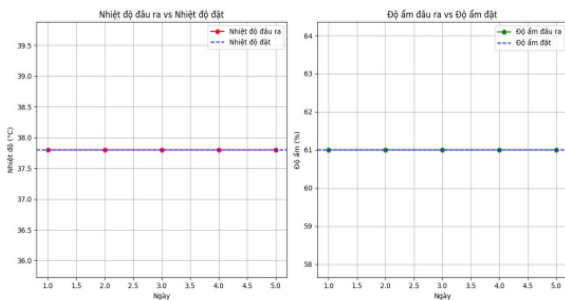
Hàm kích hoạt linear vì đầu ra là giá trị thực.

Ta có kiến trúc mạng như Hình 5:

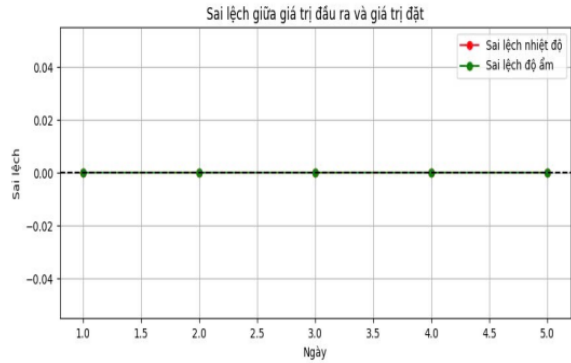


Hình 5. Cấu trúc mạng nơron

Kết quả mô phỏng hai thông số đầu ra là nhiệt độ và độ ẩm như Hình 6



Hình 6. Đặc tính nhiệt độ và độ ẩm máy áp trùng giai đoạn từ ngày 1 đến ngày 5

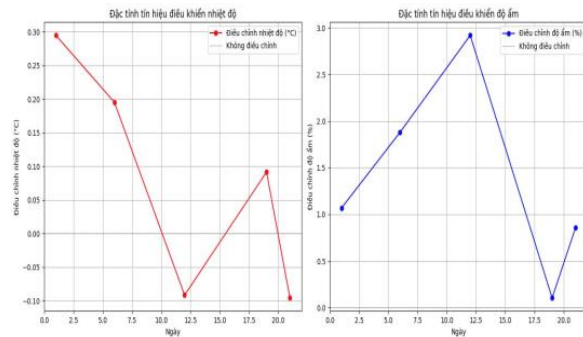


Hình 7. Sai lệch giữa giá trị đầu ra và giá trị đặt

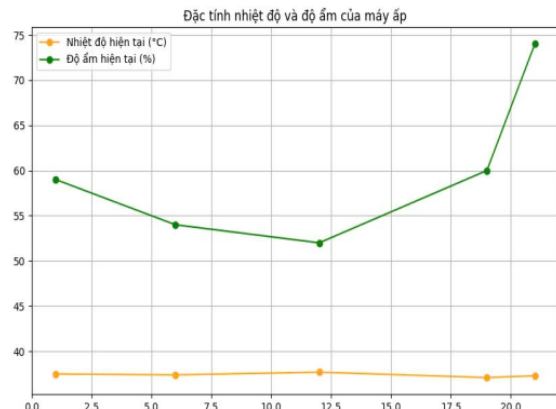
Sai lệch giữa nhiệt độ, độ ẩm của máy áp và giá trị nhiệt độ và độ ẩm đặt ở Hình 4 cho kết quả là 0 đảm bảo tính chính xác.

4.2. Kết quả mô phỏng cả chu kỳ áp trướng

Sau khi chạy chương trình, ta thu được kết quả mô phỏng như sau:



Hình 8. Đặc tính tín hiệu điều khiển nhiệt độ, độ ẩm



Hình 9. Đặc tính nhiệt độ và độ ẩm của máy áp

Kết quả mô phỏng trên Hình 9 cho thấy sử dụng mạng nơron trong việc điều khiển nhiệt độ và độ ẩm lò ấp trứng gà Đông Tảo đảm bảo độ chính xác cao. Hệ thống điều khiển thông minh đã xử lý thông tin, dữ liệu cho phép điều khiển được cả quá trình với các dữ liệu thay đổi đúng như mong muốn về độ ẩm và nhiệt độ được chia làm các giai đoạn trong 21 ngày ấp trứng gà Đông Tảo đó là từ ngày 1÷5 của chu kỳ ấp yêu cầu nhiệt độ máy 37,8 °C, độ ẩm 60÷61 %. Từ ngày 6÷11 của chu kỳ ấp yêu cầu nhiệt độ máy 37,6 °C, độ ẩm 55÷57 %. Từ ngày 12÷18 của chu kỳ ấp yêu cầu nhiệt độ máy 37,6 °C, độ ẩm 50÷53 %. Ngày thứ 19 yêu cầu nhiệt độ máy 37,2 °C, độ ẩm 60 %. Từ ngày 20÷21 của chu kỳ ấp yêu cầu nhiệt độ máy 37,2 °C, độ ẩm 70÷75 %.

Tuy nhiên so sánh kết quả trên Hình 6, Hình 7 cho thấy khi điều khiển một giai đoạn 5 ngày đầu của chu kỳ ấp trứng cho kết quả chính xác hơn so với kết quả trên Hình 9 điều khiển cả chu kỳ.

5. Kết luận

Hệ thống điều khiển tự động được phát triển nhằm tối ưu hóa quá trình ấp trứng thông qua việc duy trì nhiệt độ và độ ẩm ở mức chính xác trong từng giai đoạn phát triển của phôi. Mô hình có khả năng dự đoán và điều chỉnh kịp thời các biến động môi trường, chẳng hạn như sự thay đổi nhiệt độ bên ngoài, giúp đảm bảo điều kiện tối ưu bên trong máy ấp. Nhờ vào khả năng học và thích nghi của mạng nơron LSTM, hệ thống có thể đưa ra các điều chỉnh phù hợp theo thời gian thực, hạn chế sai số và tối ưu hóa hiệu suất hoạt động. Bên cạnh đó, hệ thống được thiết kế thông minh, linh hoạt, giúp tiết kiệm năng lượng bằng cách chỉ kích hoạt các thiết bị khi cần thiết. Điều này không chỉ giảm chi phí vận hành mà còn nâng cao hiệu quả ấp trứng, góp phần gia tăng tỷ lệ nở và cải thiện chất lượng con giống.

Tài liệu tham khảo

1. Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934. <https://arxiv.org/abs/2004.10934>
2. Boleli, I. C., Paulillo, A. C., & Maiorka, A. (2016). Poultry egg incubation: Integrating and optimizing production efficiency. *Brazilian Journal of Poultry Science*, 18(2), 1–16.
3. Chlap, P., Min, H., Vandenberg, N., Dowling, J., Holloway, L., & Haworth, A. (2021). A review of medical image data augmentation techniques for deep

learning applications. *Journal of Medical Imaging and Radiation Oncology*, 65(5), 545–563. <https://doi.org/10.1111/1754-9485.13261>

4. Islam, M. M., Tonmoy, S. S., Quayum, S., Sarker, A. R., Hani, S. U., & Mannan, M. A. (2019, January). Smart poultry farm incorporating GSM and IoT. In 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST) (pp. 277–280). IEEE.

5. Mahale, R. B., & Sonavane, S. S. (2016). Smart poultry farm: An integrated solution using WSN and GPRS based network. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 5(6), 1984–1988.

6. Massey, M. D., & Hutchings, J. A. (2021). Thermal variability during ectotherm egg incubation: A synthesis and framework. *Journal of Experimental Zoology Part A: Ecological and Integrative Physiology*, 335(1), 59–71. <https://doi.org/10.1002/jez.2400>

7. Mesquita, M. A., Mesquita, M. A., et al. (2021). Results of hatching and rearing broiler chickens in different incubation systems. *Poultry Science*, 100(1), 94–102. <https://doi.org/10.1016/j.psj.2020.09.028>

8. Nielsen, M. (2017). *Neural Networks and Deep Learning*. Retrieved June 10, 2017, from <http://neuralnetworksanddeeplearning.com/chap4.html>

9. Ohpagu, P. E., & Nwosu, A. W. (2016). Development and temperature control of smart egg incubator system for various types of egg. *European Journal of Engineering and Technology*, 4(2), 13–21.

10. Rosenblatt, F. (1957). The perceptron: A perceiving and recognizing automaton (Tech. Rep. No. 85-460-1). Cornell Aeronautical Laboratory.

11. Sutskever, I. (2012). Training recurrent neural networks (Doctoral dissertation, University of Toronto).

12. Sutskever, I., Martens, J., & Hinton, G. E. (2011). Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning* (pp. 1017–1024). 21.

13. Sepp Hochreiter, Long Short-Term Memory, November 1997; 9(8):1735-1780 DOI:10.1162/neco.1997.9.8.1735.

14. S. Reed, Z. Akata, X. C. Yan, L. Logeswaran, B. Schiele, and H. Lee, “Generative adversarial text to image synthesis,” in *Proc. 33rd Int. Conf. Machine Learning*, New York, NY, USA, vol. 48, pp. 1060–1069, 2016.