

**ĐIỀU KHIỂN VÀ GIÁM SÁT
MÔ HÌNH NHÀ KÍNH THÔNG MINH ỨNG DỤNG
IOT(INTERNET OF THINGS) VÀ THUẬT TOÁN
SUPPORT VECTOR MACHINES
CONTROL AND SUPERVISE
THE SMART GREENHOUSE MODEL USE
IOT (INTERNET OF THINGS) AND SUPPORT VECTOR MACHINES
ALGORITHM**

Trương Ngọc Hải Nam¹, Vũ Quang Huy²

¹Học viên cao học trường ĐH SPKT TPHCM

²Trường đại học Sư Phạm Kỹ Thuật TPHCM

TÓM TẮT

Từ những yêu cầu thực tế, những đòi hỏi ngày càng cao của cuộc sống đó, cộng với sự hợp tác, phát triển mạnh mẽ của mạng di động và cuộc cách mạng công nghiệp 4.0, đề tài “**Điều khiển và giám sát mô hình nhà kính thông minh sử dụng IoT (Internet of things) và thuật toán Support Vector Machines**” đã tạo nên một xu thế mới trong việc nghiên cứu nhà kính thông minh với sự kết hợp của thuật toán Support Vector Machines.

Đây cũng chính là tiền đề cho xu thế phát triển nông nghiệp công nghệ cao trong nước và trên toàn thế giới.

ABSTRACT

With the growing demand for mobile life and the industrial revolution of 4.0, I chose the subject "Control and supervise the smart greenhouse model use IoT (Internet of Things) and Support Vector Machines algorithm". The thesis has created a new trend in smart glass research with the combination of Support Vector Machines algorithm.

This is also the premise for the development of high-tech agriculture in the country and around the world.

1. GIỚI THIỆU

Ngày nay cùng với sự phát triển mạnh mẽ của các ngành khoa học kỹ thuật, công nghệ kỹ thuật điện tử mà trong đó đặc biệt là kỹ thuật điều khiển tự động đóng vai trò quan trọng trong mọi lĩnh vực khoa học kỹ thuật, quản lí, công nghiệp, cung cấp thông tin ... Bên cạnh đó còn là sự thúc đẩy sự phát triển của nền kinh tế nước nhà.

Như chúng ta đã biết, gần như tất cả các nhà kính trên thế giới đều sử dụng những công nghệ thô sơ và lạc hậu. Các công việc như bón phân, tưới cây... đều phải thông qua bàn tay con người. Đặc biệt với những vấn đề thời tiết như nhiệt độ, độ ẩm, độ sáng, tốc độ gió [1]... đều phụ thuộc nhiều vào thời tiết. Vì vậy mà chất lượng, năng suất nông sản không đạt mà lại tốn

quá nhiều nhân công, thời gian, chi phí. Nhà kính thông minh ra đời để đáp ứng nhu cầu trên.

Đề tài lấy cơ sở là dùng Web Server để điều khiển và giám sát nhà kính [2], kết hợp thuật toán Support vector machines để phân lớp dữ liệu [3], đánh giá độ chính xác phân lớp và dự đoán trạng thái nhà kính để đưa ra lệnh điều khiển cho nhà kính. Với sự kết hợp của IoT và thuật toán Support Vector Machines con người chỉ cần giám sát và không cần điều khiển bất cứ quá trình nào. Việc sử dụng Web để điều khiển thiết bị có thuận lợi là tiết kiệm chi phí, mang tính cạnh tranh. Ngoài ra, sản phẩm của đề tài này có tính mở, có thể áp dụng cho nhiều đối tượng khác nhau trong dân dụng cũng như trong công nghiệp.

Một môi trường nhà kính thông minh có thể được hiểu là một môi trường mà được trang bị hệ thống tiên tiến dành cho điều khiển đèn chiếu sáng, nhiệt độ, độ ẩm, độ PH, quạt thông gió và nhiều tính năng khác nhằm mục đích làm cho chất lượng và năng suất nông sản là cao nhất, giảm bớt chi phí nuôi trồng đồng thời tăng chất lượng an toàn nông sản [2].



Hình 1: Tổng quan các thiết bị sử dụng trong luận văn

- Nhà kính thông minh gồm 4 phần chính: Chế độ điều khiển, giao diện điều khiển, cơ sở dữ liệu và thuật toán.

+ Chế độ điều khiển bao gồm 2 chế độ: chế độ bằng tay và chế độ tự động.

+ Giao diện điều khiển gồm giao diện đăng nhập, chế độ điều khiển, tình trạng hoạt động ON/OFF của các thiết bị đặt trong hệ thống; hiển thị các bảng thông số và các đồ thị thông số cảm biến.

+ Cơ sở dữ liệu nhà kính lưu lại toàn bộ dữ liệu nhà kính để đảm bảo cho quá trình phân tích thu thập dữ liệu mọi thời điểm.

+ Phần thuật toán sử dụng thuật toán Support Vector Machines để đánh giá độ chính xác phân lớp và dự đoán, ra quyết định trạng thái nhà kính, sau đó lấy trạng thái làm cơ sở điều khiển nhà kính.

2. THUẬT TOÁN MÁY HỌC

Máy học (tiếng Anh: machine learning) là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng các kỹ thuật cho phép các hệ thống "học" tự động từ dữ liệu để giải quyết những vấn đề cụ thể. Ví dụ như các máy có thể "học" cách phân loại thư điện tử xem có phải thư rác (spam) hay không và tự động xếp thư vào thư mục tương ứng. Máy học rất gần với suy diễn thống kê (statistical inference) tuy có khác nhau về thuật ngữ [4].

Các thuật toán máy học được phân loại theo kết quả mong muốn của thuật toán. Các loại thuật toán thường dùng dùng bao gồm:

- Học có giám sát
- Học không giám sát
- Học nửa giám sát
- Học tăng cường
- Chuyển đổi
- Học cách học

2.1 Ứng dụng SVM vào nhà kính thông minh

2.1.1 Ý tưởng của phương pháp

Cho trước một tập huấn luyện, được biểu diễn trong không gian vector, trong đó mỗi tài liệu là một điểm, phương pháp này tìm ra một siêu phẳng quyết định tốt nhất có thể chia các điểm trên không gian này thành hai lớp riêng biệt tương ứng là lớp + và lớp -. Chất lượng của siêu phẳng này được quyết định bởi khoảng cách (gọi là biên) của điểm dữ liệu gần nhất của mỗi lớp đến mặt phẳng này. Khi đó, khoảng cách biên càng lớn thì mặt phẳng quyết định càng tốt, đồng thời việc phân loại càng chính xác [5][6][7].

Mục đích của thuật toán SVM là tìm được khoảng cách biên lớn nhất.

2.1.2 Nội dung

a. Cơ sở lý thuyết

SVM thực chất là bài toán tối ưu, mục tiêu của thuật toán này là tìm được một không gian F và siêu phẳng quyết định f trên F sao cho sai số phân loại là thấp nhất.

Cho tập mẫu $(x_1, y_1), (x_2, y_2), \dots, (x_f, y_f)$ với $x_i \in \mathbb{R}^n$, thuộc vào hai lớp nhãn: $y_i \in \{-1; 1\}$ là nhãn lớp tương ứng của các x_i (-1 biểu thị lớp I, 1 biểu thị lớp II).

Ta có phương trình siêu phẳng chứa vector \vec{x}_i trong không gian:

$$\vec{X}_i \cdot \vec{w} + b = 0 \quad (1)$$

Đặt:

$$f(\vec{X}_i) = \sin(\vec{X}_i \cdot \vec{w} + b) = \begin{cases} +1, & \vec{X}_i \cdot \vec{w} + b > 0 \\ -1, & \vec{X}_i \cdot \vec{w} + b < 0 \end{cases} \quad (2)$$

Như vậy $f(\vec{X}_i)$ biểu diễn sự phân lớp của \vec{X}_i vào hai lớp đã nêu. Ta nói $y_i = +1$ nếu $\vec{X}_i \in$ lớp I và $y_i = -1$ nếu $\vec{X}_i \in$ lớp II. Khi đó, để có siêu phẳng f ta phải giải bài toán sau:

Tìm $\min \|\vec{w}\|$ với W thỏa mãn điều kiện sau:

$$y_i(\sin(\vec{X}_i \cdot \vec{w} + b)) \geq 1 \text{ với } \forall i \in \overline{1, n} \quad (3)$$

Bài toán SVM có thể giải bằng kỹ thuật sử dụng toán tử Lagrange để biến đổi về thành dạng đẳng thức. Một đặc điểm thú vị của SVM là mặt phẳng quyết định chỉ phụ thuộc các Support Vector và nó có khoảng cách đến mặt phẳng quyết định là $1/\|\vec{w}\|$. Cho dù các điểm khác bị xóa đi thì thuật toán vẫn cho kết quả giống như ban đầu. Đây chính là điểm nổi bật của thuật toán SVM so với các phương pháp khác vì tất cả các dấu hiệu trong tập huấn luyện đều được dùng để tối ưu hóa kết quả.

Tóm lại: trong trường hợp nhị phân tách tuyến tính, việc phân lớp được thực hiện qua hàm quyết định $f(x) = \text{sign}(\langle w, x \rangle + b)$, hàm này thu được bằng việc thay đổi vector chuẩn w , đây là vector để cực đại hóa viên chức năng.

Việc mở rộng SVM để phân đa lớp hiện nay vẫn đang được đầu tư nghiên cứu. Có một phương pháp tiếp cận để giải quyết vấn đề này là xây dựng và kết hợp nhiều bộ phân lớp nhị phân SVM (chẳng hạn: trong quá trình luyện với SVM, bài toán phân m lớp có thể được biến đổi thành bài toán phân $2 * m$ lớp, khi đó trong mỗi hai lớp, hàm quyết định sẽ được xác định cho khả năng tổng quát hóa tối đa). Trong cách này có thể đề cập tới 2 cách một đối một và một đối tất cả.

c. Bài toán phân 2 lớp

Bài toán đặt ra là: Xác định hàm phân lớp để phân lớp các mẫu trong tương lai, nghĩa là với 1 mẫu dữ liệu mới x_i thì cần phải xác định x_i được phân vào lớp +1 hay lớp -1.

Để xác định hàm phân lớp dựa trên thuật toán SVM, ta sẽ tiến hành tìm 2 siêu phẳng song song sao cho khoảng cách y giữa chúng là lớn nhất có thể để phân tách 2 lớp này ra làm hai

phía. Hàm phân tách tương ứng với phương trình siêu phẳng tìm được.

Các điểm mà nằm trên hai siêu phẳng phân tách được gọi là các Support Vector. Các điểm này sẽ quyết định đến hàm phân tách dữ liệu.

c. Bài toán nhiều phân lớp

Để phân nhiều lớp thì kỹ thuật SVM nguyên thủy sẽ chia không gian dữ liệu thành 2 phần và quá trình này lặp lại nhiều lần. Khi đó hàm quyết định phân dữ liệu vào lớp thứ I của tập n , 2- lớp là:

$$f_i(x) = w_i^T x + b \quad (4)$$

Nhưng phân tử x là support vector sẽ thỏa điều kiện:

$$f_i(x) = \begin{cases} +1 & \text{nếu thuộc lớp } i \\ -1 & \text{nếu thuộc phần còn lại} \end{cases} \quad (5)$$

Như vậy, bài toán phân nhiều lớp sử dụng thuật toán SVM hoàn toàn có thể thực hiện giống như bài toán hai lớp. Bằng cách sử dụng bằng cách sử dụng chiến lược “một- đối- một”(one- against- one).

Giả sử bài toán cần phân loại có k lớp ($k > 2$), chiến lược “một- đối- một” sẽ tiến hành $k(k-1)/2$ lần phân lớp nhị phân sử dụng thuật toán SVM. Mỗi lớp sẽ tiến hành phân tách với $k-1$ lớp còn lại để xác định $k-1$ hàm phân tách dựa vào bài toán phân hai lớp bằng thuật toán SVM.

d. Các bước chính của SVM cho nhà kính

Thuật toán SVM yêu cầu dữ liệu được diễn tả như các vector của các số thực. Đầu vào của thuật toán SVM ta sử dụng phải là số thực.

Tiền xử lý dữ liệu: Thực hiện biến đổi dữ liệu phù hợp cho quá trình tính toán, tránh các số quá lớn mô tả các thuộc tính. Thường nên co giãn (scaling) dữ liệu để chuyển về đoạn $[-1, 1]$ hoặc $[0, 1]$.

Chọn hàm hạt nhân: Lựa chọn hàm hạt nhân chính xác để đạt được độ chính xác cao trong quá trình phân lớp.

Thực hiện việc kiểm tra chéo để xác định các tham số cho ứng dụng. Điều này cũng quyết định đến tính chính xác của quá trình phân lớp.

Sử dụng các tham số cho việc huấn luyện với tập mẫu. Trong quá trình huấn luyện sẽ sử dụng thuật toán tối ưu hóa khoảng cách giữa các siêu phẳng trong quá trình phân lớp, xác định hàm phân lớp trong không gian đặc trưng nhờ việc ánh xạ dữ liệu vào không gian đặc trưng bằng cách mô tả hạt nhân, giải quyết cho cả hai trường hợp dữ liệu là phân tách và không phân tách tuyến tính trong không gian đặc trưng.

3. XÂY DỰNG WEB SERVER

3.1 Mã nguồn Node.js

3.1.1 Giới thiệu Node.js

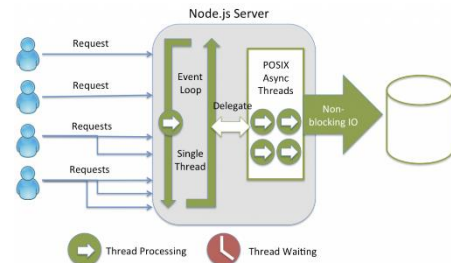
Node.js là một mã nguồn được xây dựng dựa trên nền tảng Javascript V8 Engine, nó được sử dụng để xây dựng các ứng dụng web như các trang video clip, các forum và đặc biệt là trang mạng xã hội phạm vi hẹp.

3.1.2 Các đặc tính của Node.js

- Không đồng bộ: Tất cả các API của Node.js đều không đồng bộ (non-blocking), nó chủ yếu dựa trên nền của Node.js Server và chờ đợi Server trả dữ liệu về.
- Chạy rất nhanh: Node.js được xây dựng dựa vào nền tảng V8 Javascript Engine nên việc thực thi chương trình rất nhanh.
- Đơn luồng nhưng khả năng mở rộng cao: Node.js sử dụng một mô hình luồng duy nhất với sự kiện lập.
- Không đệm: Node.js không đệm bất kỳ một dữ liệu nào và các ứng dụng này chủ yếu là đầu ra dữ liệu.
- Có giấy phép: Node.js đã được cấp giấy phép bởi MIT License.

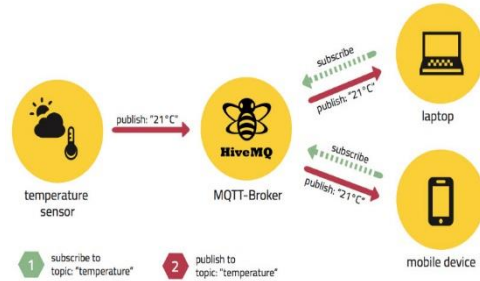
3.1.3 Hoạt động của Node.js

Node.js sử dụng non-blocking, hướng sự vào ra dữ liệu thông qua các tác vụ thời gian thực một cách nhanh chóng. Bởi vì, Node.js có khả năng mở rộng nhanh chóng, khả năng xử lý một số lượng lớn các kết nối đồng thời bằng thông lượng cao. Nếu như các ứng dụng web truyền thống, các request tạo ra một luồng xử lý yêu cầu mới và chiếm RAM của hệ thống thì việc tài nguyên của hệ thống sẽ được sử dụng không hiệu quả. Chính vì lẽ đó giải pháp mà Node.js đưa ra là sử dụng luồng đơn (Single-Threaded), kết hợp với non-blocking I/O để thực thi các request, cho phép hỗ trợ hàng chục ngàn kết nối đồng thời [8].



Hình 2: Sơ đồ tổng quát của Node.js [9]

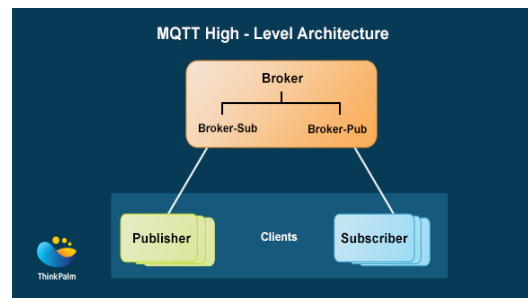
3.2 Giới thiệu giao thức MQTT



Hình 3: Ví dụ về giao thức MQTT [10]

MQTT là một giao thức truyền thông điệp (message) theo mô hình publish/subscribe (xuất bản - theo dõi), sử dụng băng thông thấp, độ tin cậy cao và có khả năng hoạt động trong điều kiện đường truyền không ổn định.

Kiến trúc mức cao (high-level) của MQTT gồm 2 phần chính là Broker và Clients.



Hình 4: Sơ đồ MQTT [11]

Trong đó, broker được coi như trung tâm, nó là điểm giao của tất cả các kết nối đến từ client. Client thì được chia thành 2 nhóm là publisher và subscriber.

Client là các software components hoạt động tại edge device nên chúng được thiết kế để có thể hoạt động một cách linh hoạt (lightweight).

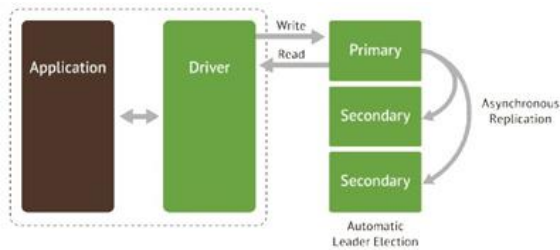
3.3 Cơ sở dữ liệu MongoDB

3.3.1 Khái niệm

MongoDB là một cơ sở dữ liệu mã nguồn mở thuộc họ nhà NoSQL được phát hành sử dụng miễn phí bởi công ty MongoDB. MongoDB là một lựa chọn tốt, khi nói đến dung lượng lưu trữ

và tốc độ, vì nó mang lại hiệu quả và độ tin cậy cao. Giải pháp NoSQL này đi kèm với việc nhúng, tự động sharding, và nhân bản trên boong cho khả năng mở rộng tốt hơn và tính sẵn sàng cao.

3.3.2 Hoạt động của MongoDB



Hình 5: Mô hình hoạt động của MongoDB [12]

MongoDB hoạt động dưới một tiến trình ngầm service luôn mở một cổng để lắng nghe các yêu cầu truy vấn, thao tác từ các ứng dụng gửi vào sau đó mới tiến hành xử lý.

3.3.3 Cơ chế phân quyền và bảo mật

MongoDB có cơ chế xác thực phân quyền và bảo mật linh hoạt để quản trị viên có thể cấp phát quyền thao tác với cơ sở dữ liệu một cách đơn giản nhất.

MongoDB có thể truy cập mà không cần xác thực khi bạn tắt chế độ xác thực quyền sở hữu với cơ sở dữ liệu

3.3.4 Ưu và nhược điểm

a. Ưu điểm

- Dữ liệu lưu trữ phi cấu trúc, không có tính ràng buộc, toàn vẹn nên tính sẵn sàng cao, hiệu suất lớn và dễ dàng mở rộng lưu trữ.

- Dữ liệu được caching (ghi đệm) lên RAM, hạn chế truy cập vào ổ cứng nên tốc độ đọc và ghi cao.

b. Nhược điểm

- Không ràng buộc, toàn vẹn nên không ứng dụng được cho các mô hình giao dịch yêu cầu độ chính xác cao.

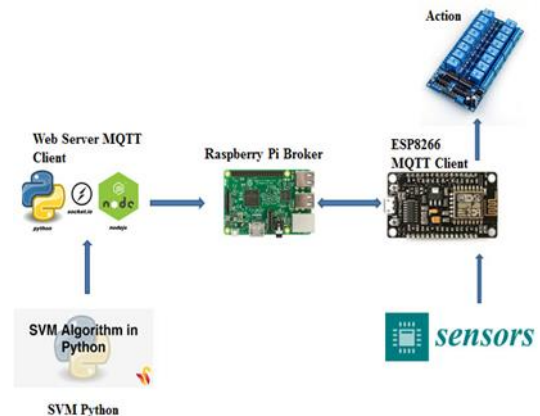
- Không có cơ chế transaction (giao dịch) để phục vụ các ứng dụng ngân hàng Dữ liệu được caching, lấy RAM làm trọng tâm hoạt động vì vậy khi hoạt động yêu cầu một bộ nhớ RAM lớn

- Như đã giới thiệu ở trên, mọi thay đổi về dữ liệu mặc định đều chưa được ghi xuống ổ cứng

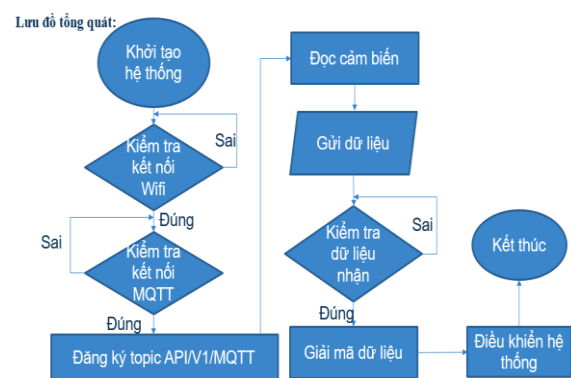
ngay lập tức vì vậy khả năng bị mất dữ liệu từ nguyên nhân mất điện đột xuất là rất cao.

4. XÂY DỰNG HỆ THỐNG

Sơ đồ tổng quát hệ thống



Hình 6: Sơ đồ tổng quát hệ thống



Hình 7: Lưu đồ tổng quát hệ thống

4.1 Xây dựng tập dữ liệu cho nhà kính thông minh

Dữ liệu được thu thập về từ hệ thống các cảm biến để định nghĩa các trạng thái của nhà kính. Trạng thái nhà kính phụ thuộc vào từng giống cây trồng.

1. Trạng thái SA(Safe Away): Trạng thái hoạt động trong nhà kính tốt nhất, các chỉ số trong mức tốt cho cây trồng.
2. Trạng thái SH(Safe Activity): Trạng thái hoạt động trong mức an toàn cho phép.
3. Trạng thái AL(Alarm): Trạng thái cảnh báo khi mức độ của các yếu tố ảnh hưởng đến nhà kính vượt mức cho phép.
4. Trạng thái EM(Emergency): Trạng thái khẩn cấp, các chỉ số vượt qua giới hạn gây ảnh hưởng lớn đến cây trồng.

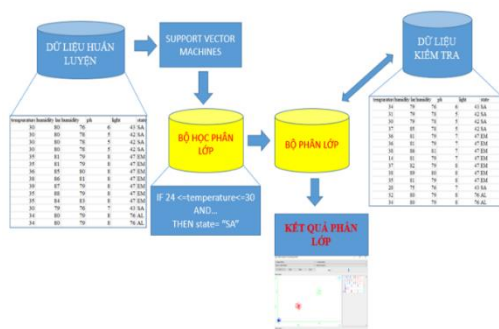
Hệ thống gồm 4 trạng thái và các thuộc tính: Nhiệt độ(TEMP), độ ẩm không khí(AIR HUMI), độ ẩm đất(LAND HUMI), độ sáng(BRIGHTNESS), độ PH(PH).

4.2 Áp dụng thuật toán Support Vector Machine trong khai phá dữ liệu nhà kính thông minh

Phân lớp dữ liệu nhà kính là một trong các phương pháp khai phá dữ liệu.

Các bước phân lớp dữ liệu:

- Bước 1: Xây dựng mô hình tập huấn luyện
- Bước 2: Sử dụng mô hình và kiểm tra tính đúng đắn của mô hình và dùng nó để phân lớp dữ liệu mới.



Hình 8: Quá trình phân lớp dữ liệu nhà kính

	A	B	C	D	E	F
1	temperaturu	humidity lan	humidity	ph	light	state
2	30	80	56	6	83	SA
3	30	80	58	5	82	SA
4	30	80	58	5	82	SA
5	30	80	58	5	82	SA
6	35	81	59	8	87	EM
7	35	81	59	8	87	EM
8	36	85	60	8	87	EM
9	38	86	61	8	87	EM
10	39	87	59	8	87	EM
11	35	88	59	8	87	EM
12	35	84	63	8	87	EM
13	30	79	56	7	83	SA
14	34	80	59	8	86	AL
15	34	80	59	8	86	AL
16	31	79	55	8	86	SH
17	31	79	55	8	86	SH
18	27	80	56	6	83	SA
19	26	80	54	5	82	SA
20	27	80	55	5	82	SA
21	28	80	58	5	82	SA
22	35	81	59	8	87	EM
23	38	81	59	8	87	EM
24	42	81	59	8	87	EM
25	45	81	59	8	87	EM
26	41	82	59	8	87	EM
27	35	82	59	8	87	EM

Bảng 1: Bảng dữ liệu huấn luyện

	A	B	C	D	E	F
1	temperaturu	humidity l	humidity	ph	light	state
2	34	79	56	6	83	SA
3	31	79	58	5	82	SA
4	30	79	58	5	82	SA
5	37	85	58	5	82	SA
6	36	81	59	7	87	EM
7	36	81	59	7	87	EM
8	38	86	61	7	87	EM
9	14	81	59	7	87	EM
10	37	82	59	8	87	EM
11	38	89	60	8	87	EM
12	35	81	59	8	87	EM
13	28	75	56	7	83	SA
14	32	80	59	8	86	AL
15	34	80	59	8	86	AL
16	24	72	55	8	86	SH
17	31	79	55	8	86	SH
18	30	80	56	6	83	SA
19	30	80	58	5	82	SA
20	30	80	58	5	82	SA
21	30	80	58	5	82	SA
22	36	81	59	8	87	EM
23	35	81	59	8	87	EM
24	33	81	59	8	87	EM
25	35	81	59	8	87	EM
26	38	87	63	8	87	EM
27	36	84	60	8	87	EM

Bảng 2: Bảng dữ liệu kiểm tra

```

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances      26          96.2963 %
Incorrectly Classified Instances    1           3.7037 %
Kappa statistic                    0.943
K&B Relative Info Score           1038.0943 %
K&B Information Score              18.3279 bits          0.6788 bits/instance
Class complexity | order 0         45.7595 bits          1.6948 bits/instance
Class complexity | scheme          27.585 bits           1.0217 bits/instance
Complexity improvement (Sf)        18.1746 bits          0.6731 bits/instance
Mean absolute error                0.2531
Root mean squared error             0.3167
Relative absolute error             76.6077 %
Root relative squared error         78.4966 %
Total Number of Instances          27
Ignored Class Unknown Instances    2
  
```

Hình 9: Kết quả phân lớp bằng thuật toán Support Vector Machines

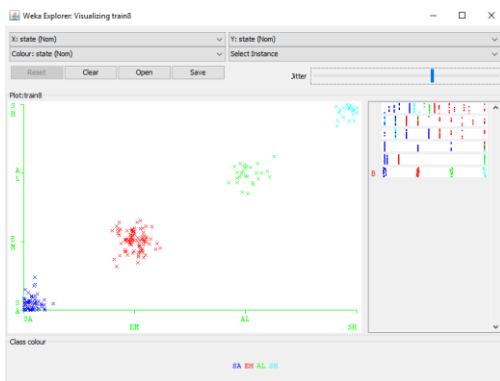
Mô hình phân lớp đạt kết quả 96.2963%, với 203 mẫu dữ liệu huấn luyện, kết quả phân lớp khi kiểm tra 29 mẫu thì mô hình phân lớp chính xác 26 mẫu, có 1 mẫu phân lớp sai và 2 mẫu chưa có nhãn lớp.

5. QUÁ TRÌNH THỰC NGHIỆM

inst#	actual	predicted	error	probability distribution
1	1:SA	1:SA	*0.5	0 0.167 0.333
2	1:SA	1:SA	*0.5	0 0.333 0.167
3	1:SA	1:SA	*0.5	0 0.333 0.167
4	1:SA	1:SA	*0.5	0.167 0.333 0
5	2:EM	2:EM	0	*0.5 0.333 0.167
6	2:EM	2:EM	0	*0.5 0.333 0.167
7	2:EM	2:EM	0	*0.5 0.333 0.167
8	2:EM	4:SH	+	0 0.167 0.333 *0.5
9	2:EM	2:EM	0	*0.5 0.333 0.167
10	2:EM	2:EM	0	*0.5 0.333 0.167
11	2:EM	2:EM	0	*0.5 0.333 0.167
12	1:SA	1:SA	*0.5	0 0.167 0.333
13	3:AL	3:AL	0	0.333 *0.5 0.167
14	3:AL	3:AL	0	0.333 *0.5 0.167
15	4:SH	4:SH	0.333	0 0.167 *0.5
16	4:SH	4:SH	0	0.167 0.333 *0.5
17	1:SA	1:SA	*0.5	0 0.167 0.333
18	1:SA	1:SA	*0.5	0 0.333 0.167
19	1:SA	1:SA	*0.5	0 0.333 0.167
20	1:SA	1:SA	*0.5	0 0.333 0.167
21	2:EM	2:EM	0	*0.5 0.333 0.167
22	?	2:EM	+	0 *0.5 0.333 0.167
23	?	2:EM	+	0 *0.5 0.333 0.167
24	2:EM	2:EM	0	*0.5 0.333 0.167
25	2:EM	2:EM	0	*0.5 0.333 0.167
26	2:EM	2:EM	0	*0.5 0.333 0.167
27	2:EM	2:EM	0	*0.5 0.333 0.167
28	1:SA	1:SA	*0.5	0 0.167 0.333
29	3:AL	3:AL	0	0.333 *0.5 0.167

Bảng 3: Khả năng ra quyết định và dự báo của thuật toán Support Vector Machines

Dựa vào 29 mẫu dữ liệu kiểm tra trong đó có 1 mẫu có nhãn sai và 2 mẫu không có nhãn. Thuật toán Support Vector Machines đã ra dự đoán kết quả của 3 nhãn trên 1 cách chính xác.



Hình 10: Sai số phân lớp

Vị trí các điểm màu cho từng nhãn lớp nằm về từng vị trí khác nhau, các điểm này càng cách xa nhau thì sai số của dữ liệu càng cao.



Hình 11: Mô hình nhà kính

Mô hình gồm phần điều khiển màn hình hiển thị lên LCD các thông số như nhiệt độ, độ ẩm không khí, độ ẩm đất, độ PH nước tưới, độ sáng toàn hệ thống.

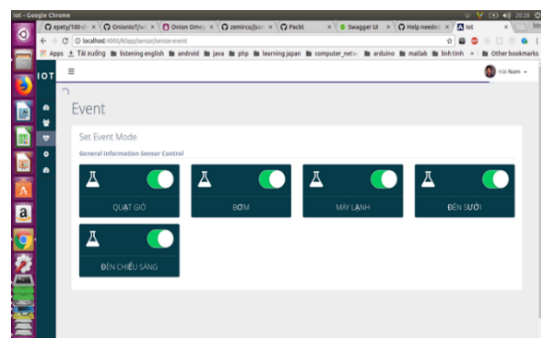
Phần cảm biến gồm: cảm biến độ ẩm đất, cảm biến độ ẩm không khí, cảm biến nhiệt, quang trở, cảm biến độ PH.

Phần cơ cấu chấp hành gồm quạt thông gió, bơm nước, đèn chiếu sáng, đèn sưởi, máy làm lạnh.

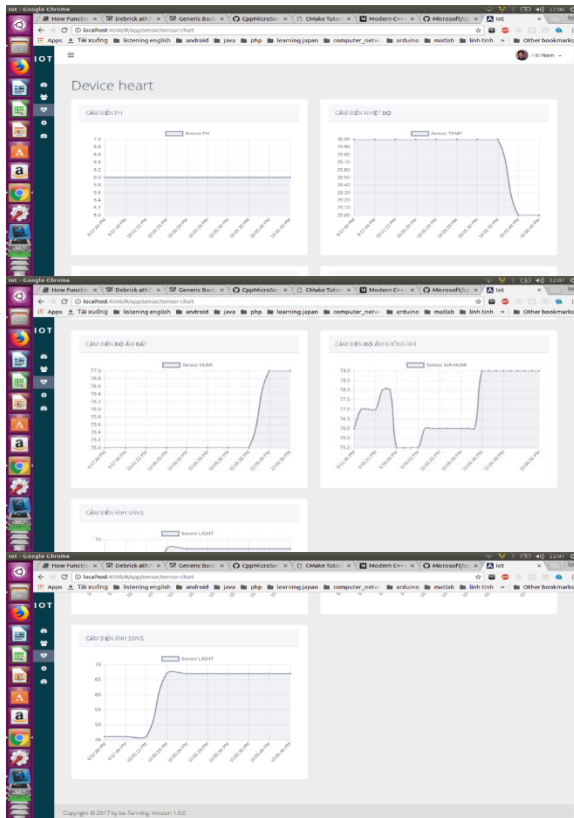
Phần trung tâm gồm bo vi điều khiển, bo giao tiếp ESP, bo nhúng và một số module...

Phần nguồn sử dụng chống nhiễu nguồn và hạ áp đảm bảo điện áp luôn ổn định, chống nhiễu từ các hệ thống xung quanh.

Giao diện điều khiển hệ thống gồm 3 phần chính: phần hiển thị các, phần điều khiển, phần báo cáo.



Hình 12: Giao diện điều khiển Web Server



Hình 13: Đồ thị quá trình thu thập dữ liệu cảm

biến. Quá trình thu thập dữ liệu cảm biến theo thời gian được hiển thị trực tiếp bằng đồ thị trực quan để đảm bảo quá trình phân tích và theo dõi hệ thống được dễ dàng hơn. Kết quả đồ thị đã cho thấy sự ổn định của hệ thống nhà kính khi sử dụng công nghệ IoT và thuật toán Support Vector Machines.

The screenshot shows a 'Sensor Table' interface with two columns of data. The left column is titled 'SENSOR PH' and the right column is titled 'SENSOR NHIỆM'. Both columns have sub-headers 'VALUE' and 'DATE'. The data is presented in a table format with alternating light and dark gray rows.

SENSOR PH		SENSOR NHIỆM	
VALUE	DATE	VALUE	DATE
8	Apr 1, 2018 12:08:02 PM	79	Apr 1, 2018 12:09:02 PM
7	Apr 1, 2018 4:15:41 PM	76	Apr 1, 2018 4:15:41 PM
8	Apr 1, 2018 4:15:31 PM	78	Apr 1, 2018 4:15:31 PM
8	Apr 1, 2018 4:15:31 PM	78	Apr 1, 2018 4:15:31 PM
8	Apr 1, 2018 4:15:29 PM	75	Apr 1, 2018 4:15:29 PM
8	Apr 1, 2018 4:15:29 PM	75	Apr 1, 2018 4:15:29 PM
8	Apr 1, 2018 4:15:29 PM	75	Apr 1, 2018 4:15:29 PM
8	Apr 1, 2018 4:15:29 PM	75	Apr 1, 2018 4:15:29 PM
8	Apr 1, 2018 4:15:29 PM	75	Apr 1, 2018 4:15:29 PM
8	Apr 1, 2018 4:15:29 PM	75	Apr 1, 2018 4:15:29 PM
8	Apr 1, 2018 4:15:29 PM	75	Apr 1, 2018 4:15:29 PM

Hình 14: Giao diện báo cáo chi tiết

6. KẾT LUẬN

Để phát triển hệ thống nhà kính thông minh, tác giả đã nghiên cứu thuật toán Support Vector Machines để huấn luyện và điều khiển. Một mô hình nhà kính đã được xây dựng và hoàn thiện để thử nghiệm thuật toán.

Để nghiên cứu thực nghiệm, việc xây dựng mô hình nhà kính thông minh để thu thập dữ liệu và đánh giá chính xác những ưu và nhược điểm của thuật toán Support Vector Machines đem lại. Hệ thống nhà kính thông minh không chỉ dừng lại ở việc điều khiển và giám sát, việc sử dụng cơ sở dữ liệu MongoDB và thuật toán SVM đã giúp cho nhà kính trở lên ổn định và thông minh hơn.

Hệ thống điều khiển và giám sát nhà kính thông minh sử dụng mã nguồn Node.js và giao thức MQTT nên hệ thống có thể xử lý hàng ngàn quá trình và xử lý một luồng dữ liệu cực lớn.

Điểm hạn chế là hệ thống chưa có thể huấn luyện mô hình bằng thuật toán Support Vector Machines theo thời gian thực.

TÀI LIỆU THAM KHẢO

- [1] VESNA DOKNÍC, 2014, Internet of Things Greenhouse Monitoring and Automation Systems, Internet of Things: Smart Devices, Processes, Services Summer term 2014.
- [2] JOHN WILEY & SONS, 2017, Internet of Things and Data Analytics Handbook, First Edition, Edited by Hwaiyu Geng.
- [3] B. BOSER, I. GUYON, V. VAPNIK, “A training algorithm for optimal margin classifiers”, Proceedings of the Fifth Annual Workshop on Computational Learning Theory (ACM), pp 144-152, 1992.
- [4] SHAI SHALEV- SHWARTZ, SHAI BEN- DAVID, 2014, Understanding Machine Learning.
- [5] E. OSUNA, R. FREUND, F. GIROSI, An improved training algorithm for Support Vector Machines, Neural Networks for Signal Processing VII -Proceedings of the 1997 IEEE Workshop, pp 276-285, New York, IEEE, 1997.
- [6] J. PLATT, Sequential minimal optimization: A fast algorithm for training Support Vector Machines, Technical Report MSR-TR-98-14, Microsoft Research, 1998.
- [7] S. DUMAIS, J. PLATT, D. HECKERMAN, M. SAHAMI, “Inductive learning algorithms and representations for text categorization”, Proceedings of Conference on Information and
- [8] Pramod Chandrayan, 2017, A Mobile & Web Development Company , Startup evangelist, Mobile App Consultant , Blockchain Enthusiast.
- [9] MIKE CANTELON, ALEX R.YOUNG, MARC HARTER, 2017, Node.js In Action.
- [10] GASTON C.HILLAR, 2017, MQTT Essentials- a Lightweight IoT Protocol.
- [11] STEPHAN JONES, FRANK M.GROOM, 2017, Enterprise Cloud Computing for Non-Engineers.
- [12] KRISTINA CHODOROW, 2013, MongoDB: The Definitive Guide: Powerful and Scalable Data Storage.
- [13] Đỗ Thanh Nghị (2008), Máy học Véc tơ hỗ trợ- Support vector machines, Khoa Công Nghệ Thông Tin, Trường Đại Học Cần Thơ.

Tác giả chịu trách nhiệm bài viết:

Họ tên: Trương Ngọc Hải Nam

Điện thoại: 0944159179

Email: hainaminventor@gmail.com

BÀI BÁO KHOA HỌC

THỰC HIỆN CÔNG BỐ THEO QUY CHẾ ĐÀO TẠO THẠC SĨ

Bài báo khoa học của học viên

có xác nhận và đề xuất cho đăng của Giảng viên hướng dẫn



Bản tiếng Việt ©, TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH và TÁC GIẢ

Bản quyền tác phẩm đã được bảo hộ bởi Luật xuất bản và Luật Sở hữu trí tuệ Việt Nam. Nghiêm cấm mọi hình thức xuất bản, sao chụp, phát tán nội dung khi chưa có sự đồng ý của tác giả và Trường Đại học Sư phạm Kỹ thuật TP. Hồ Chí Minh.

ĐỂ CÓ BÀI BÁO KHOA HỌC TỐT, CẦN CHUNG TAY BẢO VỆ TÁC QUYỀN!

Thực hiện theo MTCL & KHTHMTCL Năm học 2018-2019 của Thư viện Trường Đại học Sư phạm Kỹ thuật Tp. Hồ Chí Minh.