

TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH



LUẬN VĂN TỐT NGHIỆP(CO4313)

Xây dựng hệ thống giám sát và cảnh báo
chất lượng không khí trong môi trường
làm việc qua thiết bị di động

GVHD: PGS.TS. Phạm Trần Vũ
GVPB: ThS. Phan Đình Thế Duy
SVTH: Phạm Khánh Huy Hoàng – 1511145

TP.HỒ CHÍ MINH, THÁNG 12/2019

Mục lục

Danh mục hình vẽ	4
Danh mục bảng	4
1 Giới thiệu	4
1.1 Giới thiệu đề tài và đặt vấn đề	4
1.2 Động lực nghiên cứu	6
1.3 Mục tiêu của đề tài	6
1.4 Ý nghĩa thực tiễn	6
2 Cơ sở lý thuyết	7
2.1 Cách đánh giá chất lượng không khí	7
2.2 Mô hình Web API trong lập trình ứng dụng Web	8
2.3 Mô hình học sâu	9
2.3.1 Kiến trúc Mạng nơron nhân tạo - Artificial Neural Network (ANN)	9
2.3.2 Hoạt động của mạng nơ-ron nhân tạo	11
2.3.3 Mạng thần kinh tái phát - Recurrent neural network	13
3 Công nghệ được áp dụng trong đề tài	15
3.1 Công nghệ Internet of Things IOT	15
3.2 Phương thức truyền dữ liệu MQTT	16
3.3 Chip ESP8266	17
3.4 Nodejs Platform	19
3.5 Hệ quản trị dữ liệu MongoDB	20
3.6 ReactJS Platform	22
4 Phân tích và Thiết kế hệ thống	25
4.1 Tổng quan	25
4.2 Hệ thống module sensor phần cứng	26
4.3 Hệ thống ứng dụng Web	29
4.3.1 Phân tích yêu cầu ứng dụng	29
4.3.2 Thiết kế hệ thống ứng dụng	31
4.3.2.1 Thiết kế kiến trúc	31
4.3.2.2 Biểu đồ Use Case	31
4.3.3 Thiết kế cơ sở dữ liệu	35
4.3.4 Cài đặt ứng dụng và kết quả đạt được	37
4.4 Áp dụng Recurrent Neural Network vào bài toán giám sát	39

4.4.1	Mô hình dự báo	39
4.4.2	Chuẩn bị dữ liệu	40
4.4.3	Xử lý dữ liệu	41
4.4.4	Sử dụng thư viện BrainJS để áp dụng giải thuật Recurrent Neural Network	42
4.4.5	Dánh giá mô hình RNN	43
5	Kết quả và đánh giá	44
5.1	Ưu điểm	44
5.2	Khuyết điểm	44
6	Tổng kết và hướng phát triển	45
6.1	Tổng kết	45
6.2	Hướng phát triển	45

Danh mục hình ảnh

1.1	Nồng độ bụi PM2.5 có xu hướng gia tăng.	5
1.2	Nhiều tòa nhà ở TP HCM mờ đục, không khí cảm giác khô rát.	5
2.1	Mô hình Web Api	9
2.2	Kiến trúc mạng nơ-ron nhân tạo	10
2.3	Quá trình xử lý thông tin của một mạng nơ-ron nhân tạo	11
2.4	Một mạng lưới thần kinh tái phát	14
3.1	Minh họa Internet of Things	15
3.2	Minh họa cách MQTT truyền dữ liệu	17
3.3	Module NodeMCU v1 LUA - ESP8266ESP12E	19
3.4	Lợi thế khi dùng Nodejs Backend	20
3.5	Cơ chế hoạt động của mongodb	21
3.6	Số lượng schema của MONGODB ít hơn của RDBMS	22
3.7	Lợi thế của việc sử dụng ReactJS trong thiết kế UI	23
4.1	Mô hình tổng quan của hệ thống	25
4.2	Sensor DHT11	26
4.3	Sensor BH1750	27
4.4	Sơ đồ thiết kế phần cứng	28
4.5	Sử dụng Arduinio IDE để lập trình phần cứng	28
4.6	Hình ảnh triển khai trên thực tế của các thiết bị của dịch vụ giám sát	29
4.7	Mô hình kiến trúc 3 tầng	31
4.8	Use-case nhóm chức năng hiển thị trên bản đồ	32
4.9	Use-case nhóm chức năng hiển thị trên đồ thị	33
4.10	Use-case nhóm chức năng xem thông tin chi tiết	34
4.11	Mối quan hệ giữa thực thể NodeInfo, NodePrediction và NodeRunTime	36
4.12	Lược đồ cơ sở dữ liệu của ứng dụng	36
4.13	Màn hình thực hiện hiển thị các trạm đo trên Google Map	38
4.14	Màn hình thực hiện hiển thị các chỉ số trên đồ thị	38
4.15	Màn hình thực hiện hiển thị thông tin chi tiết của trạm đo.	39
4.16	Mô hình dự báo chỉ số không khí	40
4.17	Mẫu dữ liệu tại Madrid, Tây Ban Nha	41
4.18	Mẫu training của khí SO_2	41
4.19	Hàm lấy dữ liệu train	42
4.20	Hàm train áp dụng RRN	42
4.21	Simple RRN model	43

Danh mục bảng

2.1 Các nguồn gây ô nhiễm không khí, tác động và chỉ dẫn của WHO về các chất gây ô nhiễm	8
4.1 Bảng danh sách các tác nhân	29
4.2 Bảng các chức năng cần có của ứng dụng	30

Lời cảm ơn

Đầu tiên, em xin gửi lời cảm ơn sâu sắc đến thầy Phạm Trần Vũ, nhờ thầy hướng dẫn và định hướng đề tài mà em đã chọn. Em cảm ơn thầy đã tận tình giúp đỡ và động viên em những lúc khó khăn nhất để em có thể hoàn thành đề tài luận văn này.

Em cũng xin chân thành gửi lời cảm ơn đến quý thầy cô công tác tại Khoa Khoa học và Kỹ thuật Máy tính trường Đại học Bách Khoa TP.HCM, những người đã nhiệt tình truyền đạt kiến thức, kỹ năng trong hơn 4 năm qua để em tự tin, vững bước trên con đường tiếp theo. Em cũng xin gửi lời cảm ơn đến gia đình, bạn bè, những người đã bên cạnh động viên, giúp đỡ khi em thực hiện đề tài này.

Cuối cùng, mặc dù đã cố gắng và nỗ lực nhưng với vốn hiểu biết hạn hẹp của mình chấn chấn luận văn của em không tránh khỏi những thiếu sót và hạn chế. Vì vậy, em rất mong nhận được sự góp ý từ các thầy cô để đề tài được hoàn thiện hơn.

Hồ Chí Minh, ngày 25 tháng 12 năm 2019
Phạm Khánh Huy Hoàng

Lời cam đoan

Tôi xin cam đoan Luận văn tốt nghiệp “Xây dựng hệ thống giám sát và cảnh báo chất lượng không khí trong môi trường làm việc qua thiết bị di động.” là công trình nghiên cứu của riêng tôi dưới sự hướng dẫn của PGS.TS Phạm Trần Vũ. Nội dung nghiên cứu và các kết quả đều là trung thực và chưa từng được công bố trước đây. Các số liệu được sử dụng cho quá trình phân tích, nhận xét được chính tôi thu thập từ nhiều nguồn khác nhau và sẽ được ghi rõ trong phần tài liệu tham khảo. Ngoài ra, tôi cũng có sử dụng một số nhận xét, đánh giá và số liệu của các tác giả khác, cơ quan tổ chức khác. Tất cả đều có trích dẫn và chú thích nguồn gốc. Nếu phát hiện có bất kì sự gian lận nào, tôi xin hoàn toàn chịu trách nhiệm về nội dung Luận văn tốt nghiệp của mình. Trường Đại học Bách Khoa Thành phố Hồ Chí Minh không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện.

Người thực hiện
Phạm Khánh Huy Hoàng

Tóm tắt

Luận văn đưa ra giải pháp cụ thể trong việc truyền dữ liệu từ cảm biến lên server theo thời gian thực bằng cách ứng dụng công nghệ Internet of Things và xây dựng được ứng dụng web để giúp cho việc giám sát được những thông số môi trường trở nên dễ dàng ở những nơi sử dụng Internet. Ngoài ra, luận văn còn áp dụng machine learning để dự đoán các giá trị tiếp theo của các thông số môi trường.

Chương 1

Giới thiệu

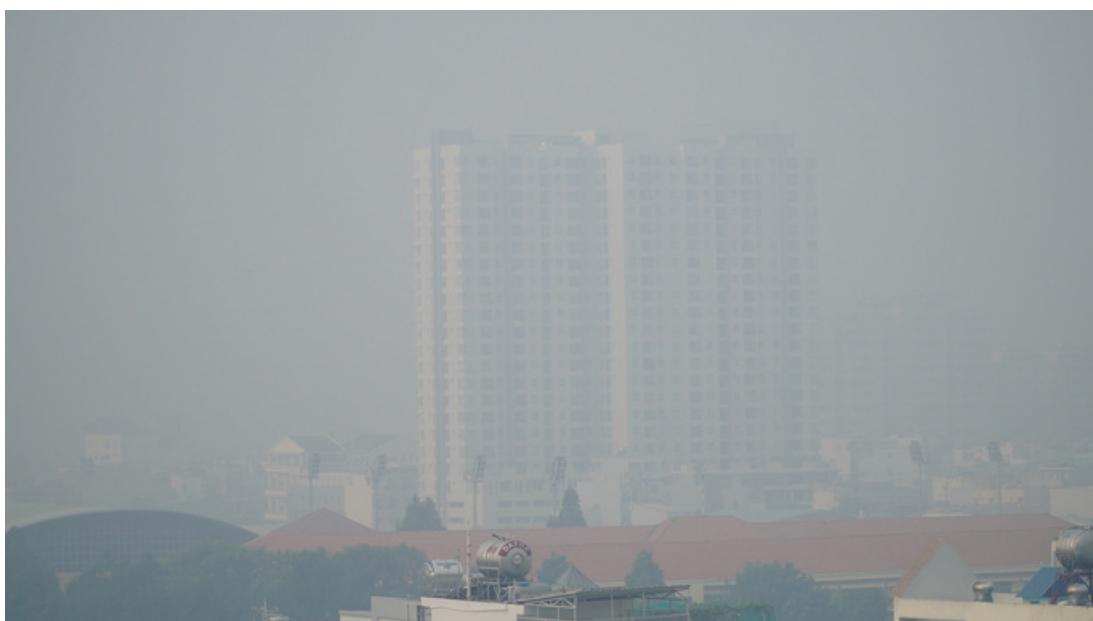
1.1 Giới thiệu đề tài và đặt vấn đề

Hiện nay, thế giới đang chạy đua trong kỉ nguyên của công nghệ và kỉ nguyên của cuộc cách mạng công nghiệp 4.0. Việc áp dụng khoa học kỹ thuật vào đời sống và công việc trở nên phổ biến và giúp ích rất lớn trong việc giải thoát sức lao động của con người. Nổi bật lên có những công nghệ đã “vượt mặt” con người ở những lĩnh vực tưởng chừng chỉ có con người mới làm được. Trong đề tài Luận văn “Xây dựng hệ thống giám sát và cảnh báo chất lượng không khí trong môi trường làm việc qua thiết bị di động.”, em hướng đến mục tiêu nghiên cứu và áp dụng công nghệ để xây dựng hệ thống quan trắc không khí tự động nhằm hỗ trợ việc thu thập dữ liệu của các tác nhân bên ngoài không khí có ảnh hưởng đến con người và môi trường sống xung quanh. Bên cạnh đó, sau khi thu thập dữ liệu ứng dụng được xây dựng với hệ thống cảnh báo cho người dùng nếu các tác nhân biến đổi đột ngột hay nồng độ của chúng vượt ngưỡng bình thường có thể ảnh hưởng xấu đến môi trường và con người xung quanh.



HÌNH 1.1: Nồng độ bụi PM2.5 có xu hướng gia tăng.

Ở Việt Nam, hệ thống quan trắc môi trường không khí cực kỳ ít ỏi. Tuy vậy, với những số liệu ít ỏi có được, bức tranh ô nhiễm không khí ở Việt Nam cũng không sáng sủa. Theo Báo cáo Môi trường Quốc gia về môi trường không khí (phát hành năm 2019) thì không khí ở các đô thị lớn có dấu hiệu ô nhiễm bụi, đặc biệt là ô nhiễm bụi mịn PM_{2.5} và ô nhiễm ozon tăng mạnh. Mới đây, công bố từ các chuyên gia Chính phủ Đức qua khảo sát tình trạng ô nhiễm môi trường tại TPHCM cho thấy, bụi mịn PM_{2.5} tăng cao, nhiều hơn gấp 23 lần so với các thành phố tại Đức. Nghiên cứu mới nhất về Chỉ số hiệu suất môi trường của Đại học Yale (Mỹ) cũng cho thấy, Việt Nam là một trong 10 quốc gia ô nhiễm không khí nhất thế giới.



HÌNH 1.2: Nhiều tòa nhà ở TP HCM mờ đục, không khí cảm giác khô rát.

1.2 Động lực nghiên cứu

Hơn lúc nào hết, việc giám sát mức độ ô nhiễm không khí và chia sẻ những thông tin này đang trở nên vô cùng quan trọng để có thể thực hiện những hành động hiệu quả. Hiện nay, những trạm đo, những thiết bị đo chỉ số không khí đang được lắp đặt ở nhiều nơi, những địa điểm mà việc giám sát là cần thiết như các thành phố lớn, các khu công nghiệp, Do đó, việc xác định những hành vi bất thường môi trường xung quanh có ý nghĩa hết sức thiết thực đến sức khỏe và đời sống. Kết quả của bài toán hoàn toàn có khả năng áp dụng vào thực tiễn để mang lại nhiều lợi ích cho xã hội. Đây là một đề tài hết sức thú vị do những tác động tích cực mà nó mang đến, bên cạnh đó đây cũng là một đề tài đầy thách thức khi đòi hỏi phải nắm bắt và vận dụng nhiều kiến thức mới có thể hoàn thành tốt đề tài.

1.3 Mục tiêu của đề tài

Đề tài hướng tới mục đích tìm hiểu, khảo sát, áp dụng và xây dựng hệ thống giám sát và cảnh báo các chỉ số chỉ hướng đến chất lượng không khí .Dánh giá chất lượng môi trường không khí trong phòng làm việc do ảnh hưởng của các tác nhân không khí khác nhau. Nghiên cứu, xây dựng phương pháp đánh giá tổng hợp chất lượng không khí. Xây dựng và phát triển hệ thống cảnh báo khi các tác nhân không khí biến đổi có khả năng gây hại cho con người.

1.4 Ý nghĩa thực tiễn

Khi khói trung tâm xử lý của IoT được tích hợp công nghệ trí tuệ nhân tạo (AI) sẽ đem lại khả năng dự báo, cảnh báo gần như là theo thời gian thực hay khi được tích hợp Bigdata với IoT sẽ mang lại khả năng phân tích xử lý dữ liệu, thống kê, lưu trữ dữ liệu rất tốt. Ứng dụng IoT mang lại những lợi ích thiết thực không chỉ cho người dân, doanh nghiệp, mà còn là các cơ quan quản lý như phân tích dữ liệu lớn trong các ứng dụng IoT như là số liệu thống kê để tham khảo phân tích các dự báo, hành vi người dùng và phương pháp xử lý dữ liệu nâng cao (bao gồm AI). Hệ thống được ứng dụng trong việc giám sát, cảnh báo mức độ ô nhiễm không khí sẽ giúp cho người dân sẽ nắm được thông tin về tình trạng môi trường không khí trong khu vực sinh sống và làm việc một cách nhanh chóng và chính xác nhất từ đó giúp họ đưa ra những quyết định phù hợp. Bên cạnh đó, khi các chỉ số môi trường không khí vượt ngưỡng cho phép, gây hại cho sức khỏe con người thì hệ thống IoT sẽ ngay lập tức phân tích xử lý và đưa ra các cảnh báo để người dân kịp thời ứng phó. Hiện nay Việt Nam chưa có tiêu chuẩn để có thể đánh giá chất lượng môi trường không khí trong nhà. Vì vậy cần có những nghiên cứu để có thể từ đó đưa ra các tiêu chuẩn về chất lượng không khí trong nhà, căn cứ vào đó chúng ta có thể đánh giá một cách chính xác chất lượng không khí, từ đó có những biện pháp giải quyết cụ thể để hạn chế được những ảnh hưởng tiêu cực do ô nhiễm không khí gây ra đối với sức khỏe con người. Ngoài ra chúng ta cũng cần có những hệ thống đo đạc một cách định kỳ thường xuyên và có khả năng cảnh báo tới mọi người một cách kịp thời, tránh những sự cố đáng tiếc có thể xảy ra

Chương 2

Cơ sở lý thuyết

2.1 Cách đáng giá chất lượng không khí

Có bốn vấn đề chính trong các khu vực đô thị xảy ra ô nhiễm môi trường không khí gây ảnh hưởng đến sức khỏe của người dân:

- Ô nhiễm không khí do các chất gây ô nhiễm hóa học và tác nhân sinh học
- Phát tán khí thải nhà kính và sự biến đổi khí hậu
- Ô nhiễm tiếng ồn
- Bức xạ và điện từ trường

Không khí bị ô nhiễm là do các chất gây ô nhiễm hóa học xảy ra trong môi trường cả trong nhà và ngoài trời, trong đó phần lớn là do môi trường trong nhà, nơi con người dành hầu hết thời gian ở đó. Tác nhân sinh học là nguyên nhân chính gây ô nhiễm không khí trong nhà. Tiếng ồn, chất phóng xạ, điện từ trường cũng là các tác nhân chính gây ô nhiễm cả môi trường ngoài trời và trong nhà. Trong đề tài này, tập trung chủ yếu vào ô nhiễm không khí ngoài trời do các hợp chất hóa học gây ra.

Các loại chất gây ô nhiễm không khí chính là các hạt bụi (PM10, PM2.5) bao gồm các phân tử các bon đen siêu nhỏ, O_3 , CO , NO_2 , các chất hữu cơ bay hơi, hiđrôcacbon và chất quang oxy hóa như ô zôn. Tuy nhiên, ở các thành phố đang phát triển, chất gây ô nhiễm không khí chính là hạt bụi và ô zôn. Về vấn đề các hạt bụi, chỉ số được báo cáo chung nhất là nồng độ PM10 (hạt có khí động học với đường kính nhỏ hơn 10 um). Ở rất nhiều thành phố Châu Á, nồng độ trung bình PM10 vượt quá 100 ug/m³ với mức cao nhất là 200 ug/m³. Hình 2.1 mô tả nguồn gây ô nhiễm không khí, tác động và chỉ dẫn của WHO về các chất gây ô nhiễm. Qua đó xác định được khoảng vượt ngưỡng của các chất gây ô nhiễm.

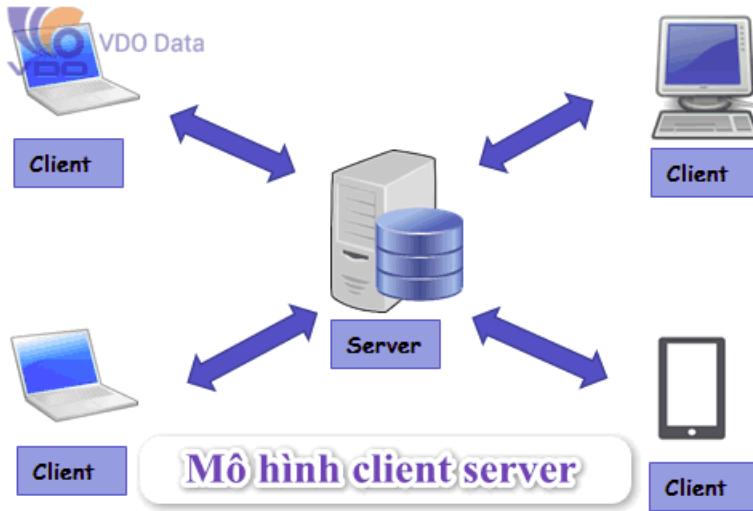
Chất gây ô nhiễm	Nguồn chính	Tác động	Chỉ dẫn về sức khỏe (WHO 2002a)
Carbon monoxide (CO)	Khí thải động cơ, các hoạt động công nghiệp	Gây độc cho người khi hít phải, CO giảm khả năng vận chuyển ôxi trong máu và tăng áp lực lên tim và phổi	10 mg/m ³ (10ppm) trên 8 tiếng; 30 mg/m ³ trên 1 tiếng (30,000 ug/m ³)
Sulphur dioxide (SO₂)	Một phần nhỏ từ các nguồn di động. Nhiệt và năng lượng sản sinh từ việc sử dụng than và dầu chứa sulphur, sulphuric acid plants	Gây trớ ngai cho con người, SO ₂ tạo phản ứng với không khí tạo ra mưa axít	20 ug/m ³ trên 24 tiếng 500 ug/m ³ trên 10 phút
Hạt bụi PM10	Đất, bụi nước biển (oceanic spray), cháy rừng, dun nấu trong nhà, phương tiện, hoạt	Tăng khả năng ung thư, trường hợp tử vong, làm nghiêm	50 ug/m ³ trên 24 tiếng 20 ug/m ³ trung bình năm
Hạt bụi PM2.5			25 ug/m ³ trên 24 tiếng

TABLE 2.1: Các nguồn gây ô nhiễm không khí, tác động và chỉ dẫn của WHO về các chất gây ô nhiễm

2.2 Mô hình Web API trong lập trình ứng dụng Web

API là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. Nó là viết tắt của Application Programming Interface – giao diện lập trình ứng dụng. API cung cấp khả năng cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dữ liệu giữa các ứng dụng. **Web API** là một phương thức dùng để cho phép các ứng dụng khác nhau có thể giao tiếp, trao đổi dữ liệu qua lại. Dữ liệu được Web API trả lại thường ở dạng JSON hoặc XML thông qua giao thức HTTP hoặc HTTPS. Web API hỗ trợ restful đầy đủ các phương thức: Get/Post/Put/Delete dữ liệu. Nó giúp bạn xây dựng các HTTP service một cách rất đơn giản và nhanh chóng. Nó cũng có khả năng hỗ trợ đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content format.

- Đầu tiên là xây dựng URL API để bên thứ ba có thể gửi request dữ liệu đến máy chủ cung cấp nội dung, dịch vụ thông qua giao thức HTTP hoặc HTTPS.
- Tại web server cung cấp nội dung, các ứng dụng nguồn sẽ thực hiện kiểm tra xác thực nếu có và tìm đến tài nguyên thích hợp để tạo nội dung trả về kết quả.
- Server trả về kết quả theo định dạng JSON hoặc XML thông qua giao thức HTTP/HTTPS.
- Tại nơi yêu cầu ban đầu là ứng dụng web hoặc ứng dụng di động, dữ liệu JSON/XML sẽ được parse để lấy data. Sau khi có được data thì thực hiện tiếp các hoạt động như lưu dữ liệu xuống Cơ sở dữ liệu, hiển thị dữ liệu...



HÌNH 2.1: Mô hình Web Api

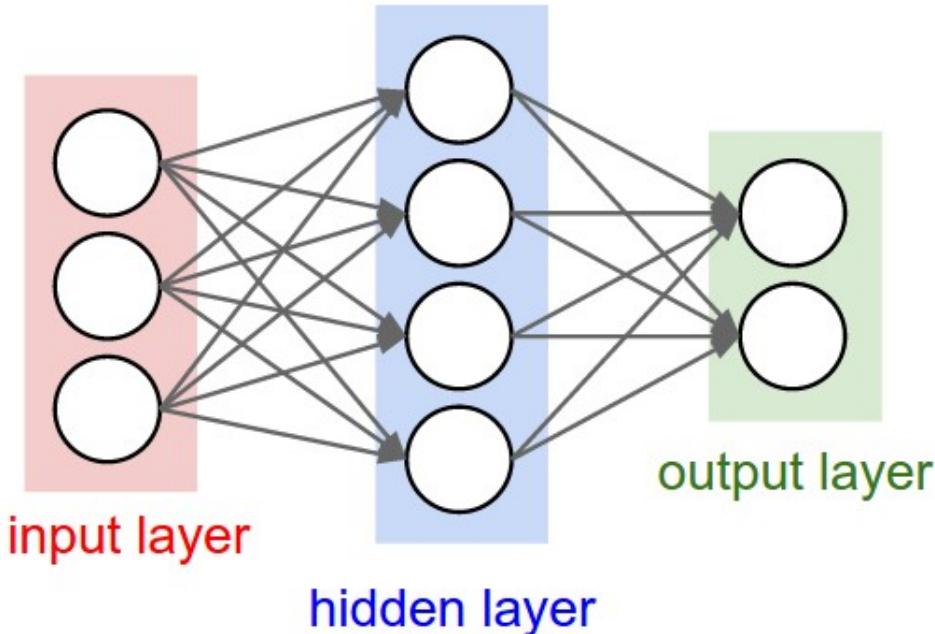
2.3 Mô hình học sâu

Trong phần này giới thiệu về cơ sở lý thuyết về mạng nơ ron nhân tạo, cách thức hoạt động của mạng nơ-ron, các phiên bản mở rộng của mạng nơ-ron nhân tạo, như: Mạng nơ-ron tái phát (RNN). Mạng nơ-ron tái phát RNN là một trong những mô hình học sâu được đánh giá có nhiều ưu điểm trong các tác vụ xử lý dữ liệu thời gian thực. Đây cũng là cơ sở thực hiện để xây dựng mô hình dự đoán trong đề tài luận văn.

2.3.1 Kiến trúc Mạng nơron nhân tạo - Artificial Neural Network (ANN)

Mạng nơ ron nhân tạo (Artificial Neural Network – ANN) là một mô hình xử lý thông tin được mô phỏng dựa trên hoạt động của hệ thống thần kinh của sinh vật, bao gồm số lượng lớn các Nơ-ron được gắn kết để xử lý thông tin. ANN hoạt động giống như bộ não của con người, được học bởi kinh nghiệm (thông qua việc huấn luyện), có khả năng lưu giữ các tri thức và sử dụng các tri thức đó trong việc dự đoán các dữ liệu chưa biết (unseen data).

Một mạng nơ-ron là một nhóm các nút nối với nhau, mô phỏng mạng nơron thần kinh của não người. Mạng nơ-ron nhân tạo được thể hiện thông qua ba thành phần cơ bản: mô hình của nơ-ron, cấu trúc và sự liên kết giữa các nơ-ron. Trong nhiều trường hợp, mạng nơ-ron nhân tạo là một hệ thống thích ứng, tự thay đổi cấu trúc của mình dựa trên các thông tin bên ngoài hay bên trong chạy qua mạng trong quá trình học.



HÌNH 2.2: Kiến trúc mạng nơ-ron nhân tạo

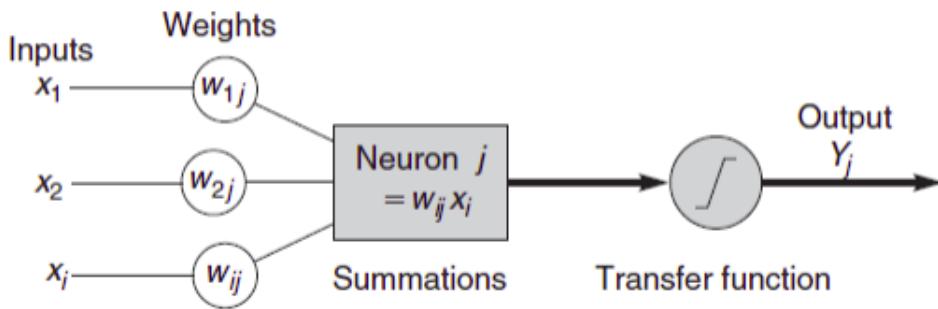
Kiến trúc chung của một ANN gồm 3 thành phần đó là **Input Layer**, **Hidden Layer** và **Output Layer**. Trong đó, lớp ẩn (Hidden Layer) gồm các nơ-ron, nhận dữ liệu input từ các Nơ-ron ở lớp (Layer) trước đó và chuyển đổi các input này cho các lớp xử lý tiếp theo. Trong một mạng ANN có thể có nhiều Hidden Layer.

Lợi thế lớn nhất của các mạng ANN là khả năng được sử dụng như một cơ chế xấp xỉ hàm tùy ý mà “học” được từ các dữ liệu quan sát. Tuy nhiên, sử dụng chúng không đơn giản như vậy, một số các đặc tính và kinh nghiệm khi thiết kế một mạng nơ-ron ANN.

- **Chọn mô hình:** Điều này phụ thuộc vào cách trình bày dữ liệu và các ứng dụng. Mô hình quá phức tạp có xu hướng dẫn đến những thách thức trong quá trình học.
- **Cấu trúc và sự liên kết giữa các nơ-ron.**
- **Thuật toán học:** Có hai vấn đề cần học đối với mỗi mạng ANN, đó là học tham số của mô hình (parameter learning) và học cấu trúc (structure learning). Học tham số là thay đổi trọng số của các liên kết giữa các nơron trong một mạng, còn học cấu trúc là việc điều chỉnh cấu trúc mạng bằng việc thay đổi số lớp ẩn, số nơ-ron mỗi lớp và cách liên kết giữa chúng. Hai vấn đề này có thể được thực hiện đồng thời hoặc tách biệt.

Nếu các mô hình, hàm chi phí và thuật toán học được lựa chọn một cách thích hợp, thì mạng ANN sẽ cho kết quả có thể vô cùng mạnh mẽ và hiệu quả.

2.3.2 Hoạt động của mạng nơ-ron nhân tạo



HÌNH 2.3: Quá trình xử lý thông tin của một mạng nơ-ron nhân tạo

- **Inputs:** Mỗi Input tương ứng với 1 đặc trưng của dữ liệu. Ví dụ như trong ứng dụng của ngân hàng xem xét có chấp nhận cho khách hàng vay tiền hay không thì mỗi input là một thuộc tính của khách hàng như thu nhập, nghề nghiệp, tuổi, số con,...
- **Output:** Kết quả của một ANN là một giải pháp cho một vấn đề, ví dụ như với bài toán xem xét chấp nhận cho khách hàng vay tiền hay không thì output là yes hoặc no.
- **Connection Weights (Trọng số liên kết):** Đây là thành phần rất quan trọng của một ANN, nó thể hiện mức độ quan trọng, độ mạnh của dữ liệu đầu vào đối với quá trình xử lý thông tin chuyển đổi dữ liệu từ Layer này sang layer khác. Quá trình học của ANN thực ra là quá trình điều chỉnh các trọng số Weight của các dữ liệu đầu vào để có được kết quả mong muốn.
- **Summation Function (Hàm tổng):** Tính tổng trọng số của tất cả các input được đưa vào mỗi Nơ-ron. Hàm tổng của một Nơ-ron đối với n input được tính theo công thức sau:

$$Y = \sum_{i=1}^n X_i W_i$$

- **Transfer Function (Hàm chuyển đổi):** Hàm tổng của một nơ-ron cho biết khả năng kích hoạt của nơ-ron đó còn gọi là kích hoạt bên trong. Các nơ-ron này có thể sinh ra một output hoặc không trong mạng ANN, nói cách khác rằng có thể output của 1 Nơ-ron có thể được chuyển đến layer tiếp trong mạng Nơ-ron theo hoặc không. Mỗi quan hệ giữa hàm tổng và kết quả output được thể hiện bằng hàm chuyển đổi.

Việc lựa chọn hàm chuyển đổi có tác động lớn đến kết quả đầu ra của mạng ANN. Hàm chuyển đổi phi tuyến được sử dụng phổ biến trong mạng ANN là hoặc sigmoid hoặc tanh.

$$Y = \sum_{i=1}^n X_i W_i$$

Trong đó, hàm tanh là phiên bản thay đổi tỉ lệ của sigmoid , tức là khoảng giá trị đầu ra của hàm chuyển đổi thuộc khoảng [-1, 1] thay vì [0,1] nên chúng còn gọi là hàm chuẩn hóa (Normalized Function).

Kết quả xử lý tại các nơ-ron (Output) đôi khi rất lớn, vì vậy hàm chuyển đổi được sử dụng để xử lý output này trước khi chuyển đến layer tiếp theo. Đôi khi thay vì sử dụng Transfer Function người ta sử dụng giá trị ngưỡng (Threshold value) để kiểm soát các output của các neuron tại một layer nào đó trước khi chuyển các output này đến các Layer tiếp theo. Nếu output của một neuron nào đó nhỏ hơn Threshold thì nó sẽ không được chuyển đến Layer tiếp theo.

Mạng nơ-ron của chúng ta dự đoán dựa trên lan truyền thẳng (forward propagation) là các phép nhân ma trận cùng với activation function để thu được kết quả đầu ra. Nếu input x là vector 2 chiều thì ta có thể tính kết quả dự đoán \hat{y} bằng công thức sau

$$\begin{aligned} z_1 &= xW_1 + b_1 \\ a_1 &= \tanh(z_1) \\ z_2 &= a_1W_2 + b_2 \\ a_2 &= \hat{y} = \text{softmax}(z_2) \end{aligned}$$

Trong đó, z_i là input của layer thứ i , a_i là output của layer thứ i sau khi áp dụng activation function. W_1, b_1, W_2, b_2 là các thông số (parameters) cần tìm của mô hình mạng nơ-ron. Huấn luyện để tìm các thông số cho mô hình tương đương với việc tìm các thông số W_1, b_1, W_2, b_2 sao cho độ lỗi của mô hình đạt được là thấp nhất. Ta gọi hàm độ lỗi của mô hình là loss function. Đối với softmax function, ta dùng crossentropy loss (còn gọi là negative log likelihood). Nếu ta có N dòng dữ liệu huấn luyện, và C nhóm phân lớp (trường hợp này là hai lớp nam, nữ), khi đó loss function giữa giá trị dự đoán \hat{y} và y được tính như sau

$$L(y, \hat{y}) = -\frac{1}{N} \sum_{n \in N} \sum_{i \in C} y_{n,i} \log \hat{y}_{n,i}$$

Ý nghĩa công thức trên nghĩa là: lấy tổng trên toàn bộ tập huấn luyện và cộng dồn vào hàm loss nếu kết quả phân lớp sai. Độ dị biệt giữa hai giá trị \hat{y} và y càng lớn thì độ lỗi càng cao.

Mục tiêu của chúng ta là tối thiểu hóa hàm lỗi này. Ta có thể sử dụng phương pháp gradient descent để tối thiểu hóa hàm lỗi. Có hai loại gradient descent, một loại với fixed learning rate được gọi là batch gradient descent, loại còn lại có learning rate thay đổi theo quá trình huấn luyện được gọi là SGD (stochastic gradient descent) hay minibatch gradient descent.

Gradient descent cần các gradient là các vector có được bằng cách lấy đạo hàm của loss function theo từng thông số $\frac{\partial L}{\partial W_1}, \frac{\partial L}{\partial b_1}, \frac{\partial L}{\partial W_2}, \frac{\partial L}{\partial b_2}$. Để tính các gradient này, ta sử dụng thuật toán backpropagation (làn truyền ngược). Đây là cách hiệu quả để tính gradient khởi điểm từ Output layer.

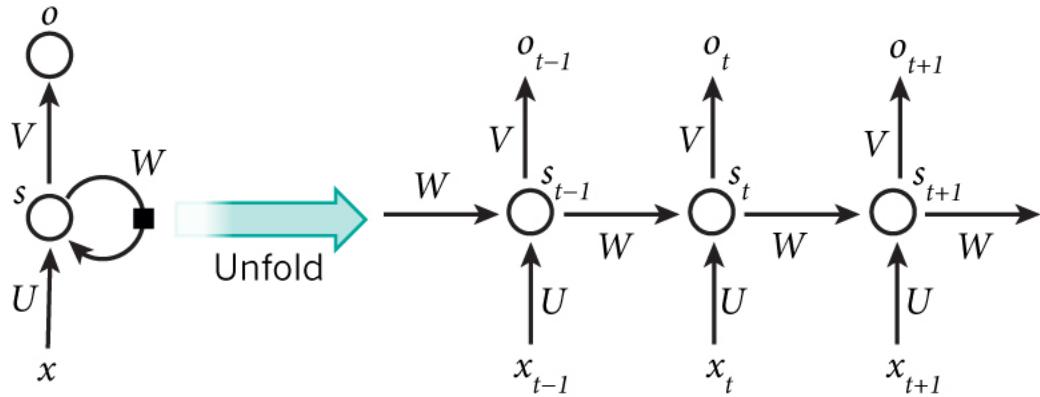
Áp dụng backpropagation ta có các đại lượng:

$$\begin{aligned}\delta_3 &= y - \hat{y} \\ \delta_2 &= (1 - \tanh^2 z_1) \circ \delta_3 W_2^T \\ \frac{\partial L}{\partial W_2} &= a_1^T \delta_3 \\ \frac{\partial L}{\partial b_2} &= \delta_3 \\ \frac{\partial L}{\partial W_1} &= x^T \delta_2 \\ \frac{\partial L}{\partial b_1} &= \delta_2\end{aligned}$$

2.3.3 Mạng thần kinh tái phát - Recurrent neural network

Mạng thần kinh tái phát (RNN) là một lớp ANN trong đó các kết nối giữa các đơn vị tạo thành một biểu đồ phôi hợp dọc theo một chuỗi. Ý tưởng của RNN đó là thiết kế một Neural Network sao cho có khả năng xử lý được thông tin dạng chuỗi (sequential information), ví dụ một câu là một chuỗi gồm nhiều từ.

Recurrent có nghĩa là thực hiện lặp lại cùng một tác vụ cho mỗi thành phần trong chuỗi. Trong đó, kết quả đầu ra tại thời điểm hiện tại phụ thuộc vào kết quả tính toán của các thành phần ở những thời điểm trước đó.



HÌNH 2.4: Một mạng lưới thần kinh tái phát

Nói cách khác, RNN là một mô hình có trí nhớ (memory), có khả năng nhớ được thông tin đã tính toán trước đó. Không như các mô hình Neural Network truyền thống đó là thông tin đầu vào (input) hoàn toàn độc lập với thông tin đầu ra (output). Về lý thuyết, RNN có thể nhớ được thông tin của chuỗi có chiều dài bất kì, nhưng trong thực tế mô hình này chỉ nhớ được thông tin ở vài bước trước đó.

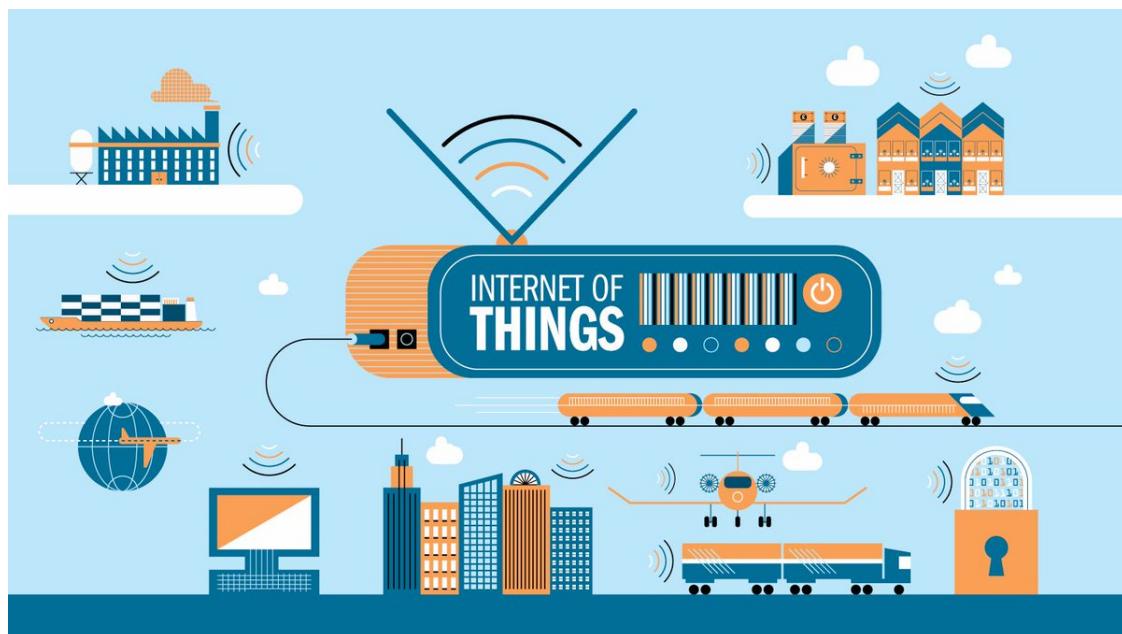
Chương 3

Công nghệ được áp dụng trong đề tài

3.1 Công nghệ Internet of Things IOT

Khái Niệm

Mạng lưới vạn vật kết nối Internet hoặc là mạng lưới thiết bị kết nối Internet viết tắt là IoT (tiếng Anh: Internet of Things) là một kịch bản của thế giới, khi mà mỗi đồ vật, con người được cung cấp một định danh của riêng mình, và tất cả có khả năng truyền tải, trao đổi thông tin, dữ liệu qua một mạng duy nhất mà không cần đến sự tương tác trực tiếp giữa người với người, hay người với máy tính. IoT đã phát triển từ sự hội tụ của công nghệ không dây, công nghệ vi cơ điện tử và Internet. Nói đơn giản là một tập hợp các thiết bị có khả năng kết nối với nhau, với Internet và với thế giới bên ngoài để thu thập thông tin về môi trường xung quanh để thực hiện một công việc cụ thể.



HÌNH 3.1: Minh họa Internet of Things

Controller đóng vai trò trung gian giữa Model và View. Xử lý các yêu cầu của người dùng. Làm việc với Model và lựa chọn View để thể hiện nội dung một cách thích hợp. Trong mô hình MVC, View chỉ hiển thị thông tin, Controller nhận, xử lý, phản hồi các yêu cầu, tương tác của người dùng và Model để xử lý các yêu cầu từ controller xuống cơ sở dữ liệu.

Đặc điểm của Internet of Things

- Tính năng kết nối liên thông (interconnectivity): Đối với IOT bất cứ thứ gì cũng có thể kết nối được với nhau nhờ mạng lưới thông tin và các cơ sở hạ tầng liên lạc tổng thể.
- Những dịch vụ có liên quan đến “Things”: Hệ thống của IOT có khả năng cung cấp các dịch vụ liên quan đến “Things”. Như bảo vệ sự nhất quán và vô tư giữa Virtual với Physical Thing. Để cung cấp được dịch vụ này, Cả công nghệ thông tin (phần mềm) và công nghệ phần cứng sẽ được thay đổi.
- Tính năng không đồng nhất: Các thiết bị trong IOT không đồng nhất vì trong nó có các phần cứng khác nhau, network cũng khác nhau. Các thiết bị giữ network sẽ có khả năng tương tác với nhau nhờ vào sự liên kết của network.
- Tính năng thay đổi kinh hoạt: Những Status của cá thiết bị sẽ được tự động thay đổi. Ngoài ra số lượng thiết bị có thể sẽ bị thay đổi.
- Với quy mô lớn: Sẽ có một khối lượng lớn các thiết bị được quản lý cùng với việc giao tiếp với nhau. Các số lượng này sẽ lớn hơn nhiều so với lượng máy tính được kết nối mạng Internet hiện nay. Tất cả các số lượng thông tin được truyền tải bởi thiết bị sẽ lớn hơn rất nhiều so với việc được truyền qua con người.

Ưu và nhược điểm của Internet of Things

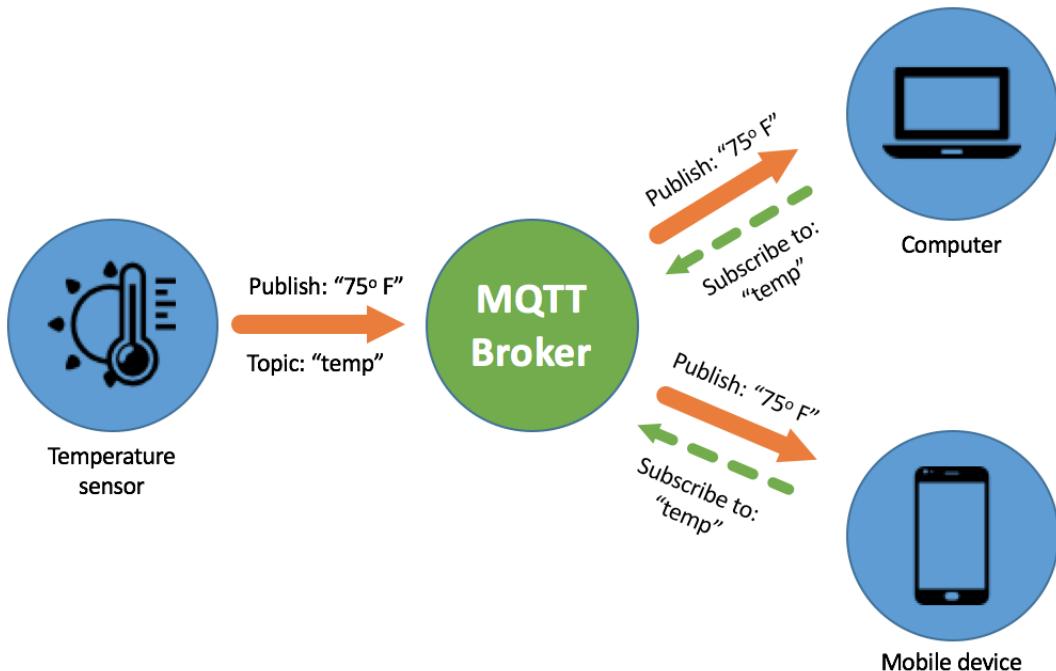
- **Ưu điểm**
 - Kết nối và truy xuất thông tin nhanh
 - Tiện dụng cho mọi hoạt động của con người
 - Tiết kiệm được thời gian khi sử dụng IOT
 - Nâng cao năng xuất trong lao động và sản xuất
- **Nhược điểm**
 - Chi phí để triển khai hệ thống hoàn thiện cao
 - Tính bảo mật về thông tin chưa cao

3.2 Phương thức truyền dữ liệu MQTT

Khái niệm

MQTT là một giao thức publish/subscribe bản tin, được thiết kế cho công nghệ M2M (Machine to Machine) gọn nhẹ. MQTT kiểm soát các gói tin header được giữ sao cho càng nhỏ càng tốt. Mỗi gói tin điều khiển MQTT bao gồm ba phần, phần header cố định, phần header thay đổi và

payload. Mỗi gói tin điều khiển MQTT có header cố định 2 byte. Không phải tất cả các gói tin điều khiển có phần header thay đổi và payload. Phần header thay đổi chứa thông tin nhận dạng của gói tin nếu nó được sử dụng bởi gói tin kiểm soát. Phần payload có dung lượng lên đến 256 MB được đính kèm trong các gói tin. Việc có một header nhỏ làm cho giao thức này phù hợp với ứng dụng IoT bằng cách giảm lưu lượng dữ liệu truyền qua mạng hạn chế băng thông.



HÌNH 3.2: Minh họa cách MQTT truyền dữ liệu

Ưu điểm và nhược điểm của MQTT

- Ưu điểm: Ít tốn băng thông, độ tin cậy cao
- Nhược điểm: Độ bảo mật thông tin cao

3.3 Chip ESP8266

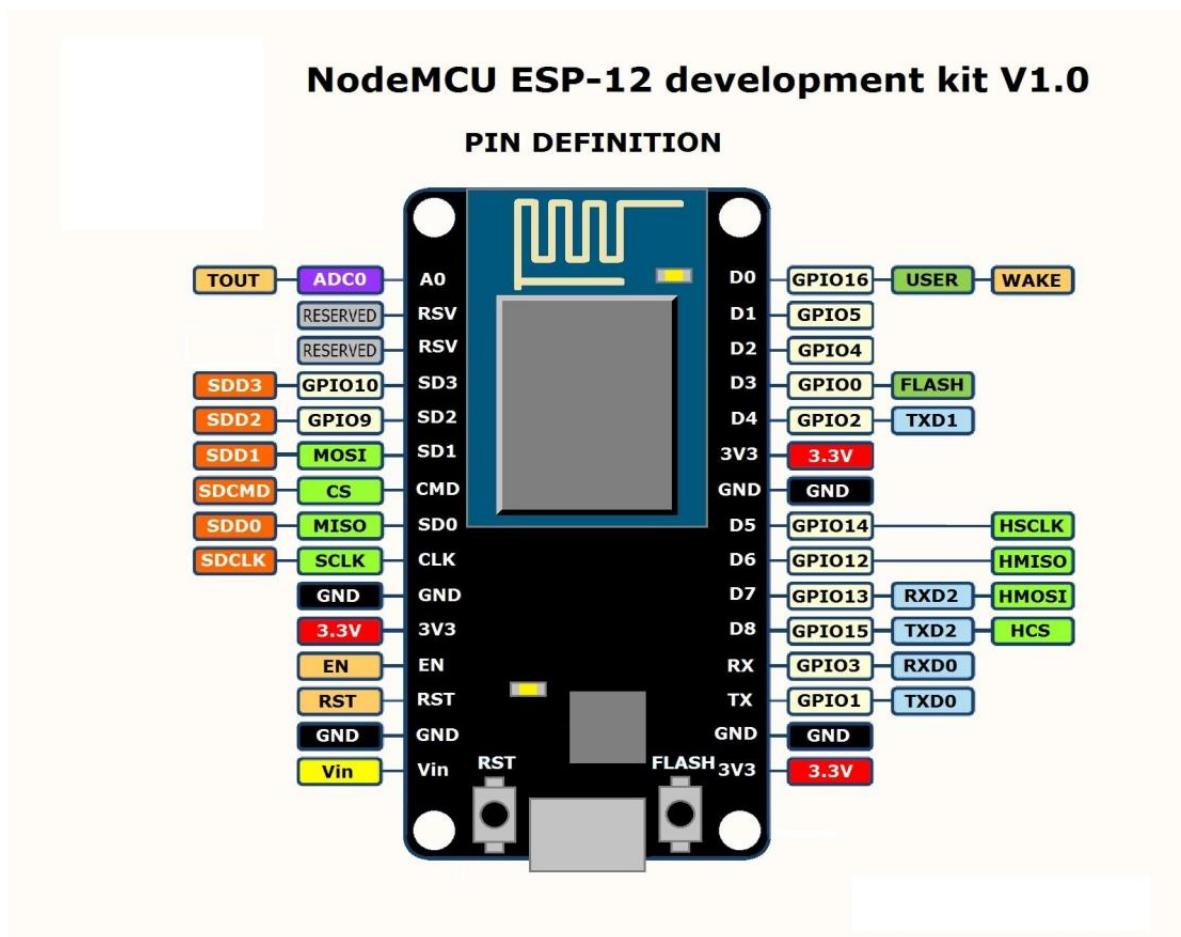
Khái quát

ESP8266 là dòng chip tích hợp Wi-Fi 2.4Ghz có thể lập trình được, giá rẻ được sản xuất bởi một công ty bán dẫn Trung Quốc: Espressif Systems. Đây là một dự án mã nguồn mở giúp hỗ trợ môi trường phát triển Arduino cho ESP8266. ESP8266 Arduino core đi kèm với thư viện kết nối WiFi hỗ trợ TCP, UDP và các ứng dụng HTTP, mDNS, SSDP, DNS Servers giúp cho việc tiếp cận và xây dựng ứng dụng rất nhanh.

Thông số phần cứng

- 32-bit RISC CPU : Tensilica Xtensa LX106 chạy ở xung nhịp 80 MHz
- Hỗ trợ Flash ngoài từ 512KiB đến 4MiB
- 64KBytes RAM thực thi lệnh
- 96KBytes RAM dữ liệu
- 64KBytes boot ROM
- Chuẩn wifi IEEE 802.11 b/g/n, Wi-Fi 2.4 GHz
- Tích hợp giao thức TCP/IP
- Hỗ trợ nhiều loại anten
- 16 chân GPIO
- Hỗ trợ SDIO 2.0, UART, SPI, I2C, PWM,I2S với DMA
- 1 ADC 10-bit
- Dải nhiệt độ hoạt động rộng : -40C – 125C

Phần mềm sử dụng lập trên máy tính cho Chip ESP8266 là Arduino, ngôn ngữ lập trình C/C++.



HÌNH 3.3: Module NodeMCU v1 LUA - ESP8266ESP12E

3.4 Nodejs Platform

Khái niệm

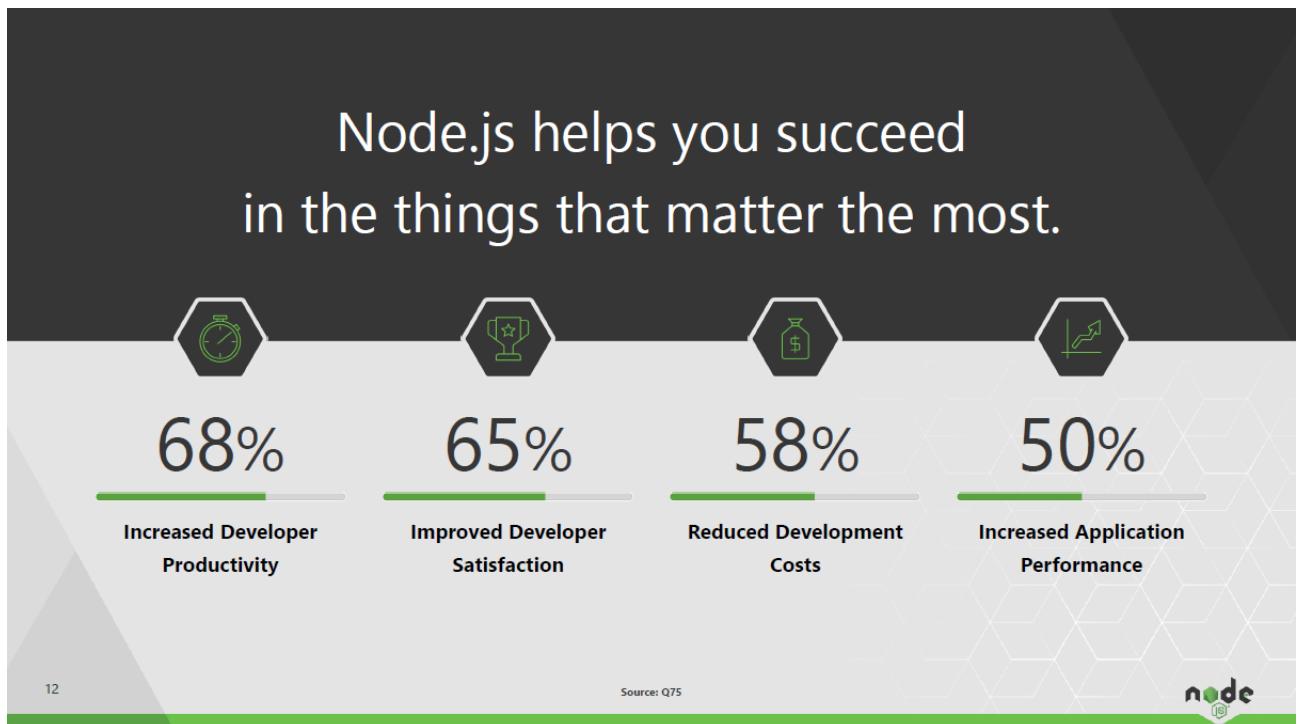
Node.js là một Javascript Run time Cross Platform được xây dựng dựa trên mã nguồn mở Google's V8 JavaScript engine cho Chrome (Browser). Node.js cho phép các lập trình viên có thể xây dựng ứng dụng Server Side, truy cập vào tài nguyên hệ thống và thực hiện được phần lớn các tác vụ hệ điều hành có thể thực hiện bằng ngôn ngữ Javascript, hoặc liên kết C++.

Nền tảng Cloud của gần như tất cả các nhà phát triển hiện nay đều hỗ trợ thực thi Node.js, điển hình như Amazon Lambda, Google Script, IBM Blumix, Microsoft Azure ...

Ngôn ngữ lập trình Javascript được cải tiến liên tục, hiện nay là EcmaScript 6 (ES5, ES2015) và đang được cải tiến rất nhanh, với nhiều ưu điểm như dễ học, xúc tích, OOP...

Một lý do Node.js được ưa chuộng nữa là đa phần các lập trình viên viết Web, Mobile đều

biết, và giờ đây, nhờ Node.js mà họ có thể triển khai các ứng dụng Server Side bằng Javascript, mà không cần dùng ngôn ngữ nào khác (như trước kia phải cần Java, PHP ...)



HÌNH 3.4: Lợi thế khi dùng Nodejs Backend

Lý do sử dụng Nodejs

Một hệ thống Internet of Things đầy đủ khá phức tạp, bao gồm thiết bị, Server xử lý kết nối, Server dữ liệu (Database), các hệ thống cân bằng tải, các hệ thống phân thích, báo cáo dữ liệu, trí tuệ nhân tạo. Server là một thành phần không thể thiếu trong hệ thống IoT. Với nhiều ưu điểm của Node.js thì nó rất phù hợp trong việc phát triển các Server cho IoT trong tương lai. Ngoài ra, Node.js được cộng đồng hỗ trợ rất nhiều, và không khó để tìm thấy 1 package cần thiết, tiết kiệm rất nhiều thời gian phát triển ứng dụng.

3.5 Hệ quản trị dữ liệu MongoDB

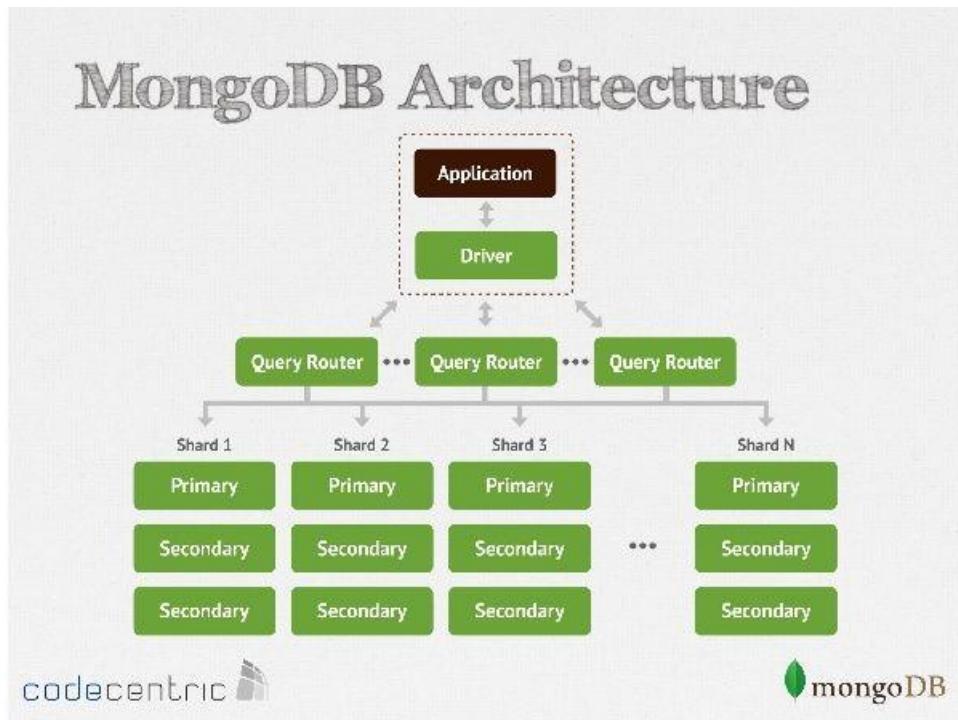
Khái niệm

MongoDB là một cơ sở dữ liệu mã nguồn mở và là cơ sở dữ liệu NoSQL(*) hàng đầu, được hàng triệu người sử dụng. MongoDB được viết bằng C++. Ngoài ra, MongoDB là một cơ sở dữ liệu đa nền tảng, hoạt động trên các khái niệm Collection và Document, nó cung cấp hiệu suất cao, tính khả dụng cao và khả năng mở rộng dễ dàng.

Cách hoạt động của MongoDB

MongoDB hoạt động dưới một tiến trình ngầm service, luôn mở một cổng (Cổng mặc định là 27017) để lắng nghe các yêu cầu truy vấn, thao tác từ các ứng dụng gửi vào sau đó mới tiến hành xử lý. Mỗi một bản ghi của MongoDB được tự động gắn thêm một field có tên **id** thuộc kiểu dữ liệu **ObjectId** mà nó quy định để xác định được tính duy nhất của bản ghi này so với bản ghi khác, cũng như phục vụ các thao tác tìm kiếm và truy vấn thông tin về sau. Trường dữ liệu **id** luôn được tự động đánh **index** (chỉ mục) để tốc độ truy vấn thông tin đạt hiệu suất cao nhất. Mỗi khi có một truy vấn dữ liệu, bản ghi được **cache** (ghi đệm) lên bộ nhớ Ram, để phục vụ lượt truy vấn sau diễn ra nhanh hơn mà không cần phải đọc từ ổ cứng.

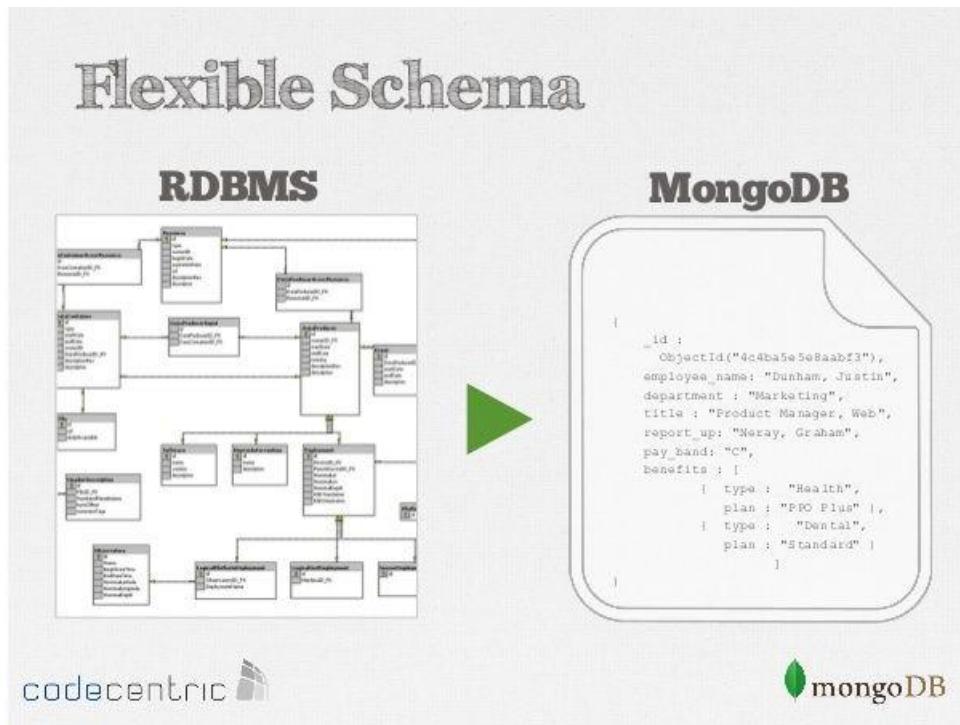
Khi có yêu cầu thêm/sửa/xóa bản ghi, để đảm bảo hiệu suất của ứng dụng mặc định MongoDB sẽ chưa cập nhật xuống ổ cứng ngay, mà sau 60 giây MongoDB mới thực hiện ghi toàn bộ dữ liệu thay đổi từ RAM xuống ổ cứng.



HÌNH 3.5: Cơ chế hoạt động của mongodb

Lợi thế khi dùng Nodejs Backend

MongoDB là phần mềm mã nguồn mở miễn phí, có cộng đồng phát triển rất lớn, hiệu năng cao với tốc độ truy vấn nhanh hơn hẳn so với các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS). Khả năng mở rộng cực lớn: việc mở rộng dữ liệu mà không phải lo đến các vấn đề như khóa ngoại, khóa chính, kiểm tra ràng buộc,... MongoDB cho phép thực hiện replication và sharding nên việc mở rộng cũng thuận lợi.

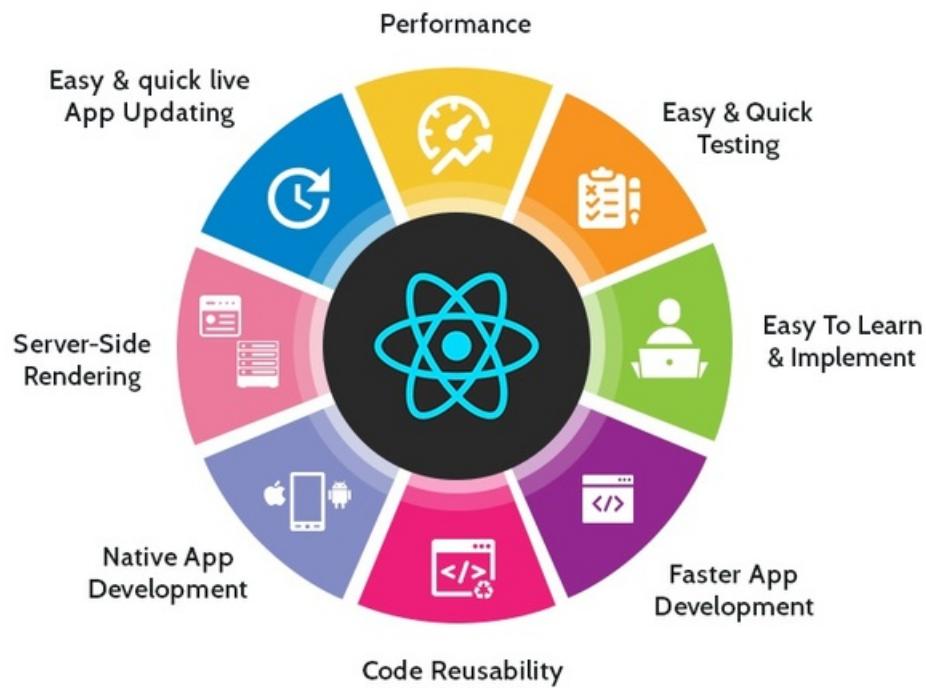


HÌNH 3.6: Số lượng schema của MONGODB ít hơn của RDBMS

3.6 ReactJS Platform

Khái niệm

ReactJS (còn được gọi là React hoặc React.js) là một thư viện JavaScript để xây dựng giao diện (UI) người dùng. Nó được duy trì bởi Facebook và một cộng đồng gồm các lập trình viên mã nguồn mở. ReactJS có thể được sử dụng trong việc phát triển các ứng dụng web hoặc di động, vì nó là tối ưu để tìm nạp dữ liệu thay đổi nhanh chóng cần được ghi lại. Mục đích cốt lõi của ReactJS không chỉ khiến cho trang web phải thật mượt mà còn phải nhanh, khả năng mở rộng cao và đơn giản. Sức mạnh của nó xuất phát từ việc tập trung vào các thành phần riêng lẻ. Chính vì vậy, thay vì làm việc trên toàn bộ ứng dụng web, ReactJS cho phép một developer có thể phá vỡ giao diện người dùng phức tạp thành các thành phần đơn giản hơn.



 TRISTATE
TECHNOLOGY

HÌNH 3.7: Lợi thế của việc sử dụng ReactJS trong thiết kế UI

Ưu điểm

- Reactjs giúp việc viết các đoạn code JS dễ dàng hơn: Nó dung cú pháp đặc biệt là JSX (Javascript mở rộng) cho phép ta trộn giữa code HTML và Javascript.
- Nó có nhiều công cụ phát triển: Khi bạn bắt đầu Reactjs, đừng quên cài đặt ứng dụng mở rộng của Chrome dành cho Reactjs. Nó giúp bạn debug code dễ dàng hơn.
- Render tầng server: Một trong những vấn đề với các ứng dụng đơn trang là tối ưu SEO và thời gian tải trang. Nếu tất cả việc xây dựng và hiển thị trang đều thực hiện ở client, thì người dùng sẽ phải chờ cho trang được khởi tạo và hiển thị lên.
- Làm việc với vấn đề test giao diện: Nó cực kì dễ để viết các test case giao diện vì virtual DOM được cài đặt hoàn toàn bằng JS.

- Hiệu năng cao đối với các ứng dụng có dữ liệu thay đổi liên tục, dễ dàng cho bảo trì và sửa lỗi.

Nhược điểm

- Reactjs chỉ phục vụ cho tầng View. React chỉ là View Library nó không phải là một MVC framework như những framework khác.
- Tích hợp Reactjs vào các framework MVC truyền thống yêu cầu cần phải cấu hình lại.

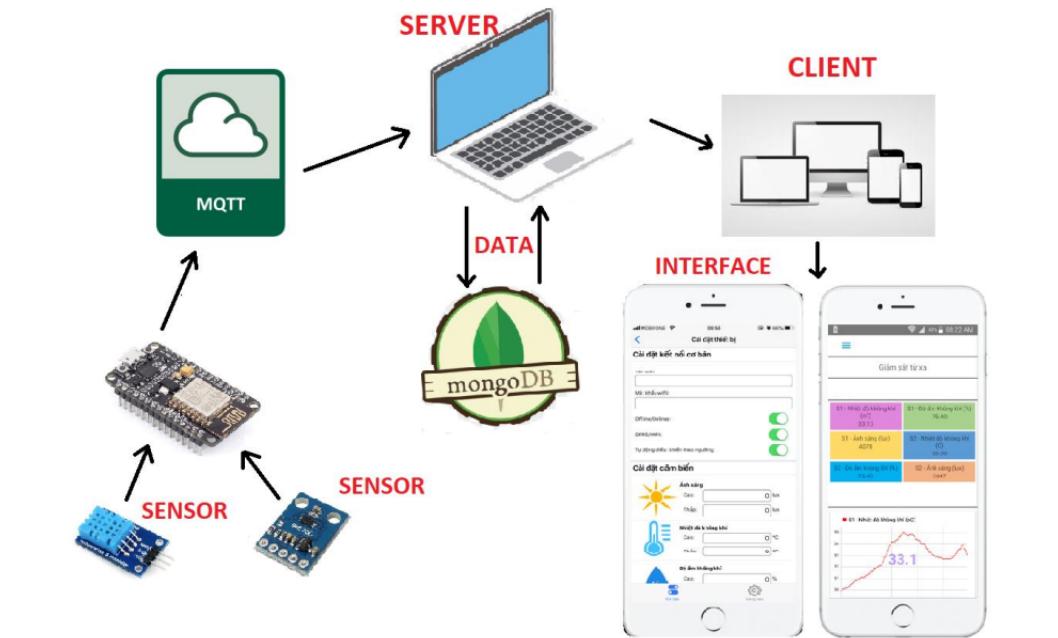
Chương 4

Phân tích và Thiết kế hệ thống

4.1 Tổng quan

Trong phạm vi đề tài, bài toán đặt ra là giám sát sự thay đổi các chỉ số môi trường và cảnh báo tình trạng biến động của các chỉ số đó trên ứng dụng di động và ứng dụng web của người dùng. Như vậy, ở đây có thể xây dựng hệ thống đầu vào của hệ thống là dữ liệu do các sensor cung cấp thông qua MQTT, từ đó máy chủ nhận được và xử lý trên, sau đó dữ liệu trả về được xử lý, lưu trữ dưới cơ sở dữ liệu và hiển thị thông qua giao diện ứng. Từ đó, xác định được hệ thống với những thành phần chính. Hệ thống được gồm có 3 phần chính như sau:

- Hệ thống thiết bị module sensor tại các điểm khảo sát
- Hệ thống ứng dụng Web
- Mô hình học máy dự báo các chỉ số

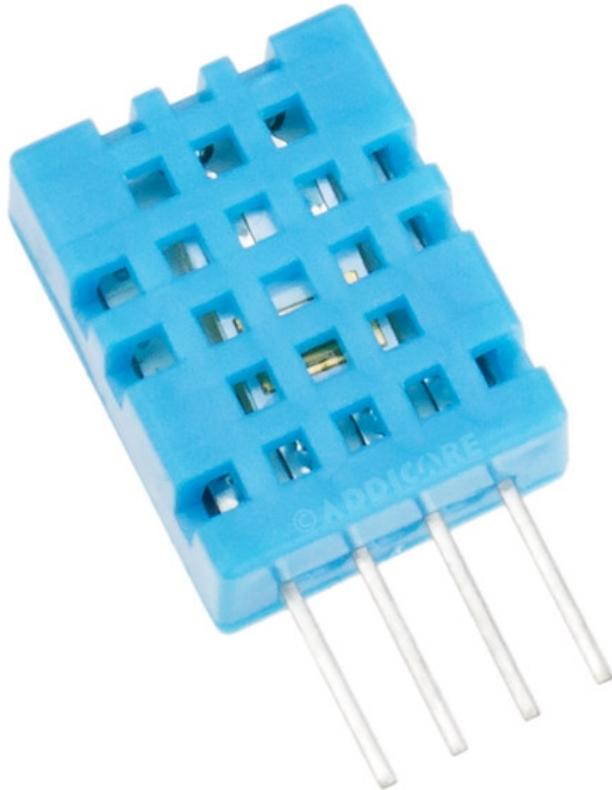


HÌNH 4.1: Mô hình tổng quan của hệ thống

4.2 Hệ thống module sensor phần cứng

Với quy mô đề tài, và tính tương tự của sự biến đổi các chỉ số ô nhiễm. Nên trong phần hiện thực hệ thống module sensor em sử dụng 2 sensor có giá thành rẻ và dễ thực hiện để mô phỏng hệ thống phần cứng. Hệ thống module sensor có sử dụng 3 linh kiện chính đó là ESP8266, DHT11 và BH1750.

DHT11 là cảm biến đo nhiệt độ độ ẩm phổ biến do giá thành rẻ và tính ứng dụng cao. Nó sử dụng chuẩn giao tiếp one-wire(sử dụng 1 dây data để truyền dữ liệu) vì vậy khi giao tiếp với vi điều khiển cần phải chú ý cài đặt thời gian timer để dữ liệu đọc không bị lỗi.



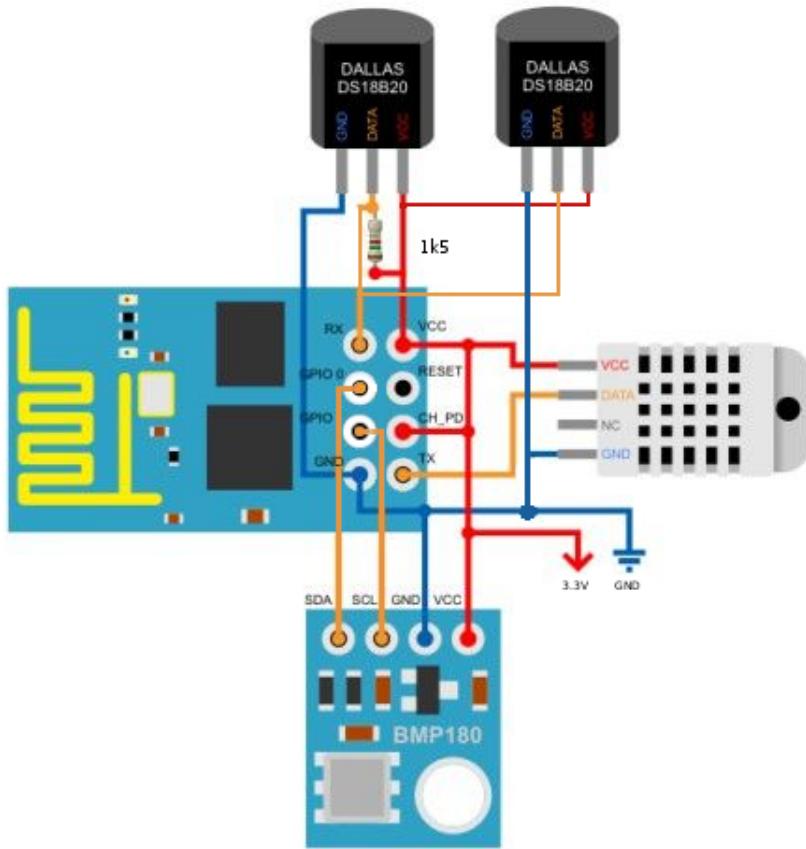
HÌNH 4.2: Sensor DHT11

BH1750FVI là một bộ cảm biến ánh sáng môi trường (Ambient Light Sensor) giao tiếp theo chuẩn I2C. IC này thích hợp thu thập dữ liệu ánh sáng môi trường xung quanh để điều chỉnh cho LCD và bàn phím điện thoại di động. Nó có thể nhận diện với người rộng độ phân giải cao (1 – 65535 lx). Đơn vị đo là Lumen (lx).



HÌNH 4.3: Sensor BH1750

Mô hình thiết kế được hiển thị qua sơ đồ bên dưới



HÌNH 4.4: Sơ đồ thiết kế phần cứng

Để lập trình module này, đê tài sử dụng Arduinio IDE với các thư viện hỗ trợ để giúp cho lập trình phần cứng dễ dàng hơn

The screenshot shows the Arduino IDE interface. On the left, a code editor window titled "test" contains C++ code for connecting to a Wi-Fi network, initializing DHT11 sensors, and publishing data to a MQTT broker. On the right, a serial monitor window titled "COM8" displays the output of the code, showing temperature and humidity readings being sent to a Thingspeak server at 115200 baud rate.

```

test
#include <Wire.h>
#include <BH1750.h>

const char* ssid      = "Hoàng Phạm";           // The SSID (name) of the Wi-Fi
const char* password  = "asdfghjkl";             // The password of the Wi-Fi network
const char* mqttServer = "mll.cloudmqtt.com";
const int mqttPort = 16524;
const char* mqttUser = "ikkwucnu";
const char* mqttPassword = "UN906syezakc";
const String device_name = "20832903";
WiFiClient espClient;
PubSubClient client(espClient);
BH1750 lightMeter(0x23);

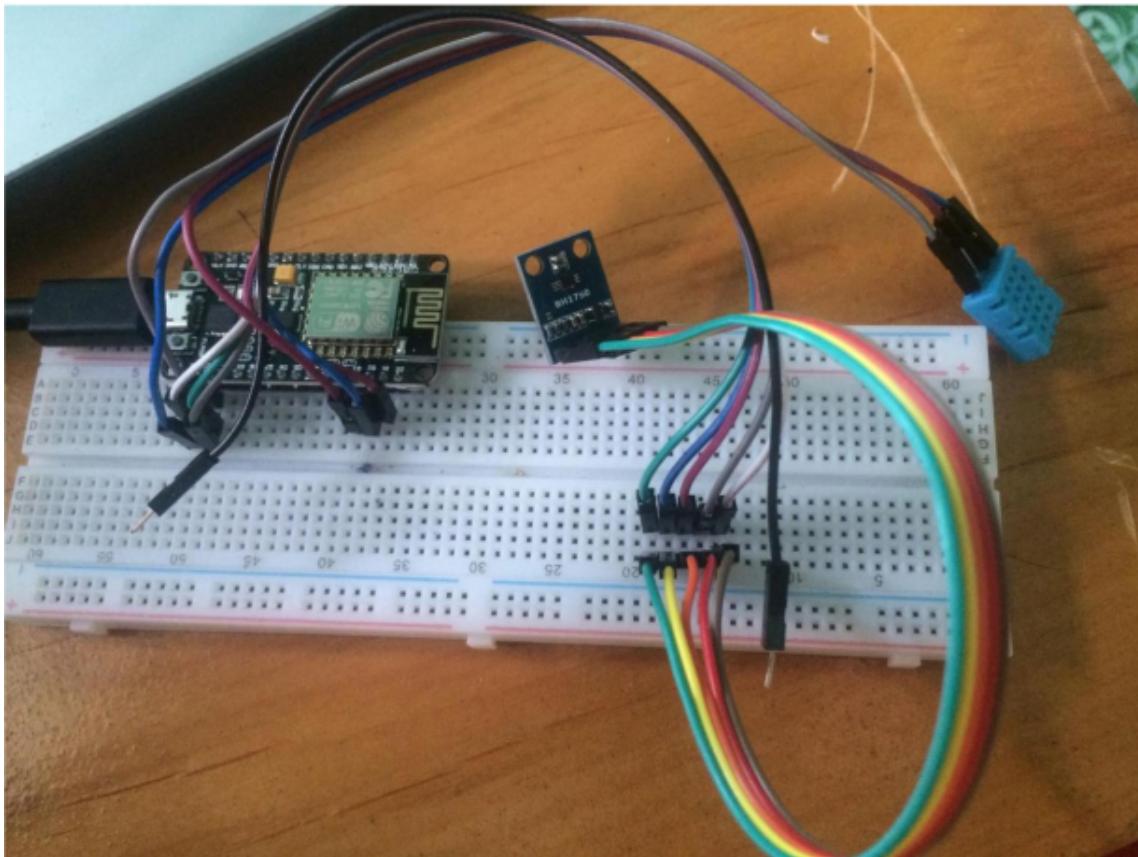
#define DHTTYPE DHT11 // DHT 11
#define dht_dpin 0
DHT dht(dht_dpin, DHTTYPE);
void setup(void)
{
    dht.begin();
    Serial.begin(115200);
    delay(10);
    WiFi.begin(ssid, password); // Connect to the network
    Serial.print("Connecting to ");
    Serial.print(ssid); Serial.println(" ...");
}

```

HÌNH 4.5: Sử dụng Arduinio IDE để lập trình phần cứng

Cứ mỗi 5 phút các sensor sẽ gửi giá trị của các chỉ số cho module Esp8266. ESP8266 thực hiện

nhiệm vụ publish các giá trị đó lên kênh của MQTT. Phía Server ứng dụng sẽ subscribe kênh đó để nhận giá trị trả về theo thời gian thực.



HÌNH 4.6: Hình ảnh triển khai trên thực tế của các thiết bị của dịch vụ giám sát

4.3 Hệ thống ứng dụng Web

Hệ thống ứng dụng đóng vai trò quan trọng cả hệ thống, bao gồm server cung cấp các API để ứng dụng có thể giao tiếp với Front-End, nhận dữ liệu MQTT để xử lý dữ liệu tiền lưu trữ và lưu trong cơ sở dữ liệu đồng thời còn truy xuất và trả về dữ liệu mà Front-End yêu cầu từ Cơ sở dữ liệu. Server sẽ chính là cầu nối giữa ứng dụng với dữ liệu của hệ thống, cung cấp môi trường giao tiếp giữa các phần trong hệ thống.

4.3.1 Phân tích yêu cầu ứng dụng

Yêu cầu chức năng đối với ứng dụng

STT	Tác nhân	Mô tả
1	Người dùng	Những người sử dụng ứng dụng qua thiết bị di động.

TABLE 4.1: Bảng danh sách các tác nhân

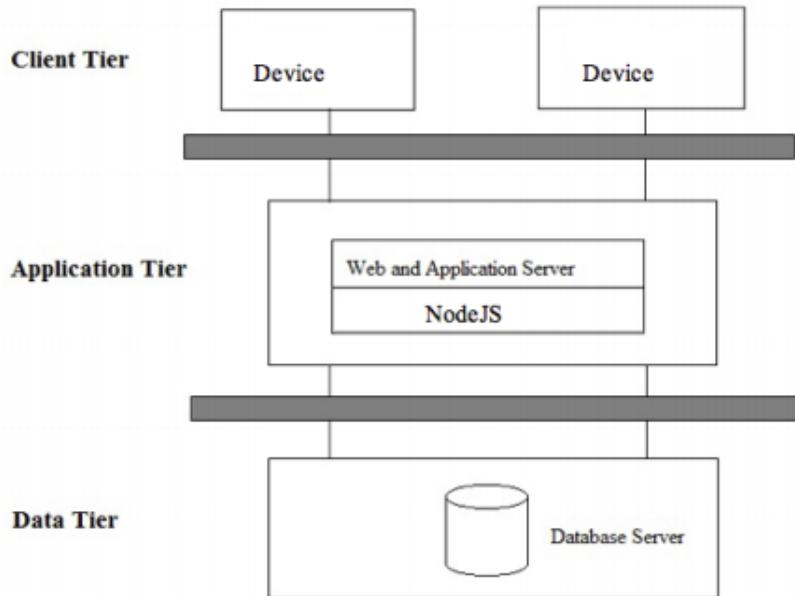
STT	Tên nhóm chức	Tên chức năng	Mô tả
1	- Nhóm chức năng hiển thị trên đồ thị	- Tải và hiển thị bản đồ.	Chức năng tải bản đồ Google Maps, tự động zoom đến vị trí hiện tại của người dùng.
		- Thông tin vị trí và trạng thái.	Chức năng hiển thị vị trí hiện tại của các trạm trên bản đồ dưới dạng Marker. Và thông tin trạng thái của trạm đó ở thời điểm hiện tại
2	- Nhóm chức năng hiển thị trên đồ thị	- Hiển thị dữ liệu thu thập được từ các trạm.	Các dữ liệu từ các trạm đo được sẽ được server gửi lên và được hiển thị trên giao diện theo dạng đồ thị
		- Dự đoán giá trị trong tương lai	Server sẽ gửi dữ liệu dự đoán giá trị của các chỉ số trong vòng 6h và sẽ được hiển thị cùng với dữ liệu được thu thập
3	- Nhóm chức năng thông tin chi tiết	- Xem chi tiết từng trạm	- Cung cấp thông tin chi tiết, đầy đủ về tình trạng của các trạm đo.
		- Cảnh báo khi có bất thường	- Cảnh báo, hướng dẫn người dùng nếu phát hiện bất thường

TABLE 4.2: Bảng các chức năng cần có của ứng dụng

Ngoài ứng dụng còn có tích hợp thêm tính năng cập nhật tình hình thời tiết tại mỗi trạm đo thông qua DarkSky Api với các thông tin về nhiệt độ, độ ẩm, tốc độ gió... Để người dùng có thể kết hợp theo dõi thời tiết.

4.3.2 Thiết kế hệ thống ứng dụng

4.3.2.1 Thiết kế kiến trúc



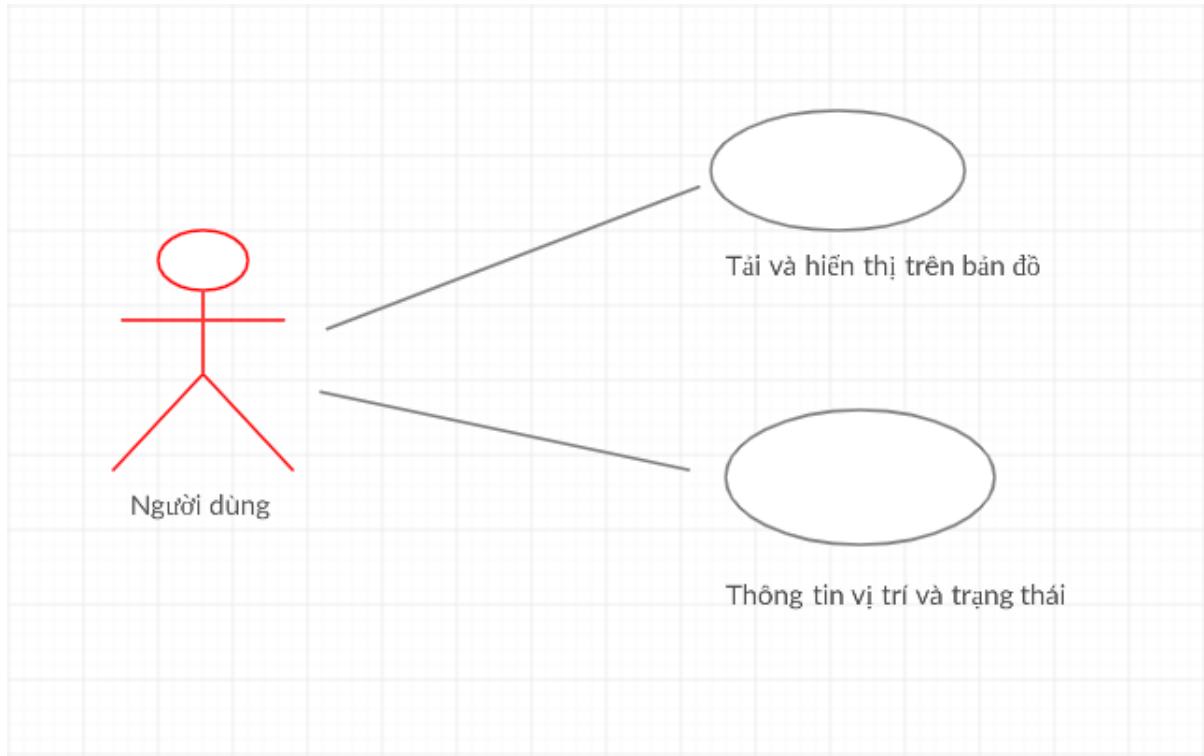
HÌNH 4.7: Mô hình kiến trúc 3 tầng

Hệ thống được xây dựng dựa trên mô hình 3 tầng như sau:

- **Tầng Client:** các thiết bị di động sử dụng trình duyệt web đóng vai trò là client, chứa ứng dụng được xây dựng bằng ngôn ngữ ReactJS. Có nhiệm vụ thực hiện các thao tác gửi request lên Server để hiển thị kết quả cho người dùng.
- **Tầng ứng dụng:** ở tầng này sẽ đảm nhận nhiệm vụ tương tác với client qua việc nhận request, sau đó gọi xuống tầng dữ liệu để thực hiện truy vấn, cuối cùng sẽ gửi lại dữ liệu kết quả cho client.
- **Tầng dữ liệu:** lưu trữ và thực hiện các thao tác truy cập, quản lý, truy vấn đến cơ sở dữ liệu, đóng vai trò trung gian giữa ứng dụng và cơ sở dữ liệu. Ở tầng dữ liệu sẽ lưu trữ các thông tin về các trạm, các giá trị đo đạc, dữ liệu dự báo

4.3.2.2 Biểu đồ Use Case

Dựa vào các chức năng đã được liệt kê, có thể xây dựng biểu đồ Use Case như sau:



HÌNH 4.8: Usecase nhóm chức năng hiển thị trên bản đồ

a) Chức năng tải và hiển thị trên bản đồ

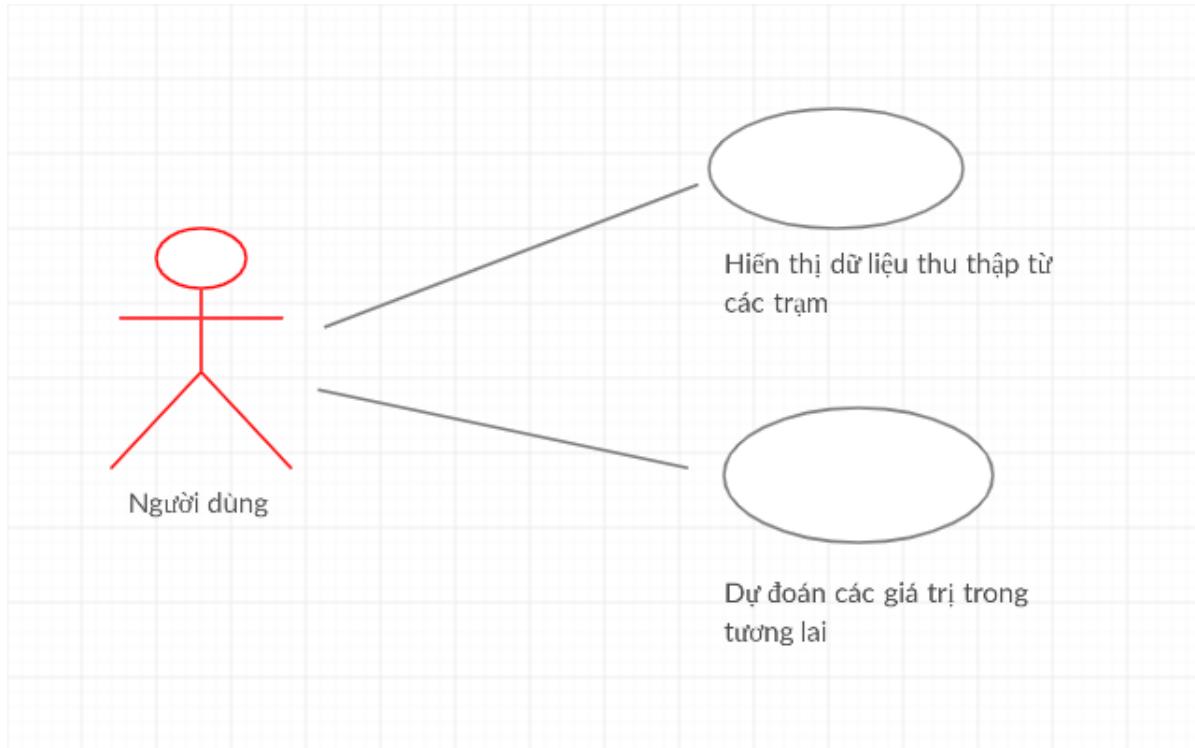
- **Tác nhân chính:** Người dùng
- **Mô tả:** Chức năng tải bản đồ Google Maps, tự động zoom đến vị trí hiện tại của người dùng.
- **Điều kiện bắt đầu:** Người dùng truy cập và đường link của trang web.
- **Điều kiện sau khi hoàn thành:** Người dùng thấy được vị trí trên bản đồ Google Maps

b) Chức năng hiển thị thông tin vị trí và trạng thái

- **Tác nhân chính:** Người dùng
- **Mô tả:** Chức năng hiển thị vị trí hiện tại của các trạm trên bản đồ dưới dạng Marker và thông tin trạng thái của trạm đó ở thời điểm hiện tại.
- **Điều kiện bắt đầu:** Người dùng chọn trạm để xem thông tin
- **Điều kiện sau khi hoàn thành:** Người dùng xem được thông tin và trạng thái của từng trạm.

- **Kịch bản sử dụng chính:**

1. Hệ thống hiển thị các trạm dưới dạng Marker
2. Người dùng chọn marker để xem thông tin và trạng thái của trạm được chọn.



HÌNH 4.9: Usecase nhóm chức năng hiển thị trên đồ thị

c) Chức năng hiển thị dữ liệu thu thập được từ các trạm

- **Tác nhân chính:** Người dùng

- **Mô tả:** Các dữ liệu từ trạm đo được sẽ được server gửi lên và được hiển thị trên giao diện theo dạng đồ thị.

- **Điều kiện bắt đầu:** Người dùng đang ở trang chủ

- **Điều kiện sau khi hoàn thành:** Người dùng thấy được đồ thị các chỉ số theo thời gian

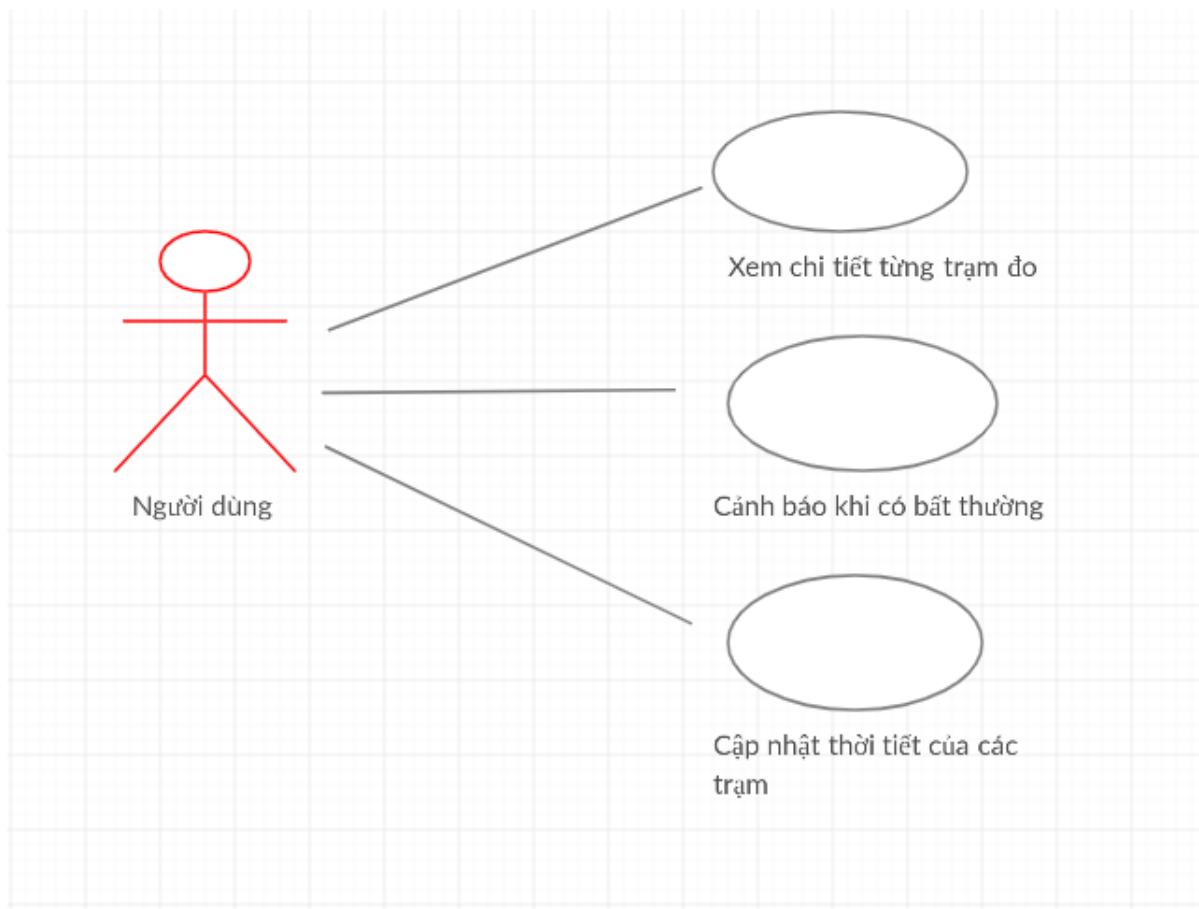
- **Kịch bản sử dụng chính:**

1. Ở Infowindow của mỗi trạm người dùng chọn See More để gửi request thực hiện chức năng
2. Dữ liệu hiển thị cho người dùng

d) Chức năng dự đoán giá trị trong tương lai

- **Tác nhân chính:** Người dùng

- **Mô tả:** Dữ liệu dự báo của mỗi chất ô nhiễm trong vòng 6h của mỗi trạm sẽ được hiển thị cùng với dữ liệu thu thập dưới dạng đồ thị.
- **Điều kiện bắt đầu:** Người dùng chọn trạm để xem thông tin
- **Điều kiện sau khi hoàn thành:** Người dùng xem được thông tin và trạng thái của từng trạm.
- **Kịch bản sử dụng chính:**
 1. Hệ thống hiển thị các trạm dưới dạng Marker
 2. Người dùng chọn marker để xem thông tin và trạng thái của trạm được chọn.



HÌNH 4.10: Usecase nhóm chức năng xem thông tin chi tiết

e) Chức năng xem chi tiết từng trạm

- **Tác nhân chính:** Người dùng
- **Mô tả:** Cung cấp thông tin chi tiết, đầy đủ về tình trạng của các trạm đo.
- **Điều kiện bắt đầu:** Người dùng chọn trạm để xem thông tin
- **Điều kiện sau khi hoàn thành:** Người dùng xem được thông tin chi tiết của trạm

- **Kịch bản sử dụng chính:**

1. Ở Infowindow của mỗi trạm người dùng chọn See More để gửi request thực hiện chức năng
2. Thông tin sẽ được hiển thị cho người dùng

f) Chức năng cảnh báo khuyến cáo

- **Tác nhân chính:** Người dùng

- **Mô tả:** Cảnh báo, hướng dẫn người dùng khi phát hiện các chỉ số bất thường

- **Điều kiện bắt đầu:** Người dùng chọn trạm để xem thông tin

- **Điều kiện sau khi hoàn thành:** Người dùng xem được thông tin chi tiết của trạm

- **Kịch bản sử dụng chính:**

1. Ở Infowindow của mỗi trạm người dùng chọn See More để gửi request thực hiện chức năng
2. Thông tin sẽ được hiển thị cho người dùng

4.3.3 Thiết kế cơ sở dữ liệu

a) Phân tích

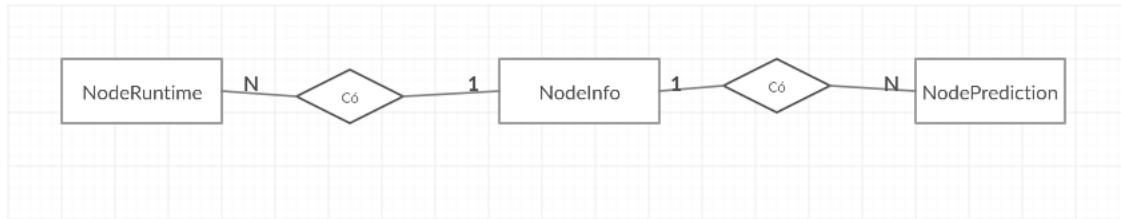
Dựa trên các chức năng của ứng dụng, cơ sở dữ liệu cần phải lưu trữ các thông tin về các trạm do, dữ liệu đo đạt theo thời gian, dữ liệu dự đoán trong tương lai. Đây tương ứng là những thực thể của hệ thống. Mỗi thực thể sẽ có các thuộc tính, là các đặc trưng mô tả được thực thể đó.

Các thuộc tính của thực thể như sau:

- Thực thể **NodeInfo** là nơi lưu trữ các thông tin cơ bản của các trạm được quan sát, gồm các trường:
 - code: mã định danh của mỗi trạm
 - station: tên của trạm tương ứng
 - lat: thông tin kinh độ
 - long: thông tin vĩ độ
- Thực thể **NodeRuntime** lưu trữ toàn bộ dữ liệu được gửi lên của các trạm:
 - stationcode: mã định danh của trạm
 - datetime: thời gian gửi lên
 - so2: thông tin về chỉ số SO_2
 - pm10: thông tin về chỉ số PM10

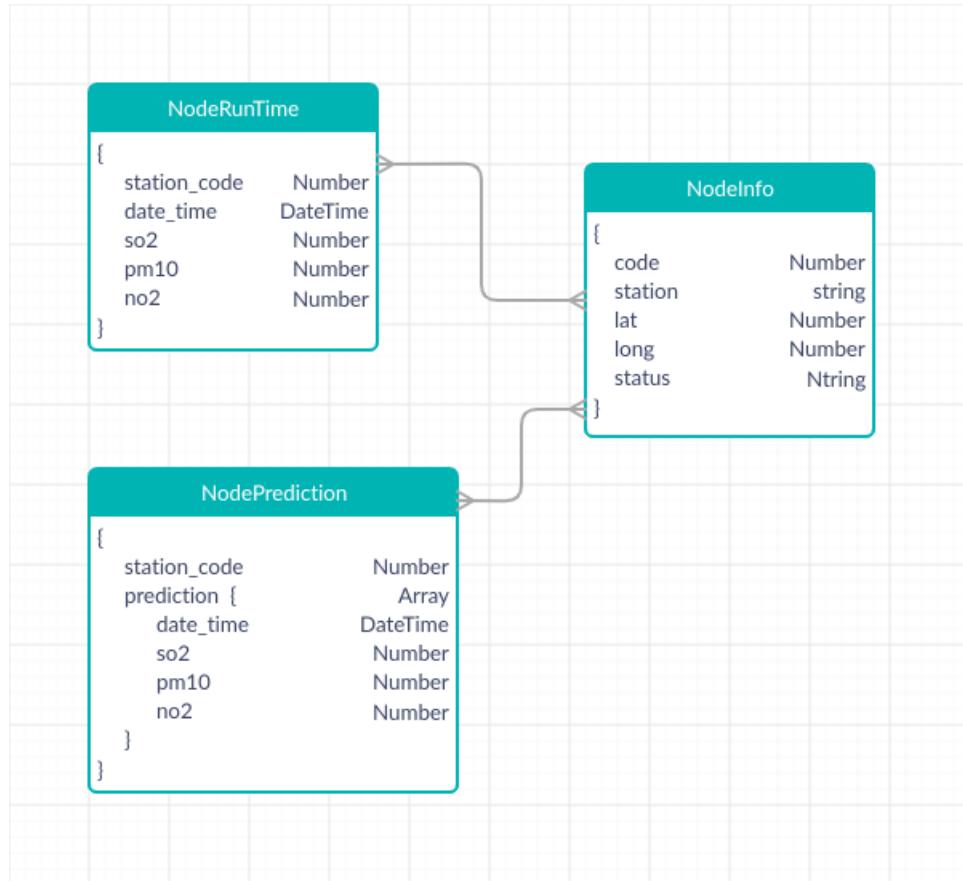
- no2: thông tin về chỉ số NO_2
- Thực thể **NodePrediction** thông tin dự đoán về các chỉ số sẽ được lưu trữ ở đây
 - stationcode: mã định danh của trạm
 - prediction: lưu trữ dưới dạng mảng các thông tin dự đoán các phần tử được tổ chức giống như ở NodeRuntime.

Quan hệ giữa các thực thể



HÌNH 4.11: Mối quan hệ giữa thực thể NodeInfo, NodePrediction và NodeRuntime

Từ những phân tích về các thực thể và các liên kết thực thể nêu trên, có thể mô tả qua lược đồ quan hệ như hình 4.9



HÌNH 4.12: Lược đồ cơ sở dữ liệu của ứng dụng

4.3.4 Cài đặt ứng dụng và kết quả đạt được

a) Quy trình xây dựng

Về phía máy chủ:

- Máy chủ được xây dựng sử dụng Node.JS phiên bản mới nhất, thực hiện các nhiệm vụ tiếp nhận yêu cầu, thực hiện các câu truy vấn đến CSDL, sau đó gửi trả lại kết quả cho máy khách.
- Cơ sở dữ liệu NoSQL được sử dụng là MongoDB có thể lưu trữ các JSON trong đó, và cấu trúc của các tài liệu này có thể thay đổi vì nó không bắt buộc như các cơ sở dữ liệu SQL có nhiệm vụ lưu trữ, xử lý truy vấn về dữ liệu. Sử dụng Mongoose (thư viện mô hình hóa đối tượng cho NodeJS và MongoDB). Nó quản lý mối quan hệ giữa dữ liệu, cung cấp sự xác nhận giản đồ và được sử dụng để dịch giữa các đối tượng trong mã và biểu diễn các đối tượng trong MongoDB.
- Các chức năng được phát triển đầy đủ như đã được phân tích

Về phía máy khách:

- Ứng dụng di động nền tảng Web được xây dựng bằng ReactJS.
- Đăng ký truy cập (get key) với việc hiển thị trên Google Map và trích xuất thông tin thời tiết
- Ứng dụng sẽ được build trực tiếp localhost.

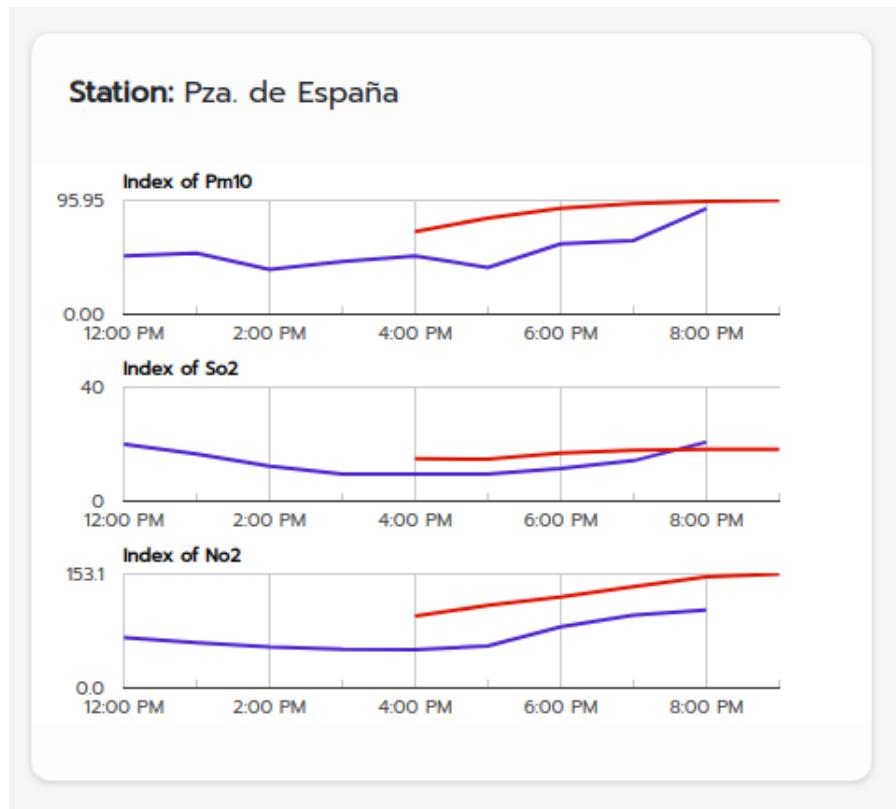
b) Bản thử nghiệm:

- Người dùng truy cập vào trang Web theo đường dẫn **localhost:3000**
- Sau khi đăng nhập hệ thống, màn hình hiển thị bản đồ tự động zoom đến vị trí trạm đo.
- Người dùng thực hiện thao tác để xem các trạm cần xem.
- Chức năng xem hiển thị trên đồ thị và xem thông tin chi tiết sẽ được thực hiện khi người dùng nhấp chọn See More trong phần InfoWindow của mỗi trạm

Sau đây là ảnh chụp màn hình khi sử dụng ứng dụng:



HÌNH 4.13: Màn hình thực hiện hiển thị các trạm đo trên Google Map



HÌNH 4.14: Màn hình thực hiện hiển thị các chỉ số trên đồ thị



HÌNH 4.15: Màn hình thực hiện hiển thị thông tin chi tiết của trạm đo.

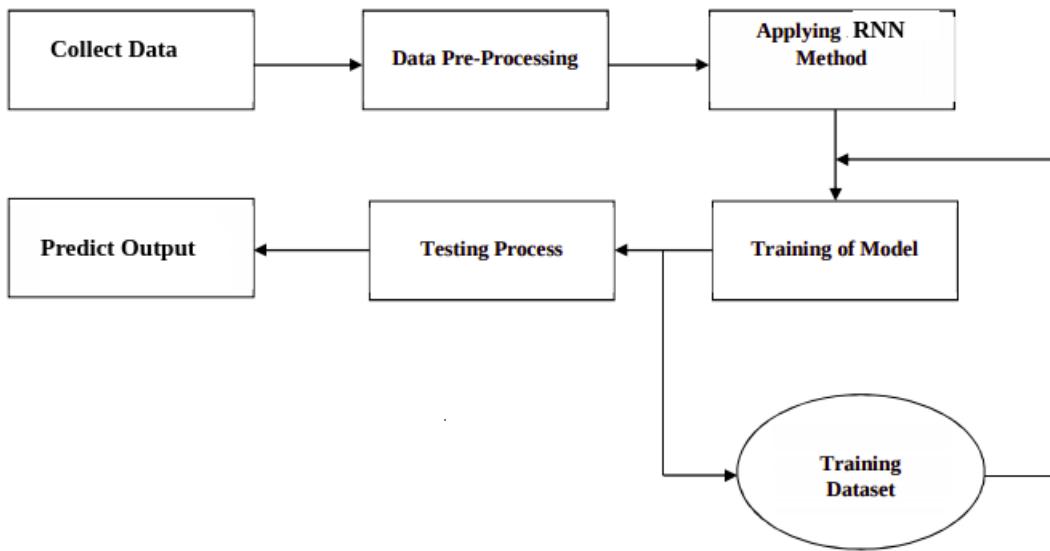
c) Kết quả thử nghiệm

Ứng dụng được xây dựng thử nghiệm thành công trên nền tảng Web, thực hiện được đầy đủ các chức năng như đã phân tích. So với những ứng dụng đã được phát triển hiện nay, ứng dụng được xây dựng trong phạm vi luận văn đã phát triển thêm các chức năng khác để hỗ trợ người dùng trong quá trình sử dụng.

4.4 Áp dụng Recurrent Neural Network vào bài toán giám sát

4.4.1 Mô hình dự báo

Để dự báo các chỉ số môi trường, ta quy nó về Time Series Analysis để có thể dễ áp dụng. Việc dự đoán các giá trị mới sẽ phụ thuộc vào các giá trị xảy ra trước đó. Mô hình dự báo được mô tả trong hình 4.15



HÌNH 4.16: Mô hình dự báo chỉ số không khí

Hệ thống bao gồm 6 module chính:

- Thu thập dữ liệu : Dữ liệu được đề tài sử dụng là mẫu chất lượng không khí tại Madrid, Tây Ban Nha năm 2003.
- Tiền xử lý: Dữ liệu sau khi thu thập sẽ được xử lý.
- Training: Sau khi có được dữ liệu đã xử lý, ta bắt đầu đưa vào model training cho máy có thể tự học.
- Testing Process: Sau khi training, thực hiện lấy dữ liệu test để kiểm tra kết quả sau khi train.
- Predict Output: Nạp dữ liệu thực vào model sau khi train để lấy kết quả dự đoán.

4.4.2 Chuẩn bị dữ liệu

Dữ liệu là một yếu tố bắt buộc trong tất cả các bài toán Machine Learning. Để thực hiện xây dựng model giám sát ta cần dữ liệu tin cậy để cho máy có thể học được, ở đây em dùng tập dữ liệu về chỉ số ô nhiễm không khí của các chất ở Tây Ban Nha, năm 2003

date	CO	NO_2	O_3	PM10	SO_2	station
2003-03-01 01:00:00	1.72000002861023	73.9000015258789#	10.5500001907349	55.209990844727	24.299992370605	28079001
2003-03-01 01:00:00	1.45000004768372	0	72.1100006103516#1	6.7199979019165	52.389993896484	14.229995422363#
2003-03-01 01:00:00	1.57000005245209	80.559975585938#	21.049992370605	63.2400016784668	17.879991607666	28079035
2003-03-01 01:00:00	2.45000004768372	78.370002746582#	4.2199979019165	67.839993378906	24.899996185303	28079004
2003-03-01 01:00:00	3.2599999046326	96.25#	8.46000003814697	95.7799987792969	18.75	28079039
2003-03-01 01:00:00	1.94000005722046##0	90.30000030517578#	9.9499980926514	95.1500015258789#	29.2700004577637##	28079006
2003-03-01 01:00:00	1.3799999523163	0	89.5800018310547#	7.1999980926514	54	23.709990844727#
2003-03-01 01:00:00	1.58000004291534	0	93.639993896484#	4.1900005722046	26.6200008392334	17.739997711182#
2003-03-01 01:00:00						28079008
2003-03-01 01:00:00	1.9199995708466	71.8399963378906#	5.3299992370606	39.3600006103516	21.639993896484#	28079038
2003-03-01 01:00:00	1.33000004291534	0	87.9199981689453#	4.25	52.2700004577637	24.700007629395#
2003-03-01 01:00:00	2.1800000667572	89.8499984741211#	8.4099984741211	78.0599975585938	34	28079012
2003-03-01 01:00:00	1.1399998569489	0	73.870002746582#	7.2199979019165	63.2700004577637	7.889998664856#
2003-03-01 01:00:00	4.6799998233862	42.189998626709#	7.5799992370606	103.5	16.5	28079014
2003-03-01 01:00:00	1.44000005722046#	0	91.0100021362305#	1.1900005722046	50.5400009155273	20.0400009155273##
2003-03-01 01:00:00	1.25	70.5699996948242#	9.68000030517578	67.9499969482422	14.829999237061	28079016
2003-03-01 01:00:00	1.6399998569489	45.4700012207031#	6.1900005722046	81.970001207031	11.979995422363	28079017
2003-03-01 01:00:00	1.85000002384186	1	94.5100021362305#	7.5100002288184	52	16.8500003814697#
2003-03-01 01:00:00	1.74000000953674	62.2599983215332#	2.64000010490417	58.0299987792969	17.909998474121	28079019
2003-03-01 01:00:00	1.5399996185303	73.2399978637695#	2.6700007629395	67.6900024414063	20.790009155273	28079036
2003-03-01 01:00:00	1.73000001907349	6.9000009536743#	7.7699998092651	72.309975585938	26.1900005340573	28079021
2003-03-01 01:00:00	1.75	88.4800033569336#	7.5300020980835	63.599984741211	23.389993896484#	28079022
2003-03-01 01:00:00	1.32000005245209#	0	84.8499984741211#	5.4299982833862	75.0100021362305	13.199998092651##
2003-03-01 01:00:00	1.2699998092651##0	54.25##	6.53999986185303	53.0099983215332#	8.80000019073486##	28079024
2003-03-01 01:00:00	0.959999978542328##	89.3300018310547##	6.6300001144092	69.629997253418#	13.529997329712#	28079025
2003-03-01 01:00:00		0	95.120002746582#	7.7199979019165	61.479995422363	9.689995803833#
						28079026

HÌNH 4.17: Mẫu dữ liệu tại Madrid, Tây Ban Nha

Bộ dữ liệu được thu thập từ nhiều trạm và mỗi lần đo cách nhau 1h gồm các chỉ số về các chất gây ô nhiễm như CO, NO₂, PM10, SO₂.

4.4.3 Xử lý dữ liệu

Dữ liệu sẽ được tiền xử lý. Bỏ qua đối với các giá trị null. Vì các giá trị có bản chất giống nhau ở các chất nên ở đây ta chỉ cần sử dụng một mẫu dữ liệu là SO₂ để làm mẫu cho việc training. Đối với việc training dữ liệu, ta cần các giá trị input và output theo chuỗi. Ở đây, ta xử dụng input là chuỗi array các giá trị nồng độ SO₂ làm input và output cũng là 1 chuỗi array các giá trị nồng độ SO₂ sau đó 1h.

Input	Output
15.97	15.9
15.9	15.83
15.83	15.85
15.85	16.24
16.24	16.69
16.69	17.22
17.22	17.72
17.72	18.25

HÌNH 4.18: Mẫu training của khí SO₂

Đoạn code dưới đây hiện thực cách lấy mẫu dữ liệu training.

```

function createDataTrain(data, factor){
    var xTrain = []
    data.forEach((element) => {
        if (element[factor] > 0) xTrain.push([element[factor]])
    })

    var yTrain = []
    xTrain.forEach((element, index) => {
        if(index === 0) return
        yTrain.push(element)
    })
    var xTest = xTrain.pop();
    var dataTrain = [];
    yTrain.forEach((element, index) => {
        dataTrain.push({ input: xTrain[index], output: yTrain[index] })
    })
}

return {dataTrain, xTest}
}

```

HÌNH 4.19: Hàm lấy dữ liệu train

4.4.4 Sử dụng thư viện BrainJS để áp dụng giải thuật Recurrent Neural Network

Để áp dụng thuật toán Recurrent Neural Network, ở đây em sử dụng thư viện có BrainJS là một thư viện có sẵn và có khả năng training dữ liệu. Brain là một thư viện JavaScript được tạo ra cho mạng neural network cấp cao và dễ dàng. Brain.js nhanh chóng tạo ra một Neural Network đơn giản bằng ngôn ngữ cấp cao nơi bạn có thể tận dụng số lượng lớn các thư viện nguồn mở. Với một bộ dữ liệu tốt và một vài dòng mã, bạn có thể tạo ra một số chức năng thực sự thú vị. Brain xử lý gần như tất cả các thiết lập cho bạn, cho phép bạn chỉ lo lắng về các quyết định cấp cao.

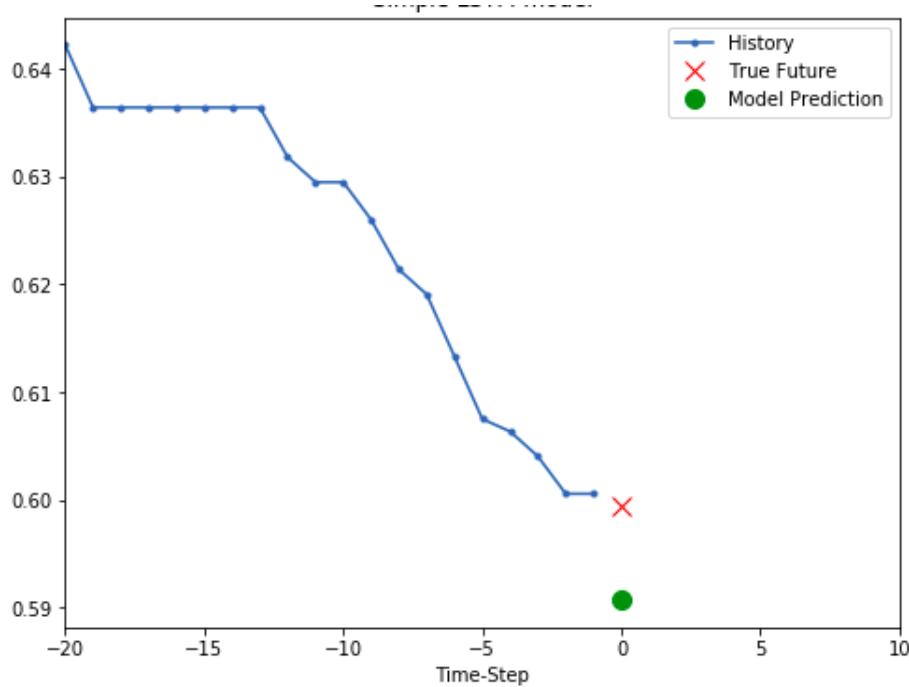
```

function predictNode(data, factor) {
    var {dataTrain, xTest} = createDataTrain(data, factor);
    const net = new brain.recurrent.RRN([
        inputSize: 1,
        hiddenLayers: [3],
        outputSize: 1
    ]);
    net.train(dataTrain, { log: false, iterations: 600 });
    const output = net.forecast(xTest, 6 );
    return output
}

```

HÌNH 4.20: Hàm train áp dụng RNN

Hàm train được áp dụng với kích thước dữ liệu train là 200 data, inputSize là 1, outputSize là 1, số lớp ẩn giữa output và input là 3, số lần lặp training là 600. Kết quả được hiển thị như hình.



HÌNH 4.21: Simple RNN model

4.4.5 Đánh giá mô hình RNN

Do chưa đi sâu tìm hiểu về các cách đánh giá của các mô hình học máy nên trong phần này em chỉ đánh giá tổng quát các kết quả dự báo. Với dữ liệu được đánh giá là chỉ số của SO₂. Đánh giá sơ bộ các kết quả dự đoán dựa trên độ lệch của giá trị dự đoán và giá trị thực trong vòng liền tiếp (sau 1h) thì độ chính xác 70-80%. Độ chính xác giảm dần sau các lần dự đoán độ chính xác sau 3h là 30-40%. Với kết quả thu được khá khả quan nhưng vẫn chưa tối ưu. Trong hướng phát triển của đề tài ở giải đoạn sau cần tập trung nghiên cứu, phân tích phần này để có những dự báo chính xác hơn nữa.

Chương 5

Kết quả và đánh giá

Luận văn đã trình bày cách thiết kế, xây dựng cũng như hiệu quả của hệ thống giám sát và cảnh báo chất lượng không khí thông qua thiết bị di động. Những chức năng tính khả thi cao trong ứng dụng thực tế. Các kết quả đã đạt được như sau:

- Xây dựng được hệ thống giám sát với module phần cứng ESP 8266 và các sensor.
- Hiện thực ứng dụng sử dụng Nodejs + Mongodb phần server và ReactJS ở front-end.
- Tích hợp giải thuật Recurrent Neural Network vào ứng dụng để dự đoán các chỉ số trong tương lai.

5.1 Ưu điểm

- Hệ thống được xây dựng tương đối đầy đủ các chức năng giám sát và cảnh báo như yêu cầu đã đặt ra. Đây cũng là chức năng chính của ứng dụng, là mục đích chúng tôi thực hiện từ lúc bắt đầu cho đến khi kết thúc quá trình làm luận văn
- Sử dụng nhiều công nghệ mới với tốc độ xử lý cao, mang lại hiệu quả lớn.
- Hệ thống được thiết kế có thể mở rộng để phát triển lớn hơn.

5.2 Khuyết điểm

- Việc em xác định mục tiêu ban đầu của đề tài chưa đúng nên dẫn đến quá trình thực hiện đề tài bị chậm.
- Do thời gian giới hạn và năng lực còn hạn chế nên việc áp dụng machine learning vào bài toán giám sát còn nhiều hạn chế.
- Chưa triển khai hệ thống vào thực tiễn để có đánh giá nhận xét khách quan

Chương 6

Tổng kết và hướng phát triển

6.1 Tổng kết

Trong lĩnh vực môi trường, công nghệ IoT có thể hỗ trợ cải thiện hiệu quả và nâng cao khả năng thu thập dữ liệu môi trường, phân tích, dự báo, quản lý nguồn lực và hỗ trợ ra quyết định về quản lý môi trường. Với xu hướng đó, việc xây dựng một ứng dụng di động phục vụ việc giám sát, cảnh báo chất lượng không khí sẽ có tính khả thi cao trong ứng dụng thực tế.

Về mặt công nghệ, luận văn đã nghiên cứu các nền tảng phổ biến phù hợp để xây dựng ứng dụng. Với những ưu điểm về phổ biến và cách tiếp cận đơn giản khóa luận đã lựa chọn phát triển ứng dụng trên nền tảng Web. Về phía máy chủ, lựa chọn xây dựng trên nền tảng Node.JS với ưu điểm là mã nguồn mở, được viết bằng ngôn ngữ JavaScript. Bên cạnh đó, máy chủ sử dụng hệ quản trị cơ sở dữ liệu Mongoddb với ưu điểm nổi bật hơn so với các cơ sở dữ liệu khác là lưu trữ dữ liệu cung cấp hiệu suất cao, tính sẵn sàng cao và tự động mở rộng.

Quá trình thực hiện đề tài trải qua nhiều giai đoạn. Việc triển khai thực hiện cơ bản đúng với nội dung nguyên cứu. Kết quả thu được một hệ thống giám sát và cảnh báo chất lượng không khí nhưng vẫn còn ở mức đơn giản do không có đủ thời gian để nguyên cứu hệ thống với dữ liệu lớn, đa dạng, chưa đi sâu vào giải thuật dự đoán các chỉ số và độ chính xác của việc chạy giải thuật chưa được cao. Với hệ thống này, em mong muốn sẽ phần nào hỗ trợ con người trong việc theo dõi chất lượng không khí xung quanh.

6.2 Hướng phát triển

Trong tương lai, cần mở rộng đề tài với việc giám sát các chỉ số cấp thiết mang tính thời sự hơn, nguyên cứu sự tương tác, tác động lẫn nhau giữa các chỉ số để hệ thống cảnh báo tốt hơn. Hệ thống cần được xây dựng để đáp ứng với số lượng sensor lớn và việc dự đoán ở các chỉ số ở tương lai chuẩn xác hơn. Hướng đến mục tiêu, đưa hệ thống vào đời thực để hỗ trợ con người trong phòng tránh ô nhiễm không khí.

Tài liệu tham khảo

- [1] S. Poornima * and M. Pushpalatha *Advanced Weather Forecasting Prediction using Deep Learning*.Department of Computer Science Engineering, SRM Institute of Science and Technology, Kattangulathur 603203, India
- [2] Tổng quan về Recurrent Neural Network
<https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks>
- [3] Advanced Weather Forecasting Prediction using Deep Learning
<https://www.ijraset.com/fileserve.php?FID=24817>
- [4] Quản lý chất lượng không khí.
https://www.sutp.org/files/contents/documents/resources/A_Sourcebook/SB5_Environment%20and%20Health/GIZ_SUTP_SB5a_Air-Quality-Management_VN.pdf
- [5] Lập trình ESP8266 Arduino.
<https://arduino.esp8266.vn/#lap-trinh-esp8266-arduino>