|  |  |
| --- | --- |
|  |  |
| **HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**  **---------------------------------------** | |
|  | |
|  | |
| **Nguyễn Quang Biên** | |
|  | |
|  | |
| **NGHIÊN CỨU VÀ TRIỂN KHAI THUẬT TOÁN PHÁT HIỆN**  **BẤT THƯỜNG CHO HỆ THỐNG IoT TẠI BIÊN** | |
|  | |
| **ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT**  **(*Theo định hướng ứng dụng*)**  HÀ NỘI- NĂM 2024 | |
|  | |
|  |  |
| **HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**  **---------------------------------------** | |
|  | |
|  | |
| **Nguyễn Quang Biên** | |
|  | |
| **NGHIÊN CỨU VÀ TRIỂN KHAI THUẬT TOÁN PHÁT HIỆN**  **BẤT THƯỜNG CHO HỆ THỐNG IoT TẠI BIÊN** | |
| **Chuyên ngành : Kỹ thuật Điện tử** | |
| **Mã số: 8.52.02.03**    **ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT**  ***(Theo định hướng ứng dụng)***  NGƯỜI HƯỚNG DẪN KHOA HỌC  **TS. NGUYỄN NGỌC MINH** | |
| HÀ NỘI- NĂM 2024 | |

# LỜI CAM ĐOAN

Tôi cam đoan tất cả nội dung trong đề án *“Nghiên cứu và triển khai thuật toán phát hiện bất thường cho hệ thống IoT tại biên”* là công trình nghiên cứu của riêng tôi dưới sự hướng dẫn, chỉ bảo, góp ý của TS. Nguyễn Ngọc Minh, giảng viên Khoa Kỹ thuật điện tử 1, Học viện Công nghệ Bưu chính Viễn thông. Các số liệu, kết quả nêu trong đề án tốt nghiệp là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác và không sao chép dưới mọi hình thức. Các tài liệu tham khảo trong đề án được trích dẫn đầy đủ, rõ ràng.

Tôi xin chịu mọi trách nhiệm về công trình nghiên cứu của riêng mình !

**Tác giả đề án**

**Nguyễn Quang Biên**

# LỜI CẢM ƠN

Lời đầu tiên và trên hết, em xin bày tỏ lời cảm ơn sâu sắc đến người hướng dẫn khoa học TS. Nguyễn Ngọc Minh – Giảng viên, khoa Kỹ thuật điện tử 1, Học viện công nghệ bưu chính viễn thông, người không chỉ hướng dẫn trực tiếp về mặt khoa học mà còn hỗ trợ em về mọi mặt và động viên để em có thể hoàn thành đề án này sau hơn một năm học Thạc sĩ. Hơn nữa, chính những kiến thức, sự chỉ bảo tận tâm, sự động viên cổ vũ của thầy đã giúp em có nhiều cơ hội, dự định học lên những cấp cao hơn.

Em cũng xin chân thành cảm ơn toàn thể giảng viên của khoa Kỹ thuật điện tử 1 và khoa Đào tạo Sau Đại học đã luôn giúp đỡ, luôn nhiệt tâm giảng dạy, hướng dẫn và chia sẻ những kiến thức và kinh nghiệm quý báu cho em trong suốt những năm học vừa qua về lĩnh vực ICT nói chung và lĩnh vực Điện tử nói riêng. Đối với em, đây là một hành trình tương đối khó khăn và đầy thử thách, cả trong thời gian học tập cũng như nghiên cứu. Nhưng nhờ sự giúp đỡ và tận tình chỉ dạy của các thầy cô trong quá trình học Thạc sĩ tại trường, em đã có thêm những kiến thức, được tạo nền tảng, dạy cách tư duy, định hướng để em có thể hoàn thành quá trình học tập và thực hiện đề án này và giúp em có thể tự tin hơn trong những hành trang phía trước của mình. Em sẽ mãi khắc ghi công ơn dạy dỗ này của các thầy cô trong khoa Kỹ thuật điện tử 1 và khoa Đào Tạo Sau Đại Học.

Em xin cảm ơn các anh chị và các bạn học viên của lớp M22CQDT02-B đã luôn sát cánh và đồng hành cùng em trong quá trình học tập tại trường và hoàn thành đề án.

Cuối cùng, em dành những lời yêu thương nhất đến gia đình em, mẹ và chị gái em. Sự động viên, giúp đỡ và sự hi sinh, nhẫn nại của họ là động lực mạnh mẽ giúp tôi vượt qua mọi khó khăn để hoàn thành đề án này.

Xin chân thành cảm ơn!

# MỤC LỤC

[LỜI CAM ĐOAN i](#_Toc166052967)

[LỜI CẢM ƠN ii](#_Toc166052968)

[MỤC LỤC iii](#_Toc166052969)

[DANH MỤC CHỮ VIẾT TẮT v](#_Toc166052970)

[DANH MỤC BẢNG vii](#_Toc166052971)

[DANH MỤC HÌNH ẢNH viii](#_Toc166052972)

[MỞ ĐẦU 1](#_Toc166052973)

[CHƯƠNG 1 – TỔNG QUAN VỀ IoT, ĐIỆN TOÁN BIÊN VÀ KỸ THUẬT PHÁT HIỆN BẤT THƯỜNG 3](#_Toc166052974)

[1.1 Tổng quan về IoT (Internet of Things) 3](#_Toc166052975)

[1.1.1 Lịch sử phát triển, thành tựu của công nghệ IoT 3](#_Toc166052976)

[1.1.2 Các vấn đề trong IoT 6](#_Toc166052977)

[1.2 Giới thiệu chung về mô hình điện toán biên trong IoT 8](#_Toc166052978)

[1.3 Tổng quan về hệ thống phát hiện bất thường 13](#_Toc166052979)

[1.4 Tổng quan về hệ thống phát hiện bất thường cho IoT tại biên 17](#_Toc166052980)

[1.4.1 Phân loại phát hiện sự bất thường trong IoT 17](#_Toc166052981)

[1.4.2 Các thách thức trong phát hiện sự bất thường cho IoT 20](#_Toc166052982)

[1.5 Kết luận chương 1 21](#_Toc166052983)

[CHƯƠNG 2 – NGHIÊN CỨU MỘT SỐ THUẬT TOÁN PHÁT HIỆN BẤT THƯỜNG CHO IOT 22](#_Toc166052984)

[2.1 Giới thiệu chung 22](#_Toc166052985)

[2.2 Một số thuật toán phát hiện bất thường cho IoT 24](#_Toc166052986)

[2.2.1 Autoencoder 25](#_Toc166052987)

[2.2.2 Mạng Nơ-Ron sâu (DNN) 29](#_Toc166052988)

[2.2.3 Mạng Nơ-Ron tích chập (CNN) 33](#_Toc166052989)

[2.2.4 SVM 36](#_Toc166052990)

[2.3 Một số cảm biến cho hệ thống IoT tại biên 38](#_Toc166052991)

[2.3.1 Cảm biến gia tốc ADXL345 39](#_Toc166052992)

[2.3.2 Cảm biến giám sát điện áp và dòng điện DC INA3221 40](#_Toc166052993)

[2.4 Tập dữ liệu sử dụng trong mô hình phát hiện bất thường tại biên 41](#_Toc166052994)

[2.5 Kết luận chương 2 44](#_Toc166052995)

[CHƯƠNG 3 – ÁP DỤNG THUẬT TOÁN PHÁT HIỆN BẤT THƯỜNG CHO HỆ THỐNG NHÚNG TẠI BIÊN 45](#_Toc166052996)

[3.1 Đề xuất xây dựng mô hình hệ thống phát hiện bất thường cho IoT tại biên. 45](#_Toc166052997)

[3.2 Thiết lập thử nghiệm hệ thống phát hiện bất thường dựa trên thuật toán 45](#_Toc166052998)

[3.3 Mô phỏng, thử nghiệm giao thức đề xuất 45](#_Toc166052999)

[3.4 Đánh giá kết thực nghiệm 45](#_Toc166053000)

[3.3.1 Các tham số đánh giá 45](#_Toc166053001)

[3.3.2 Đánh giá kết quả thử nghiệm mô hình đề xuất 45](#_Toc166053002)

[3.5 Kết luận chương 3 45](#_Toc166053003)

[KẾT LUẬN VÀ KIẾN NGHỊ 46](#_Toc166053004)

[DANH MỤC TÀI LIỆU THAM KHẢO 47](#_Toc166053005)

# DANH MỤC CHỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Từ viết tắt** | **Nghĩa Tiếng Anh** | **Nghĩa Tiếng Việt** |
| AI | Artificial Intelligence | Trí tuệ nhân tạo |
| CNN | Convolutional Neural Network | Mạng nơ-ron tích chập |
| CPU | Central Processing Unit | Bộ phận xử lý trung tâm |
| CSO | Cuckoo Search Optimization | Tối ưu hóa tìm kiếm Cuckoo |
| DNN | Deep Neural Network | Mạng nơ-ron sâu |
| DoS | Denial of Service | Tấn công từ chối dịch vụ |
| DDoS | Distributed Denial of Service | Tấn công từ chối dịch vụ phân tán |
| DT | Decision Tree | Cây quyết định |
| EMS | Event Management System | Hệ thống quản lý sự kiện |
| FA | Firefly Algorithm | Thuật toán đom đóm |
| FFDNN | Feed Forward Deep Neural Networks | Mạng thần nơ-ron chuyển tiếp |
| FTP | [File Transfer Protocol](https://en.wikipedia.org/wiki/File_Transfer_Protocol) | Giao thức truyền file |
| GA | Genetic Algorithm | Thuật toán di truyền |
| HTTP | Hypertext Transfer Protocol | Giao thức truyền siêu văn bản |
| IDS | Intrusion Detection System | Hệ thống phát hiện xâm nhập |
| IoT | Internet of Things | Internet vạn vật |
| IP | Internet Protocol | Giao thức Internet |
| KNN | K - Nearest Neighbor | K-Láng giềng gần nhất |
| LR | Logistic Regression | Hồi quy logistic |
| LR-CSO | Logistic Regression- Cuckoo Search Optimization | Hồi quy tuyến tính-Tối ưu hóa tìm kiếm Cuckoo |
| LR-FA | Logistic Regression-Firefly Algorithm | Hồi quy tuyến tính-Thuật toán đom đóm |
| LR-GA | Logistic Regression-Genetic Algorithm | Hồi quy tuyến tính-Thuật toán di truyền |
| LR-PSO | Logistic Regression-Particle Swarm Optimization | Hồi quy tuyến tính-Tối ưu hóa bầy đàn |
| LSTM | Long Short-Term Memory | Trí nhớ ngắn hạn |
| ML | Machine Learning | Học máy |
| POP | Post Office Protocol | Giao thức bưu điện |
| PSO | Particle Swarm Optimization | Tối ưu hóa bầy đàn |
| RNN | Recurrent Neural Networks | Mạng nơ-ron hồi quy |
| SMTP | Simple Mail Transfer Protocol | Giao thức truyền mail đơn giản |
| SVM | Support Vector Machine | Máy vectơ hỗ trợ |
| TCP | Transmission Control Protocol | Giao thức điều khiển truyền dẫn |
| UDP | User Datagram Protocol | Giao thức gói dữ liệu người dùng |
| WSN | Wireless Sensor Network | Mạng cảm biến không dây |

# DANH MỤC BẢNG

[Bảng 2.1 Tóm tắt các kịch bản IoT độc hại và lành tính **Error! Bookmark not defined.**](#_Toc136525118)

[Bảng 3.1 Tất cả loại tấn công của tập dữ liệu IoT-23 **Error! Bookmark not defined.**](#_Toc137479858)6

[Bảng 3.2 Tập dữ liệu sử dụng trong mô hình đề xuất được trích xuất trong tập dữ liệu IoT-23 **Error! Bookmark not defined.**](#_Toc137479859)

[Bảng 3.3 Tham số mô hình **Error! Bookmark not defined.**](#_Toc137479860)

[Bảng 3.4 Bảng ma trận nhầm lẫn đa lớp (Confusion Matrix) **Error! Bookmark not defined.**](#_Toc137479861)

[Bảng 3.5 Hiệu suất của mô hình đề xuất trên tập dữ liệu IoT-23 **Error! Bookmark not defined.**](#_Toc137479862)

[Bảng 3.6 Hiệu suất của mô hình đề xuất theo từng loại tấn công **Error! Bookmark not defined.**](#_Toc137479863)

# DANH MỤC HÌNH ẢNH

[Hình 1. 1: Mô hình điện toán biên 9](#_Toc166053006)

[Hình 1. 2: Ví dụ bất thường điểm trong một tập dữ liệu trong không gian hai chiều. 14](#_Toc166053007)

[Hình 1. 3: Ví dụ về bất thường bối cảnh 16](#_Toc166053008)

[Hình 1. 4: Ví dụ về bất thường tập hợp. 17](#_Toc166053009)

[Hình 1. 5: Tổng quan về phân loại bất thường 18](#_Toc166053010)

[Hình 2.1 Thuật toán SVM ………………………………………………………..**Error! Bookmark not defined.**](#_Toc137479975)

[Hình 2.2 Sơ đồ mô tả thuật toán Random Forest **Error! Bookmark not defined.**](#_Toc137479976)

[Hình 2.3 Mạng nơ-ron bốn lớn với hai lớp ẩn **Error! Bookmark not defined.**](#_Toc137479977)

[Hình 3.1 Mô hình hệ thống ………………………………………………………..**Error! Bookmark not defined.**](#_Toc137479985)

[Hình 3.2 Tập dữ liệu IoT-23 ban đầu với 23 đặc trưng **Error! Bookmark not defined.**](#_Toc137479986)

[Hình 3.3 Tập dữ liệu IoT-23 sau khi tiền xử lý dữ liệu **Error! Bookmark not defined.**](#_Toc137479987)

[Hình 3.4 Dữ liệu sau khi phân cụm và được trực quan hóa **Error! Bookmark not defined.**](#_Toc137479988)

[Hình 3.5 Biểu đồ huấn luyện của phương pháp LR-GA (bao gồm cả *accuracy* và *loss*) **Error! Bookmark not defined.**](#_Toc137479989)

[Hình 3.6 Biểu đồ huấn luyện của phương pháp LR-PSO (bao gồm cả *accuracy* và *loss*) **Error! Bookmark not defined.**](#_Toc137479990)

[Hình 3.7 Biểu đồ huấn luyện của phương pháp LR-CSO (bao gồm cả *accuracy* và *loss*) **Error! Bookmark not defined.**](#_Toc137479991)

[Hình 3.8 Biểu đồ huấn luyện của phương pháp LR-FA (bao gồm cả *accuracy* và *loss*) **Error! Bookmark not defined.**](#_Toc137479992)

# MỞ ĐẦU

Sự phát triển nhanh chóng của công nghệ đã giúp thu nhỏ các thiết bị nhúng và giảm tiêu thụ năng lượng, cùng với sự tiến bộ trong việc tối ưu hóa các thuật toán học máy (ML), đã mở ra một triển vọng mới trong lĩnh vực Internet vạn vật (IoT). Trong bối cảnh này, học máy cực nhỏ (TinyML) đã nổi lên, yêu cầu triển khai các thuật toán ML trực tiếp trên các thiết bị IoT. Việc triển khai các thuật toán ML ngay trên các thiết bị IoT giúp phân tích dữ liệu ngay tại chỗ, hay còn gọi là Edge Computing với mục tiêu tối ưu hóa hệ thống tính toán đám mây. Thay vì gửi dữ liệu từ các thiết bị IoT về trung tâm dữ liệu hoặc máy chủ đám mây để xử lý, Edge Computing cho phép việc xử lý dữ liệu diễn ra ngay tại các thiết bị, trước khi dữ liệu được chuyển đi. Điều này mang lại nhiều lợi ích, bao gồm giảm độ trễ, tăng cường bảo mật và quản lý dữ liệu hiệu quả hơn. Sự kết hợp giữa IoT, TinyML và Edge Computing tạo ra một hệ sinh thái thông minh và linh hoạt. TinyML, hay Machine Learning nhúng, cho phép các thiết bị IoT thực hiện các nhiệm vụ học máy ngay tại chỗ mà không cần kết nối đến một máy chủ đám mây. Điều này không chỉ giúp giảm độ trễ trong việc phản ứng và xử lý dữ liệu mà còn giảm bớt gánh nặng cho hệ thống mạng. Qua đó, các thiết bị IoT giờ đây có khả năng hoạt động độc lập một cách đáng tin cậy mà không cần phải phụ thuộc vào việc truy cập dữ liệu từ các dịch vụ đám mây, đồng thời vẫn cung cấp các dịch vụ học máy chính xác. Đặc biệt là trong những môi trường thường xuyên có kết nối không đầy đủ.

Với sự tiến bộ trong công nghệ hệ thống nhúng và phát triển đáng kể của ngành công nghiệp Internet of Things, việc áp dụng Machine Learning để phát hiện các sự bất thường trên thiết bị IoT ngay tại hệ thống nhúng hạn chế tài nguyên đã trở thành một giải pháp hứa hẹn. Tuy nhiên, triển khai Machine Learning trên các thiết bị có tài nguyên hạn chế không đơn giản. Đó là bởi vì đòi hỏi tối ưu hóa, xử lý hiệu quả, độ tin cậy cao, và bảo trì mô hình trong môi trường có sức mạnh tính toán thấp. Đáp ứng nhu cầu này, đã xuất hiện nhiều nền tảng hỗ trợ triển khai Machine Learning trực tiếp trên các thiết bị nhúng, trong đó Edge Impulse là một trong những nền tảng nổi bật. Edge Impulse là một nền tảng phát triển và triển khai các ứng dụng AI và Machine Learning trực tiếp trên các thiết bị nhúng. Qua đó, Edge Impulse có thể thu thập dữ liệu từ cảm biến trên thiết bị, hỗ trợ đào tạo mô hình Machine Learning và triển khai chúng trực tiếp lên thiết bị, giúp giảm thiểu độ trễ và lưu lượng dữ liệu mạng. Đồng thời, trong đề án này tập trung vào việc triển khai thuật toán phát hiện bất thường cho hệ thống IoT tại biên, với mục tiêu đảm bảo kích thước mô hình nhỏ để chạy trên các thiết bị nhúng hạn chế tài nguyên và đồng thời đảm bảo độ chính xác và thời gian phát hiện nhanh nhất.

Nội dung đề án tốt nghiệp “***Nghiên cứu và triển khai thuật toán phát hiện bất thường cho hệ thống IoT tại biên***” được trình bày theo 3 chương như sau:

* **Chương 1: Tổng quan về IoT, điện toán biên và kỹ thuật phát hiện bất thường:** Khái quát lịch sử phát triển và những thành tựu mà IoT, tổng quan về hệ thống phát hiện bất thường cho IoT tại biên
* **Chương 2: Nghiên cứu một số thuật toán phát hiện bất thường cho IoT:** Trình bày một số thuật toán phát hiện bất thường cho Iot, trình bày một số cảm biến và cách thức thu thập dữ liệu trên các cảm biến này cho thực nghiệm ở chương 3.
* **Chương 3: Áp dụng thuật toán phát hiện bất thường cho hệ thống nhúng tại biên**: Đề xuất một mô hình TinyML để phát hiện bất thường và triển khai nó trên hệ thống nhúng IoT, sau đó thực hiện thử nghiệm để đánh giá độ chính xác và hiệu suất của mô hình.

# CHƯƠNG 1 – TỔNG QUAN VỀ IoT, ĐIỆN TOÁN BIÊN VÀ KỸ THUẬT PHÁT HIỆN BẤT THƯỜNG

## Tổng quan về IoT (Internet of Things)

### 1.1.1 Lịch sử phát triển, thành tựu của công nghệ IoT

Internet of Things (IoT – Kết nối internet vạn vật) là một thuật ngữ chỉ một mạng lưới các thiết bị vật lý được trang bị cảm biến, chip và phần mềm, cho phép chúng thu thập và trao đổi dữ liệu thông qua mạng internet mà không cần sự can thiệp trực tiếp của con người.

Khái niệm về các thiết bị kết nối đã có từ năm 1832 khi điện báo điện từ đầu tiên được thiết kế. Điện báo cho phép liên lạc trực tiếp giữa hai máy thông qua việc truyền tín hiệu điện. Tuy nhiên, lịch sử thực sự của IoT bắt đầu từ việc phát minh ra Internet một thành phần rất thiết yếu vào cuối những năm 1960, sau đó phát triển nhanh chóng trong những thập kỷ tiếp theo. Các giai đoạn phát triển IoT chính gồm:

* Năm 1980: Một máy bán Coca-Cola đầu tiên được kết nối tại Đại học Carnegie Melon, đánh dấu bước đầu tiên trong việc sử dụng internet sơ khai để kiểm tra trạng thái và sẵn có của đồ uống.
* Giai đoạn từ 1990 - 1999: Năm 1990, John Romkey lần đầu tiên kết nối máy nướng bánh mì với Internet bằng giao thức TCP/IP. Một năm sau, các nhà khoa học của Đại học Cambridge nảy ra ý tưởng sử dụng nguyên mẫu máy ảnh web đầu tiên để theo dõi lượng cà phê có sẵn trong bình cà phê của phòng thí nghiệm của họ. Năm 1999 Kevin Ashton đặt ra thuật ngữ “Internet of Things”. Là một nhà công nghệ có tầm nhìn, Ashton đã thuyết trình cho Procter & Gamble, nơi ông mô tả IoT là công nghệ kết nối nhiều thiết bị với sự trợ giúp của thẻ RFID để quản lý chuỗi cung ứng.
* Năm 2000: Vào đầu thế kỷ 21, thuật ngữ "Internet of Things" trở nên phổ biến trên các phương tiện truyền thông như The Guardian, Forbes và Boston Globe. Sự tăng cường quan tâm đến công nghệ IoT đã dẫn đến Hội nghị quốc tế đầu tiên về Internet vạn vật tại Thụy Sĩ vào năm 2008, nơi các chuyên gia từ 23 quốc gia thảo luận về RFID, truyền thông không dây và mạng cảm biến. Các tiến bộ như tủ lạnh kết nối internet của LG Electronics vào năm 2000 và robot Nabaztag vào năm 2005 cũng đã đóng vai trò quan trọng trong việc thúc đẩy sự phát triển của IoT.
* Giai đoạn từ 2008 - 2009: Theo Cisco IBSG\*, IoT được hình thành từ năm 2008 đến năm 2009, khi số lượng máy móc được kết nối vượt quá số lượng con người trên hành tinh. Hiện tại, có khoảng 21,5 tỷ thiết bị được kết nối trên thế giới – gần gấp ba lần số người trên hành tinh.
* Giai đoạn từ 2013 - 2014: Trong năm 2014 khái niệm “smart city” ra đời. Smart Docklands ở Dublin đã cung cấp nền tảng cho các nhà đổi mới thử nghiệm các giải pháp công nghệ tiên tiến nhằm giải quyết các thách thức của thành phố, chẳng hạn như thùng rác thông minh, cảm biến giám sát mực nước lũ và cảm biến giám sát âm thanh thành phố.
* Giải đoạn 2018 - Nay: Từ năm 2018 đến nay, Người ta cũng nhận thấy rằng khái niệm IoT đã chuyển từ thông tin, dịch vụ và người dùng sang tích hợp hệ thống lớn. Giai đoạn này liên quan đến việc sử dụng Trí tuệ nhân tạo để xử lý thông tin, đưa ra quyết định và tạo ra tác động đến cuộc sống của con người (tức là phân tích dữ liệu và Học máy).

Internet of Things là một xu hướng công nghệ quan trọng, ngày càng có nhiều biến thể của IoT như sự kết hợp của AI vào các thiết bị IoT (AIoT) để phân tích dữ liệu hoặc phát hiện các bất thường trên thiết bị và đưa ra cảnh báo sớm. Sự hội tụ của AI và IoT sẽ cho phép các trường hợp sử dụng mới trong các ngành như phân tích dữ liệu nâng cao, cải thiện bảo trì dự đoán và tự động hóa theo thời gian thực, cũng như sẽ có tác động lớn đến tính bền vững với sự trợ giúp của giám sát tiêu thụ năng lượng theo thời gian thực được tối ưu hóa. Với sự phát triển to lớn của các thiết bị IoT và dữ liệu được tạo ra bởi các thiết bị đó, AI trở thành một cơ chế quan trọng để quản lý và tăng thêm giá trị cho dữ liệu do IoT tạo ra. IoT càng phát triển mạnh thì sẽ thúc đẩy chất lượng cuộc sống của con người lên cao và từ đó đạt được nhiều thành tựu bao gồm:

* Quản lý năng lượng hiệu quả hơn: IoT và AIoT đã giúp tối ưu hóa việc sử dụng năng lượng thông qua việc tự động điều chỉnh hệ thống điện, chiếu sáng, và hệ thống điều hòa không khí.
* Giảm thiểu lãng phí tài nguyên: Công nghệ IoT và AIoT đã giúp giảm thiểu lãng phí tài nguyên như nước, điện và nguyên liệu, bằng cách cung cấp thông tin chính xác và tự động hóa quy trình sản xuất và vận hành.
* Giao thông thông minh: Hệ thống giao thông thông minh dựa trên IoT và AIoT giúp giảm kẹt xe, cải thiện an toàn giao thông và giảm thời gian di chuyển của người dân thông qua phân tích dữ liệu và tối ưu hóa luồng giao thông.
* Chăm sóc sức khỏe cá nhân: Các thiết bị y tế thông minh kết nối IoT và AIoT giúp người dùng theo dõi sức khỏe của họ, cung cấp dữ liệu phân tích và cảnh báo sớm về các vấn đề sức khỏe tiềm ẩn.
* Nâng cao an ninh và an toàn: Hệ thống an ninh thông minh dựa trên IoT và AIoT cung cấp giám sát liên tục và tự động phát hiện sự cố, giúp ngăn chặn tội phạm và các tình huống nguy hiểm khác.
* Nông nghiệp thông minh: Các thiết bị IoT đã giúp nâng cao năng suất và hiệu suất trong nông nghiệp thông qua việc theo dõi và quản lý thông tin về đất đai, thời tiết, và sức khỏe của cây trồng.
* Phá hiện bất thường trên các thiết bị công nghiệp: Các thiết bị phát hiện bất thường IoT nhỏ gọn được lắp trên các động cơ, máy móc công nghiệp hoặc tại các nhà máy đã giúp phần giảm thiểu các lỗi nghiêm bằng cách phát hiện bất thường và cảnh báo tức thời.

Dù IoT (Internet of Things) đã mang lại nhiều lợi ích cho xã hội và doanh nghiệp, nhưng vẫn còn một số vấn đề cần được giải quyết để tối ưu hóa sự phát triển và triển khai của công nghệ này:

* Bảo mật và riêng tư: Một trong những vấn đề lớn nhất của IoT là bảo mật và riêng tư dữ liệu. Việc kết nối nhiều thiết bị thông qua internet mở ra nhiều cơ hội cho các cuộc tấn công mạng và việc xâm nhập vào hệ thống, đặc biệt là khi thông tin cá nhân được thu thập và chia sẻ qua các thiết bị kết nối.
* Tiêu thụ năng lượng: Một số thiết bị IoT tiêu thụ năng lượng lớn, đặc biệt là những thiết bị hoạt động liên tục và cần kết nối internet. Điều này có thể dẫn đến tăng chi phí điện và gây ra tác động tiêu cực đến môi trường.
* Tiêu chuẩn và tương thích: Sự phát triển của IoT đã dẫn đến việc sử dụng nhiều giao thức và tiêu chuẩn khác nhau, tạo ra khó khăn trong việc tương thích và tích hợp giữa các thiết bị và hệ thống khác nhau.
* Quản lý dữ liệu lớn: Sự phát triển nhanh chóng của IoT đã tạo ra lượng dữ liệu lớn đòi hỏi cần có các giải pháp hiệu quả để lưu trữ, quản lý và phân tích dữ liệu một cách hiệu quả.
* Vấn đề về độ tin cậy và ổn định: Một số thiết bị IoT có thể gặp phải vấn đề về độ tin cậy và ổn định, gây ra sự cố hoặc ngừng hoạt động đột ngột, đặc biệt trong môi trường hoạt động khắc nghiệt.
* Khả năng mở rộng và tích hợp: Các hệ thống IoT thường cần có khả năng mở rộng và tích hợp với các hệ thống và thiết bị mới một cách dễ dàng, điều này đặc biệt quan trọng trong môi trường kinh doanh đang thay đổi nhanh chóng.

### 1.1.2 Các vấn đề trong IoT

Trong hơn hai thập kỷ, Internet of Things (IoT) đã làm thay đổi cách hoạt động của nhiều ngành công nghiệp bằng việc cung cấp khả năng giám sát, phân tích và điều khiển thiết bị từ xa cho cả doanh nghiệp và người tiêu dùng. Sự phát triển không ngừng của các thiết bị IoT đã đưa vào sử dụng hàng tỷ thiết bị được kết nối trên khắp thế giới. Tuy nhiên, mặc dù IoT mở ra nhiều cơ hội mới, nhưng cũng mang lại những thách thức không nhỏ cho các nhà sản xuất, nhà phát triển và người tiêu dùng. Những thách thức và vấn đề quan trọng nhất mà IoT đang đối diện bao gồm:

* Vấn đề về an ninh, bảo mật vẫn là một thách thức hàng đầu đối với IoT. Khi các thiết bị IoT tiếp tục tăng lên, chúng trở nên ngày càng là mục tiêu hấp dẫn đối với các cuộc tấn công mạng. Đảm bảo an ninh cho cả thiết bị và dữ liệu là điều cực kỳ quan trọng để ngăn chặn các vụ vi phạm. Một số vấn đề cụ thể về an ninh và thách thức trong IoT bao gồm:
* Lấy cắp dữ liệu: Tin tặc nhắm đến dữ liệu nhạy cảm từ IoT, đe dọa quyền riêng tư và kiếm soát các thiết bị của người dùng.
* Các lỗ hổng bảo mật: Nhiều thiết bị IoT thiếu các cơ chế phát hiện và vá các lỗ hổng bảo mật một cách đúng đắn, khiến chúng trở nên dễ bị khai thác.
* Lo ngại về quyền riêng tư: IoT thu thập một lượng lớn dữ liệu cá nhân, gây ra lo ngại về quyền riêng tư và vấn đề tuân thủ pháp luật.
* Bảo mật mạng: Bảo vệ các mạng IoT đòi hỏi mã hóa, bảo vệ cổng và kiểm soát truy cập.
* Vấn đề về quản lý dữ liệu đóng vai trò quan trọng đối với Internet of Things (IoT). Các hệ thống IoT tạo ra lượng dữ liệu lớn, đặt ra những thách thức đáng kể trong việc xử lý, lưu trữ và phân tích thông tin này trong thời gian thực. Để đảm bảo hoạt động liền mạch của các hệ sinh thái IoT, các giải pháp quản lý dữ liệu hiệu quả là cần thiết. Điều này bao gồm việc triển khai các chiến lược quản lý dữ liệu mạnh mẽ, từ việc thu thập, lưu trữ, truyền tải đến phân tích dữ liệu để giải quyết những thách thức này. Các nhà phát triển cần đầu tư vào các giải pháp lưu trữ dữ liệu có khả năng mở rộng và hiệu quả, sử dụng các công cụ phân tích dữ liệu tiên tiến và thiết lập các giao thức truyền dữ liệu an toàn. Khả năng xử lý dữ liệu thời gian thực là cần thiết để có thể rút ra thông tin hành động một cách kịp thời.
* Vấn đề về hiệu suất năng lượng là một vấn đề cốt lõi không thể thiếu. Nhiều thiết bị IoT phụ thuộc vào nguồn năng lượng pin, vì vậy việc cải thiện hiệu suất năng lượng để kéo dài tuổi thọ pin và giảm thiểu tác động môi trường trở nên cực kỳ quan trọng. Thách thức này bao gồm:
  + Thiết kế phần cứng tối ưu: Để đảm bảo hiệu quả về năng lượng, các thiết bị IoT cần được thiết kế với các phần cứng tiết kiệm năng lượng cùng các hệ thống quản lý năng lượng hiệu quả, nhằm tối ưu hóa sự tiêu thụ năng lượng và kéo dài thời gian sử dụng pin.
  + Hiệu quả truyền dữ liệu: Các giao thức truyền dữ liệu hiệu quả, như MQTT, đóng vai trò quan trọng trong việc giảm tiêu thụ năng lượng trong quá trình trao đổi dữ liệu.
  + Edge Computing: Xử lý dữ liệu tại điểm xử lý trước khi truyền tới đám mây (Cloud) có thể giảm đáng kể năng lượng cần thiết cho quá trình truyền dữ liệu.
  + Các phương pháp tự nạp năng lượng: Sử dụng các công nghệ thu năng lượng như tấm pin mặt trời hoặc máy phát năng lượng từ chuyển động có thể hỗ trợ việc nạp lại pin của các thiết bị IoT một cách bền vững.

## Giới thiệu chung về mô hình điện toán biên trong IoT

Điện toán biên (Edge computing), hay còn gọi là tính toán biên, là là một phần mở rộng của điện toán đám mây (Cloud Computing) mang tính toán và lưu trữ dữ liệu đến gần hơn với nguồn tạo dữ liệu tại chỗ. Trong hệ thống IoT, điện toán biên là việc xử lý và phân tích dữ liệu từ các thiết bị IoT tại chỗ hoặc ở gần chỗ tạo ra dữ liệu hơn. Thay vì truyền dữ liệu đến máy chủ trung tâm hoặc đám mây để xử lý, dữ liệu được xử lý cục bộ tại chỗ hoặc ở rìa mạng. Điều này giúp giảm độ trễ, tăng hiệu suất và nâng cao bảo mật. Điện toán biên kết hợp với IoT là một cuộc cách mạng lớn. Các thiết bị IoT tạo ra một lượng lớn dữ liệu, và việc truyền toàn bộ dữ liệu này lên đám mây có thể gây trễ và tăng chi phí. Edge computing cho phép xử lý dữ liệu tại chỗ, giảm độ trễ và tăng hiệu suất. Ví dụ, trong một tòa nhà văn phòng, một cảm biến IoT có thể phát hiện nhiệt độ trong phòng vượt quá ngưỡng. Thay vì gửi dữ liệu lên đám mây để xử lý, dữ liệu có thể được xử lý tại chỗ. Điều này giúp hệ thống phản hồi nhanh hơn, chẳng hạn bằng cách kích hoạt hệ thống điều hòa không khí hoặc mở cửa sổ mà không phải chờ đợi đám mây.

Trong hệ thống điện toán biên, dữ liệu được xử lý và lưu trữ gần nguồn thu thập, tạo môi trường tính toán phân tán. Điện toán biên kết hợp tính linh hoạt của điện toán đám mây với tính cục bộ của các thiết bị IoT, đóng vai trò quan trọng trong cung cấp xử lý và lưu trữ dữ liệu. Nó giúp giảm độ trễ trong truyền dữ liệu, tăng cường bảo mật và hiệu suất hệ thống, đồng thời giảm tải và chi phí vận hành cho các trung tâm dữ liệu. Trong mô hình này, các thiết bị cuối của IoT như cảm biến và máy tính nhúng đóng vai trò chính trong xử lý dữ liệu và thực hiện các tác vụ tính toán, tăng cường tính tự lập của hệ thống IoT..

A screenshot of a computer

Description automatically generated

Hình 1. 1: Mô hình điện toán biên

Trong mô hình điện toán biên của IoT (hình 1.1), đó là một cấu trúc phân tán của hệ thống IoT, nơi sự tính toán được thực hiện trực tiếp trên các thiết bị đầu cuối như cảm biến, bộ điều khiển và máy tính nhúng, được gọi là biên thay vì được chuyển đến các trung tâm dữ liệu tập trung hoặc đám mây. Mô hình này nhằm mục đích giảm thiểu độ trễ mạng và tăng cường tốc độ phản hồi của hệ thống, giúp cho các ứng dụng IoT được xử lý một cách hiệu quả hơn. Có ba cách chính để triển khai mô hình tính toán biên trên hệ thống IoT như sau:

* + Điện toán biên tại thiết bị IoT (On premise): Xử lý dữ liệu trực tiếp trên các thiết bị IoT mà không cần truyền dữ liệu lên đám mây là phương pháp tối ưu nhất của điện toán biên. Tại đây, các thiết bị và giải pháp IoT có thể nhanh chóng phân tích dữ liệu mà không cần phải chờ đợi kết quả từ đám mây hoặc tiêu tốn băng thông internet. Sau khi hoàn tất quá trình xử lý dữ liệu, chỉ kết quả hoặc dữ liệu đầu ra mới được gửi lên đám mây. Phương pháp này giúp tiết kiệm băng thông, giảm chi phí sử dụng tài nguyên đám mây và rút ngắn thời gian truyền dữ liệu. Ví dụ, hệ thống hỗ trợ lái xe trong ô tô có thể tránh việc gửi video lớn lên đám mây bằng cách xử lý dữ liệu trực tiếp trên thiết bị cục bộ và chỉ chuyển những dữ liệu cần thiết lên đám mây.
  + Điện toán biên kết hợp (Hybrid edge computing): Trong phương pháp này, thiết bị Điện toán biên phát hiện và chọn lọc các phần quan trọng của dữ liệu thu được trước khi gửi chúng lên đám mây để tiếp tục xử lý. Việc này không chỉ giới hạn băng thông và thời gian cần thiết đến một mức nhất định mà còn đảm bảo sự tối ưu hóa hiệu suất và tài nguyên của điện toán đám mây. Ví dụ, một hệ thống hỗ trợ lái xe có thể ghi và xử lý một lượng lớn dữ liệu video với hiệu suất giới hạn. Tại đây, các phần dữ liệu tiềm năng quan trọng được giữ lại để điện toán đám mây có thể tiến hành phân tích sâu hơn. Phân chia vai trò của quá trình xử lý này thành hai phần tận dụng tối ưu từ cả hai lựa chọn.
  + Điện toán biên trên mây (On-cloud computing): Đây là một mô hình tính toán không tích hợp điện toán biên vào các ứng dụng IoT. Tất cả dữ liệu thu được được gửi trực tiếp đến đám mây để lưu trữ, xử lý và ghi lại kết quả phân tích. Với việc hoàn toàn phụ thuộc vào đám mây, điều này có thể gây ra chi phí cao, tiêu tốn lượng băng thông lớn, mất thời gian khi tải lên, và tăng nhu cầu về không gian lưu trữ.

Cách thức hoạt động của điện toán biên như sau:

* + Tạo dữ liệu: Điện toán biên bắt đầu với việc tạo ra dữ liệu tại các thiết bị biên như cảm biến IoT, xe tự hành, hoặc các thiết bị di động. Những thiết bị này thu thập một lượng lớn dữ liệu trong thời gian thực.
  + Xử lý cục bộ: Thay vì gửi toàn bộ dữ liệu đến một trung tâm dữ liệu từ xa, điện toán biên thực hiện việc xử lý và phân tích ban đầu tại biên. Điều này được thực hiện bằng cách sử dụng các thiết bị biên hoặc máy chủ biên được đặt gần nguồn dữ liệu. Việc xử lý cục bộ giúp giảm độ trễ và cho phép ra quyết định trong thời gian thực.
  + Lọc và phân tích dữ liệu: Tại biên thiết bị, dữ liệu được lọc và phân tích dựa trên các quy tắc hoặc thuật toán được xác định trước. Quá trình lọc này giúp giảm lượng dữ liệu cần được truyền đến đám mây hoặc trung tâm dữ liệu, tối ưu hóa việc sử dụng băng thông.
  + Truyền dữ liệu có ích: Chỉ có dữ liệu liên quan và có giá trị được truyền đến đám mây hoặc trung tâm dữ liệu trung tâm để phân tích và lưu trữ lâu dài. Điều này giúp giảm thiểu lượng dữ liệu gửi qua mạng và yêu cầu băng thông.
  + Tích hợp với đám mây: Dữ liệu đã được xử lý và lọc có thể được tích hợp với các dịch vụ hoặc ứng dụng dựa trên đám mây cho phân tích nâng cao, học máy, hoặc lưu trữ. Việc tích hợp này cho phép có cái nhìn sâu sắc hơn, tương quan dữ liệu, và lưu trữ dữ liệu lâu dài.
  + Ra quyết định thời gian thực: Điện toán biên cho phép ra quyết định trong thời gian thực bằng cách xử lý dữ liệu cục bộ và giảm thời gian trễ tới một máy chủ từ xa. Điều này rất quan trọng đối với các ứng dụng yêu cầu phản hồi tức thì, như xe tự hành, tự động hóa công nghiệp, hoặc hệ thống giám sát quan trọng.

Các thành phần cơ bản của một mô hình điện toán biên bao gồm:

* + Thiết bị biên (Edge Devices): Là các phần cứng thông minh được đặt gần nguồn tạo ra dữ liệu trong hệ thống IoT. Chúng bao gồm cảm biến, máy tính nhúng, camera và thiết bị di động, hoạt động như bộ não của hệ thống, thu thập dữ liệu đa dạng từ môi trường xung quanh. Ví dụ, các cảm biến đo lường nhiệt độ, độ ẩm, ánh sáng và áp suất, trong khi camera ghi lại hình ảnh hoặc video. Thiết bị IoT cung cấp thông tin về trạng thái và điều khiển các thiết bị khác. Điều này giúp hệ thống thu thập dữ liệu từ nhiều nguồn và tạo ra nguồn thông tin phong phú.
  + Máy Chủ Biên (Edge Servers): Là trung tâm xử lý dữ liệu cục bộ. Chúng thực hiện xử lý và phân tích dữ liệu từ các thiết bị biên để hỗ trợ quyết định thời gian thực và giảm độ trễ. Máy chủ biên xử lý lượng lớn dữ liệu từ thiết bị biên và thực hiện các thuật toán phức tạp để phân tích dữ liệu và trích xuất thông tin quan trọng, từ đó tối ưu hóa việc thu thập dữ liệu và cải thiện hiệu suất của hệ thống IoT.
  + Nút Biên (Edge Node): Là thuật ngữ phổ biến trong IoT, chỉ các thiết bị và cơ sở hạ tầng có khả năng tính toán và xử lý dữ liệu tại vị trí gần nguồn tạo ra dữ liệu. Điều này tối ưu hóa hiệu suất và đáp ứng nhanh chóng cho các ứng dụng IoT. Nút biên giúp tối ưu hóa quá trình thu thập và xử lý dữ liệu bằng cách phân phối tính toán gần nguồn tạo ra dữ liệu, giảm độ trễ, tải mạng và yêu cầu băng thông. Nó cung cấp khả năng phản ứng nhanh chóng và quyết định thời gian thực, tạo ra một hệ thống linh hoạt, hiệu quả và có thể mở rộng được.
  + Máy chủ (Cloud Server): Trong hệ thống này, máy chủ có thể là một đám mây công cộng hoặc riêng tư, hoặc thậm chí là một trung tâm dữ liệu. Chúng đóng vai trò quan trọng trong việc lưu trữ và chạy các ứng dụng dành cho việc điều phối và quản lý các nút biên khác nhau. Các đám mây này không chỉ cung cấp không gian lưu trữ mà còn chạy các ứng dụng quản lý để hỗ trợ các hoạt động của các nút biên. Trong quá trình xử lý dữ liệu, các công việc ở biên, ở thiết bị cuối và trên đám mây sẽ tương tác với nhau, tạo ra một môi trường tích hợp và đồng bộ.
  + Cổng biên (Edge gateway): Đóng vai trò quan trọng trong hệ thống Edge Computing, giúp kết nối các thiết bị IoT với mạng và hệ thống điện toán biên. Với sự phát triển nhanh chóng của Internet of Things (IoT), việc thu thập và xử lý dữ liệu ngay tại nguồn trở nên ngày càng quan trọng. Cổng biên là cầu nối giữa các thiết bị IoT và hệ thống điện toán biên, chịu trách nhiệm thu thập dữ liệu từ các thiết bị, xử lý nó và truyền tải đến hệ thống điện toán biên hoặc các ứng dụng đám mây.

## Tổng quan về hệ thống phát hiện bất thường

Phát hiện bất thường (Anomaly Detection), còn được biết đến là phát hiện điểm ngoại lai hoặc phát hiện dữ liệu không bình thường, là quá trình phân tích dữ liệu nhằm xác định các điểm dữ liệu có các đặc điểm đáng kể khác biệt so với phần lớn dữ liệu còn lại hoặc không tuân theo các quy luật phân phối của dữ liệu bình thường. Các điểm dữ liệu này, thường được gọi là điểm ngoại lai, nhiễu, sai số hoặc ngoại lệ, có thể biểu thị các sự kiện hoặc trạng thái không phù hợp, không thông thường trong hệ thống hoặc quá trình mà dữ liệu đang được thu thập từ. Mục tiêu của phát hiện bất thường là nhận diện và hiểu rõ các điểm này để có thể xử lý kịp thời và đưa ra các biện pháp phòng tránh hoặc cải thiện hệ thống. Khi tiến hành phân tích dữ liệu, việc xử lý các dữ liệu bất thường thường gây ra thách thức trong việc đưa ra nhận định và dự đoán. Trong nhiều trường hợp, dấu hiệu của sự bất thường có thể chỉ ra vấn đề nghiêm trọng đang diễn ra, gây tổn thất lớn và đòi hỏi nhiều thời gian và công sức để khắc phục. Ví dụ, trong môi trường công nghiệp, các máy móc như động cơ điện thường hoạt động liên tục và có nguy cơ gặp sự cố, gây ra thiệt hại lớn cho công ty. Phát hiện dấu hiệu bất thường sớm có thể giúp giảm thiểu chi phí và ngăn chặn các sự cố nghiêm trọng. Một ví dụ khác là khi ngân hàng phát hiện sự bất thường trong giao dịch thẻ tín dụng, họ có thể từ chối giao dịch để bảo vệ quyền lợi của khách hàng và duy trì uy tín.

Phát hiện các biến đổi không bình thường đóng vai trò quan trọng vì thường chứa thông tin có giá trị và mang ý nghĩa trong nhiều ngữ cảnh thực tế. Trong thực tế, các biến đổi không bình thường hiện diện dưới nhiều hình thái khác nhau, do đó, phụ thuộc vào ứng dụng cụ thể, chúng ta cần sử dụng các phương pháp phát hiện khác nhau. Có ba dạng chính của phát hiện bất thường như sau:

A diagram of a number of dots

Description automatically generated

Hình 1. 2: Ví dụ bất thường điểm trong một tập dữ liệu trong không gian hai chiều.

* Bất thường điểm (point anomalies): Bất thường điểm là một khái niệm quan trọng trong phát hiện bất thường, chỉ các điểm hoặc sự kiện trong dữ liệu mà phân bố hoặc đặc điểm của chúng khác biệt đáng kể so với các điểm khác. Các bất thường điểm thường là sự kiện hiếm hoặc không tuân theo mô hình phân phối thông thường và có thể xuất hiện trong nhiều lĩnh vực như kiểm soát chất lượng sản phẩm hoặc phát hiện gian lận tín dụng. Để phát hiện bất thường điểm, thường sử dụng các phương pháp thống kê và học máy, so sánh các đặc điểm của mỗi điểm dữ liệu với một ngưỡng hoặc mô hình dự đoán để xác định liệu điểm đó có phải là bất thường hay không.

Trong Hình 1.2 hiển thị một ví dụ về bất thường điểm trong một tập dữ liệu hai chiều. Trong hình này, các điểm O1 và O2 được biểu diễn là nằm ngoài vùng bình thường được đại diện bởi các tập dữ liệu C1 và C2. Điều này chỉ ra rằng O1 và O2 là những điểm bất thường, vì chúng không thuộc về phân phối chung của các tập dữ liệu bình thường. Điều này có thể được hiểu là các điểm này có đặc điểm hoặc hành vi không phù hợp hoặc không phản ánh các mẫu dữ liệu thông thường. Một ví dụ về bất thường điểm có thể là trong lĩnh vực của giám sát hoạt động mạng. Giả sử chúng ta đang theo dõi lưu lượng dữ liệu trên một mạng máy tính, và thông thường lưu lượng này có thể thay đổi theo thời gian, nhưng vẫn duy trì trong một phạm vi nhất định. Tuy nhiên, một ngày nào đó, lưu lượng dữ liệu đột ngột tăng cao đột ngột lên mức rất cao so với các mức trung bình trước đó. Trong trường hợp này, điểm dữ liệu đại diện cho lưu lượng dữ liệu trong khoảng thời gian đó sẽ là một bất thường điểm.

* Bất thường ngữ cảnh (contextual anomalies): là một trường hợp có thể được coi là bất thường trong một số bối cảnh cụ thể. Điều này có nghĩa là việc quan sát cùng một điểm thông qua các bối cảnh khác nhau sẽ không phải lúc nào cũng cho chúng ta dấu hiệu của hành vi bất thường. Sự bất thường theo ngữ cảnh được xác định bằng cách kết hợp các đặc điểm theo ngữ cảnh và hành vi. Thông thường, các yếu tố thời gian và không gian được sử dụng phổ biến nhất trong việc xác định bất thường theo ngữ cảnh, trong khi các yếu tố hành vi thường liên quan đến các yếu tố đặc trưng của miền dữ liệu đang được nghiên cứu, như số tiền chi tiêu hoặc nhiệt độ trung bình, hoặc các yếu tố định lượng khác. Vì vậy mỗi trường hợp dữ liệu được xác định bằng cách sử dụng bởi các thuộc tính sau:
  + Thuộc tính ngữ cảnh: Thuộc tính ngữ cảnh như thời gian và không gian đóng vai trò quan trọng trong việc xác định bất thường ngữ cảnh. Thời gian có thể chỉ ra các sự kiện xảy ra vào thời điểm không phù hợp hoặc không thường xuyên, trong khi không gian có thể liên quan đến vị trí hoặc môi trường không thường xuyên. Đặc điểm địa lý như địa điểm hoặc khu vực cũng có thể cung cấp thông tin ngữ cảnh quan trọng cho việc phát hiện bất thường.
  + Thuộc tính hành vi: đóng vai trò trong việc phát hiện các biến đổi không bình thường trong mô hình dữ liệu. Sự biến đổi trong hành vi hoặc xu hướng cũng có thể chỉ ra sự xuất hiện của các sự kiện không thường xuyên hoặc không phù hợp.

A graph of a function

Description automatically generated

Hình 1. 3: Ví dụ về bất thường bối cảnh

Trong biểu đồ nhiệt độ ba năm liên tiếp như được mô tả trong Hình 1.3, thời điểm t2 được xem là một điểm bất thường trong ngữ cảnh cụ thể của chuỗi dữ liệu. Điều này là do vào thời gian này trong chuỗi thời gian, nhiệt độ không tuân theo xu hướng chung của các thời điểm trước đó. Ví dụ, tại thời điểm t1, một sự kiện tương tự đã xảy ra vào tháng 12 của hai năm trước đó, tuy nhiên, việc xảy ra vào tháng 6 không phải là một kịch bản phổ biến. Điều này tạo ra sự khác biệt đáng kể so với các dữ liệu trước đó, và do đó, thời điểm t2 được xem xét là một điểm bất thường trong bối cảnh của chuỗi dữ liệu nhiệt độ theo thời gian.

* Bất thường tập hợp (collective anomalies): là một loại bất thường xuất hiện khi một nhóm các điểm dữ liệu cùng một lúc thể hiện một hành vi không phù hợp hoặc không thường xuyên. Khác với bất thường điểm và bất thường ngữ cảnh, bất thường tập hợp không chỉ xảy ra ở một điểm hoặc một cá thể, mà là một tập hợp các sự kiện hoặc mẫu dữ liệu được gom nhóm lại. Ví dụ, giả sử một hệ thống giám sát giao thông ghi lại lưu lượng xe cộ qua một điểm giao cắt trong một thành phố. Trong ngày thường, lưu lượng xe qua điểm giao cắt này dao động trong khoảng từ 500 đến 800 xe mỗi giờ. Tuy nhiên, vào một ngày đặc biệt, như ngày lễ hay sự kiện đặc biệt, lưu lượng xe cộ tăng lên đáng kể, đạt mức khoảng 2000 xe mỗi giờ. Trong trường hợp này, sự tăng đột ngột của lưu lượng xe không chỉ đơn thuần là một điểm bất thường mà là một bất thường tập hợp, vì nó ảnh hưởng đến một nhóm các sự kiện giao thông liên quan trong cùng một thời điểm và khu vực.

A graph showing a red line

Description automatically generated

Hình 1. 4: Ví dụ về bất thường tập hợp.

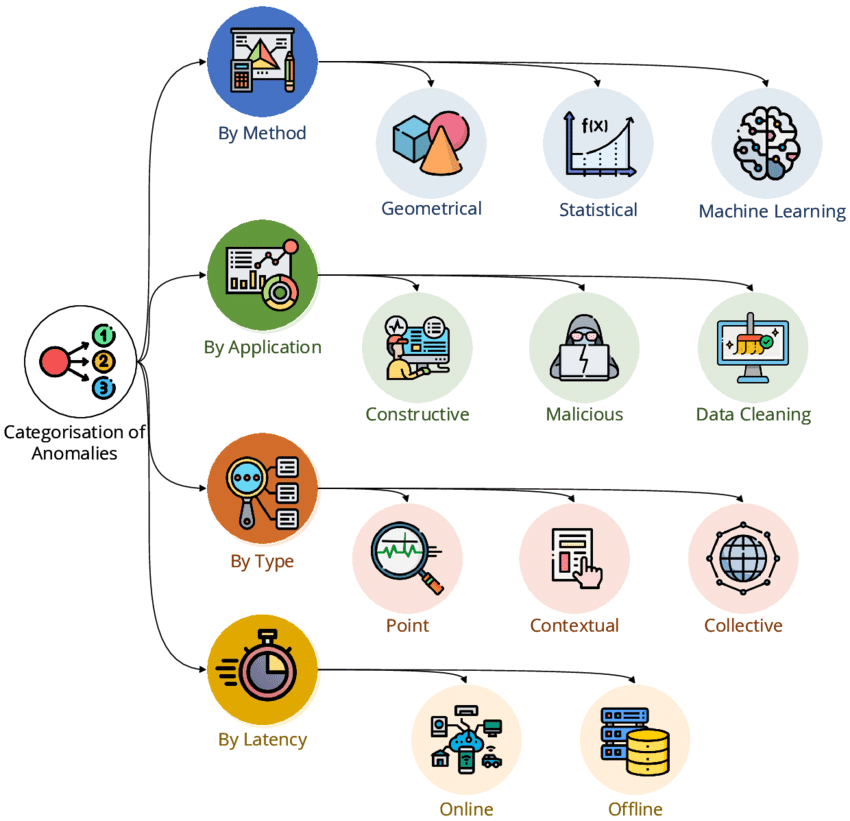
Trong Hình 1.4, chúng ta quan sát một đường biểu đồ màu đỏ bắt đầu từ tháng 5 và kéo dài đến tháng 8. Giá trị của đường này đột ngột tăng lên so với các tháng khác, và do đó được coi là bất thường. Sự gia tăng đột biến này kéo dài trong một khoảng thời gian dài, từ tháng 5 đến tháng 8, chứ không chỉ là một sự kiện cụ thể. Điều này làm cho hiện tượng này trở thành một loại bất thường tập hợp, vì nó ảnh hưởng đến một chuỗi các giá trị trong một khoảng thời gian liên tục.

## Tổng quan về hệ thống phát hiện bất thường cho IoT tại biên

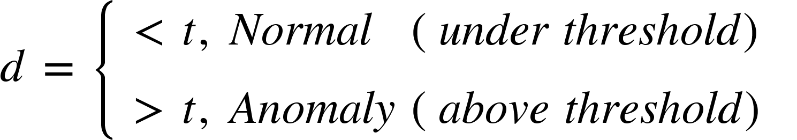
### 1.4.1 Phân loại phát hiện sự bất thường trong IoT

Trong Hình 1.5 cho thấy một cái nhìn tổng quan về các dạng phát hiện bất thường. Các phương pháp phát hiện sự bất thường trong IoT được phân loại thành bốn dạng: Theo Phương pháp (By Method), Theo Ứng dụng (By Application), Theo Loại Sự bất thường (By Anomaly Type), Theo Độ trễ (By Latency). Cụ thể các dạng phát hiện bất thường như sau:

* Theo Phương pháp (By Method): Phân loại sự bất thường theo phương pháp bao gồm hình học, thống kê và học máy. Các phương pháp hình học hoạt động dựa trên giả định rằng dựa trên khoảng cách và mật độ biểu diễn một tập dữ liệu, dữ liệu dự kiến và dữ liệu bất thường sẽ được phân tách. Trong tập hợp các điểm dữ liệu, ý tưởng của các kỹ thuật cô lập hoặc dựa trên mật độ là sự bất thường, nhưng bất thường này xuất hiện trong các khu vực ít dày đặc. Các phương pháp này sử dụng một ngưỡng tĩnh hoặc động 't' trên khoảng cách ước lượng 'd' để phân loại các sự bất thường, cụ thể như sau:



Hình 1. 5: Tổng quan về phân loại bất thường



Các phương pháp thống kê như phương pháp thể tích tối thiểu và “exponential smoothing” cố gắng xây dựng các mô hình dữ liệu bình thường và dự đoán dữ liệu trong tương lai. Dữ liệu bất thường là những điểm không tuân thủ mô hình. Mặc dù các phương pháp này được sử dụng rộng rãi, các giải pháp học máy và học sâu trở nên cần thiết để xử lý các tình huống phức tạp hơn. Việc lựa chọn mô hình phụ thuộc vào loại dữ liệu, với LSTM và transformer cho dữ liệu tuần tự, và CNN cho dữ liệu không tuần tự. Thuật toán cố gắng phân biệt giữa hành vi bình thường và bất thường bằng cách sử dụng SVM hoặc LSTM trong dữ liệu truyền theo luồng.

* Theo Ứng dụng (By Application): Phân loại sự bất thường theo ứng dụng là xây dựng (constructive), phá hủy (destructive) và làm sạch dữ liệu (data cleaning).
  + Xây dựng (constructive): hệ thống sẽ sử dụng các mô hình học máy và học sâu để xây dựng các mô hình dự đoán và nhận biết hành vi bất thường từ dữ liệu đầu vào. Ví dụ, trong việc giám sát hành vi hàng ngày của người cao tuổi, một mô hình có thể được huấn luyện để nhận diện các hoạt động không bình thường như rơi hoặc mất cân đối từ dữ liệu cảm biến.
  + Phá hủy (destructive): Các biện pháp bảo mật sẽ phát hiện và ngăn chặn các cuộc tấn công từ xa hoặc từ chối dịch vụ. Hệ thống sẽ sử dụng các thuật toán phát hiện xâm nhập để phát hiện các hành vi đáng ngờ trong dữ liệu mạng và ngăn chặn chúng trước khi gây ra thiệt hại.
  + Làm sạch dữ liệu (data cleaning): Các mô hình học máy và học sâu sẽ được sử dụng để loại bỏ các dữ liệu nhiễu và không mong muốn từ tập dữ liệu đầu vào. Các kỹ thuật như mạng nơ-ron tích chập sâu có thể được áp dụng để lọc ra các điểm dữ liệu không bình thường và cải thiện chất lượng dữ liệu.
* Theo Loại Sự bất thường (By Anomaly Type): Có ba loại bất thường phổ biến là điểm, ngữ cảnh và tập hợp. Một điểm bất thường là khi một điểm dữ liệu lệch khỏi hành vi dự kiến, như việc phát hiện gian lận thẻ tín dụng. Bất thường ngữ cảnh xảy ra khi một trường hợp có thể được xem là bất thường trong một số ngữ cảnh cụ thể. Một sự bất thường ngữ cảnh được phát hiện khi cả hai đặc điểm ngữ cảnh và hành vi được xem xét cùng nhau. Ví dụ, trong việc vi phạm luật giao thông, các sự bất thường thay đổi tùy thuộc vào thông tin vị trí địa lý. Bất thường tập hợp nhìn vào toàn bộ tập dữ liệu. Một ví dụ là việc sử dụng điện tâm đồ để giám sát và phát hiện các sự bất thường hoặc vấn đề trong tim người.
* Theo Độ trễ (By Latency): Khả năng mở rộng và độ trễ của một thuật toán phát hiện quyết định xem phương pháp được thực hiện trực tuyến trong giai đoạn thu thập dữ liệu hay sau đó trong giai đoạn lưu trữ. Thuật toán trực tuyến có thể xử lý tuần tự thông tin, từ một điểm dữ liệu đơn lẻ đến một cửa sổ, mà không cần truy cập vào toàn bộ dữ liệu đầu vào. Các phương pháp trực tuyến truyền thống bao gồm các kỹ thuật dựa trên khoảng cách, mật độ, độ lệch và góc. Ngược lại, các thuật toán ngoại tuyến có truy cập vào dữ liệu hoàn chỉnh và thường sử dụng các thuật toán phức tạp và tốn nhiều tài nguyên tính toán để giải quyết vấn đề trong khoảng thời gian hợp lý.

### Các thách thức trong phát hiện sự bất thường cho IoT

Trong những năm gần đây, ứng dụng các kỹ thuật học máy (Machine Learning) phát hiện bất thường để bảo hệ, cảnh báo sớm cho các hệ thống IoT đã có những bước phát triển manh mẽ. Các mô hình học máy được huấn luyện trên tập dữ liệu bình thường và bất thường, sau đó được sử dụng để xây dựng một mô hình có khả năng phân loại và phát hiện các điểm bất thường trong dữ liệu. Tuy nhiên, việc phát triển các phương pháp phát hiện bất thường trên các hệ thống IoT đầy thách thức do một số yếu tố như:

* Dữ liệu không chuẩn: Dữ liệu từ các thiết bị IoT có thể không đồng nhất về cấu trúc và định dạng, điều này làm cho việc phát hiện bất thường trở nên phức tạp. Sự không thống nhất này có thể làm giảm hiệu suất của các thuật toán phát hiện bất thường.
* Sự thay đổi môi trường: Môi trường hoạt động của các thiết bị IoT thường thay đổi đột ngột, ví dụ như sự biến đổi thời tiết, nhiễu mạng, hoặc thậm chí là sự can thiệp từ con người. Những thay đổi này có thể tạo ra các bất thường không mong muốn trong dữ liệu.
* Sự thiếu hụt dữ liệu: Trong một số trường hợp, dữ liệu từ các thiết bị IoT có thể bị mất hoặc không đủ để đào tạo mô hình phát hiện bất thường. Sự thiếu hụt dữ liệu có thể làm giảm độ chính xác của quá trình phát hiện bất thường.
* Tính toán và tài nguyên hạn chế: Trong một số trường hợp, các thiết bị IoT có tính toán và tài nguyên hạn chế, làm cho việc triển khai các thuật toán phức tạp để phát hiện bất thường trở nên khó khăn. Đòi hỏi sự tinh giản và tối ưu trong các giải pháp phát hiện bất thường.
* Tính đa dạng của dữ liệu IoT: Dữ liệu từ các loại thiết bị IoT khác nhau có thể đa dạng, bao gồm dữ liệu từ cảm biến, camera, bộ cảm biến và hệ thống điều khiển. Việc kết hợp và phân tích dữ liệu đa dạng này đòi hỏi các kỹ thuật phát hiện bất thường phức tạp và linh hoạt.
* Thách thức về cập nhật và duy trì: Các thiết bị IoT thường được triển khai trong môi trường phức tạp và khó tiếp cận, làm cho việc cập nhật và duy trì các phần mềm và firmware trở nên khó khăn. Sự thiếu cập nhật có thể tạo ra các lỗ hổng bảo mật và làm giảm khả năng phát hiện bất thường.

## Kết luận chương 1

Trong Chương 1 của đề án đã giới thiệu về sự phát triển của IoT và điện toán biên, cùng với các ứng dụng và tiềm năng của chúng. Mô hình điện toán biên trong IoT đã được giới thiệu, mở ra một cánh cửa mới trong việc xử lý dữ liệu và tương tác với các thiết bị. Hơn nữa, ở chương này còn giới thiệu tổng quan về hệ thống phát hiện bất thường, đề cập đến khái niệm và phân loại các bất thường. Các hệ thống này đóng vai trò quan trọng trong việc phát hiện các sự kiện không bình thường trong môi trường IoT, từ đó tăng cường hiệu suất và an toàn. Cuối cùng, tổng quan về hệ thống phát hiện bất thường cho IoT tại biên, mở ra cơ hội hiểu sâu hơn về cách triển khai các giải pháp phát hiện bất thường trong IoT.

# CHƯƠNG 2 – NGHIÊN CỨU MỘT SỐ THUẬT TOÁN PHÁT HIỆN BẤT THƯỜNG CHO IOT

## Giới thiệu chung

Trong những năm gần đây, cách mạng công nghiệp 4.0 đã tiến triển đáng kể, dẫn đến sự gia tăng mạnh mẽ của việc áp dụng máy móc để thay thế công việc truyền thống. Sự triển khai các dây truyền sản xuất tự động đã mang lại nhiều lợi ích, bao gồm việc tăng năng suất và giảm chi phí nhân công, nhờ khả năng hoạt động liên tục trong thời gian dài. Tuy nhiên, nếu xảy ra sự cố gây hỏng máy, dây chuyền sản xuất có thể bị gián đoạn, ảnh hưởng đến năng suất và gây thiệt hại kinh tế cho các công ty. Các sự gián đoạn này bắt nguồn từ các hành vi không mong đợi và không đúng chuẩn có thể xảy ra trong môi trường sản xuất, được gọi là các sự bất thường. Trong các hệ thống máy móc công nghiệp, các sự bất thường có thể phát sinh từ sự hỏng hóc của thiết bị, điều kiện môi trường không ổn định và sự thay đổi trong điều kiện vận hành. Do đó, việc phát hiện các sự bất thường ngay từ ban đầu là rất quan trọng để ngăn chặn sự hỏng hóc của thiết bị, giảm thiểu thời gian ngừng hoạt động và giảm chi phí sửa chữa. Sự phát triển của các hệ thống IoT đã mở ra khả năng thu thập lượng dữ liệu lớn từ các máy móc công nghiệp, tạo nên nguồn dữ liệu đủ lớn để phát hiện các sự cố bất thường. Hệ thống IoT bao gồm cảm biến và hệ thống giám sát có khả năng thu thập dữ liệu về nhiệt độ, áp suất, rung động và tiêu thụ điện năng. Dữ liệu này cho phép các kỹ sư và chuyên gia giám sát theo dõi tình trạng của các máy móc công nghiệp trong thời gian thực và thực hiện các biện pháp sửa chữa kịp thời để chẩn đoán và giải quyết vấn đề cơ bản trước khi nó gây ra sự gián đoạn đáng kể đối với quá trình sản xuất. Mặc dù vậy, việc xử lý lượng dữ liệu lớn và phức tạp được tạo ra bởi các hệ thống IoT tạo ra thách thức trong việc phát hiện các sự bất thường một cách thủ công, việc này làm tăng thêm nhu cầu bảo dưỡng, sửa chữa và dự phòng.

Trong nhưng năm gần đây sự phát triển vượt bậc của học máy (ML) đã mở ra nhiều hướng nghiên cứu mới và ứng dụng được vào nhiều lĩnh vực khác nhau. Một trong nhưng ứng dụng quan trong đó là phát hiện bất thường trên các dữ liệu. Sử dụng học máy có thể giải quyết được thách thức của việc xử lý lượng dữ liệu lớn và phức tạp từ các hệ thống IoT. Cụ thể, các thuật toán học máy có khả năng tự động phân tích dữ liệu từ các cảm biến trên các thiết bị công nghiệp, giúp tự động hóa quá trình phát hiện sự bất thường. Lợi thế của việc ứng dụng học máy (ML) vào phát hiện bất thường trong các máy móc công nghiệp nằm ở việc có khả năng phân tích và xử lý lượng dữ liệu lớn, giúp tăng cường khẳ năng phát hiện các sự bất thường một cách chính xác và hiệu quả nhất. Ngoài ra, nếu kết hợp học máy (ML) vào các hệ thống IoT và thực hiện xử lý tại biên còn giúp xử lý dữ liệu tại chỗ trước khi truyền đến trung tâm. Điều này giúp giảm bớt gánh nặng cho hạ tầng mạng và giảm độ trễ trong việc đưa ra các quyết định quan trọng. Bằng cách phát hiện và xử lý các dữ liệu không bình thường ngay tại điểm thu thập, hệ thống có thể tự động áp dụng các biện pháp bảo vệ hệ thống máy công nghiệp hoặc thông báo cảnh báo mà không cần phải chờ đợi phản hồi từ trung tâm. Điều này đặc biệt hữu ích trong các ứng dụng đòi hỏi phản

A diagram of a machine learning model

Description automatically generated

Hình 2. 1: Tổng quan về IoT và ứng dụng ML vào phát hiện sự bất thường trong công nghiệp

ứng nhanh như trong các môi trường sản xuất, nơi mà mỗi giây đều quan trọng và mọi sự cố đều cần được xử lý ngay lập tức để tránh gây ra sự gián đoạn trong quá trình sản xuất. uy nhiên, thách thức lớn nhất đối với việc triển khai các thuật toán học máy (ML) vào các hệ thống IoT tại biên là sự hạn chế về tài nguyên của các nút IoT. Thường thì các nút IoT được tích hợp từ các cảm biến có hạn chế về tài nguyên và được sử dụng cùng với các hệ thống nhúng có chi phí thấp. Các thuật toán học máy thường đòi hỏi khá nhiều tài nguyên tính toán và bộ nhớ để thực hiện quá trình huấn luyện và dự đoán. Tuy nhiên, các nút IoT thường chỉ có khả năng xử lý và lưu trữ hạn chế, điều này có thể làm giảm hiệu suất và độ chính xác của các thuật toán học máy khi triển khai trên các hệ thống IoT tại biên. Để vượt qua thách thức này, cần phải phát triển các phương pháp tiên tiến để tối ưu hóa việc triển khai thuật toán học máy trên các nút IoT, bao gồm cả việc tối ưu hóa thuật toán và quản lý tài nguyên hiệu quả.

Để khắc phục các hạn chế về tài nguyên và triển khai hiệu quả các thuật toán học máy trên các hệ thống IoT tại biên. Đề án sẽ nghiên cứu và triển khai một mô hình học máy có thể phân biệt được các bất thường trong một động cơ điện. Ứng dụng một số thuật toán như CNN và DNN để phân loại các sự cố bất thường dựa trên tập dự liệu thu được từ các cảm biến gắn trên động cơ điện.

## Một số thuật toán phát hiện bất thường cho IoT

Trong lĩnh vực IoT, việc phát hiện bất thường là một phần không thể thiếu để đảm bảo tính an toàn và hiệu suất của hệ thống. Có hai loại chính của thuật toán phát hiện bất thường: có giám sát và không giám sát.

* Trong nhóm có giám sát: Các thuật toán như Support Vector Machine (SVM), Random Forest, và mạng nơ-ron sâu (Deep Neural Networks - DNN) thường được sử dụng. SVM có khả năng phân loại các mẫu là bình thường hoặc bất thường dựa trên dữ liệu được gán nhãn. Random Forest là một phương pháp tạo ra nhiều cây quyết định và kết hợp kết quả để xác định trạng thái bất thường của dữ liệu. DNN, đặc biệt là các mô hình sâu như Convolutional Neural Networks (CNN), có thể học biểu diễn phức tạp của dữ liệu và phát hiện các biểu hiện bất thường thông qua quá trình huấn luyện với dữ liệu có nhãn.
* Trong nhóm không giám sát: Các thuật toán như K-means Clustering, Isolation Forest, và autoencoders cũng rất phổ biến. K-means Clustering phân loại dữ liệu vào các cụm và phát hiện các cụm có tính chất bất thường. Isolation Forest xây dựng cây quyết định để tách các điểm dữ liệu và xác định các điểm ít phân loại hoặc phân loại sớm có thể là bất thường. Autoencoders, đặc biệt là trong mạng nơ-ron sâu, có thể học biểu diễn của dữ liệu và đánh giá mức độ mất mát tái tạo, với mức độ mất mát cao hơn có thể cho thấy các trường hợp bất thường.

Trong các nghiên cứu thì có 12 kỹ thuật, thuật toán ML/DL được áp dụng để phát hiện sự bất thường thông qua tinyML. Trong đó CNN là thuật toán được sử dụng nhiều nhất, chiếm 18,8% trong số các nghiên cứu. Autoencoder, LSTM, DNN, Isolation Forest và Gaussian Mixture Model (GMM) lần lượt theo sau, với tỷ lệ 13,6% mỗi loại [3]. Trong phần này sẽ mô tả kỹ hơn về ba thuật toán đang được sử dụng nhiều nhất là Autoencoder, DNN, CNN và thuật toán SVM cũng là một thuật toán được sử dụng nhiều trong các bài toán phát hiện bất thường.

### Autoencoder

Autoencoder là một loại mạng neuron truyền thẳng không giám sát (unsupervised feedforward neural network), tức là không yêu cầu dữ liệu huấn luyện được gán nhãn. Nó được thiết kế để tự động học cách biểu diễn dữ liệu một cách hiệu quả bằng cách tái tạo dữ liệu đầu vào với độ tương đồng cao nhất có thể. Trong quá trình huấn luyện, mạng autoencoder học cách biểu diễn dữ liệu dưới dạng một biểu diễn gọn gàng và ý nghĩa, thường là thông qua việc giảm chiều dữ liệu. Ngoài việc được sử dụng rộng rãi trong việc giảm chiều dữ liệu và khử nhiễu từ ảnh, autoencoder cũng được áp dụng trong việc phát hiện bất thường trong dữ liệu. Trong phát hiện sự bất thường, sau khi mạng autoencoder đã được huấn luyện với dữ liệu bình thường, nó có thể phát hiện ra các điểm dữ liệu không phù hợp hoặc bất thường trong dữ liệu mới. Điều này thực hiện thông qua việc so sánh dữ liệu mới với các biểu diễn của dữ liệu bình thường mà mạng đã học được. Các điểm dữ liệu có sự khác biệt đáng kể so với dữ liệu bình thường có thể được coi là bất thường và cần phải được kiểm tra hoặc A diagram of a dna structure

Description automatically generated

Hình 2. 2: Cấu trúc của mạng Deep Autoencoder.

tái tạo xử lý. Mục đích của Autoencoder ứng dụng vào phát hiện bất thường sẽ là cố gắng tái tạo dữ liệu đầu vào sao cho giống nhất với dữ liệu huấn luyện. Dựa vào thông tin này, thuật toán chỉ cần cung cấp dữ liệu “normal” để học và không cần cung cấp dữ liệu dạng “abnormal”. Sau đó, Autoencoder phân tích lỗi và tính lỗi tái tạo (reconstruction error) trên cả tập dữ liệu “normal” lẫn “abnormal”, nếu độ lỗi tái tạo nhỏ hơn, đó chứng tỏ Autoencoder đã tái tạo tập dữ liệu bình thường một cách chính xác; ngược lại, nếu độ lỗi tái tạo cao hơn, có thể dữ liệu đầu vào không thuộc lớp bình thường, mà là bất thường. Vì vậy, độ lỗi tái tạo có thể hiểu như một biểu đồ histogram, và chỉ cần thiết lập một ngưỡng để phân biệt giữa hai loại lỗi của dữ liệu: lỗi từ dữ liệu bình thường và lỗi từ dữ liệu bất thường. Điều này biến bài toán thành một bài toán phân loại nhị phân, nơi mục tiêu là phân biệt giữa dữ liệu bình thường và dữ liệu bất thường dựa trên độ lỗi tái tạo từ Autoencoder.

* Cấu trúc của Autoencoder bao gồm 3 phần chính như Hình 2.2 là Encoder, Bottleneck, Decoder cụ thể như sau:
  + Encoder (Bộ mã hóa): Phần này đảm nhận nhiệm vụ biến đổi dữ liệu đầu vào từ không gian chiều cao thành một biểu diễn nén trong không gian chiều thấp hơn. Quá trình này thường được thực hiện thông qua một chuỗi các lớp neural network, như các lớp convolutional hoặc fully connected, để trích xuất các đặc trưng quan trọng từ dữ liệu.
  + Bottleneck (hoặc Code): Đây là phần của mạng autoencoder có kích thước nhỏ nhất, nơi mà dữ liệu được biểu diễn dưới dạng một biểu diễn nén và gọn gàng nhất. Bottleneck chính là trung tâm của autoencoder, nơi mà thông tin quan trọng nhất của dữ liệu được lưu trữ và biểu diễn.
  + Decoder (Bộ giải mã): Sau khi thông tin đã được nén ở phần bottleneck, phần Decoder nhận biểu diễn nén và cố gắng tái tạo lại dữ liệu đầu vào ban đầu. Quá trình này thường được thực hiện thông qua một chuỗi các lớp neural network ngược lại so với phần Encoder, như các lớp convolutional transpose hoặc fully connected, với mục tiêu là tái tạo dữ liệu đầu vào càng giống nhất có thể.
* Mối quan hệ giữa Bộ Mã Hóa (Encoder), Phần Chặn (Bottleneck) và Bộ Giải Mã (Decoder):
  + Bộ Mã Hóa (Encoder) là phần của mạng nơ-ron mà nén đầu vào thành một biểu diễn nhỏ gọn, thường được gọi là Phần Chặn (Bottleneck). Điều này thường được thực hiện thông qua một loạt các khối tích hợp và mô-đun, chức năng chính của chúng là biến đổi đầu vào ban đầu thành một không gian biểu diễn ít chiều hơn mà vẫn bảo toàn những đặc trưng quan trọng nhất của dữ liệu. Bottleneck, hay còn gọi là Phần Chặn, đóng vai trò quan trọng nhất trong mạng nơ-ron và thường là phần nhỏ nhất của nó. Nhiệm vụ của Phần Chặn là hạn chế lưu lượng thông tin đi từ Bộ Mã Hóa đến Bộ Giải Mã, chỉ cho phép thông tin quan trọng nhất đi qua.
  + Bottleneck được thiết kế để tối ưu hóa thông tin mà một hình ảnh có thể sở hữu, từ đó thiết lập biểu diễn tri thức của dữ liệu đầu vào. Cấu trúc của Bộ Mã Hóa-Giải Mã (Encoder-Decoder) cho phép trích xuất tối đa thông tin từ một hình ảnh và xác định các mối quan hệ hữu ích giữa các đặc điểm khác nhau trong mạng. Bottleneck hóa các biểu diễn nén của dữ liệu đầu vào, ngăn chặn mạng nơ-ron ghi nhớ quá nhiều về dữ liệu và trang bị quá nhiều trên dữ liệu. Đặc biệt, khi Bottleneck càng nhỏ, rủi ro quá tải càng giảm.
  + Bộ Giải Mã (Decoder) là phần của mạng nơ-ron mà tái tạo đầu ra từ biểu diễn nhỏ gọn của Phần Chặn. Đầu vào của Bộ Giải Mã là một biểu diễn tri thức đã được nén từ Bộ Mã Hóa, vì vậy Bộ Giải Mã thực hiện vai trò như một "bộ giải nén", tái tạo lại hình ảnh từ các thuộc tính ẩn của nó.
* Một số Hyper-parameter quan trong trong Autoencoder bao gồm:

1. Kích thước mã (Code size): Đây là một tham số cực kỳ quan trọng được sử dụng để xác định kích thước của Bottleneck. Kích thước của Bottleneck quyết định mức độ nén thông tin trong mô hình. Việc chọn kích thước phù hợp cho Bottleneck là một yếu tố quan trọng.
2. Số lớp (Number of layers): Giống như các mạng nơ-ron khác, số lớp là một tham số quan trọng để điều chỉnh độ sâu của bộ mã hóa và bộ giải mã trong Autoencoder. Một mô hình sâu hơn có thể mang lại khả năng biểu diễn phức tạp hơn, nhưng đồng thời cũng tăng độ phức tạp tính toán và kích thước của mô hình.
3. Số lượng nút trên mỗi lớp (Number of nodes per layer): Số lượng nút trên mỗi lớp quyết định số lượng trọng số sẽ được sử dụng trên mỗi lớp. Thông thường, số lượng nút giảm dần theo từng lớp để đảm bảo tính chất của dữ liệu đầu vào được bảo toàn qua các lớp.
4. Hàm mất mát tái tạo (Reconstruction Loss): Hàm mất mát là một phần không thể thiếu trong mạng nơ-ron và phụ thuộc vào loại dữ liệu mà mô hình sẽ xử lý. Ví dụ, khi làm việc với ảnh, các hàm mất mát phổ biến bao gồm Mean Square Error (MSE) và L1 Loss. Trong trường hợp dữ liệu nhị phân như MNIST, Binary Cross Entropy có thể là lựa chọn tốt hơn. Việc chọn hàm mất mát phù hợp sẽ ảnh hưởng đến hiệu suất và khả năng học của mô hình.

Autoencoder thuật toán mạnh mẽ để phát hiện bất thường trong dữ liệu IoT. Bằng cách huấn luyện trên dữ liệu bình thường, nó học cách tái tạo dữ liệu một cách chính xác. Khi có dữ liệu mới, nếu độ lỗi tái tạo vượt quá ngưỡng, chúng ta có thể kết luận rằng dữ liệu đó có thể là bất thường. Tính linh hoạt và hiệu quả của autoencoder khi áp dụng vào phát hiện bất thường trong dữ liệu IoT làm cho nó trở thành một công cụ quan trọng trong việc duy trì an ninh và giám sát trong các hệ thống IoT phức tạp. Đồng thời, sự tiện lợi của việc tự động hóa quy trình phát hiện bất thường giúp tiết kiệm thời gian và công sức của các nhà phát triển và quản trị hệ thống.

### Mạng Nơ-Ron sâu (DNN)

Deep Neural Network (DNN), một trong những mạng nơ-ron nhân tạo tiên tiến, đóng vai trò quan trọng trong lĩnh vực Học Sâu (Deep Learning). DNN không chỉ dự đoán và phân loại dữ liệu mới chưa từng xuất hiện trong tập mẫu, mà còn mở ra nhiều ứng dụng đa dạng. DNN được sử dụng để thực hiện nhiều nhiệm vụ, bao gồm phân loại ảnh, nhận dạng đối tượng, nhận dạng giọng nói, dịch máy, nhận dạng chữ viết tay, và thậm chí là lái xe tự động. Nó cũng được áp dụng trong dự báo và dự đoán, tìm kiếm và gợi ý. Sự linh hoạt và hiệu suất của DNN đã tạo ra sự tiến bộ đáng kể trong nhiều lĩnh vực ứng dụng khác nhau. Trong lĩnh vực IoT, việc phát hiện các bất thường là vô cùng quan trọng với các hệ thống IoT. Mạng DNN có khả năng tự học và điều chỉnh để nhận biết các dữ liệu bình thường từ các cảm biến, từ đó có thể phát hiện ra những dữ liệu bất thường và đưa ra các cảnh báo sớm. Việc này không chỉ giúp ngăn chặn các sự cố, mà còn cung cấp cơ sở cho quy trình bảo dưỡng dựa trên thực tế, tối ưu hóa hiệu suất và độ tin cậy của các hệ thống IoT. Mặc dù, việc huấn luyện một mô hình DNN có thể tốn rất nhiều thời gian và tài nguyên tính toán, đặc biệt là khi số lượng lớp và nơ-ron trong mô hình rất lớn. Trong luận văn có sử dụng một phần của mạng DNN để phát hiện hiện các bất thường trên động cơ điện.

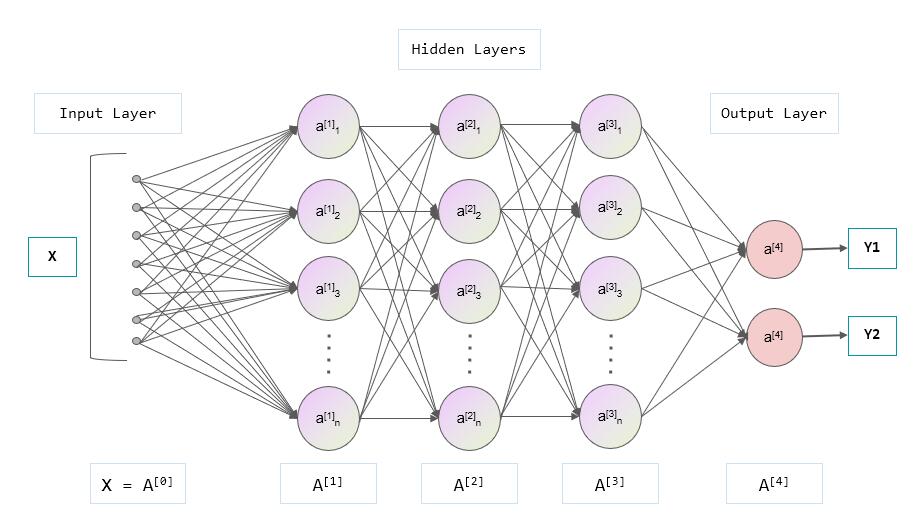
* Khái niệm

Mạng DNN là viết tắt của "Deep Neural Network" (Mạng Nơ-ron Sâu). Đây là một dạng của mạng nơ-ron nhân tạo (ANN) được cấu trúc thành nhiều lớp, trong đó thông tin được chuyển tiếp qua các lớp từ lớp đầu vào đến lớp đầu ra thông qua một loạt các phép biến đổi tuyến tính và phi tuyến tính. Trong mạng DNN, các nơ-ron được kết nối thành các lớp, trong đó mỗi nơ-ron có một tập trọng số (weights) và một độ lệch (biases) đi kèm. Quá trình huấn luyện mạng DNN thường sử dụng các thuật toán tối ưu hóa như Gradient Descent hoặc các biến thể như Stochastic Gradient Descent (SGD), Adam, hoặc RMSprop đóng vai trò quan trọng. Nhưng quá trình này không chỉ đơn giản là việc tính toán gradient của hàm mất mát (loss function) và cập nhật trọng số và độ lệch. Nó là một quy trình phức tạp của sự học, trong đó mỗi cập nhật trọng số và độ lệch đều phản ánh sự độ chính xác ngày càng tốt hơn về dữ liệu, và từ đó, mô hình hóa chính xác hơn. Cùng với backpropagation, một kỹ thuật cho phép tính toán gradient một cách hiệu quả qua các lớp của mạng, quá trình này rất quan trọng với việc huấn luyện mạng nơ-ron. Trong mỗi bước của quá trình huấn luyện, việc lựa chọn thuật toán tối ưu hóa phù hợp là một vấn đề quan trọng, có thể quyết định sự hội tụ nhanh chóng của mạng hoặc sự ổn định của quá trình huấn luyện. Adam, SGD, hay RMSprop, mỗi thuật toán mang lại một cách tiếp cận khác nhau, với những ưu điểm và hạn chế riêng.

Perceptrons đóng vai trò như các mô-đun cơ bản để xử lý dữ liệu đầu vào và tạo ra đầu ra tương ứng trong mạng Neural Hồi Quy Sâu (DNN). Mỗi perceptron nhận đầu vào từ các mô-đun trước đó, tính tổng có trọng số của chúng và sau đó áp dụng một hàm kích hoạt để tạo ra đầu ra. Trọng số trong perceptron được điều chỉnh để tạo ra đầu ra của tầng và làm cho mô hình có khả năng học các mẫu và tương quan trong dữ liệu. Trong kiến trúc DNN, các perceptron được tổ chức thành các tầng và kết nối với nhau. Tầng đầu tiên được gọi là tầng đầu vào, nhận đầu vào từ dữ liệu. Các tầng ẩn nằm ở giữa thực hiện tính toán và truyền dẫn thông tin qua mạng. Tầng cuối cùng là tầng đầu ra, tạo ra đầu ra của mạng. Perceptrons trong DNN chủ yếu được sử dụng để tạo ra các biểu diễn phi tuyến tính và học các mẫu phức tạp trong dữ liệu. Bằng cách kết hợp nhiều perceptron và các tầng ẩn, DNN có khả năng học các đặc trưng phức tạp và giải quyết các bài toán phức tạp như xử lý ngôn ngữ tự nhiên, thị giác máy tính và nhận dạng giọng nói.

* Kiến trúc

Mạng Neural Hồi Quy Sâu (DNN) được cấu thành từ một kiến trúc phức tạp, bao gồm sáu thành phần chính được tổ chức một cách cẩn thận để xử lý thông tin đầu vào và tạo ra dự đoán chính xác. Mỗi thành phần của kiến trúc này đóng vai trò quan trọng trong việc xây dựng một mô hình mạng nơ-ron mạnh mẽ và linh hoạt. Trong Hình 2.3 mô tả mạng nơ-ron năm lớp bao gồm một lớp đầu vào, ba lớp ẩn và một lớp đầu ra. Mỗi mạng nơ-ron thì gồm các lớp chính như sau:



Hình 2. 3: Mạng nơ-ron năm lớp với ba lớp ẩn.

* Tầng Đầu Vào (Input Layer): Là nơi tiếp nhận dữ liệu đầu vào từ các nguồn như hình ảnh, văn bản, hoặc âm thanh. Mỗi nơ-ron trong tầng này đại diện cho một đặc trưng hoặc biến đầu vào cụ thể. Số lượng nơ-ron phụ thuộc vào số lượng và loại biến đầu vào của bài toán. Ví dụ, nếu xử lý hình ảnh kích thước 100x100 pixel, mỗi pixel có thể tương ứng với một nơ-ron trong tầng này. Điều này giúp mạng DNN biểu diễn và xử lý dữ liệu đầu vào từ nhiều nguồn để tạo ra các đặc trưng hữu ích cho việc học và dự đoán.
* Tầng Ẩn (Hidden Layers): Nằm ở giữa tầng đầu vào và tầng đầu ra, đóng vai trò quan trọng trong việc thực hiện các tính toán và truyền thông tin qua mạng. Mỗi tầng ẩn được định nghĩa trước với một số lượng nơ-ron cụ thể. Trong các tầng ẩn này, mỗi nơ-ron tính toán thông tin từ các nơ-ron trong tầng trước đó bằng cách sử dụng các trọng số và hệ số bias. Điều này giúp tạo ra một mạng nơ-ron có khả năng học và biểu diễn các đặc trưng phức tạp từ dữ liệu đầu vào.
* Tầng Đầu Ra (Output Layer): Hay còn gọi là tầng cuối cùng của mạng Neural Hồi Quy Sâu (DNN), chịu trách nhiệm tạo ra các dự đoán cuối cùng của mô hình. Số lượng nơ-ron trong tầng này thường phụ thuộc vào loại bài toán cụ thể mà mô hình đang giải quyết. Ví dụ, trong một bài toán phân loại nhị phân, chỉ cần một nơ-ron để đưa ra dự đoán. Tuy nhiên, đối với các bài toán phân loại nhiều lớp, số lượng nơ-ron trong tầng này sẽ phụ thuộc vào số lượng lớp cần phân loại. Tầng Đầu Ra chính là nơi mà mô hình DNN đưa ra dự đoán cuối cùng dưới dạng các giá trị đầu ra phù hợp với bài toán.
* Quá trình xử lý
* Đầu vào (Input): Dữ liệu đầu vào được chuẩn bị và tiền xử lý trước khi đưa vào mạng neural. Điều này có thể bao gồm chuẩn hóa dữ liệu, mã hóa one-hot (nếu cần) và các phương pháp tiền xử lý khác.
* Lan truyền tiến (Feedforward): Dữ liệu đầu vào được truyền qua mạng neural từ lớp đầu vào cho đến lớp đầu ra. Mỗi nơ-ron trong mỗi lớp tính tổng trọng số của các kết nối đến từ các nơ-ron trong lớp trước đó, sau đó áp dụng một hàm kích hoạt để tạo ra đầu ra của nơ-ron.
* Tính toán lỗi (Error Computation): Một hàm lỗi (loss function) được sử dụng để đánh giá hiệu suất của mạng neural dựa trên đầu ra dự đoán và các nhãn thực tế. Thông thường, sự khác biệt giữa đầu ra dự đoán và giá trị thực tế được sử dụng để tính toán hàm lỗi. Một số hàm lỗi phổ biến bao gồm: Cross-Entropy Loss, Mean Squared Error (MSE), Hinge Loss.
* Phản hồi ngược (Backpropagation): Quá trình backpropagation là quá trình tính toán độ dốc của hàm lỗi theo các tham số của mạng, từ lớp đầu ra đến lớp đầu vào. Độ dốc này sau đó được sử dụng để cập nhật trọng số của các kết nối trong mạng neural thông qua một thuật toán tối ưu hóa như gradient descent hoặc các biến thể của nó.
* Cập nhật trọng số (Weight Update): Các trọng số của mạng neural được cập nhật dựa trên độ dốc tính được từ quá trình backpropagation. Quá trình này có thể được thực hiện bằng cách sử dụng các thuật toán tối ưu hóa để điều chỉnh trọng số một cách tối ưu nhằm giảm thiểu hàm lỗi.
* Lặp lại (Iteration): Quá trình truyền tiến, tính toán lỗi, backpropagation và cập nhật trọng số được lặp lại nhiều lần thông qua các vòng lặp (epochs) để mạng neural được huấn luyện đủ lâu để học được biểu diễn tốt của dữ liệu đầu vào.
* Đánh giá (Evaluation): Sau khi mạng neural đã được huấn luyện, nó được đánh giá bằng cách sử dụng dữ liệu kiểm tra để đo lường hiệu suất của việc huấn luyện trên các dữ liệu thử nghiệm. Điều này giúp đảm bảo rằng mạng neural có khả năng tổng quát hóa tốt cho các dữ liệu mới.

### Mạng Nơ-Ron tích chập (CNN)

Mạng CNN đã không chỉ là một phần quan trọng trong lĩnh vực thị giác máy tính mà còn có ứng dụng tiềm năng lớn trong việc phát hiện bất thường trong hệ thống IoT. Trong các hệ thôgns IoT việc phát hiện các sự kiện không bình thường có thể giúp dự đoán và ngăn chặn các sự cố trước khi chúng gây ra hậu quả lớn. CNN có thể được sử dụng để xử lý dữ liệu từ các thiết bị biên trong hệ thống IoT, như các cảm biến, máy tính nhúng, hoặc camera. Dữ liệu từ các thiết bị này thường bao gồm các thông số như nhiệt độ, áp suất, độ ẩm, hoặc hình ảnh, video. Sử dụng CNN có thể huấn luyện mô hình để phát hiện các biến đổi bất thường trong dữ liệu này, chỉ ra khi nào một sự kiện nào đó không phù hợp với mô hình dự kiến. Ví dụ, trong một hệ thống giám sát môi trường, CNN có thể được áp dụng để phát hiện các biến đổi bất thường trong nồng độ khí thải hoặc chất lượng không khí. Khi dữ liệu từ cảm biến cho thấy một sự thay đổi lớn và đột ngột, mô hình CNN có thể cảnh báo về một tình huống bất thường, cho phép người quản lý hệ thống can thiệp kịp thời. Ngoài ra, CNN cũng có thể được sử dụng để phát hiện các hành vi bất thường trong giao thông, như tai nạn hoặc vi phạm luật giao thông. Bằng cách phân tích dữ liệu từ camera giám sát, mô hình CNN có thể nhận biết các hành vi không phù hợp, giúp cảnh báo về các tình huống nguy hiểm trên đường. Như vậy, sử dụng mạng CNN trong hệ thống IoT không chỉ giúp cải thiện khả năng giám sát và quản lý mà còn tăng cường khả năng phát hiện và ứng phó với các tình huống bất thường, từ đó giảm thiểu rủi ro và tăng cường an toàn cho các hệ thống và cộng đồng.

* Khái niệm

Mạng nơ-ron tích chập (CNN) là một loại mô hình học máy, được xem là phù hợp nhất cho việc phân tích dữ liệu trực quan. CNN, thường được gọi là convnet, sử dụng các nguyên tắc từ đại số tuyến tính, đặc biệt là các phép toán tích chập, để trích xuất các đặc trưng và nhận biết các mẫu trong hình ảnh. Mặc dù chủ yếu được thiết kế để xử lý hình ảnh, CNN cũng có thể được điều chỉnh để hoạt động với dữ liệu âm thanh và tín hiệu khác. Kiến trúc của CNN được lấy cảm hứng từ cách mà não bộ con người - đặc biệt là vỏ não thị giác - hoạt động để nhận biết và xử lý các tín hiệu thị giác. Trong CNN, các tế bào thần kinh nhân tạo được tổ chức một cách hiệu quả để diễn giải thông tin hình ảnh, cho phép mô hình xử lý toàn bộ hình ảnh một cách hiệu quả. Với khả năng xuất sắc trong việc nhận diện các đối tượng, CNN thường được sử dụng trong các ứng dụng thị giác máy tính như nhận dạng hình ảnh và phát hiện đối tượng. Các ứng dụng phổ biến bao gồm hệ thống xe tự lái, nhận dạng khuôn mặt và phân tích hình ảnh y tế. Trong khi các mạng nơ-ron truyền thống thường yêu cầu xử lý hình ảnh theo từng phần hoặc sử dụng hình ảnh được phân đoạn, CNN có cách tiếp cận toàn diện hơn, cho phép nó vượt trội trong các tác vụ liên quan đến hình ảnh. Điều này làm cho CNN trở thành lựa chọn hàng đầu cho xử lý hình ảnh và, ở mức độ thấp hơn, xử lý giọng nói và âm thanh.

* Kiến trúc

Một Mạng nơ-ron tích chập (CNN) thường được cấu trúc với một chuỗi các lớp quan trọng như Hình 2.4. Cụ thể, các lớp bao gồm Lớp đầu vào (Input layer), Lớp tích chập (Convolutional layer), Lớp gộp (Pooling layer), Lớp kết nối đầy đủ (Fully connected layer) và Lớp đầu ra (Output layer). Các lớp này cùng liên kết để chuyển đổi dữ liệu đầu vào thành dự đoán hoặc phân loại ở đầu ra. Chi tiêt các lớp như sau:

A diagram of a diagram of a process

Description automatically generated with medium confidence

Hình 2. 4: Kiến trúc CNN đơn giản.

* Lớp đầu vào (Input layer): Lớp này là nơi tiếp nhận dữ liệu đầu vào, thường là một tập hợp các đặc trưng hoặc dữ liệu số. Dữ liệu đầu vào có thể được biểu diễn dưới dạng ma trận hoặc vectơ, tùy thuộc vào loại dữ liệu được sử dụng. Ví dụ, đối với dữ liệu âm thanh, mỗi âm thanh có thể được biểu diễn dưới dạng vectơ đặc trưng, trong khi đối với dữ liệu văn bản, mỗi câu có thể được mã hóa dưới dạng một vectơ từ.
* Lớp tích chập (Convolutional layer): Lớp này thực hiện việc áp dụng các bộ lọc (filters) lên dữ liệu đầu vào để trích xuất các đặc trưng cụ thể. Các bộ lọc này có thể nhận diện các mẫu hoặc cấu trúc trong dữ liệu đầu vào, tương tự như việc nhận diện cạnh, góc, hoặc các đặc điểm đặc trưng khác. Kết quả của quá trình này là một bản đồ đặc trưng (feature map) được tạo ra từ việc áp dụng các bộ lọc lên dữ liệu.
* Lớp gộp (Pooling layer): Lớp này được sử dụng để giảm kích thước của bản đồ đặc trưng, giúp giảm chi phí tính toán và nguy cơ overfitting. Phổ biến nhất là lớp gộp tối đa (max pooling), trong đó giá trị lớn nhất trong mỗi vùng cụ thể của bản đồ đặc trưng được chọn làm đại diện cho vùng đó. Quá trình gộp còn có thể giúp giữ lại các đặc trưng quan trọng trong quá trình học.
* Lớp kết nối đầy đủ (Fully connected layer): Sau khi đã trích xuất các đặc trưng từ dữ liệu và giảm kích thước của chúng, thông tin được chuyển đến các lớp kết nối đầy đủ để thực hiện việc phân loại hoặc dự đoán. Mỗi nút trong lớp này kết nối với tất cả các nút của lớp trước đó, tạo ra một mạng kết nối đầy đủ. Các lớp này thường sử dụng các hàm kích hoạt như ReLU (Rectified Linear Unit) để tạo tính phi tuyến tính và khả năng học của mạng.
* Lớp đầu ra (Output layer): Lớp này chứa các nút đại diện cho các lớp phân loại, mỗi nút đại diện cho một lớp hoặc một loại của đối tượng. Đối với bài toán phân loại, hàm kích hoạt thường được sử dụng là softmax để tính toán xác suất của mỗi lớp, cho biết xác suất của đối tượng thuộc vào từng lớp.

### SVM

SVM (Support Vector Machine) là một trong những thuật toán quan trọng trong lĩnh vực học máy, được sử dụng phổ biến cho cả các bài toán phân loại và hồi quy. Đặc điểm độc đáo của SVM là khả năng xử lý cả hai loại bài toán này thông qua việc tối ưu hóa một đường ranh giới (decision boundary) trong không gian đặc trưng sao cho cực đại hoặc cực tiểu tạo thành một hàm mất mát nhất định. SVM có thể được hiểu như một ranh giới phân chia giữa các nhóm dữ liệu trong không gian đặc trưng. Một ranh giới phân loại tốt là đường ranh giới cách xa nhất có thể từ các điểm dữ liệu của các nhóm, được gọi là margin. SVM cố gắng tối đa hóa margin này để tạo ra một ranh giới phân loại tốt nhất.

Trong quá trình huấn luyện, SVM xác định các điểm dữ liệu quan trọng nhất, được gọi là các vector hỗ trợ (support vectors), chúng đóng vai trò quan trọng trong việc xác định ranh giới phân loại. SVM chỉ phụ thuộc vào các vector hỗ trợ để xác định ranh giới phân chia, điều này làm giảm đáng kể độ phức tạp của mô hình. Hình 2.5 cho thấy tập dữ liệu được chia thành hai lớp và bài toán nằm trong không gian hai chiều. Ở đây siêu mặt phẳng được biểu diễn dưới dạng đường thẳng nét liền to đậm.

A diagram of a technical support

Description automatically generated

Hình 2. 5: Thuật toán SVM

Một trong những khái niệm quan trọng nhất của SVM là kernel. Kernel cho phép SVM xử lý các vấn đề phân loại phi tuyến tính bằng cách ánh xạ dữ liệu từ không gian đặc trưng ban đầu vào một không gian chiều cao hơn, nơi phân loại có thể được thực hiện một cách tuyến tính. Các loại kernel phổ biến bao gồm linear, polynomial, radial basis function (RBF), và sigmoid… Hàm mất mát (loss function) trong SVM đo lường sự mất mát của mô hình dự đoán so với thực tế và được sử dụng để đánh giá hiệu suất của SVM trong quá trình huấn luyện. Hàm mất mát thường được sử dụng là hàm hinge loss, được thiết kế để tối ưu hóa margin. Ngoài ra, SVM có các tham số siêu (hyperparameters) cần được điều chỉnh để tối ưu hóa hiệu suất của mô hình. Các tham số này bao gồm tham số regularization (C), loại kernel, gamma cho kernel RBF, và các tham số liên quan đến hàm mất mát. Điều chỉnh các tham số này là quan trọng trong quá trình huấn luyện SVM và có thể ảnh hưởng đến hiệu suất của mô hình trên dữ liệu mới. Hàm mất mát thường được sử dụng là hàm hinge loss, được thiết kế để tối ưu hóa margin. Ngoài ra, SVM có các tham số siêu (hyperparameters) cần được điều chỉnh để tối ưu hóa hiệu suất của mô hình. Các tham số này bao gồm tham số regularization, loại kernel, gamma cho kernel RBF, và các tham số liên quan đến hàm mất mát. Điều chỉnh các tham số này là quan trọng trong quá trình huấn luyện SVM và có thể ảnh hưởng đến hiệu suất của mô hình trên dữ liệu mới.

Mặc dù Deep Learning và Neural Networks đang phát triển mạnh mẽ, SVM vẫn giữ vững tầm quan trọng của mình trong ngành học máy, được sử dụng rộng rãi trong nhiều ngành công nghiệp với độ chính xác và hiệu suất cao.

## Một số cảm biến cho hệ thống IoT tại biên

Cảm biến đóng vai trò quan trọng trong cả các hệ thống IoT nói chung và hệ thống phát hiện bất thường trên các động cơ công nghiệp cụ thể. Trong hệ thống IoT, cảm biến không chỉ thu thập dữ liệu từ môi trường xung quanh mà còn giúp giám sát và điều khiển các thiết bị một cách tự động và thông minh. Dữ liệu từ các cảm biến cũng có thể được sử dụng để phát hiện các biểu hiện không bình thường và tiên đoán sự cố, giúp hệ thống IoT ngăn chặn các vấn đề trước khi chúng xảy ra. Trong hệ thống phát hiện bất thường trên các động cơ công nghiệp, cảm biến là công cụ quan trọng để giám sát trạng thái và hoạt động của động cơ. Dữ liệu từ các cảm biến như áp suất, nhiệt độ và rung động cung cấp thông tin quan trọng về trạng thái của động cơ và giúp phát hiện sớm các vấn đề có thể xảy ra. Điều này giúp tối ưu hóa hiệu suất và bảo dưỡng của động cơ, từ đó gia tăng tuổi thọ và giảm thiểu chi phí bảo trì.

Dựa trên các nghiên cứu được chọn lọc, Đơn vị đo lường quán tính (IMU) là loại dữ liệu được sử dụng rộng rãi nhất để phát hiện sự bất thường trong các ứng dụng TinyML, chiếm 20% trong 18 bài báo sử dụng các thuật toán ML/DL để phát hiện bất thường trên MCU [3]. Trong một nghiên cứu khác đã chỉ ra Cảm biến dòng điện (current sensor) được sử dụng trong 29 nghiên cứu [2]. Hình 2.6 trình bày rằng đa số các cảm biến được áp dụng hiện nay trong việc phát hiện bất thường trên các hệ thống IoT và trong việc phát hiện bất thường cho các máy móc công nghiệp chủ yếu tập trung vào hai loại cảm biến: IMU và cảm biến dòng điện. Vì vây trong luận án này sẽ tập trung vào sử dụng hai cảm biến trên để phát hiện bất thường cho động cơ điện.

|  |  |
| --- | --- |
| A graph of blue bars with white text  Description automatically generated | A colorful pie chart with numbers  Description automatically generated |

Hình 2. 6: Minh họa phân bổ các cảm biến và các kiểu dữ liệu được sử dụng nhiều trong các nghiên cứu [2], [3].

### Cảm biến gia tốc ADXL345

A close-up of a blue circuit board

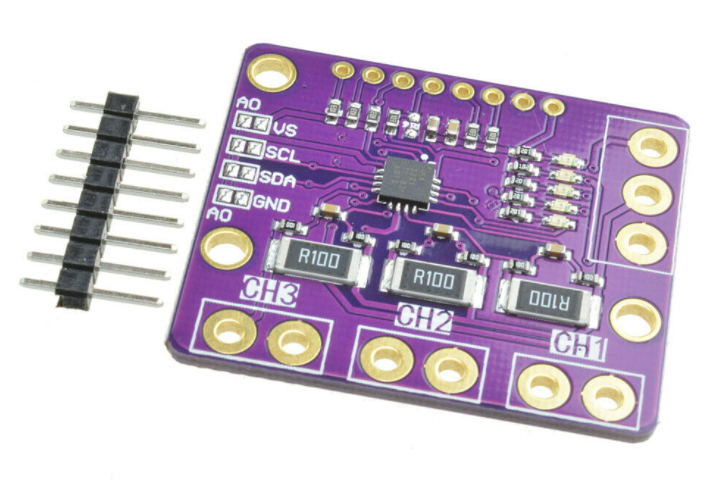
Description automatically generated

Hình 2. 7: Cảm biến gia tốc ADXL345.

ADXL345 la module cảm biến độ nghiêng 3 trục, tiêu thụ năng lượng thấp với độ phân giải cao (13 bit) đo lên đến ± 16g. Dữ liệu đầu ra kỹ thuật số được định dạng dưới dạng bổ sung 16bit hai chiều và có thể truy cập thông qua giao diện kỹ thuật số SPI (3 hoặc 4 dây) hoặc I2C. Cảm biến gia tốc góc ADXL345 có thể được sử dụng để phát hiện bất thường trong hoạt động của động cơ trong các ứng dụng như máy móc, thiết bị công nghiệp hoặc các phương tiện di động như ô tô, xe máy, hay máy bay. Bằng cách gắn cảm biến này vào động cơ, ta có thể sử dụng các thông số như gia tốc và góc quay để giám sát trạng thái hoạt động của động cơ và phát hiện sớm các dấu hiệu bất thường, từ đó đưa ra cảnh báo và ngăn chặn sự cố trước khi nó xảy ra. Khi động cơ hoạt động bình thường, ADXL345 sẽ ghi nhận dữ liệu gia tốc và góc quay theo một mô hình đã được xác định trước. Bất kỳ biến đổi nào đáng ngờ, như sự dao động lớn, thay đổi đột ngột trong góc quay, hoặc biến đổi ngoài phạm vi dự kiến, có thể được xem xét là dấu hiệu của sự cố hoặc hỏng hóc. Khi phát hiện bất thường, hệ thống có thể kích hoạt cảnh báo hoặc ngừng hoạt động động cơ để tránh những hậu quả nghiêm trọng.

* Thông số kỹ thuật:
* Điện áp hoạt động: 3.3VDC / 5VDC
* Dòng điện tiêu thụ: 23uA
* Độ phân giải: 10bit (độ phân giải tối đa 13 bit)
* Giao tiếp: SPI hoặc I2C
* Kích thước: 14 x 19mm
* Nhiệt độ hoạt động: -40 đến +85 độ C

### Cảm biến giám sát điện áp và dòng điện DC INA3221



Hình 2. 8: Module giám sát điện áp và dòng điện DC INA3221.

Module giám sát điện áp và dòng điện INA3221 có ba kênh theo dõi từ CH1-CH3, được thiết kế với chuẩn giao tiếp I2C và SMBUS. INA3221 có khả năng giám sát cả điện áp shunt và điện áp cung cấp cho bus, và cung cấp các tính năng như lập trình thời gian chuyển đổi và chế độ lấy trung bình cho các tín hiệu này. Đặc biệt, INA3221 cung cấp cảnh báo quan trọng và cảnh báo để phát hiện nhiều điều kiện nằm ngoài phạm vi có thể lập trình cho mỗi kênh. Cảm biến INA3221 có khả năng cảm nhận dòng điện trên các bus trong khoảng từ 0 V đến 26 V. Nó hoạt động với nguồn cung cấp từ 2,7 V đến 5,5 V và tiêu thụ dòng điện cung cấp khoảng 350 µA (tiêu chuẩn). INA3221 có thể hoạt động trong phạm vi nhiệt độ từ -40°C đến +125°C. Giao diện I2C và SMBUS có bốn địa chỉ có thể lập trình, cung cấp linh hoạt trong việc tích hợp vào các hệ thống đa cảm biến. INA3221 được ứng dụng trong phát hiện bất thường của động cơ điện DC bằng cách giám sát dòng điện và điện áp trên các bus của động cơ. Bằng cách này, INA3221 có thể đo và theo dõi mức dòng điện tiêu thụ qua động cơ cũng như điện áp cung cấp cho động cơ. Dựa vào các dữ liệu thu thâp từ cảm biến có thể tạo được ra các tập dữ liệu bình thường hoặc bất thường của động cơ từ đó có thể phát hiện được bất thường.

* Thông số kỹ thuật:
* Số kênh: 3 kênh (giám sát dòng điện và điện áp bus).
* Giao tiếp: Chuẩn I2C và SMBUS.
* Dải điện áp đo: 0v đến 26V.
* Dòng điện tiêu thụ: 350 μA
* Địa chỉ I2C: Có bốn địa chỉ có thể lập trình được.

## Tập dữ liệu sử dụng trong mô hình phát hiện bất thường tại biên

Tập dữ liệu cho luận văn được thu thập từ 5 quạt điện DC như Hình 2.9 với kích thước 8x8x2.5 và hoạt động ở điện áp 5V. Sự đa dạng trong việc thu thập dữ liệu từ các quạt điện này là một yếu tố quan trọng, vì mỗi quạt có các điều kiện hoạt động và vận hành riêng biệt, không thể hoàn toàn tương đồng. Bằng cách thu thập dữ liệu từ nhiều nguồn, không chỉ tạo ra một tập dữ liệu đa chiều mà còn phản ánh chính xác hơn hiệu suất của các ứng dụng sử dụng dữ liệu này. Đồng thời, việc sử dụng dữ liệu từ nhiều nguồn giúp loại bỏ các yếu tố thiên vị có thể tồn tại khi chỉ sử dụng một nguồn dữ liệu duy nhất, nâng cao tính tin cậy của tập dữ liệu.



Hình 2. 9: Thiết quạt được sử dụng để thu thập dữ liệu.

Tập dữ liệu được thu từ hai cảm biến đã được giới thiệu ở phần 2.3 là cảm biến ADXL345 và cảm biến INA3221. Cảm biến ADXL345 được sử dụng để đo gia tốc chuyển động của quạt theo ba trục Ax, Ay và Az, từ đó cho phép theo dõi các biến động và rung động của quạt. Trong khi đó, cảm biến INA3221 được dùng để đo dòng điện tiêu thụ của quạt. Các cảm biến này được lấy mẫu với tần số 50Hz và được lưu dưới định dạng file CSV. Bảng 2.1 cung cấp chi tiết về các tập dữ liệu và các trạng thái tương ứng của từng tập dữ liệu. Bảng này chứa thông tin về các trạng thái của quạt, bao gồm trạng thái bình thường và bất thường, cùng với số lượng tệp dữ liệu, thời gian thu thập và kích thước của mỗi trạng thái. Trong bảng có đề cập đến sáu tệp chính bao gồm Quạt Tắt (off), Quạt Bật (on), Gõ (tapping), Quạt Rung (mount fall), Lỗi cánh/Kẹt cánh (blade fault), Gãy cánh quạt (broken balde) kịch bản chi tiết thu các tập dữ liệu như sau:

* Quạt Tắt(off): Trạng thái khi quạt không hoạt động, ngừng vận hành. Phương pháp thu thập bắt đầu bằng việc ghi lại dữ liệu từ cảm biến dòng điện INA3221 để xác định dòng điện tiêu thụ, phải cho kết quả là ~ 0A khi quạt tắt. Ghi lại dữ liệu từ cảm biến gia tốc ADXL345 để kiểm tra và xác định trạng thái không có rung động của quạt.
* Quạt Bật(off): Trạng thái quạt hoạt động bình thường. Phương pháp thu thập là kích hoạt quạt và ghi lại dữ liệu từ cảm biến INA3221 và ADXL345 trong quá trình hoạt động. Ghi lại các dữ liệu như dòng điện tiêu thụ, gia tốc, và bất kỳ thông số nào liên quan đến vận hành của quạt sau đó lưu vào file CSV.
* Gõ (Tapping): Trạng thái khi một lực nhỏ được áp dụng lên quạt, mô phỏng các tác động nhỏ từ bên ngoài. Phương pháp thu thập là sử dụng các vật hoặc dùng tay gõ trực tiếp lên quạt để tạo các rung động từ đó ghi lại các biến đổi trong dữ liệu thu về từ cảm biến ADXL345.

Bảng 2. 1: Bảng chi tiết tập dữ liệu sử dụng để phát hiện bất thường tại biên

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên tập dữ liệu | Trạng thái | Số lượng tệp | Thời gian(phút) | Kích thước(MB) |
| 1 | Quạt Tắt(off) | Bình thường | 100 | 60 | 7,8 |
| 2 | Quạt Bật(on) | Bình thường | 150 | 90 | 11 |
| 3 | Gõ(tapping) | Bất thường | 140 | 65 | 6,48 |
| 4 | Quạt Rung  (mount fall) | Bất thường | 136 | 62 | 6,34 |
| 5 | Lỗi cánh/Kẹt cánh  (blade fault) | Bất thường | 100 | 60 | 4,49 |
| 6 | Gãy cánh quạt (broken balde) | Bất thường | 80 | 90 | 8,95 |

* Quạt Rung (Mount Fall): Trạng thái khi quạt bị rung lắc không ở một vị trí cố định. Phương pháp thu thập là thực hiện dùng tay rung lắc quạt qua lại từ đó ghi lại các biến đổi trong dữ liệu thu về từ cảm biến ADXL345.
* Lỗi Cánh/Kẹt Cánh (Blade Fault): Trạng thái khi cánh của quạt bị lỗi hoặc kẹt, gây ra sự giảm hiệu suất hoặc nguy cơ hỏng hóc. Phương thức thu thập là tạo ra các tình huống làm kẹt cánh quạt khi đang hoạt động bình thường từ đó ghi lại các biến đổi trong dữ liệu thu về từ cảm biến INA3221.
* Gãy Cánh Quạt (Broken Blade): Trạng thái khi một hoặc nhiều cánh của quạt bị gãy hoặc hỏng. Phương pháp thu thập là thực hiện bẻ một hoặc nhiều cánh quạt và ghi lại dữ liệu từ cảm biến ADXL345.

## Kết luận chương 2

Trong chương 2 của đề án tập trung vào những kỹ thuật phát hiện bất thường phổ biến nhất dành cho IoT nói chung và IoT tại biên nói riêng. Các thuật toán được giới thiệu bao gồm SVM, CNN, DNN và Autoencoder, những công cụ quan trọng để xử lý dữ liệu phức tạp và nhận diện các biểu hiện không bình thường trong các hệ thống IoT. Bên cạnh đó, chương này cũng giới thiệu và đề xuất hai loại cảm biến phổ biến nhất trong các ứng dụng IoT, đó là cảm biến giám sát điện áp và dòng điện DC INA3221 cùng cảm biến gia tốc ADXL345. Những cảm biến này cung cấp thông tin quan trọng về trạng thái hoạt động và môi trường của các thiết bị IoT, từ đó làm cơ sở cho quá trình phát hiện bất thường và quản lý hiệu suất của hệ thống. Hơn nữa, chương này giới thiệu tập dữ liệu được áp dụng trong việc xây dựng mô hình phát hiện bất thường, chuẩn bị cho việc triển khai và ứng dụng các mô hình này vào chương 3. Điều này làm nổi bật vai trò quan trọng của việc sử dụng dữ liệu thực tế để phát triển và kiểm thử các giải pháp phát hiện bất thường trong môi trường IoT đa dạng và phức tạp.

# CHƯƠNG 3 – ÁP DỤNG THUẬT TOÁN PHÁT HIỆN BẤT THƯỜNG CHO HỆ THỐNG NHÚNG TẠI BIÊN

## Đề xuất xây dựng mô hình hệ thống phát hiện bất thường cho IoT tại biên.

## Thiết lập thử nghiệm hệ thống phát hiện bất thường dựa trên thuật toán

## Mô phỏng, thử nghiệm giao thức đề xuất

Trong phần này sẽ tiến hành thiết lập thử nghiệm giải pháp phát hiện xâm nhập IDS đề xuất trên tập dữ liệu nhỏ IoT-23.

## Đánh giá kết thực nghiệm

### 3.3.1 Các tham số đánh giá

### 3.3.2 Đánh giá kết quả thử nghiệm mô hình đề xuất

## Kết luận chương 3

Chương cuối của đề án, một giải pháp IDS thông minh sử dụng các thuật toán học máy để phát hiện tấn công một cách hiệu quả đã được đề xuất. Đồng thời, mô hình hệ thống đã được thử nghiệm và đánh giá. Hệ thống IDS dựa trên mô hình DNN với quá trình xử lý dữ liệu bằng thuật toán K-means và các thuật toán *heuristic* kết hợp với LR đều cho kết quả tương đối tốt, nhưng LR-GA cho kết quả tốt nhất đối với hệ thống IoT tại biên mạng khi chỉ phải quyết định các loại tấn công dựa trên 8 đặc tính mà vẫn đảm bảo độ chính xác (*precision*) lên đến 99%.

# KẾT LUẬN VÀ KIẾN NGHỊ

Sau quá trình làm đề án “*Nghiên cứu hệ thống phát hiện xâm nhập (IDS) thông minh cho IoT tại biên mạng*”, quyển đề án đã trình bày bám sát nội dung của đề cương đã được duyệt bao gồm 3 chương:

* Chương 1: Tổng quan về IoT, điện toán biên và kỹ thuật phát hiện xâm nhập
* Chương 2: Giải pháp phát hiện xâm nhập thông minh cho IoT tại biên mạng
* Chương 3: Đề xuất giải pháp IDS thông minh tại biên mạng

Đề án đã đề xuất một mô hình IDS dựa trên DNN với phương pháp K-means và các thuật toán *heuristic* (GA, PSO, CSO và FAO) kết hợp với LR để phát hiện các cuộc tấn công và đánh giá hiệu suất của chúng. Mô hình đề xuất sử dụng phương pháp K-means phân cụm dữ liệu để giảm kích thước dữ liệu. Các thuật toán *heuristic* được tối ưu hóa bởi LR nhằm trích xuất ra các đặc trưng quan trọng đồng thời sử dụng kỹ thuật trọng số lớp (class-weight) để ngăn chặn các cuộc tấn công mất cân bằng. Các cuộc tấn công được phân biệt dựa vào DNN. Mô hình đề xuất đã được kiểm tra bằng nhiều kích thước mẫu khác nhau trên bộ dữ liệu IoT-23. Kết quả thử nghiệm chứng minh rằng, đối với tất cả các phương pháp độ chính xác phát hiện cuộc tấn công của mô hình được đề xuất đều tốt với độ chính xác *precision*, *recall* và *F1-score* lên đến 99% với số lượng mẫu khác nhau. Ngoài ra, thuật toán GA kết hợp với LR trên mô hình IDS dựa trên DNN để phát hiện tấn công là hiệu quả nhất khi so sánh các thuật toán *heuristic* do số lượng đặc trưng được chọn ra tương đối ít (8 đặc trưng) nhưng vẫn đảm bảo độ chính xác, từ đó giảm thời gian tính toán. Do đó, khả năng ứng dụng của mô hình đề xuất có thể được sử dụng trên các thiết bị biên mạng có nguồn tài nguyên hạn chế, đặc biệt là mô hình sử dụng thuật toán GA, có thể triển khai thực tế trên các thiết bị tiên tiến thực sự trong các hệ thống IoT nông nghiệp.

Các kết quả chính yếu của nghiên cứu liên quan trong đề án đã được công bố 01 bài trên tạp chí Mendel:

[1] **T. T. Thuy**, L. Thuan, N. Duc, and H. Minh, “A Study on Heuristic Algorithms Combined With LR on a DNN-Based IDS Model to Detect IoT Attacks”, *mendel*, vol. 29, no. 1, pp. 62-70, Jun. 2023.

# DANH MỤC TÀI LIỆU THAM KHẢO

Phát hiện bất thường (Anomaly Detection), còn được biết đến là phát hiện điểm ngoại lai hoặc phát hiện dữ liệu không bình thường, là quá trình phân tích dữ liệu nhằm xác định các điểm dữ liệu có các đặc điểm đáng kể khác biệt so với phần lớn dữ liệu còn lại hoặc không tuân theo các quy luật phân phối của dữ liệu bình thường. Các điểm dữ liệu này, thường được gọi là điểm ngoại lai, nhiễu, sai số hoặc ngoại lệ, có thể biểu thị các sự kiện hoặc trạng thái không phù hợp, không thông thường trong hệ thống hoặc quá trình mà dữ liệu đang được thu thập từ. Mục tiêu của phát hiện bất thường là nhận diện và hiểu rõ các điểm này để có thể xử lý kịp thời và đưa ra các biện pháp phòng tránh hoặc cải thiện hệ thống. Khi tiến hành phân tích dữ liệu, việc xử lý các dữ liệu bất thường thường gây ra thách thức trong việc đưa ra nhận định và dự đoán. Trong nhiều trường hợp, những dấu hiệu của sự bất thường có thể chỉ ra một vấn đề nghiêm trọng đang diễn ra, có thể gây ra tổn thất lớn và đòi hỏi nhiều thời gian và công sức để khắc phục. Ví dụ, trong môi trường công nghiệp, đặc biệt là trong các nhà máy sản xuất, các máy móc, như các động cơ điện, thường hoạt động liên tục trong nhiều ngày và có nguy cơ gặp sự cố, gây ra thiệt hại lớn cho công ty. Do đó, việc phát hiện các dấu hiệu bất thường sớm có thể giúp giảm thiểu chi phí và ngăn chặn các sự cố nghiêm trọng hơn. Một ví dụ khác là khi ngân hàng phát hiện sự bất thường trong giao dịch thẻ tín dụng, họ có thể từ chối giao dịch để bảo vệ quyền lợi của khách hàng và duy trì uy tín của mình.