 **BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

------------------------------

A red circle with a star and a book

Description automatically generated

**BÁO CÁO THỰC TẬP**

**TỐT NGHIỆP ĐẠI HỌC**

**Đề tài:**

**“ỨNG DỤNG PHƯƠNG PHÁP HỌC MÁY TRONG BÀI TOÁN NHẬN DẠNG NHỮNG REQUEST BẤT THƯỜNG**

**”**

**Người hướng dẫn : TS. NGUYỄN MINH HẢI**

**Sinh viên thực hiện : VÕ TIẾN ĐẠT**

**Mã số sinh viên : N19DCAT016**

**Lớp : D19CQAT01-N**

**Khóa : 2019 - 2024**

**Hệ : ĐẠI HỌC CHÍNH QUY**

**TP. HCM, tháng ... năm 2023**

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

--------------------------------------



**BÁO CÁO THỰC TẬP TỐT NGHIỆP ĐẠI HỌC**

**Đề tài:**

**ỨNG DỤNG PHƯƠNG PHÁP HỌC MÁY TRONG BÀI TOÁN NHẬN DẠNG NHỮNG REQUEST BẤT THƯỜNG**

**Người hướng dẫn : TS. NGUYỄN MINH HẢI**

**Sinh viên thực hiện : VÕ TIẾN ĐẠT**

**Mã số sinh viên : N19DCAT016**

**Lớp : D19CQAT01-N**

**Khóa : 2019 - 2024**

**Hệ : ĐẠI HỌC CHÍNH QUY**

**TP. HCM, tháng ... năm 2023**

**LỜI CẢM ƠN**

Lời đầu tiên, cho em xin phép được gửi lời tri ân sâu sắc đến các thầy cô trường Học viện Công nghệ Bưu chính Viễn thông cơ sở tại TP.HCM đã tận tình dẫn dắt và truyền đạt cho em rất nhiều kiến thức quý báu trong trong những năm học vừa qua.

Đặc biệt, em xin gửi lời cảm ơn chân thành nhất tới ThS.Nguyễn Minh Hải, thầy đã truyền đạt kiến thức, hướng dẫn em nghiên cứu và thực hành trong suốt quá trình thực hiện đề tài. Em không chỉ tiếp thu thêm được nhiều kiến thức mới mà còn học được tinh thần và thái độ làm việc nghiêm túc từ thầy. Đó sẽ là những hành trang cần thiết cho quá trình làm việc trong tương lai.

Em xin chân thành cảm ơn ban lãnh đạo Công ty TNHH giải pháp công nghệ SGOD đã chấp nhận cho em thực tập trực tiếp tại công ty và tạo điều kiện cho em tham gia các buổi đào tạo, các khóa học trực tuyến và trực tiếp, và áp dụng những kiến thức vào thực tế mà em đã được học được tại trường, cũng như được đào tạo sau đó tại công ty.

Vì đây là lần đầu em được tiếp xúc với môi trường làm việc thực tế tại công ty nên bài làm của em sẽ không tránh được việc có nhiều thiếu sót. Em rất mong nhận được sự thông cảm và đóng góp ý kiến của quý thầy cô và phía công ty cùng toàn thể các bạn.

Sau cùng, em xin chúc quý thầy cô khoa Công nghệ thông tin 2, thầy Nguyễn Minh Hải, và những anh chị ở công ty thực tập đã nhiệt tình giúp đỡ em trong thời gian thực tập vừa qua thật dồi dào sức khỏe để có thể tiếp tục truyền đạt kiến thức cho thế hệ mai sau, cũng như đào tạo ra được nhiều lớp trẻ tài năng cho ngành.

|  |  |
| --- | --- |
|  | TP Hồ Chí Minh, ngày 13 tháng 08 năm 2023  **SINH VIÊN THỰC HIỆN ĐỀ TÀI**  **Võ Tiến Đạt** |

**MỤC LỤC**

[**LỜI CẢM ƠN** i](#_Toc142804087)

[**MỤC LỤC** ii](#_Toc142804088)

[**DANH MỤC THUẬT NGỮ, CHỮ VIẾT TẮT** iv](#_Toc142804089)

[**DANH MỤC HÌNH VẼ** v](#_Toc142804090)

[**LỜI MỞ ĐẦU** 1](#_Toc142804091)

[**CHƯƠNG 1: CƠ SỞ LÝ THUYẾT** 2](#_Toc142804092)

[**1.1.** **Tổng quan về HTTP request** 2](#_Toc142804093)

[**1.1.1.** **Các phương thức của HTTP** 2](#_Toc142804094)

[**1.1.2.** **Cấu trúc của HTTP request** 3](#_Toc142804095)

[**1.2.** **Một số thuật toán máy học** 3](#_Toc142804096)

[**1.2.1.** **Mạng nơ ron nhân tạo** 3](#_Toc142804097)

[**1.2.2.** **Cây quyết định** 5](#_Toc142804098)

[**1.2.3.** **K-mean Cluster** 6](#_Toc142804099)

[**1.3.** **Các dạng phân phối** 7](#_Toc142804100)

[**1.3.1.** **Phân phối đều** 7](#_Toc142804101)

[**1.3.2.** **Phân phối đều liên tục** 8](#_Toc142804102)

[**1.3.3.** **Phân phối poison** 9](#_Toc142804103)

[**1.3.4.** **Phân phối chi bình phương** 10](#_Toc142804104)

[**1.3.5.** **Phân phối Student** 11](#_Toc142804105)

[**CHƯƠNG 2: TÌM HIỂU CÁC LOẠI TẤN CÔNG BẰNG REQUEST** 15](#_Toc142804106)

[**2.1.** **Tấn công từ chối dịch vụ** 15](#_Toc142804107)

[**2.1.1.** **DOS** 15](#_Toc142804108)

[**2.1.2.** **DDOS** 16](#_Toc142804109)

[**2.3.** **SQL Injection** 20](#_Toc142804110)

[**2.3.1.** **Tấn công SQL Injection** 20](#_Toc142804111)

[**2.3.2.** **Cách phòng tránh tấn công SQL Injection.** 22](#_Toc142804112)

[**2.4.** **Cross Site Scripting (XSS)** 23](#_Toc142804113)

[**2.4.1.** **Tấn công XSS** 23](#_Toc142804114)

[**2.4.2.** **Phòng chống** 25](#_Toc142804115)

[**CHƯƠNG 3: THIẾT KẾ MÔ HÌNH VÀ ỨNG DỤNG THUẬT TOÁN HỌC MÁY NHẬN DIỆN BẤT THƯỜNG REQUEST** 27](#_Toc142804116)

[**3.1.** **Giới thiệu chương** 27](#_Toc142804117)

[**3.2.** **Các thuộc tính của một log webserver** 27](#_Toc142804118)

[**3.2.1.** **Giới thiêu về Webserver** 27](#_Toc142804119)

[**3.2.2.** **Các thuộc tính được log trong hệ thống webserver** 28](#_Toc142804120)

[**3.3.** **Thuật toán đề xuất** 29](#_Toc142804121)

[**3.3.1.** **Interquartile Range** 29](#_Toc142804122)

[**3.3.2.** **Isolation Forest** 29](#_Toc142804123)

[**3.3.3.** **Median Absolute Deviation** 30](#_Toc142804124)

[**3.3.4.** **KNN (K- Nearest Neighbor algorithm)** 30](#_Toc142804125)

[**3.3.5.** **HBOS(Histogram Based Outlier Detection)** 30](#_Toc142804126)

[**3.3.6.** **LOF** 31](#_Toc142804127)

[**3.3.7.** **OCSVM** 31](#_Toc142804128)

[**3.3.8.** **Nhận xét chung** 32](#_Toc142804129)

[**3.4.** **Các bước thực hiện** 33](#_Toc142804130)

[**3.4.1.** **Sơ lược về mô hình** 33](#_Toc142804131)

[**3.4.2.** **Xây dựng một webserver** 33](#_Toc142804132)

[**3.4.3.** **Lựa chọn thuật toán** 35](#_Toc142804133)

[**KẾT LUẬN** 37](#_Toc142804134)

[**DANH MỤC TÀI LIỆU THAM KHẢO** 38](#_Toc142804135)

# 

# **DANH MỤC THUẬT NGỮ, CHỮ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **Viết tắt** | **Tiếng Anh** | **Tiếng Việt** |
| DDoS | Distributed Denial of Service | Tấn công từ chối dịch vụ |
| DOS | Denial of Service | Từ chối dịch vụ |
| HBOS | Histogram Based Outlier Detection |  |
| KNN | K-Nearest Neighbors |  |
| IF | Isolation Forest |  |
| MAD | Median Absolute Deviation |  |
| LOF | Local Outlier Factor |  |
| OCSVM | One-Class Support Vector Machine |  |
| IR | Interquatile Range | Khoảng phân vị |

# 

# **DANH MỤC HÌNH VẼ**

[*Hình 1. 1: Phân cụm K-means 14*](#_Toc142035370)

[*Hình 1. 2: Đồ thị phân phối đều 15*](#_Toc142035371)

[*Hình 1. 3: Đồ thị phân phối đều liên tục 16*](#_Toc142035372)

[*Hình 1. 4: Đồ thị poission 17*](#_Toc142035373)

[*Hình 1. 5: Đồ thị phân phối chi bình phương 18*](#_Toc142035374)

[*Hình 1. 6: Phân phối student 19*](#_Toc142035375)

[*Hình 1. 7: Đồ thị phân phối chuẩn 20*](#_Toc142035376)

[*Hình 2. 1: Sơ đồ phân loại các kiểu tấn công DDoS 23*](#_Toc142035377)

[*Hình 2. 2: Kiến trúc attack kiểu Agent – Handler 23*](#_Toc142035378)

[*Hình 2. 3: Kiến trúc attack network kiểu IRC-based 24*](#_Toc142035379)

[*Hình 2. 4: Phân loại kiểu tấn công DDoS 25*](#_Toc142035380)

[*Hình 3. 1: Phát hiện các điểm bất thường dựa trên Interquartile Range 32*](#_Toc142035381)

[*Hình 3. 2: Phát hiện các điểm bất thường dựa trên Isolation Forest 33*](#_Toc142035382)

[*Hình 3. 3: Phát hiện những điểm bật thường dựa trên thuật toán Median Absolute Deviatation 33*](#_Toc142035383)

[*Hình 3. 4: Phát hiện bất thường dựa trên thuật toán KNN 34*](#_Toc142035384)

[*Hình 3. 5: Phát hiện bất thường dựa trên HBOS 34*](#_Toc142035385)

[*Hình 3. 6: Phát hiện bất thường dựa trên thuật toán LOF 35*](#_Toc142035386)

[*Hình 3. 7: Đồ thị áp dụng thuật toán OCSVM 35*](#_Toc142035387)

[*Hình 3. 8: Mô hình sơ lược về quá trình chạy 36*](#_Toc142035388)

[*Hình 3. 9: File docker compose để dựng webserver 37*](#_Toc142035389)

[*Hình 3. 10: Kết quả sau khi chạy lệnh 37*](#_Toc142035390)

[*Hình 3. 11: hiển thị webserver 38*](#_Toc142035391)

[*Hình 3. 12: Kết quả sau khi chạy lệnh 38*](#_Toc142035392)

[*Hình 3. 13: Code triển khai nhận biết những ngày có số lượng request bất thường 39*](#_Toc142035393)

# **LỜI MỞ ĐẦU**

1. **Tính cấp thiết của đề tài**

Với thời đâị công nghiệp 4.0 hiện nay, cùng với sự bùng nổ về Trí tuệ nhân tạo đã ngày càng ứng dụng ở nhiều lĩnh vực trong đời sống. Việc quản lý và đảm bảo internet dẫn đến hạ tầng, công nghệ thông tin càng ngày càng lớn và ngày nay càng có nhiều cuộc tấn công bằng nhiều hình thức khác nhau nhằm phá hoại hệ thống và gây lỗi thiết bị về lâu dài sẽ ảnh hưởng đến sự an toàn và hiệu năng của hệ thống. Với việc ngày nay nhiều hệ thống đã có xây dựng những hệ thống quản lý log và phân tích dữ liệu để có thể nhận diện các cuộc tấn công cũng như nhận diện bất thường. Đó cũng là lý do em chọn đề tài “ỨNG DỤNG HỌC MÁY VÀO NHẬN DẠNG REQUEST BẤT THƯỜNG” để có thể phát triển dự báo hoặc cảnh báo sớm những sự cố bất thường trong những hệ thống hiên nay.

1. **Mục đích nghiên cứu**

Mục tiêu chính: Dựa vào dữ liệu log từ webserver được những log nào bình thường, log nào bất thường, áp dụng mô hình máy học để nhận biết

Từ mục tiêu chính trên, kết quả của của bài đồ án thực tập đạt được như sau:

* Tìm hiểu tổng quan về các đặc điểm của request.
* Tìm hiểu về một số thuật toán học máy ứng dụng vào nhận dạng các request bất thường
* Tìm hiều về các dạng phân phối

1. **Đối tượng và phạm vi nghiên cứu**

**Đối tượng nghiên cứu**

* Tìm hiểu về log của webserver
* Tìm hiểu về số lượng request cao bất thường

**Phạm vi nghiên cứu**

* Xây dựng webserver để log lại các request
* Phân tích dữ liệu của những request và tìm ra những đặc trưng
* Tìm hiều những thuật toán để áp dụng nhận diện các request bất thường

1. **Phương pháp nghiên cứu**

Phương pháp luận: Dựa trên lý thuyết

Phương pháp đánh giá bằng mô phỏng thực nghiệm:Xây dựng mô hình mô phỏng và nhận biết bất thường

# 

# **CHƯƠNG 1: CƠ SỞ LÝ THUYẾT**

* 1. **Tổng quan về HTTP request**

HTTP request là thông tin được gửi từ client lên server, để yêu cầu server tìm hoặc xử lý một số thông tin, dữ liệu mà client muốn. HTTP request có thể là một file text dưới dạng XML hoặc Json mà cả hai đều có thể hiểu được

* + 1. **Các phương thức của HTTP**
       1. Phương thức GET

GET là phương thức được client gửi dữ liệu lên server thông qua đường đẫn URL nằm trên thanh địa chỉ Brower. Server sẽ nhận đường dẫn đó và phân tích trả về kết quả cho bạn. Hơn nữa, nó là một phương thức được sử dụng phổ biến mà không cần request body

Ví dụ khi bạn mở trang web lên, client sẽ gửi một phương thức GET lên server để truy xuất nội dụng hiển thị của trang web. Nó tương đương với thao tác đọc.

Một số đặc điểm chính của phương thức GET là:

* Giới hạn độ dài của các giá trị là 255 kí tự
* Chỉ hỗ trợ các dữ liệu kiểu dữ liệu String
* Có thể lưu vào bộ nhớ cache
* Một số tham số truyền vào được lưu trữ trong lịch sử trình duyệt
* Có thể được bookmark ( đánh dấu rồi xem lại) do được lưu trong lịch sử trình duyệt
  + - 1. Phương thức POST

Phương thức POST là phương thức gửi dữ liệu đến server giúp bạn có thể thêm mới dữ liệu hoặc cập nhật dữ liệu đã có vào có sở dữ liệu

Chúng ta sẽ cần những thông tin cần thêm hoặc cập nhật trong phần body request.

Một số đặc điểm chính của POST là:

* Dữ liệu cần thêm hoặc cập nhật không được hiển thị trong URL của trình duyệt.
* Dữ liệu không được lưu trong lịch sử trình duyệt.
* Không có hạn chế về độ dài dữ liệu
* Hỗ trợ nhiều kiểu dữ liệu như: String, binary, integers.
  + - 1. Phương thức PUT

Cách hoạt động cũng tương tự như POST nhưng nó chỉ được sử dụng để cập nhật dữ liệu đã có trong database. Khi sử dụng nó, bạn phải sửa toàn bộ dữ liệu của một đối tượng.

* + - 1. Phương thức PATCH

Phương thức PATCH là được sử dụng khi bạn cần cập nhật một phần của tài nguyên, chẳng hạn như một trường hay một thuộc tính. Nó không được sử dụng khi bạn cần cập nhật một bộ tài nguyên, vì điều này có thể được thực hiện bằng phương thức PUT

Để sử dụng phương thức PATCH, bàn cần chỉ định phương thức PATCH trong yêu cầu HTTP của mình và sau đó là nội dung của cập nhật. Nội dung của cập nhật có thể được cung cấp dưới dạng JSON, XML hoặc bất kỳ định dạng nào khác mà máy chủ có thể hiểu

Phương thức PATCH là một phương thức hưu ích cho việc cập nhật, tài nguyên một cách linh hoạt và hiệu quả

* + - 1. Phương thức DELETE
      2. Phương thức HEAD
    1. **Cấu trúc của HTTP request**

Một HTTP Request bao gồm:

* Request line
* Body Request

Một ví dụ về HTTP request đơn giản:

* + - 1. Request Line

Request line là dòng đầu tiên trong HTTP Request. Nó bao gồm 3 phần:

* Phương thức HTTP được sử dụng
* URI giúp xác định các tài nguyên mà client yêu cầu
* Phiên bản của giao thức HTTP

Một request line sẽ có định dạng như sau: GET/ BookStore/v1/Books HTTP/1.1

* + - 1. Request Header

Request Header giúp client có thể gửi yêu cầu lên server. Mỗi yêu cầu sẽ kèm theo các thông số và thông số đó được gọi là Header Parameters. Trình duyệt và server sẽ dựa vào thông số của header nay để trả dữ liệu và hiển thị dữ liệu cho phù hợp

Các thông số mà các bạn có thể gặp thường xuyên như:

User-Agent: cho phép server xác định ứng dụng, hệ thống điều hành, nhà cung cấp và phiên bản.

Connection: kiểm soát kết nối mạng. Nói cách khác, cho phép dừng hoặc tiếp tục kết nối sau khi server thực hiện xong yêu cầu.

Cache-Control: chỉ định chính sách bộ nhớ đệm cảu trình duyệt.

Accept-Language: cho biết tất cả các ngôn ngữ (tự nhiên) mà client có thể được.

* + - 1. Request Body

Cho phép client gửi đến yêu cầu bổ sung cần server thực hiện như: tạo mới hoặc cập nhật dữ liệu mà không thể truyền trên Header Parameters.

Request body thường được sử dụng trong phương thức POST, PUT, PATCH.

* 1. **Một số thuật toán máy học**
     1. **Mạng nơ ron nhân tạo**

Một mạng lưới thần kinh sinh học bao gồm một nhóm các tế bào thần kinh liên kết về mặt hóa học hoặc chức năng. Một nơ-ron duy nhất có thể được kết nối với nhiều nơ-ron khác và tổng số nơ-ron và kết nối trong một mạng có thể lớn. Các kết nối, được gọi là khớp thần kinh, thường được hình thành từ sợi trục đến đuôi gai, mặc dù có thể có các khớp thần kinh đuôi gai và các kết nối khác [9]. Ngoài tín hiệu điện, có những hình thức tín hiệu khác phát sinh từ sự khuếch tán chất dẫn truyền thần kinh. Trí tuệ nhân tạo, mô hình nhận thức và mạng nơ-ron là những mô hình xử lý thông tin được lấy cảm hứng từ cách hệ thống thần kinh sinh học xử lý dữ liệu. Mạng nơ-ron là một chuỗi các thuật toán cố gắng nhận ra các mối quan hệ cơ bản trong một tập hợp dữ liệu thông qua một quá trình bắt chước cách bộ não con người hoạt động. Theo nghĩa này, mạng nơ-ron đề cập đến hệ thống nơ-ron, có thể là hữu cơ hoặc nhân tạo trong tự nhiên. Mạng nơron có thể thích ứng với việc thay đổi đầu vào; để mạng tạo ra kết quả tốt nhất có thể mà không cần thiết kế lại các tiêu chí đầu ra. Khái niệm về mạng nơ-ron, có nguồn gốc từ trí tuệ nhân tạo, đang nhanh chóng trở nên phổ biến trong sự phát triển của các hệ thống giao dịch. Trí tuệ nhân tạo và mô hình nhận thức cố gắng mô phỏng một số đặc tính của mạng nơ-ron sinh học. Trong lĩnh vực trí tuệ nhân tạo, mạng nơ-ron nhân tạo đã được ứng dụng thành công để nhận dạng giọng nói, phân tích hình ảnh và điều khiển thích ứng, nhằm tạo ra các tác nhân phần mềm (trong máy tính và trò chơi điện tử) hoặc robot tự động. Trong lịch sử, máy tính kỹ thuật số phát triển từ mô hình von Neumann và hoạt động thông qua việc thực hiện các lệnh rõ ràng thông qua quyền truy cập vào bộ nhớ của một số bộ xử lý. Mặt khác, nguồn gốc của mạng nơ-ron dựa trên những nỗ lực lập mô hình xử lý thông tin trong các hệ thống sinh học. Không giống như mô hình von Neumann, tính toán mạng nơ-ron không tách biệt bộ nhớ và xử lý. Lý thuyết mạng lưới thần kinh vừa giúp xác định rõ hơn cách thức hoạt động của các tế bào thần kinh trong não vừa cung cấp cơ sở cho những nỗ lực tạo ra trí thông minh nhân tạo. Mạng nơron (NN), trong trường hợp các nơron nhân tạo được gọi là mạng nơron nhân tạo (ANN) hoặc mạng nơron mô phỏng (SNN), là một nhóm các nơron tự nhiên hoặc nhân tạo được kết nối với nhau sử dụng mô hình toán học hoặc tính toán để xử lý thông tin dựa trên cách tiếp cận liên kết để tính toán. Trong hầu hết các trường hợp, ANN là một hệ thống thích ứng thay đổi cấu trúc của nó dựa trên thông tin bên ngoài hoặc nội bộ truyền qua mạng. Nói một cách thực tế hơn, mạng nơ-ron là công cụ ra quyết định hoặc mô hình dữ liệu thống kê phi tuyến tính. Chúng có thể được sử dụng để mô hình hóa các mối quan hệ phức tạp giữa đầu vào và đầu ra hoặc để tìm các mẫu trong dữ liệu. Mạng nơ-ron nhân tạo bao gồm một mạng lưới các phần tử xử lý đơn giản (nơ-ron nhân tạo) có thể thể hiện hành vi toàn cục phức tạp, được xác định bởi các kết nối giữa các phần tử xử lý và các tham số phần tử. Tế bào thần kinh nhân tạo lần đầu tiên được đề xuất vào năm 1943 bởi Warren McCulloch, một nhà sinh lý học thần kinh và Walter Pitts, một nhà logic học, người lần đầu tiên cộng tác tại Đại học Chicago. Một loại mạng nơ-ron nhân tạo cổ điển là mạng Hopfield tái diễn. Khái niệm mạng nơ-ron dường như lần đầu tiên được Alan Turing đề xuất trong bài báo Intelligent Machinery năm 1948. Tiện ích của các mô hình mạng nơron nhân tạo nằm ở chỗ chúng có thể được sử dụng để suy ra một hàm từ các quan sát và cũng có thể sử dụng nó. Mạng nơ-ron không giám sát cũng có thể được sử dụng để học các biểu diễn của đầu vào nắm bắt các đặc điểm nổi bật của phân phối đầu vào. Mạng nơ-ron, trong thế giới tài chính, hỗ trợ phát triển các quy trình như dự báo chuỗi thời gian, giao dịch theo thuật toán, phân loại chứng khoán, mô hình hóa rủi ro tín dụng và xây dựng các chỉ số độc quyền và các công cụ phái sinh giá cả. Mạng thần kinh hoạt động tương tự như mạng thần kinh của não người. Một "nơ-ron" trong mạng nơ-ron là một hàm toán học thu thập và phân loại thông tin theo một kiến trúc cụ thể. Mạng này có sự tương đồng mạnh mẽ với các phương pháp thống kê như phân tích đường cong và phân tích hồi quy. Mạng nơ-ron được sử dụng rộng rãi, với các ứng dụng cho hoạt động tài chính, lập kế hoạch doanh nghiệp, giao dịch, phân tích kinh doanh và bảo trì sản phẩm. Mạng nơ-ron cũng đã được áp dụng rộng rãi trong các ứng dụng kinh doanh như các giải pháp nghiên cứu tiếp thị và dự báo, phát hiện gian lận và đánh giá rủi ro. Mạng nơ-ron đánh giá dữ liệu giá cả và tìm ra cơ hội để đưa ra quyết định thương mại dựa trên phân tích dữ liệu. Các mạng có thể phân biệt sự phụ thuộc lẫn nhau phi tuyến tinh vi và các mẫu mà các phương pháp phân tích kỹ thuật khác không làm được. Theo nghiên cứu, độ chính xác của mạng nơ-ron trong việc đưa ra dự đoán giá cổ phiếu là khác nhau. Một số mô hình dự đoán giá cổ phiếu chính xác từ 50 đến 60 phần trăm trong khi những mô hình khác dự đoán chính xác 70 phần trăm trong tất cả các trường hợp. Một số người đã cho rằng cải thiện 10% hiệu quả là tất cả những gì nhà đầu tư có thể yêu cầu từ mạng nơ-ron. Sẽ luôn có các tập dữ liệu và các lớp nhiệm vụ được phân tích tốt hơn bằng cách sử dụng các thuật toán đã phát triển trước đó. Thuật toán không quá quan trọng; chính dữ liệu đầu vào được chuẩn bị kỹ lưỡng về chỉ số được nhắm mục tiêu sẽ quyết định cuối cùng mức độ thành công của mạng nơ-ron.

* + 1. **Cây quyết định**

Cây quyết định là một công cụ hỗ trợ quyết định sử dụng mô hình quyết định dạng cây và các hệ quả có thể xảy ra của chúng, bao gồm cả kết quả sự kiện may rủi, chi phí tài nguyên và tiện ích. Đó là một cách để hiển thị một thuật toán chỉ chứa các câu lệnh điều khiển có điều kiện. Cây quyết định thường được sử dụng trong nghiên cứu hoạt động, đặc biệt là trong phân tích quyết định, để giúp xác định chiến lược có nhiều khả năng đạt được mục tiêu nhất, nhưng cũng là một công cụ phổ biến trong học máy. [10]

Cây quyết định là một cấu trúc giống như lưu đồ, trong đó mỗi nút bên trong đại diện cho một "thử nghiệm" trên một thuộc tính (ví dụ: lật xu xảy ra trước), mỗi nhánh biểu thị kết quả. kết quả của bài kiểm tra và mỗi lá đại diện cho một lớp nhãn (quyết định được đưa ra sau khi tính toán tất cả các thuộc tính). Các đường dẫn từ gốc để biểu diễn kiểu luật phân loại.

Trong phân tích quyết định, cây quyết định và sơ đồ ảnh hưởng có liên quan chặt chẽ được sử dụng như một công cụ hỗ trợ ra quyết định trực quan và phân tích, nơi các giá trị kỳ vọng (hoặc tiện ích kỳ vọng) của các lựa chọn thay thế cạnh tranh được tính toán. Một cây quyết định bao gồm ba loại nút

* Các nút quyết định - thường được biểu diễn bằng hình vuông
* Các nút cơ hội - thường được biểu thị bằng các vòng tròn
* Các nút kết thúc - thường được biểu diễn bằng hình tam giác

Một cây có thể được “học” bằng cách tách tập nguồn thành các tập con dựa trên kiểm tra giá trị thuộc tính. Quá trình này được lặp lại trên mỗi tập con dẫn xuất theo cách đệ quy được gọi là phân vùng đệ quy. Quá trình đệ quy được hoàn thành khi tất cả các tập con tại một nút đều có cùng giá trị của biến mục tiêu hoặc khi việc tách không còn thêm giá trị vào các dự đoán. Việc xây dựng bộ phân loại cây quyết định không yêu cầu bất kỳ kiến thức miền hoặc thiết lập tham số nào, và do đó thích hợp cho việc khám phá kiến thức khám phá. Cây quyết định có thể xử lý dữ liệu chiều cao. Nhìn chung bộ phân loại cây quyết định có độ chính xác tốt. Quy nạp cây quyết định là một cách tiếp cận quy nạp điển hình để tìm hiểu kiến thức về phân loại. Cây quyết định phân loại các cá thể bằng cách sắp xếp chúng theo cây từ gốc đến một số nút lá, điều này cung cấp sự phân loại của cá thể. Một thể hiện được phân loại bằng cách bắt đầu từ nút gốc của cây, kiểm tra thuộc tính được chỉ định bởi nút này, sau đó di chuyển xuống nhánh cây tương ứng với giá trị của thuộc tính như trong hình trên. Quá trình này sau đó được lặp lại đối với cây con bắt nguồn từ nút mới.

Điểm mạnh của phương pháp cây quyết định là:

* Cây quyết định có thể tạo ra các quy tắc dễ hiểu.
* Cây quyết định thực hiện phân loại mà không cần tính toán nhiều.
* Cây quyết định có thể xử lý cả biến liên tục và biến phân loại.
* Cây quyết định cung cấp một dấu hiệu rõ ràng về các trường nào là quan trọng nhất để dự đoán hoặc phân loại. Điểm yếu của phương pháp cây quyết định:
* Cây quyết định ít thích hợp hơn cho các nhiệm vụ ước tính trong đó mục tiêu là dự đoán giá trị của một thuộc tính liên tục.
* Cây quyết định dễ mắc lỗi trong các bài toán phân loại với nhiều lớp và số lượng ví dụ huấn luyện tương đối nhỏ
* Cây quyết định có thể tốn kém về mặt tính toán để đào tạo. Quá trình trồng cây quyết định rất tốn kém về mặt tính toán. Tại mỗi nút, mỗi trường phân tách ứng cử viên phải được sắp xếp trước khi có thể tìm thấy trường phân tách tốt nhất của nó. Trong một số thuật toán, kết hợp các trường được sử dụng và phải thực hiện tìm kiếm để có trọng số kết hợp tối ưu. Các thuật toán cắt tỉa cũng có thể tốn kém vì nhiều cây con ứng cử viên phải được hình thành và so sánh.
  + 1. **K-mean Cluster**

K-Means Clutering là một thuật toán học không giám sát đơn giản và phổ biến được sử dụng để giải quyết các vấn đề phân cụm trong học máy hoặc khoa học dữ liệu. Thông thường, các thuật toán không giám sát đưa ra suy luận từ tập dữ liệu chỉ sử dụng các vector đầu vào mà không đề cập đến các kết quả đã biết hoặc được gán nhãn

Mục tiêu của K-means rất đơn giản: nhóm các điểm dữ liệu tương tự lại với nhau. Để đạt mục tiêu này, K-means tìm kiếm một số lượng cố định (k) các cụm trong một tập dữ liệu, nhóm các tập dữ liệu không được gắn thành các cụm khác nahu. Ở đây K là số lượng cụm được xác dịnh cần được tạo trong quá trình này, nếu K = 2, sẽ có hai cụm, và đối với K =3, có ba cụm

Cum được đề cập đến một tập hợp các điểm dữ liệu được tổng hợp lại với nhau vì có những điểm tương đồng nhất đinh. Centroid là vị trí đại diện cho trung tâm của cụm. Mọi điểm dữ liệu được phân bổ cho từng cụm với yêu cầu là tổng khoảng cách giữa điểm dữ liệu và cụm tương ứng của chúng là nhỏ nhất. Nói cách khác, thuật toán K-means xác định k số centroid và sau đó phân bổ mọi điểm dữ liệu cho cụm gần nhất, đồng thời giữ các centroid càng nhỏ càng tốt. Ý nghĩa trông K-means đề cập giá trị trung bình của dữ liệu; túc là tìm ra điểm trung tâm.

A diagram of a diagram of a number of dots

Description automatically generated

Hình 1. 1: Phân cụm K-means

Để sử dụng dữ liệu huấn luyện, quá trình K-means trong Khai phá dữ liệu bắt đầu với nhóm đầu tiên bao gồm các ngẫu nhiên trung tâm được chọn, được sử dụng làm điểm bắt đầu cho tất cả các cụm và sau đó thực hiện các phép tính lặp đi lặp lại để tối ưu hóa vị trí của các trung tâm.

Thuật toán tạm dừng tạo và tối ưu hóa các cụm khi:

* Các centroid đã ổn định không có thay đổi về giá trị của chúng vì việc phân nhóm đã thành công.
* Đã đạt được số lần lặp xác định.

Hoạt động của thuật toán K-Means được giải thích theo các bước dưới đây

* Bước 1: Chọn số K để quyết định số lượng cụm.
* Bước 2: Chọn K điểm hoặc trọng tâm ngẫu nhiên. (Nó có thể khác với tập dữ liệu đầu vào).
* Bước 3: Gán mỗi điểm dữ liệu cho trung tâm gần nhất của chúng, sẽ tạo thành các cụm K được xác định trước.
* Bước 4: Tính toán phương sai và đặt một trung tâm mới của mỗi cụm.
* Bước 5: Lặp lại các bước thứ ba, có nghĩa là chỉ định lại mỗi điểm dữ liệu cho trung tâm gần nhất mới của mỗi cụm.
* Bước 6: Nếu có bất kỳ sự phân công lại nào xảy ra, hãy chuyển sang bước 4, sau đó chuyển đến hoàn tất.
* Bước 7: Mô hình đã sẵn sàng

Hiệu suất của thuật toán phân cụm K-mean phụ thuộc vào các cụm hiệu quả cao mà nó tạo thành. Nhưng chọn số lượng cụm tối ưu là một nhiệm vụ lớn. Có một số cách khác nhau để tìm số lượng cụm tối ưu, phương pháp thích hợp nhất để tìm số lượng cụm hoặc giá trị của K. Phương pháp được đưa ra dưới đây:

* 1. **Các dạng phân phối** 
     1. **Phân phối đều**

Tổng quát, biến ngẫu nhiên phân phối đều rời rạc X có thể nhận giá trị trong một tập vô hạn, nhưng ở đây ta chỉ đề cập đến trường hợp X nhận hữu hạn giá trị nguyên dương từ 1 đến n. Mỗi giá trị có cùng xác suất xảy ra là  . Kí hiệu  với (a,b) là khoảng giá trị của X.

* Hàm xác suất
* Trung bình
* Phương sai
* Hàm sinh moment    

A screenshot of a graph

Description automatically generated

Hình 1. 2: Đồ thị phân phối đều

* + 1. **Phân phối đều liên tục**

Biến ngẫu nhiên có phân phối đều liên tục nhận giá trị trên đoạn [a,b]. Xác suất X nhận bất kì giá trị nào thuộc khoảng (a,b) đều bằng . Kí hiệu X có phân phối đều trên khoảng (a,b) là 

|  |  |
| --- | --- |
| * Hàm mật độ |  |
| * Trung bình |  |
| * Phương sai |  |
| * Hàm sinh moment |  |

A screenshot of a graph

Description automatically generated

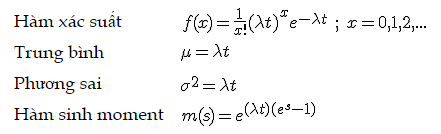
**Hình 3: Đồ thị phân phối đều liên tục**

Hình 1. 3: Đồ thị phân phối đều liên tục

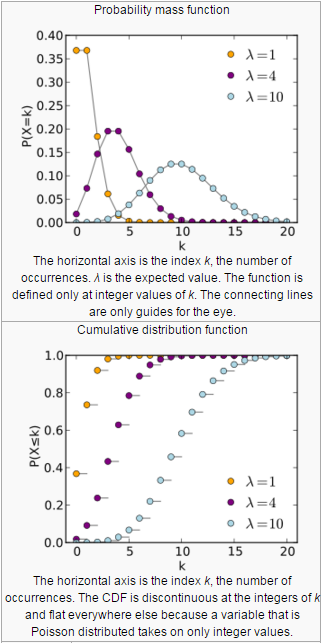
* + 1. **Phân phối poison**

Trong lý thuyết xác suất và thống kê, phân phối Poisson là một phân phối xác suất rời rạc. Nó khác với các phân phối xác suất rời rạc khác ở chỗ thông tin biết không phải là xác suất để một sự kiện xảy ra thành công trong một lần thử như trong phân phối Bernoulli hay là số lần mà sự kiện đó xảy ra trong n lần thử như trong phân phối nhị thức, mà chính là trung bình số lần xảy ra thành công của một sự kiện trong một khoảng thời gian nhất định. Gía trị trung bình này được gọi là lambda, kí hiệu là . Phân phối Poisson còn được dùng cho khoảng mà đơn vị khác thời gian như: khoảng cách, diện tích hay thể tích. Một ví dụ cổ điển là sự phân rã hạt nhân của các nguyên tử.

Khi những sự kiện xảy ra một cách ngẫu nhiên đều đặn với tỷ lệ là   sự kiện trên một đơn vị thời gian, khi đó biến ngẫu nhiên X đếm số sự kiện xảy ra trong khoảng thời gian độ dài t, có phân phối Poisson.



Hình 1.4: công thức của poison

H

Hình 1. 4: Đồ thị poission

* + 1. **Phân phối chi bình phương**

Trong lý thuyết xác suất thống kê, phân phối Chi-bình phương ()  với k bậc tự do (degrees of freedom, df) là phân phối của tổng k bình phương biến ngẫu nhiên có phân phối chuẩn hóa độc lập và là một trong những phân phối thường được sử dụng trong thống kê suy luận như kiểm định giả thuyết thống kê hay xây dựng khoảng tin cậy.

|  |  |
| --- | --- |
| * Hàm mật độ | A black background with white dots  Description automatically generated   với |
| * Trung bình |  |
| * Phương sai |  |
| * Hàm sinh moment |  |

A graph of different colored lines

Description automatically generated

Hình 1. 5: Đồ thị phân phối chi bình phương

* + 1. **Phân phối Student**

Nếu  và  và độc lập với  thì .  Phân phối Student có hình dạng đối xứng hình chuông giống với phân phối chuẩn nhưng phần đuôi của phân phối Student "nặng" hơn phân phối chuẩn, nghĩa là nó có nhiều giá trị phân bố xa giá trị trung bình hơn phân phối chuẩn.  Trong khi phân phối chuẩn mô tả toàn bộ tổng thể, phân phối Student mô tả mẫu được lấy từ một tổng thể đầy đủ, do đó với mỗi cỡ mẫu khác nhau ta sẽ có phân phối Student cho mẫu khác nhau và khi cỡ mẫu càng lớn thì phân phối Student sẽ càng giống với phân phối chuẩn.

|  |  |
| --- | --- |
| * Hàm mật độ | A black background with white dots  Description automatically generated |
| * Trung bình |  |
| * Phương sai | khi |

A screenshot of a graph

Description automatically generated

Hình 1. 6: Phân phối student

* + 1. **Phân phối chuẩn**

Phân phối chuẩn, còn gọi là phân phối Gauss, là một phân phối xác suất cực kì quan trọng trong nhiều lĩnh vực. Nó là họ phân phối có dạng tổng quát giống nhau, chỉ khác tham số vị trí (giá trị trung bình μ) và tỉ lệ (phương sai σ2).

Phân phối chuẩn chuẩn hóa (standard normal distribution) là phân phối chuẩn với giá trị trung bình bằng 0 và phương sai bằng 1 (đường cong màu đỏ trong hình bên phải). Phân phối chuẩn còn được gọi là đường cong chuông (bell curve) vì đồ thị của mật độ xác suất có dạng chuông.

|  |  |
| --- | --- |
| * Hàm mật độ |  |
| * Trung bình |  |
| * Phương sai |  |
| * Hàm sinh moment |  |

A screenshot of a graph

Description automatically generated

Hình 1. 7: Đồ thị phân phối chuẩn

# 

# **CHƯƠNG 2: TÌM HIỂU CÁC LOẠI TẤN CÔNG BẰNG REQUEST**

* 1. **Tấn công từ chối dịch vụ**
     1. **DOS**
        1. Giới thiệu khái quát về DoS
* Dos có thể mô tả như hành động ngăn cản những người dùng hợp pháp của một dịch vụ nào đó truy cập và sử dụng dịch vụ. Nó bao gồm cả việc làm tràn ngập mạng, làm mất kết nối với dịch vụ,… mà mục đích cuối cùng là làm cho server không thể đáp ứng các yêu cầu dịch vụ từ các client. DoS có thể làm ngưng hoạt động của một máy tính, một mạng nội bộ, thậm chí cả một hệ thống mạng rất lớn. Thực chấp DoS là kẻ tấn công chiếm dụng một lượng lớn tài nguyên mạng như băng thông, bộ nhớ… và làm mất khả năng xử lý các yêu cầu dịch vụ từ các client khác.
  + - 1. Các cách thức tấn công

Phá hoại dựa trên tính giới hạn hoặc không thể phục hồi của tài nguyên mạng

* Thông qua kết nối:

Tấn công SYN flood: Lợi dụng các thức hoạt động của kết nối TCP/ IP, hacker bắt đầu quá trình thiết lập một kết nối TPC/IP với mục tiêu muốn tấn công nhưng sẽ phá vớ kết nối ngay sau khi quá trình SYN, SYN ACK hoàn tất, khiến cho mục tiêu rơi vào trạng thái chờ đợi ( đợi gói tin ACK từ phía yêu cầu thiết lập kết nối) và liên tục gửi gói tin SYN ACK để thiết lập kết nối. Một cách khác là giả mạo địa chỉ IP nguồn và các gói tin yêu cầu thiết lập kết nối SYN và cũng như trường hợp trên, máy tính đích cũng rơi vào trạng thái chờ vì các gói tin SYN ACK không thể đi đến đích do địa chỉ IP nguồn và không có thật. Cách thức này có thể được các hacker áp dụng đẻ taansc ông một hệ thống mạng có băng thông lớn hơn hệ thống của hacker.

* Lợi dụng tài nguyên của nạn nhân
* Sử dụng băng thông
* Sử dụng các nguồn tài nguyên khác
* Tấn công kiểu Smurf Attack
* Tấn công kiểu Tesar Drop:
* Phá hoại hoặc chỉnh sửa thông tin cấu hình
  + - 1. Các cách phòng chống
* Dos có thể làm tiêu tốn rất nhiều thời gian cũng như tiền bạc, vì vậy, cần phải có những biến pháp phòng chống
* Mô hình hệ thống được xây dựng hợp lý, tránh phụ thuộc lẫn nhau quá mức dễ dấn đến một bộ phận gặp sự cố làm cả hệ thống trục trặc.
* Thiết lập password bảo vệ các thiệt bị hay các nguồn tài nguyên quan trọng.
* Thiết lập các mức xác thực đối với người dùng cũng như các nguồn tin trên mang ( các thông tin cập nhật định tuyến giữa các router cũng nên thiết lập ở chế độ xác thực)
* Xây dựng hệ thống lọc thông tin trên router, firewall,.. và hệ thống bảo vệ chống lại SYN flood.
* Chỉ chấp nhận các dịch vụ cần thiết, tạm thời dùng các dịch vụ chưa có yêu cầu cung cấp hoặc không sử dụng.
* Xây dựng hệ thống định múc, giới hạn cho người sử dụng ngan ngửa trường hợp người dùng có ác ý muốn lợi dụng các tài nguyên trên server để tấn công chính server hay mạng, server khác.
* Liên tục cập nhật, nghiên cứu, kiểm tra để phát hiện các lỗ hổng bảo mật và có biện pháp khắc phục kịp thời.
* Sử dụng các biện pháp kiểm tra hoạt động của hệ thống một cách liên tục để phát hiện những hành động bất thường.
* Xây dựng hệ thống dự phòng.
  + 1. **DDOS**
* DDoS là kỹ thuật tấn công làm các ISP lo âu, giới hacker chính thống không công nhận DDoS là kỹ thuật tấn công chính thống. Thế nhưng Black hat đang có rất nhiều ưu thế khi triển khai khi tấn công bằng kỹ thuật DDoS. Việc phòng ngừa và ngăn chặn DDoS vẫn còn đang thực hiện ở mực độ khắc phục hậu quả và truy tìm thủ phạm.
  + - 1. **Các giai đoạn của một cuộc tấn công kiểu DDoS**

1. **Giai đoạn chuẩn bị**

* Chuẩn bị là công cụ quan trọng của các cuộc tấn công, công cụ này thông thường hoạt động theo mồ hình client-server. Hacker có thể viết phần mềm này hay download một cách dễ dàng, theo thống kê tạm thời có khoảng hơn 10 cộng cụ DDoS được cung cấp miễn phí trong mạng ( các công cụ này sẽ phân tích chi tiết các phần sau)
* Kế tiếp, dùng các kỹ thuật hack khác để nắm trọn quyền một số host trên mạng tiến hành cài đặt các software cần thiết trên các host này, việc cấu hình và thử nghiệm toàn bộ attack-network ( bao gồm mạng lưới các máy đã bị lợi dụng cùng với các software đã được thiết lập trên đó, máy của hacker hoặc một số máy khác đã được thiết lập như điểm phát động tấn công) cũng sẽ được thời điểm trong giai đoạn này.

1. **Giai đoạn xác định mục tiêu và thời điểm**

* Sau khi xác định mục tiêu lần cuối, hacker sẽ có hoạt động điều chỉnh attack network chuyển hướng tấn công về mục tiêu.
* Yếu tố thời điểm quyết định mức độ thiệt hại và tốc độ đáp ứng của mục tiêu đối với cuộc tấn công.

1. **Phát động tấn công và xóa dấu vết**

* Đúng thời điểm đã định, hacker phát động tấn công từ máy của mình, lệnh tấn công này có thể đi qua nhiều cấp mới đến host thực sự tấn công. Toàn attack-network ( có thể lên đến hàng ngàn máy), sẽ vắt cạn năng lực của server mục tiêu liên tục, ngăn chặn không cho nó hoạt động như thiết kế.
* Sau một khoảng thời gian tấn công thích hợp, hacker tiến hành xóa mọi dấu vết có thể truy ngược đến mình, việc này đòi hỏi trình khá cao và không tuyệt đối cần thiết
  + - 1. **Kiến trúc tổng quan của DDoS attack-network**

Nhìn chung DDoS attack-network có hai mô hình chính:

* Mô hình Agent – Handler
* Mô hình IRC – Based

A diagram of a computer network

Description automatically generated

Hình 2. 1: Sơ đồ phân loại các kiểu tấn công DDoS

1. Mô hình Agent – Handler

Mô hình này Attack – network gồm 3 thành phần: Agent, Client và Handler

* Client: là software cơ sở để hacker điều khiển mọi loại hoạt động của attack-network
* Handler: là một thành phần software trung gian giữa Agent và Client
* Agent: là thành phần software thực hiện sự tấn công mục tiêu, nhận điều khiển client thông qua Handler.

A diagram of a victim

Description automatically generated

Hình 2. 2: Kiến trúc attack kiểu Agent – Handler

* Attacker sẽ từ Client giao tiếp với Handler để xác định số lượng Agent đang online, điều chỉnh thời điểm tấn công và cập nhật các Agent. Tùy theo cách attacker cấu hình attack-network, các Agent sẽ chịu sự quản lý của một hay nhiều Handler.
* Thông thường Attacker sẽ đặt Handler software trên một Router hay một server có lượng traffic lưu thông nhiều. Việc này nhằm mục đích làm cho cá giao tiếp giữa Client, Handler và Agent khó bị phát hiện. Cá giao tiếp này thông thường xảy ra trên các protocol TCP, UDP và ICMP. Chủ nhân thực sự của các Agent thông thường không hề hay biết họ đạng bị lợi dụng vào cuộc tấn công DDoS, do họ không thể đủ kiến thức hoặc các chương trình Backdoor Agent chỉ sử dụng rất ít tài nguyên hệ thống làm cho hầu như không thể thấy ảnh hưởng gì đến hiệu năng của hệ thống.

1. Mô hình IRC – Based:

* Internet Relay Chat ( IRC) là một hệ thống online chat multiuser, IRC cho phép User tạo một kết nối đến multipoint đến nhiều user khác và chat thời gian thực. Kiến trúc của IRC network bao gồm nhiều IRC server trên khắp internet, giao tiếp với nhau trên nhiều kênh (channel), IRC network cjo phép uer tạo ba loại channel, public, private, serect.
* Public channel: Cho phép user của channel đó thấy IRC name và nhận được message của mọi user khác trên cùng channel.
* Private channel: được thiết kế để giao tiếp với các đối tượng cho phép. Không cho phép các user không cùng channel thấy IRC name và message trên channel. Tuy nhiên, nếu user ngoài channel dùng một số lệnh channel locator thì có thể biết được sự tồn tại của private channel đó.
* Secret channel: tương tự private channel nhưng không thể xác định bằng channel locator.

A diagram of a network

Description automatically generated

Hình 2. 3: Kiến trúc attack network kiểu IRC-based

* IRC- based network cũng tương tự như Agent – Handler netowork nhưng mô hình này sử dụng các kênh giao tiếp IRC làm phương tiện giao tiếp giữa Client và Agent. Sử dụng mô hình này, attacker còn có thêm một số lợi thế khác như:

Các giao tiếp dưới dạng chat message làm cho việc phát hiện chúng là vô cùng khó khăn

IRC traffic có thể di chuyển trên mạng với số lượng lớn mà không bị nghi ngờ.

Không cần phài duy trì danh sách các Agent, hacker chỉ cần logon vào IRC server là đã có thể nhận được report về trạng thái các Agent do các channel gửi về.

Sau cùng: IRC cũng là một môi trường file sharing tạo điều kiện phát tán các Agent code trên nhiều máy khác.

* + - 1. **Phân loại tấn công kiểu DDoS**
* Nhìn chung, có rất nhiều biến thể của kỹ thuật tấn công DDoS nhưng nếu nhìn dưới góc độ chuyên môn thì có thẻ chia các biển thể thành hai loại dựa trên mục đích tấn công: Làm cạn kiệt băng thông và làm cạn kiệt tài nguyên của hệ thông.

A diagram of a computer system

Description automatically generated

Hình 2. 4: Phân loại kiểu tấn công DDoS

* 1. **SQL Injection**
     1. **Tấn công SQL Injection**
        1. Tấn công SQL Injection là gì?
* Khi triển khai các ứng dụng web trên Internet, nhiều người vẫn nghĩ rằng việc đảm bảo an toàn, bảo mật nhằm giảm thiểu tối đã khả năng bị tấn công từ các tin tặc chỉ đơn thuần tập trung vào các vấn đề như chọn hệ điều hành, hệ quản trị cơ sở dữ liệu, webserver sẽ chạy ứng dụng,… mà quên mất rằng ngay cả bản thân ứng dụng chạy trên đó cũng tiềm ẩn một lỗ hổng bảo mật rất lớn. Một trong số các lỗ hổng đó là SQL Injection. Tại Việt Nam, đã qua thời kỳ các quản trị web lơ là việc quét virus, cập nhập các bản vá lỗi từ các phần mềm hệ thống, nhưng việc chăm sóc các lỗi của các ứng dụng lại rất ít được quan tâm. Đó là lý do tại sao trong thời gian vừa qua, không it website tại Việt Nam bị tấn công và đa sô đều là lỗi SQL Injection. Vậy SQL Injection là gì ?
* SQL Injection là một kỹ thuật cho phép những kẻ tấn công lợi dụng lỗ hoognr trong việc kiểm tra dữ liệu nhập trong các ứng dụng web và các lỗi thông báo lỗi của hệ quản trị cơ sở dữ liệu để tiêm vào (inject) và thi hành các lệnh SQL bất hợp pháp (không được người phát triển ứng dụng lường trước ). Hậu quả cuẩ nó là rất tai hại vì nó cho phép nhũng kẻ tấn công có thể thực hiện các thao tác cơ sở dữ liệu, hiệu chỉnh, do có toàn quyền trên cơ sở dữ liệu của ứng dụng, thậm chí web có dữ liệu được quản lý bằng các hệ quản trị cơ sở dữ liệu như SQL Server, MySQL, Oracle, DB2, Sysbase
  + - 1. Các dạng tấn công SQL Injection
* Có 4 dạng thông thường bao gồm: vượt qua kiểm tra lúc đăng nhập, sử dụng câu lệnh SELECT, sử dụng câu lệnh INSERT, sử dụng các Stored-procedured
* Để biết các website bán hàng sử dụng CSDL SQL ta sử dụng các soft hoặc các công cụ tìm lỗi. Hoặc các công cụ tìm kiếm như Google, Và dùng các Dork tìm kiểm như
* Để biết website nào dính lỗi SQL Injection ta thêm dấu “’” vào sau thanh địa chỉ
* Với dạng tấn công này, tin tặc có thể dễ dàng vượt qua các trang đăng nhập nhờ vào lỗi khi dùng các câu lệnh SQL thao tác trên cơ sở dữ liệu của ứng dụng web.
* Xét một ví dụ điển hình, thông thường để cho phép người dùng truy cập vào các trang web được bảo mật, hệ thống thường xuyên xây dựng trang đăng nhập để yêu cầu người dùng nhập thông tin về tên đăng nhập và mật khẩu. Sau khi người dùng nhập thông tin vào, hệ thống sẽ kiểm tra tên đăgn nhập và mật khẩu có hợp lệ hay không để quyết định cho phép hay hay từ chối thực hiện tiếp. Trong trường hợp này, người ta có thể dùng hai trang, một trang HTML để hiển thị form nhập liệu và một trang ASP dùng để xử lý thông tin nhập từ phía người dùng

**Dạng tấn công sử dụng câu lệnh SELECT**

* Dạng tấn công này phức tạp hơn. Để thực hiện được kiểu tấn công này, kẻ tấn công phải có khả năng hiểu và lợi dụng các sở hở trong các thông báo lỗi từ hệ thống để dò tìm các điểm yếu khởi đầu cho việc tấn công. Xét một ví dụ rất thường gặp trong website về tin tức. Thông thường, sẽ có một trang nhận ID của tin cần hiển thị rồi sau đó truy vấn nội dung của tin có ID này.

**Dạng tấn công sử dụng câu lệnh INSERT**

* Thông thường các ứng dụng web cho phép người dùng đăng kỹ một tài khoản để tham gia. Chức năng không thể thiếu là sau khi đăng ký thành công, người dùng cso thể xem và hiệu chỉnh thông tin của mình. SQL Injection có thể được dùng khi hệ thống không kiểm tra tính hợp lệ của thông tin nhập vào.

**Dạng tấn công sử dụng Stored-procedured**

* Việc tấn công bằng Stored-procedure sẽ gây tắc hại rấ lớn nếu ứng dụng được thực thi với quyền quản trị hệ thống ‘sa’
* Ví dụ: nếu ta thay đoạn mã tiêm vào dạng ‘;EXEC xp\_cmdshell’ cmd.exe dir C: ‘. Lúc này hệ thống sẽ thực hiện lệnh liệ kê thưc mục trên ổ địa C:\ cài đật Server – Việc phá hoại kiểu nào tùy thuộc vào cau lệnh đằng sau cmd.exe. Nếu cài SQL Server ở chế độ mặc định thì SQL Server chạy trên nền SYSTEM, tương đương mức truy cập ở windows. Có thẻ dùng master..xp\_cmdshell để thi hành lệnh từ xa:

Exe master..xp\_cmdshell ‘ping 10.10.1.2’

Thử dùng nháy đôi nếu nháy đơn không làm được việc này

**Dạng tấn công SQL Injection nâng cao**

* Chuỗi ký tự không có dấu nháy đơn:
* Những nhà lập trình có thể bảo vệ ứng dụng của họ bằng cách loại bỏ tất cả các dấu nháy, thông thường loại bỏ dấu nháy bằng cachs thay một dấu nháy thành 2 dấu nháy
* Rõ ràng là, nó ngăn chặn được tất cả những kiểu tấn công trên. Tuy nhiên nếu muốn tạo ra một chuỗi giá trị mà không dùng các dấu nháy, có thể dùng hàm “char()”
* Tấn công 2 tầng:
* Mặc dù ứng dụng đã thay thế dấu nháy đơn nhưng vấn còn khả năng bị chèn đoạn mã SQL
  + 1. **Cách phòng tránh tấn công SQL Injection.**

Như vậy, có thể lỗi SQL Injection khai thác những bất cẩn của các lập trình viên phát triển ứng dụng web khi xử lý các dữ liệu nhập vào để xây dựng câu lệnh SQL. Tác hại từ lỗi SQL injection tùy thuộc vào môi trường và cách cấu hình hệ thống. Nếu ứng dụng sử dụng quyền dbo ( quyền của người sở hữu cơ sở dữ liệu – owner) khi thao tác dữ liệu, nó có thể xóa toàn bộ các bảng dữ liệu liệu, tạo các bảng dữ liệu mới, … Nếu ứng dụng sử dụng quyền sa ( quyền quản lý hệ thống ), nó có thể điểu khiển toàn bộ hệ quản trị cơ sở dữ liệu và với quyền hạn rộng lớn như vậy nó có thể tạo ra các tài khoản bất hợp pháp người dùng bất hợp pháp để điểu khiển hệ thống của bạn.

* Trong hầu hết các trình duyệt, những kí tự nên được mã hóa trên địa chỉ URL trước khi được sử dụng.
* Việc tấn công theo SQL Injection dựa vào những câu thông báo lỗi do đó việc phòng chống hay nhất vấn là không cho hiển thị những thông điệp lỗi cho người dùng bằng cách thay thế những lỗi thông báo bằng 1 trang do người phát triển thiết kế mỗi khi lỗi xảy ra trên ứng dụng.
* Kiểm tra kĩ giá trị nhập của người dùng, thay thế những ký tự như ‘; … Hãy loại bỏ các kĩ tự meta như “,”, \, /, và các kí tự extend như NULL, CR, LF, … trong các string nhận được từ:
* Dữ liệu nhập do người dùng đệ trình
* Các tham số URL
* Các giá trị từ cookie
* Đối với các giá trị numeric, hãy chuyển nó sang integer trước khi thực hiện câu truy vấn SQL, hoặc dùng ISNUMERIC để chắc chắn nó là một integer.
* Dùng thuật toán để mã hóa dữ liệu

1. **Kiểm tra dữ liệu**
2. **Khóa chặt SQL Server**

Đây là một danh sách các công việc cần làm để bảo vệ SQL Server:

* Xác định các phương pháp kết nối đến Server
* Dùng tiện ích Network Utility để kiểm tra rằng chỉ có các thư viện mạng đang dùng là hoạt động
* Kiểm tra tất cả các tài khoản có trông SQL Server
* Chỉ tạo tài khoản có quyền thấp cho các ứng dụng
* Loại bỏ những tài khoản không cần thiết
* Kiểm tra các đối tượng tồn tại
* Kiểm tra những tài khoản nào có thể truy xuất đến những đối tượng nào
* Kiểm tra các phiên làm việc trên server
* Thay đổi startup và chạy SQL Server ở mức người dùng quyền hạn thấp trong SQL Server Security

1. **Thiết lập cầu hình an toàn cho hệ quản trị cơ sở dữ liệu**

* Cần có cơ chế kiểm soát chặt chẽ và giới hạn quyền xử lý dữ liệu đến tài khoản người dùng mà ứng dụng web đang sử dụng. Các ứng dụng thông thường nên tránh dùng đến các quyền như dbo hay sa. Quyền càng bị hạn chế, thiết hại càng ít.
* Ngoài ra để tránh các nguy cơ từ SQL Injection Attack, nên chú ý loại bỏ bất kì thông tin kĩ thuật nào chứa những thông điệp chuyển xuổng cho người dùng khi ứng dụng có lỗi. Các thông báo lỗi thông thường tiết lộ các kĩ thuật cho phép kẻ tấn công biết được điểm yếu của hệ thống.
  1. **Cross Site Scripting (XSS)**
     1. **Tấn công XSS**
* Cross- Site Scripting (XSS) là một trong những kĩ thuật tấn công phổ biến nhất hiện nay, đồng thời nó cũng là một trong những vấn đề bảo mật quan trọng đối với các nhà phát triển web và cả những người sử dụng web. Bất kì một website nào cho phép người sử dụng đăng thông tin mà không có sự kiểm tra chặt chẽ các đoạn mã nguy hiểm thì đều có thể tiềm ẩn các lỗi XSS.
* Cross-site Scripting hay còn được gọi tắt XSS ( thay vì gọi tắt là CSS để tránh nhầm lẫn với CSS – Casading Style Sheet của HTML) là một kĩ thuật tấn công bằng cách chèn các website động (ASP, PHP, CGI, JSP, …) những thẻ HTML hay những đoạn mã script nguy hiểm có thể gây nguy hại cho người sử dụng khác. Trong đó, những đoạn mã nguy hiểm được chèn vào hấu hết được viết bằng các Clint-Site Script như Javascript, JSScript, DHTML và cũng có thể là cả các thẻ HTML. Kĩ thuật tấn công XSS đã nhanh chóng trở thành một trong những lỗi phổ biến nhất của Web Applications và mối đe dọa của chúng đối với những người sử dụng ngày càng lớn. Người chiến thắng trong cuộc thi eWeek OpenHack 2022 là người đã tìm ra 2 XSS mới. Phải chăng mỗi nguy hiểm từ XSS đã ngày càng được mọi người chú ý tới.
  + - 1. **Hoạt động của XSS**
* Về cơ bản XSS cũng như SQL Injection hay Source Injection nó cũng là các yêu cầu được gửi từ máy client tới server nhằm chèn vào đó các thông tin vượt quá tầm kiểm soát của server. Nó có thể làm một request được gửi từ các form dữ liệu hoặc cũng có thể đỏ chỉ là các URL như là
* Và rất có thể trình duyệt của bạn sẽ hiện lên thông báo “XSS was found!”. Các đoạn mã trong thẻ script không hề bị giới hạn bới chúng hoàn toàn có thể thay thế bằng một file nguồn trên một server khác thông qua thuộc tính src của thẻ script. Cũng là vì lẽ đó mà chúng ta chưa thể lường hết mức độ nguy hiểm của các lỗi XSS.
* Nhưng nếu như các kĩ thuật tấn công khác có thể làm thay đổi được dữ liệu nguồn của webserver ( mã nguồn, cầu trúc, cơ sở dữ liệu) thì XSS chỉ gây tổn hại đối với website ở phía client mà nạn nhân trực tiếp là những người khách duyệt site đó. Tất nhiên đối khi các hacker cũng sử dụng kĩ thuật này để deface các website nhưng đó vẫn chỉ tấn công vào bề mặt của website. Thật vậy, XSS là những Client- Side cript, những đoạn mã này sẽ chỉ chạy bởi trình duyệt phái client do đó XSS không làm ảnh hưởng đến hệ thống website nằm trên server. Mục tiêu tấn công của XSS không ai khác chính là những người sử dụng khác trên website, khi họ vô tình vào các trang có chữa đoạn mã nguy hiểm do các hacker để lại họ có thể bị chuyển tới website khác, đặt lại homepage, hay nặng hơn là mất mật khẩu, mất cookie thậm chí máy tính bạn có thể bị cài loại virus, backdoor, room.
  + - 1. **Cách tấn công**

1. **Scan lỗ hổng XSS của ứng dụng web**

* Cách 1: Sửu dụng nhiều chương trình do quét lỗi của ứng dụng web, ví dụ như chương trình Web Vulnerability Scanner để do quét XSS.
* Cách 2: Thực hiện 5 bước:
* Bước 1: Mở website cần kiểm tra
* Bước 2: Xác định các chỗ cần kiểm tra XSS. Một site bất kỳ bao giờ cũng có các phần:
  + - Search, Error Message, web form. Chủ yếu lỗi XSS nằm ở phần này, nói chung XSS có thể xảy ra ở chỗ nào mà người dùng có thể nhập dữ liệu vào vào sau đó nhận được một cái gì đó. Ví dụ chúng ta nhập mỗi chuỗi XSS.
* Bước 3: Xác minh khả năng site có bị lỗi XSShay không bằng cách xem các thông tin trả về. Ví dụ như chúng ta thấy thế này: ‘ Không tìm thấy XSS’, hay là “tài khoản XSS không chính xác”. Đăng nhập với XSS không thành công … thì khi đó khả năng chỗ đó bị đính xSS là rất cao.
* Bước 4: Khi đã xác định chỗ có khả năng bị đính lỗi XSS thì chúng ta chèn những đoạn code của chúng ta vào để thử tiếp, ví dụ như sau:

Chèn đoạn code này: **<script>alert(‘XSS’)</script>** vào ô bị lỗi và nhấn nút Login, nều chúng ta nhận được một popup có chữ ‘XSS’ thì 100% bị dính XSS. Nhưng xin chú ý, thỉnh thoảng vấn có trường hợp website đó bị đính XSS những vẫn không xuất hiện cái popup thì buộc bạn phải View Source nỏ ra để xem. Khi view source nhwos kiểm tra dòng này **<script>alert(‘XSS’)</script>**  nếu có thì hết chạy, XSS đây rồi.

1. **Tấn công**

* Thật ra thì có nhiều kỹ thuật tấn công dựa trên lỗi XSS này, chủ yếu là sau khi đã biết cách tìm lỗ hổng thì mọi người sẽ có một mưu mô cho cách tấn công của mình. Ở đây mình xin giới thiệu đến các bạn một kỹ thuật mà mình đã thực hiện thành công trên trang modle của khoa công nghệ thông tin KHTN. Kỹ thuật toàn ăn cắp password.
* Sau khi đã xác minh một điều chắc chắn rằng một moodle bị lỗi XSS ơ chỗ đăng nhập.
* Tôi lập tức viết ngay một ứng dụng nhỏ rồi up lên một cái host free, ứng dụng này sẽ có nhiệm vụ nhận thông tin về mssv và password gửi về và ghi xuống file txt
* Bước 1: Tôi tạo một mail giả dạng nói là: Diễn đàn tuyển dụng của Intel, mời các bạn nào qua. Rồi tạo ra một đường link giả. Trong tích tắc này trang này sẽ gắn một cái đoạn script độc có nhiệm vụ lấy về username và pasword sau khi đăng nhập vầ gắn vào cái trang thật ( Vì trang thật bị lỗi XSS nên cho phép chúng ta gắn mã độc, gắn ở đây có nghĩa là khi chúng ta view source code của trang lên, chúng ta sẽ thấy có một đoạn script của chúng ta nằm ở đâu, rồi sau đó redirect sang trang thật ngay lập tức để khỏi bị nghi ngờ.
* Bước 2: Người dùng vào mail, tưởng thât, click vào link và thấy chạy đúng trang mooodle.
* Bước 3: Họ đăng nhập, khi đó ứng dụng sẽ chạy biện dịch từ trên xuống và tất nhiên sẽ chạy luôn cả script mà chúng ta đã cài, khi đó MSSV và password sẽ được lấy về để gửi cho một cái trang trên server mà chúng ta dựng ra.
* Bước 4: Ứng dụng server của ta nhận được mssv và password, ghi ra file txt.
* Bước 5: Kết thúc quá trình tấn công, chúng ta có một danh sách các tài khoản của sinh viên.
  + 1. **Phòng chống**
* Như đã đề cập ở trên, một tấn công XSS chỉ thực hiện được khi gửi một trang web cho trình duyệt web của nạn nhân có kèm theo mã script độc của kẻ tấn công. Vì vậy, những người phát triển web có thể bảo vệ website của mình khoi bị lợi dụng thông qua những tấn công XSS này, đảm bảo những trang phát sinh động không chứa các tag của script bằng cách lọc và xác nhận hợp lý các dữ liệu đầu vào từ phía người dùng hoặc mã hóa (encoding) và lọc các giá trị xuất cho người dùng.
  + - 1. **Lọc**
* Luôn luôn lọc các dữ liệu nhập từ phía người dùng bằng các lọc các kí tự meta ( kĩ tự đặc biệt) được định nghĩa trong đặc tả của HTML. Mỗi trường hợp nhập liệu bào gồm cả tham số liên kết sẽ được kiểm trả để phát hiện trong thẻ script.
  + - 1. **Mã hóa**
* Lỗi XSS có thể tránh được khi máy chủ Web đảm bảo những trang phát sinh được mã hóa ( encoding) thích hợp để ngăn chặn chạy các script không mong muốn.
* Mã hóa phía máy chủ là một tiến trình mà tất cả nội dung phát sinh động sẽ đi qua một hàm mã hóa nơi mà các thẻ script sẽ được thay thế bởi mã của nó.
* Nói chung, việc mã hóa được khuyến khích sử dụng vì nó không yêu cầu bạn phải đưa ra quyết định những kí tự nào là hợp lệ hoặc không hợp lệ. Tuy nhiên khả năng thực thi một số máy chủ.

# 

# **CHƯƠNG 3: THIẾT KẾ MÔ HÌNH VÀ ỨNG DỤNG THUẬT TOÁN HỌC MÁY NHẬN DIỆN BẤT THƯỜNG REQUEST**

## **Giới thiệu chương**

Trong sự phát triển của công nghệ hiện nay, có rất hàng ngàn cuộc tấn công mạng hằng ngày, hàng giờ để nhắm mục địch phá hoại hay thỏa mãn cá nhân. Để nhắm nâng cao tính bảo mật và giảm thiểu những rủi ro không đáng có cho những hệ thống hiện nay. Nhiều hệ thống log ra đời để mục đích lưu và phân tích những trường hợp tấn công có thể có nguy cơ sẽ đến trong tương lại và nhiều webserver có đã tích hợp ghi nhận log để những log cần phân tích. Vậy thì một webserver log lại như thế nào? Có thể log được những thuộc tính gì? Và những log đó có thể giúp ích được gì tránh các cuộc tấn công tương tự trong trương lại không. Thì ngay dưới đây sẽ giải thích về những điều đó?

* 1. **Các thuộc tính của một log webserver**
     1. **Giới thiêu về Webserver**

Webserver là máy chủ cài đặt các chương trình phục vụ các ứng dụng web có khả năng tiếp nhận request từ các trình duyệt web và gửi phản hổi đến client thông qua giao thức HTTP hoặc các giao thức khác.

1. **Nginx**

Nginx là một webserver nhẹ, không chiếm nhiều tài nguyên của hệ thống, Nginx còn là một reverse proxy mã nguồn mở. Nginx khá là ổn đinh, cấu hình đơn giản và hiệu suất cao.

Nginx là một phần mềm mã nguồn mở miễn phí, được phát hành rộng rãi theo giấy phép BSD, Nginx được phát trineer bằng nhiểu ngôn ngữ. Nginx có các tính

1. **Apache HTTP Server**

Apache là webserver được sử dụng rộng rãi nhất thế giới. Apache được phát triển và duy trì bởi một cộng đồng mã nguồn mở dưới sự bảo trợ Apache Software. Apache được phát hành với giấy phép Apache License là được sử dụng tự do, miễn phí

1. **Internet Information Services**

IIS do Microsoft phát triển, sản phẩm này được tích hợp cùng với hệ điều hành windows. Trong IIS bao gồm nhiều dịch vụ như: dịch vụ Webserver, dịch vụ FTP Server. Tính đến thời điểm tháng 5 năm 2015 thì số lượng trang web sử dụng máy chủ IIS gần 248 triệu trang web,

Tất cả tinhsa năn của webserver được quản lý độc lập dọ đó chúng ta có thể dễ dàng thêm loại bỏ hoặc thay thế các tính nắng của webserver.

Nhờ được tích hợp ASP.NET IIS có thể sử dụng toàn bộ sức mạng ASP. NET Module ASP.NET làm cho máy chủ phát triển nhanh chóng nhờ vào giao diện quen thuộc và các dịch vụ ứng dụng của ASP.NET

1. **Apache Tomcat**

**Apache Tomcat** là một java servlet được phát triển bởi Apache Software Foundation. Tomcat thực thi các ứng dụng Java Servlet và JSP. Tomcat cung cấp một máy chủ HTTP cho ngôn ngữ Java thuần túy.

**Apache Tomcat** rất ổn định và có tăt cả các tính năng của một ứng dụng web thương mại nhưng lại đi theo giấy phép mã nguồn mở của Apache. Tomcat cũng cung cấp một số tổ chức năng bổ sung như tomcat manager application, speciallized realm imlementation và tomcat valves.

1. **Lighttpd**

**Lighttpd** là một phần mềm mã nguồn mỡ, an toàn và linh hoạt, đặc biệt miễn phí và được phân phối theo giấy phép BSD, LIghttpd được viết bới Jan Kneschke. Lightpd chiếm ít tài nguyên, và memory thấp, CPU nhỏ. Nó được phát triển bởi ngôn ngũ C. chạy trên hệ điều hành Linux, Windows, MacOS

* + 1. **Các thuộc tính được log trong hệ thống webserver**
* Địa chỉ IP của client: địa chỉ IP của client được sử dụng để xác định nguồn gốc của yêu cầu truy cập, giúp phân tích lưu lượng mạng và phát hiện cuộc tấn công
* Thời gian yêu cầu: Thời gian yêu cầu dược ghi lại để phân tích thời gian hoạt động và tải trang của trang web
* URL: URL của trang web được yêu cầu cũng được ghi lại, cho phép phân tích lưu lượng truy cập và đánh giá hiệu quả của các trang web cụ thể
* Phương thức HTTP: phướng thức HTTP được sử dụng trong yêu cầu cũng được ghi lại, cho phép xác định phương thức GET, POST hay các phương thức khác.
* Mã trạng thái HTTP: mã trạng thái HTTP được ghi lại để xác định kết quả của yêu cầu, bao gồm mã trạng thái thông báo thành công lỗi hay chuyến hướng.
* Kích thước phản hồi: Kích thước phản hồi của yêu cầu cũng được ghi lại chó phép xác định lưu lượng truy cập và đánh giá hiệu quả của trang web server.
* User Agent: User Agent của client cũng được ghi lại trong log, cho phép phân tích các trình duyệt và thiết bị được sử dụng để truy cập vào trang web.
* Thông tin về cookies. Các thông tin về cookies đươc sử dụng khi truy cập vào trang web cũng được ghi lại trong log.
* Thông tin về phiên làm việc. Các thông tin về phiên làm việc của người dùng trên trang web cũng có thể được ghi lại trong log.

## **Thuật toán đề xuất**

Với ý tưởng xem số lượng request trong một ngày và áp dụng một số giải thuật để nhận biết được một số request bất thường. Có những thuật toán được đề xuất như sau:

* Sử dụng Interquartile
* Thuật toán Isolation Forest
* Thuật toán Median Absolute Deviation
* KNN ( K – Nearest Neighbor algorithm)
* HBOS
* LOF
* OCSVM
  + 1. **Interquartile Range**

**A graph with red and green dots

Description automatically generated**

Hình 3. 1: Phát hiện các điểm bất thường dựa trên Interquartile Range

Với bảng đồ thị như trên có nhận xét như sau:

* + Số lượng request trên 111.000 đều là những điểm bất thường
    1. **Isolation Forest**

**A graph with red and green dots

Description automatically generated**

Hình 3. 2: Phát hiện các điểm bất thường dựa trên Isolation Forest

Với bảng sơ đồ hình trên có nhận xét như sau:

* + Số lượng 50.000 request một ngày thì có thể thì chỉ phát hiện được 2 điểm bất thường
  + Với một ngày trên 50.000 request đều được xem là bất thường hết
    1. **Median Absolute Deviation**

**A graph with red green and blue dots

Description automatically generated**

Hình 3. 3: Phát hiện những điểm bật thường dựa trên thuật toán Median Absolute Deviatation

Với bảng sơ đồ trên có nhận xét như sau:

* + Với những ngày có trên 150.000 request mỗi ngày thì đều là điểm bất thường
    1. **KNN (K- Nearest Neighbor algorithm)**

**A graph with green and red dots

Description automatically generated**

Hình 3. 4: Phát hiện bất thường dựa trên thuật toán KNN

Với hình này có nhận xét như sau:

* + Với những điểm có số lượng request trong một ngày trên 150.000 chỉ phát hiện được một điểm bình thường và còn lại là không bất thường
    1. **HBOS(Histogram Based Outlier Detection)**

**A graph with green dots

Description automatically generated**

Hình 3. 5: Phát hiện bất thường dựa trên HBOS

Với biểu đồ trên ta có nhận xét như sau:

* + Không phát hiện được các điểm bất thường
    1. **LOF**

**A graph with green and red dots

Description automatically generated**

Hình 3. 6: Phát hiện bất thường dựa trên thuật toán LOF

Với hình này có nhận xét như sau:

* + Với những điểm có số lượng request trong một ngày trên 150.000 chỉ phát hiện được một điểm bình thường và còn lại là không bất thường
    1. **OCSVM**

**A graph with green dots

Description automatically generated**

**Hình 14: đồ thị áp dụng thuật toán OCSVM**

Hình 3. 7: Đồ thị áp dụng thuật toán OCSVM

* + 1. **Nhận xét chung**
* Với 7 thuật toán trên ta có nhận xét như sau:
  + Với thuật toán OCSVM và thuật toán HBOS không nhận dạng được các điểm bất thường
  + Với thuật toán LOF và thuật toán KNN đều nhận biết được số lượng điểm bất thường và bình thường giống nhau
  + Với phương pháp tứ phân vị (Interquatile Range) thì có thể tìm thấy được phân bố điểm bất thường và bình thường rõ ràng
  + Với biểu đồ Isolation Forest không thể thấy được phân bố rõ ràng giữa điểm bất thường và bình thường
  + Với biểu đồ Median Absolute Deviation nhận biết số lượng điểm bất thường ít hơn số lượng điểm bất thường sử dụng phương pháp tứ phân vị

## **Các bước thực hiện**

* + 1. **Sơ lược về mô hình**

A screenshot of a computer screen

Description automatically generated

Hình 3. 8: Mô hình sơ lược về quá trình chạy

Mô tả:

* + Khi người dùng gửi một yêu cầu đến server
  + Với mỗi request này sẽ được đi qua webserver gồm có 2 chức năng như sau:
    - Thực hiện các yêu cầu tiếp khi cấu hình trong file cấu hình của webserver
    - Được log lại trong file access.log
  + Và webserver cũng sẽ trả về response sau khi wordpress trả về cho webserver
  + Bước analysis là bước phân tích log do người phân tích hoặc máy phân tích và sau đó trả về kết quả
    1. **Xây dựng một webserver**

Công nghệ: Docker

File docker compose:

A screenshot of a computer program

Description automatically generated

Hình 3. 9: File docker compose để dựng webserver

**Cách chạy:**

* + Vào đúng thư mục chưa file docker-compose.yml
  + Chạy lệnh sau **docker compose up**
  + Sau đó vào đường dẫn

**Kết quả:**

A screen shot of a computer

Description automatically generated

Hình 3. 10: Kết quả sau khi chạy lệnh

A screenshot of a computer

Description automatically generated

Hình 3. 11: hiển thị webserver

* + 1. **Lựa chọn thuật toán**

Với những nhận xét ở mục 3.3.8 thì tôi chọn phương pháp Interquartile Range ( tứ phân vị với dataset này)

Với Interquatile Range chọn **mức cao nhất của số lượng request bình thường là 111532 request**

Ý tưởng thuật toán như sau:

B1: Tính số lượng request trong một ngày

B2: Với số lượng request trong một ngày lớn hơn **mức cao nhất của số lượng request bình thường**  thì được xem là bất thường

A computer screen shot of a program

Description automatically generated

Hình 3. 12: Kết quả sau khi chạy lệnh

A screenshot of a computer program

Description automatically generated

Hình 3. 13: Code triển khai nhận biết những ngày có số lượng request bất thường

**KẾT LUẬN**

1. **Kết luận:**

Sau khi thực hiện đề tài “Ứng dụng phương pháp máy học trong bài toán nhận dạng những request bất thường”, em đã tìm hiểu về các dạng phân phối chuẩn và một vài thuật toán áp dụng trong nhận dạng request bất thường. Từ những lý thuyết em tìm hiểu và nghiên cứu cùng với một số công nghệ đang có hiện này em đã dựng mô hình nhận diện request bất thường

Trong báo cáo này của em nói đến một số lý thuyết liên quan đến thống kê và những mô hình máy học em đề xuất. Và trong bài báo cáo em đã thể hiện được điều sau:

* + Xây dựng được mô hình log cơ bản của webserver
  + Ứng dụng một vài thuật toán máy học để tìm ra đăc trưng của tập dữ liệu liên quan đến những dạng request bất thường
  + Và chọn ra được các đặc điểm để có thể nhận diện được những ngày bất thường

1. **Hạn chế**

Bởi những hạn chế về năng lực cũng như thời gian, đề tài chỉ thể hiện được những điều trên. Và những điều hiện tại em đã nhận thấy và chưa thể hiện được:

* + Cần tìm hiểu nhiều thuật toán và đề xuất mô hình và đánh giá để mang lại các kết quá chỉnh xác hơn về nhận diện bất thường trong request
  + Và chưa thiết kế được hệ thống cảnh báo nhận diện bất thường các request bất thường trong một thời gian ngắn

1. **Hướng phát triển:**

Tiếp tực nghiên cứu để tăng khả năng nhận diện các request bất thường để có độ chính xác ngày càng nhận diện tốt hơn và thiết kế hệ thống nhận diện các request bất thường trong thời gian ngắn

# **DANH MỤC TÀI LIỆU THAM KHẢO**

**Tiếng Anh:**

[1]. Thomas Defard, Aleksandr Setkov, Angelique Loesch, Romaric Audigier. PaDiM: a Patch Distribution Modeling Framework for Anomaly Detection and Localization

[2]. Tianyuan Lu, Lei Wang \* and Xiaoyong Zhao, *Review of Anomaly Detection Algorithms for Data Streams*

[3]. David Guthrie, Louise Guthrie, Ben Allison, Yorick Wilks, *Unsupervised Anomaly Detection*

**Danh mục các website tham khảo:**

[4]. <https://www.scribbr.com/statistics/normal-distribution/>

[5]. <https://www.analyticsvidhya.com/blog/2021/06/>

[6]. <https://meeyland.com/chuyen-doi-so/anomaly-detection-la-gi-cac-uu-nhuoc-diem-khi-su-dung-anomaly-detecrion/>

[7]. <https://www.dathoangblog.com/2019/01/anomaly-detection.html>

[8]. 5 Anomally Detection Algorithms every Data Scientist should know https://towardsdatascience.com/5-anomaly-detection-algorithms-every-data-scientist-should-know-b36c3605ea16