HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA AN TOÀN THÔNG TIN

———————*———————

ĐỒ ÁN TỐT NGHIỆP

Đề tài: ỨNG DỤNG HỌC MÁY TRONG PHÁT HIỆN XÂM NHẬP MẠNG

Giảng viên hướng dẫn: **TS. Phạm Hoàng Duy**
Sinh viên: **Hoàng Anh Phi**
Lớp: **D13CQAT02-B**
Khóa: **2013-2018**
Hệ: **Chính Quy**

HÀ NỘI 12-2017

NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM
(Của người hướng dẫn)

. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .

Điểm: . . . . . (bằng chữ . . . . .)

Đồng ý/Không đồng ý cho sinh viên bảo vệ trước hội đồng chấm đồ án tốt nghiệp?

Hà Nội, ngày . . . . . tháng 12 năm 2017
CÁN BỘ - GIẢNG VIÊN HƯỚNG DẪN
(Ký, ghi rõ họ tên)

LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn tới các thầy, cô trong Khoa Công nghệ thông tin 1 nói chung và các thầy Bộ môn An toàn thông tin nói riêng tại Học viện Công nghệ Bưu chính Viễn thông, những người trong hơn 4 năm vừa qua đã truyền đạt cho em bao kiến thức, kinh nghiệm quý báu, những hành trang cho em bước vào tương lai.

Em xin trân trọng cảm ơn thầy Nguyễn Mạnh Sơn vì đã tạo điều kiện cho em được tham gia các cuộc thi Tin học sinh viên, một trong những trải nghiệm quý báu nhất thời sinh viên của em.

Đặc biệt, em xin được bày tỏ lòng biết ơn sâu sắc tới giảng viên, TS. Phạm Hoàng Duy, người đã tận tình hướng dẫn và chỉ bảo tôi tận tình trong quá trình làm đồ án này. Những lời khuyên của thầy trong suốt quá trình hoàn thành đồ án này đã giúp em có thêm nhiều kiến thức cũng như kinh nghiệm trong lĩnh vực nghiên cứu khoa học.

Em cũng xin chân thành cảm ơn anh Phạm Xuân Cường, Trung tâm Không gian mạng Viettel đã trang bị cho em rất nhiều kiến thức nền tảng trong lĩnh vực Học máy để em có thể hoàn thành được đồ án này.

Cuối cùng, em xin cảm ơn gia đình, bạn bè, những người luôn ở bên cạnh, quan tâm, tạo điều kiện cho em để có thể hoàn thành được đồ án này!

Hà Nội, ngày 12 tháng 12 năm 2017

Tác giả

Hoàng Anh Phi

Mục lục

Danh sách bảng

Danh sách hình vẽ

Danh mục từ viết tắt

STT	Từ viết tắt	Tiếng Anh	Tiếng Việt/Giải thích
1	IDS	Intrusion detection system	Hệ thống phát hiện xâm nhập
2	NIDS	Network-based Intrusion detection system	Hệ thống phát hiện xâm nhập mạng
3	HIDS	Host-based Intrusion detection system	Hệ thống phát hiện xâm nhập máy tính cá nhân
4	FTP	File Transfer Protocol	Giao thức truyền tập tin
5	HTTP	HyperText Transfer Protocol	Giao thức truyền tải siêu văn bản
6	SMTP	Simple Mail Transfer Protocol	Giao thức truyền tải thư tin đơn giản
7	DNS	Domain Name System	Hệ thống tên miền
8	NBA	Network behavior analysis	Phân tích hành vi mạng
9	SVM	Support Vector Machine	Máy vector hỗ trợ
10	kNN	K-neighbour neighbor	Giải thuật k hàng xóm gần nhất
11	CART	classification and regression trees	Cây phân loại và hồi quy
12	JVM	Java Virtual Machine	

MỞ ĐẦU

Trong thập kỷ trước, sự phát triển vượt bậc của máy tính cùng với việc giảm giá thành của các thiết bị tính toán đã được đoán trước. Ngày nay, không chỉ các công ty, tập đoàn lớn mà ngay cả cá nhân cũng có thể sở hữu một chiếc máy tính. Để làm việc thuận lợi và hiệu quả, các máy tính được liên kết với nhau sử dụng hệ thống mạng (network). Một trong những hệ thống mạng lớn nhất hiện nay là Internet cho phép một người sử dụng máy tính để trao đổi các thông điệp với các máy tính khác trên Internet. Người dùng làm việc trên Internet được hưởng lợi từ rất nhiều các ứng dụng thuận tiện như World Wide Web (WWW) và e-mail. Tuy nhiên, việc kết nối mở cũng tiềm ẩn những nguy cơ. Xâm nhập, tấn công mạng luôn là mối đe dọa thường trực tới tài sản, uy tính thậm chí là tính mạng tới các cá nhân, tổ chức, doanh nghiệp. Vì vậy việc phát hiện và ngăn chặn cách cuộc tấn công, xâm nhập mạng luôn là mối ưu tiên hàng đầu trong lĩnh vực anh toàn thông tin ngày nay.

Có rất nhiều các phương pháp đã được các nhà khoa học trên thế giới nghiên cứu và ứng dụng vào thực tế giúp phát hiện nhanh chóng và chính xác các cuộc tấn công, xâm nhập mạng. Một trong những phương pháp phổ biến và mạnh mẽ nhất là phương pháp phát hiện xâm nhập dựa trên học máy. Học máy không phải là lĩnh vực mới mẻ mà đã được nghiên cứu từ thế kỷ trước. Tuy nhiên, trong những năm gần đây, sự phát triển của các công nghệ tính toán khiến cho học máy có những thành tựu to lớn. Phương pháp phát hiện xâm nhập mạng dựa trên học máy có thể học dữ liệu từ các cuộc tấn công đã biết, từ đó dự đoán các cuộc tấn công mới. Trong phạm vi kiến thức, đồ án sẽ tập trung nghiên cứu và xây dựng mô hình phát hiện xâm nhập mạng dựa trên học máy, cụ thể với thuật toán Boosted Tree.

Đồ án được chia 3 chương với nội dung như sau:

- **Chương 1: Tổng quan về phát hiện xâm nhập mạng**
  Chương này sẽ giới thiệu tổng quan về các vấn đề an toàn thông tin mạng, các biện pháp phòng chống xâm nhập cũng như các kỹ thuật phát hiện xâm nhập mạng.
- **Chương 2: Mô hình Boosted Tree**
  Giới thiệu về phương pháp học máy có giám sát, tổng quan về phương pháp học Ensemble Learning và một mô hình cụ thể của phương pháp này là Boosted Tree.
- **Chương 3: Mô hình Boosted Tree cho phát hiện xâm nhập mạng**
  Áp dụng mô hình đã nghiên cứu vào tập dữ liệu UNSW-NB15.
- **Chương 4: Tổng kết**
  Tổng kết bài toán, tóm tắt những kết quả đã đạt được và còn chưa đạt được. Từ đó đề xuất mục tiêu hướng tới cũng như hướng nghiên cứu, phát triển tiếp theo.

Chương 1

TỔNG QUAN VỀ PHÁT HIỆN XÂM NHẬP MẠNG

Trong chương 1, đồ án trình bày cái nhìn tổng quan về các vấn đề về an toàn thông tin mạng, từ đó đưa ra cái nhìn tổng quan đồng thời nêu lên sự cần thiết của bài toán phát hiện xâm nhập mạng, khái niệm của phát hiện xâm nhập mạng.

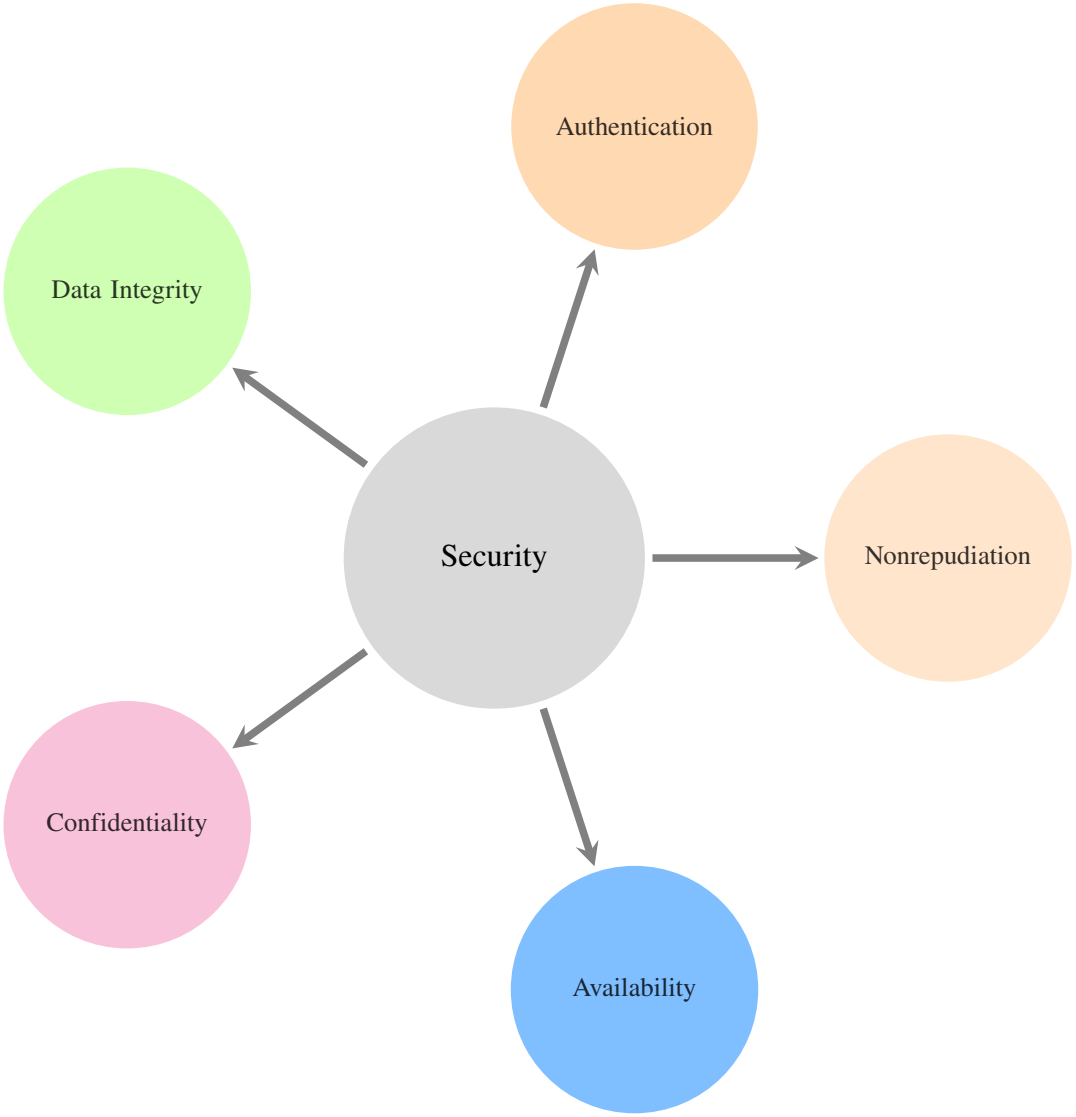
Bên cạnh đó, đồ án nêu lên các phương pháp phát hiện xâm nhập mạng đang được nghiên cứu và áp dụng hiện nay và đóng góp của đồ án đã thực hiện được.

1.1 Các vấn đề về an toàn thông tin mạng

1.1.1 Mục tiêu của việc đảm bảo an toàn thông tin

Với sự phát triển mạnh mẽ của công nghệ thông tin hiện nay, thông tin nắm vai trò cốt lõi, mang tính sống còn đối với các cá nhân, tổ chức, doanh nghiệp và đo đó chúng cũng trở thành mục tiêu hàng đầu của các kẻ tấn công. Thông tin ở đây có thể là thông tin về khách hàng, thông tin về chi tiết kinh doanh hay các thông tin mật. Điều này đặt ra một vấn đề quan trọng đó chính là phải đảm bảo an toàn thông tin trong suốt quá trình vận hành hệ thống mạng. Đảm bảo an toàn thông tin cho hệ thống mạng là đảm bảo các yếu tố sau:

- Tính xác thực (Authentication): Tính xác thực đảm bảo quá trình truyền tin giữa hai bên đều xác nhận được đối phương mà mình đang giao tiếp. Các phương pháp đảm bảo tính xác thực là : mật khẩu, chữ kí số, vân tay, mống mắt, ...
- Tính toàn vẹn (Data Integrity): Tính toàn vẹn dữ liệu đảm bảo quá trình truyền tin giữa hai bên dữ liệu còn nguyên vẹn, không bị sửa đổi, mất mát. Các biện pháp đảm bảo tính toàn vẹn là kiểm soát truy nhập chặt chẽ, xác thực quyền với đối tượng truy nhập dữ liệu, ...
- Tính bí mật (Confidentiality): Tính bí mật đảm bảo trong quá trình truyền tin, thông tin không bị lộ cho bên thứ ba có thể nghe lén được.
- Tính sẵn sàng (Availability): Tính sẵn sàng đảm bảo trong quá trình sử dụng, các tài nguyên của hệ thống luôn đáp ứng được nhu cầu của người sử dụng hợp pháp. Các yêu cầu luôn được xử lý kịp thời, không bị gián đoạ
- Tính chống chối bỏ (Nonrepudiation): Tính chống chối bỏ ngăn chặn các đối tượng phủ nhận hành vi đã thực hiện đối với hệ thống. Khi một sự cố an toàn thông tin xảy ra thì đây là một yếu tố quan trọng trong quá trình điều tra số.



Hình 1.1: Các vấn đề về an toàn thông tin

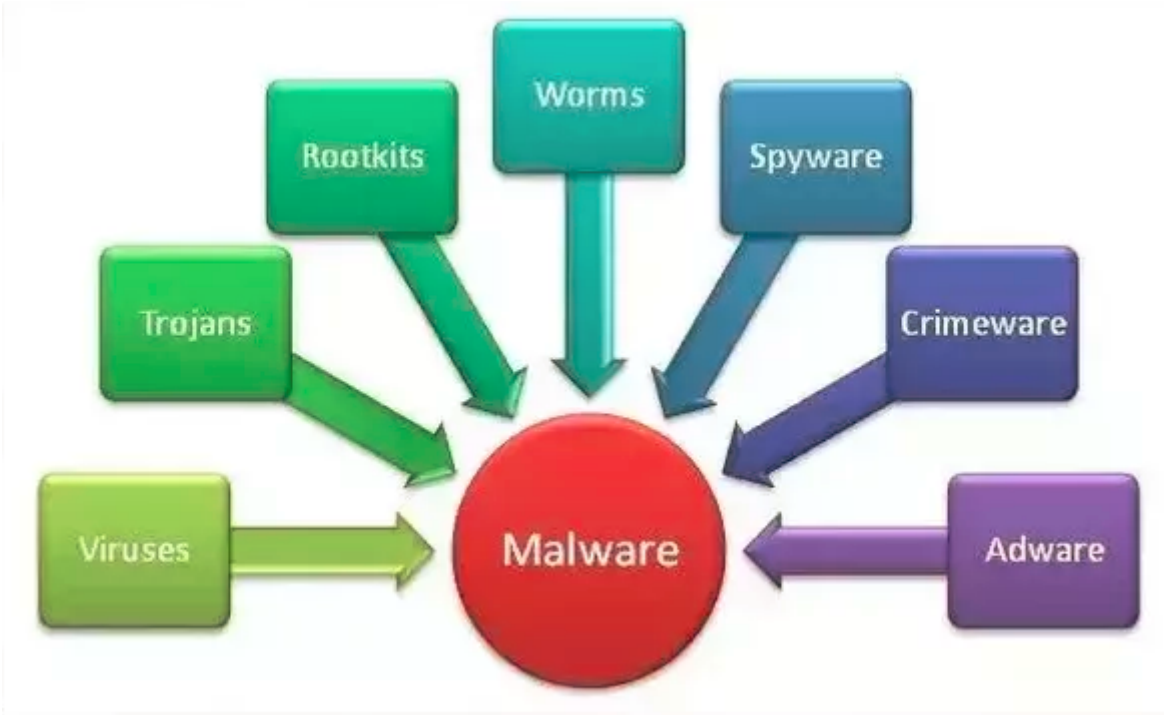
1.1.2 Tấn công mạng

Một cuộc tấn công không gian mạng là bất kỳ hình thức tấn công nào của các quốc gia, cá nhân, nhóm hoặc tổ chức nhắm vào các hệ thống thông tin máy tính, cơ sở hạ tầng, mạng máy tính hoặc các thiết bị máy tính cá nhân bằng nhiều cách khác nhau của các hành vi độc hại thường có nguồn gốc từ một nguồn giấu tên, mà đánh cắp, thay đổi, hoặc hủy hoại một mục tiêu cụ thể bằng cách hack vào một hệ thống dễ bị tổn thương.

Các hình thức tấn công mạng tuy ngày càng tinh vi và phức tạp nhưng vẫn có thể chia thành 2 nhóm chính: tấn công bằng phần mềm độc hại và tấn công tài nguyên mạng

Tấn công bằng phần mềm độc hại

Phần mềm độc hại hay còn gọi là Malware là một loại phần mềm hệ thống do các tay tin tặc hay các kẻ nghịch ngợm tạo ra nhằm gây hại cho các máy tính. Tùy theo cách thức mà tin tặc dùng, sự nguy hại của các loại phần mềm ác ý có khác nhau từ chỗ chỉ hiển thị các cửa sổ hù dọa cho đến việc tấn công chiếm máy và lây lan sang các máy khác như là virus trong cơ thể của các sinh vật.



Hình 1.2: Malware

Dưới đây là một vài loại Malware phổ biến hiện nay:

- **Viruses**: Là một chương trình phần mềm có khả năng tự sao chép chính nó từ đối tượng lây nhiễm này sang đối tượng khác (đối tượng có thể là các file chương trình, văn bản, máy tính), thường dùng thực hiện mục đích không tốt. Virus có thể lây vào máy tính qua email, qua các file tải về từ Internet hay copy từ usb và các máy tính khác về... Email là con đường lây lan virus chủ yếu và phổ biến nhất hiện nay. Virus cũng có thể lợi dụng các lỗ hổng phần mềm để xâm nhập từ xa, cài đặt, lây nhiễm lên máy tính một cách âm thầm. Phạm vi phá hoại của virus là rất lớn. Thông thường nhất, các virus thường gây ra mất mát dữ liệu, hư hỏng phần mềm và hư hỏng cả hệ điều hành.

- **Worms**: Có khả năng tự nhân bản trên chính nó mà không cần cấy vào một tập tin lưu trữ. Chúng còn thường sử dụng Internet để lây lan, do đó gây thiệt hại nghiêm trọng cho một mạng lưới về tổng thể, trong khi virus thường chỉ nhắm vào các tập tin trên máy tính bị nhiễm. Worm lây lan chủ yếu là do các lỗ hổng bảo mật của hệ thống.

- **Trojans**: Là những chương trình hoạt động núp dưới danh nghĩa một phần mềm hữu ích khác, và sẽ thực hiện các khi chương trình giả danh được kích hoạt bởi người sử dụng nhằm đánh cắp thông tin cá nhân, mở các cổng để hacker đột nhập, biến máy tính bị nhiễm thành nguồn phát tán thư rác hoặc trở thành công cụ tấn công một website nào đó. Không như Worm, Trojan horse không có khả năng tự nhân bản để lây lan, cũng như khả năng tự thực thi như virus.

- **Rootkits**: Chủ động "tàng hình" khỏi cặp mắt của người dùng, hệ điều hành và các chương trình anti-virus/anti-malware, rootkit là phần mềm độc hại rất khó bị phát hiện. Rootkit có thể được cài đặt bằng nhiều cách bao gồm việc khai thác lỗ hổng trong hệ điều hành hoặc lấy quyền quản trị máy tính.

- **Spyware**: hay phần mềm gián điệp là thuật ngữ thường được sử dụng để chỉ các phần mềm thực hiện hành vi nhất định như quảng cáo, thu thập thông tin người dùng hoặc thay đổi cấu hình máy tính của người dùng, nói chung là không có sự đồng thuận của người dùng.

- **Crimeware**: Một số nhà cung cấp sử dụng thuật ngữ "crimeware" để chỉ phần mềm độc hại được sử dụng để phạm tội, thường là một tội phạm liên quan đến lợi ích tài chính. Cũng giống như malware, crimeware là một phạm trù rộng gồm hàng loạt các phần mềm độc khác.
- **Adware**: Adware là một loại phần mềm độc hại tải xuống hoặc hiển thị pop-up quảng cáo trên thiết bị của người dùng. Thông thường, Adware không lấy cắp dữ liệu từ hệ thống, nhưng nó buộc người dùng phải xem những quảng cáo mà họ không muốn trên hệ thống. Một số hình thức quảng cáo cực kì gây khó chịu cho người dùng đó là tạo ra pop-up trên trình duyệt mà không thể đóng lại được. Đôi khi người dùng tự lây nhiễm adware được cài đặt mặc định khi tải về những ứng dụng khác mà không hề hay biết.

Tấn công tài nguyên mạng

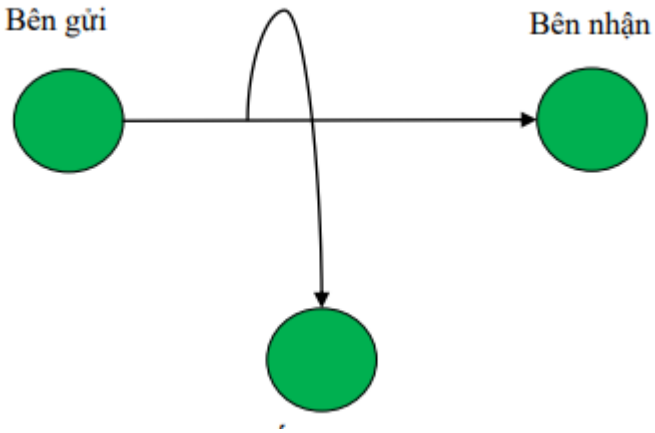
Quá trình truyền và gửi thông tin giữa 2 thực thể trong mạng có thể gặp các kiểu tấn công sau:

Hình 1.3: Luồng dữ liệu bình thường

- **Tấn công gián đoạn (Interuption)**: Đây là phương pháp tấn công phá vỡ tính sẵn sàng của hệ thống. Khi bị tấn công gián đoạn, việc truyền tin giữa 2 thực thể trong mạng sẽ gặp khó khăn. Cách tấn công này đôi khi còn gọi là tấn công từ chối dịch vụ (Denial of Services - DoS). Một vài cách tấn công phổ biến là: SYN flood, Smurf, Ping of Death,...

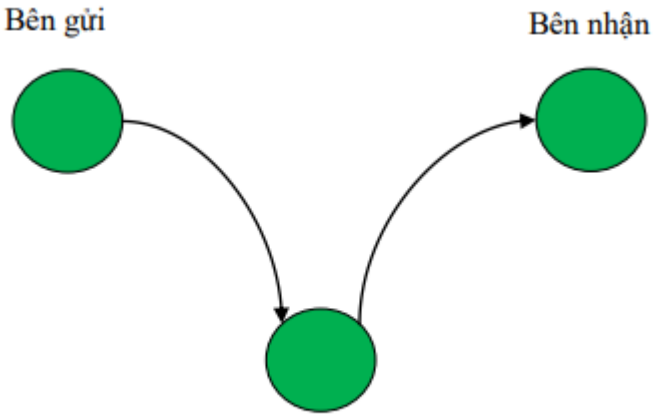
Hình 1.4: Tấn công gián đoạn

- **Tấn công nghe trộm (Interception)**: Kiểu tấn công này nhằm mục đích phá hủy tính bí mật của hệ thống. Thông tin được truyền đi giữa 2 thực thể trong mạng bị bên thứ ba nghe lén và thu thập. Một số phương pháp phổ biến dạng này là: Scan port, Packet sniffer,...



Hình 1.5: Tấn công nghe trộm

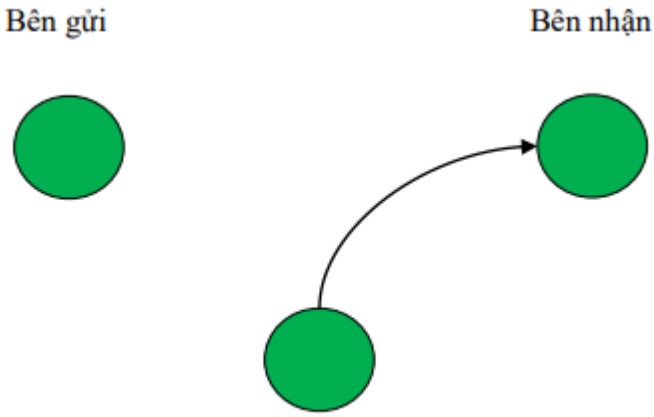
- **Tấn công sửa đổi (Modification)**: Đây là kiểu tấn công nhằm phá vỡ tính toàn vẹn của hệ thống. Dữ liệu trên kênh truyền sẽ bị sửa đổi so với ban đầu khiến cho thông tin bên nhận có được sẽ bị sai lệch so với khi gửi đi.



Hình 1.5 Tấn công thay đổi

Hình 1.6: Tấn công sửa đổi

- **Tấn công giả mạo (Fabrication)**: Kiểu này tấn công vào tính xác thực của hệ thống. Kẻ tấn công sẽ tìm cách mạo danh để gửi các thông điệp độc hại hoặc vượt qua các khâu xác thực.

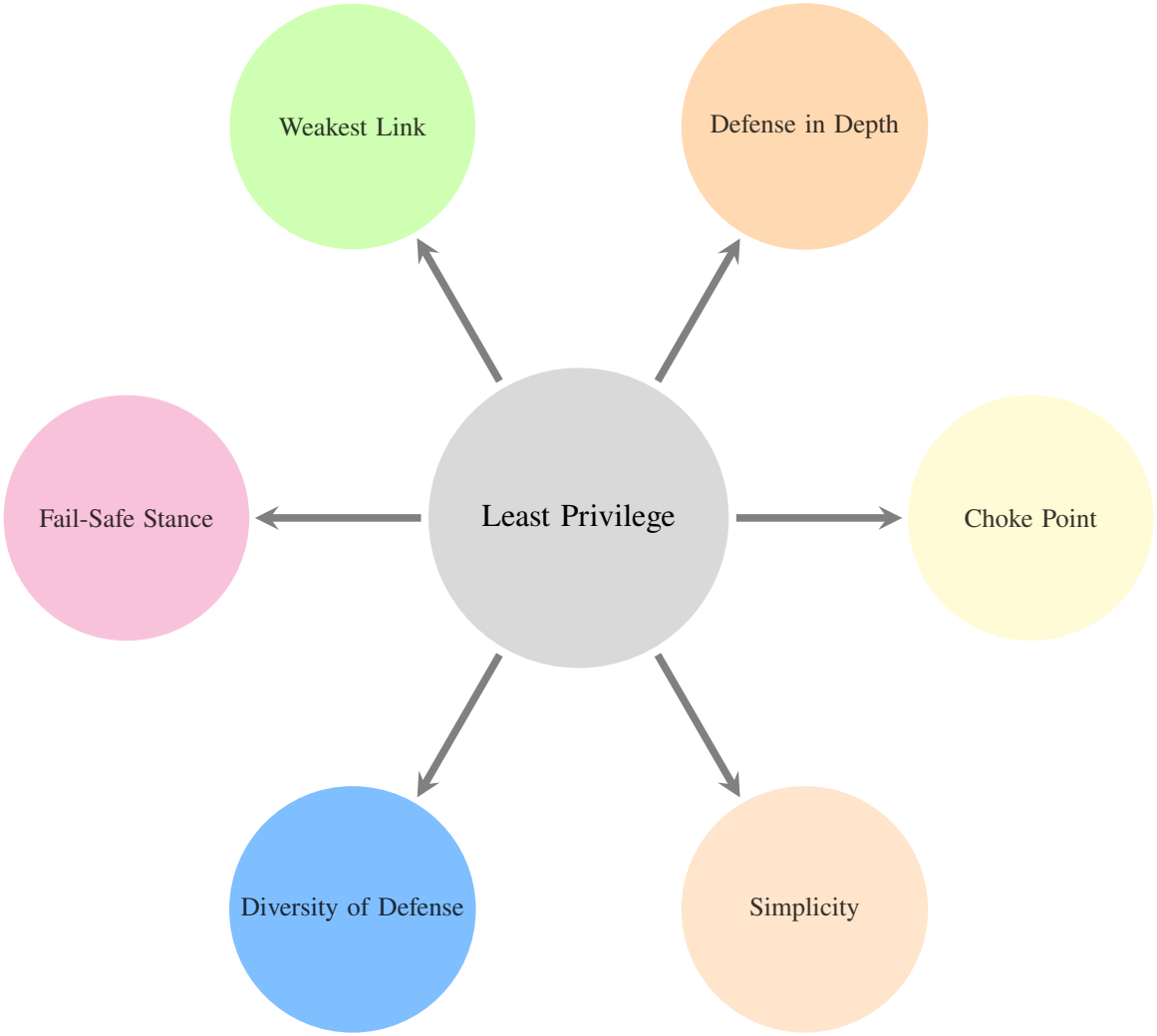


Hình 1.7: Tấn công giả mạo

1.1.3 Các biện pháp phòng chống tấn công, xâm nhập mạng

Chiến lược an toàn hệ thống

Các hệ thống mạng không nên sử dụng một phương pháp an toàn duy nhất mà nên có nhiều cơ chế án toàn khác nhau để chúng hỗ trợ lẫn nhau và có thể đảm bảo an toàn ở mức cao.



Hình 1.8: Các chiến lược an toàn hệ thống

- **Quyền tối thiểu (Least Privilege)**: Đây là nguyên tắc cơ bản nhất của an toàn thông tin. Quyền tối thiểu có nghĩa là bất kỳ đối tượng nào (người dùng, quản trị viên, chương trình, hệ thống, hoặc bất cứ điều gì) chỉ nên có các đặc quyền mà đối tượng cần thực hiện nhiệm vụ được giao - không được nhiều hơn. Quyền tối thiểu là một nguyên tắc quan trọng để hạn chế các cuộc tấn công và thiệt hại gây ra từ các cuộc tấn công đó.
- **Phòng thủ theo chiều sâu (Defense in Depth)**: Không phụ thuộc vào một cơ chế an toàn duy nhất, dù mạnh; thay vào đó, cài đặt nhiều cơ chế hỗ trợ nhau.
- **Điểm thắt (Choke Point)**: buộc các kẻ tấn công sử dụng một kênh hẹp, mà quản trị viên có thể giám sát và kiểm soát. Có thể có nhiều ví dụ về những điểm thắt trong thực tế: trạm thu phí trên cầu, máy an ninh tại siêu thị, quầy bán vé tại rạp chiếu phim.
- **Liên kết yếu nhất (Weakest Link)**: Chiến lược này dựa trên nguyên tắc: " Một dây xích chỉ chắc tại mắt duy nhất, một bức tường chỉcứng tại điểm yếu nhất".

  Kẻ phá hoại thường tìm những chỗyếu nhất của hệ thống để tấn công, do đó ta cần phải gia cố các yếu điểm của hệ thống. Thông thường chúng ta chỉ quan tâm đến kẻ tấn công trên mạng hơn là kẻ tiếp cận hệ thống, do đó an toàn vật lý được coi là yếu điểm nhất trong hệ thống.
- **Lập trường thất bại an toàn(Fail-Safe Stance)**: Trong phạm vi có thể, hệ thống nên có cơ chế thất bại an toàn. Có nghĩa là khi hệ thống bị sụp đổ, nó sẽ từ chối truy cập từ người dùng bất hợp

pháp và cả người dùng hợp pháp cho đến khi lỗi được khắc phục. Đây là một hình thức đánh đổi chấp nhận được.

- **Phòng thủ đa dạng (Diversity of Defense)**: Hơi giống so với phòng thủ theo chiều sâu, nhưng phòng thủ đa dạng có một chút khác biệt. Đó là hệ thống không những cần được phòng thủ nhiều tầng mà mỗi tầng lại phải có nhiều phương pháp phòng thủ khác nhau.
- **Đơn giản hóa (Simplicity)**: Hệ thống phải được đơn giản hóa. Điều này xuất phát từ 2 lý do. Thứ nhất, hệ thống càng đơn giản thì càng dễ hiểu. Sẽ không đánh giá được một hệ thống có an toàn hay không nếu không hiểu rõ về nó. Thứ hai, các chương trình càng phức tạp thì nguy cơ lỗi càng cao hơn. Đây là một vấn đề lớn trong an toàn thông tin.

Các biện pháp ngăn chặn xâm nhập

Không thể có một giải pháp an toàn tuyệt đối nên người ta thường phải sử dụng đồng thời nhiều mức bảo vệ khác nhau tạo thành nhiều hàng rào chắn đối với các hoạt động xâm nhập. Việc bảo vệ thông tin trên mạng chủyếu là bảo vệ thông tin cất giữ trong máy tính, đặc biệt là các máy chủ trên mạng. Bởi thế ngoài một số biện pháp nhằm chống thất thoát thông tin trên đường truyền mọi cố gắng tập trung vào việc xây dựng các mức rào chắn từ ngoài vào trong cho các hệ thống kết nối vào mạng.

- **Tường lửa**: Tường lửa được xem như một công cụ dùng để hạn chế truy cập vật lý tới máy tính. Chỉ có các thông điệp từ các host đã được xác thực mới được truyền qua tường lửa.
- **Bảo vệ vật lý**: Ngăn cản các truy nhập vật lý vào hệ thống. Thường dùng các biện pháp truyền thống như ngăn cấm tuyệt đối người không phận sự vào phòng đặt máy tính, dùng ổ khoá trên máy tính hoặc các máy trạm không có ổ đĩa...
- **Mã hóa dữ liệu**: Mã hóa có nhiệm vụ chuyển các thông điệp sang dạng không thể đọc được trước khi truyền đi. Trong quá trình truyền tải trong mạng, thông điệp luôn ở dạng không thể đọc được và chỉ có bên nhận mới có thể giải mã thông điệp đó.
- **Xác thực**: Mỗi người sử dụng muốn được tham gia vào mạng để sử dụng tài nguyên đều phải được đăng ký tên và mật khẩu trước.
- **Phân quyền truy nhập**: Lớp bảo vệ trong cùng nhằm kiểm soát các tài nguyên của mạng và quyền hạn trên tài nguyên đó. Dĩ nhiên là kiểm soát được các cấu trúc dữ liệu càng chi tiết càng tốt. Hiện tại việc kiểm soát thường ở mức file. Tùy vào từng người dùng cụ thể mà có những quyền khác nhau: đọc, ghi, sửa xóa file.
- **Hệ thống phát hiện xâm nhập mạng**: Phát hiện xâm nhiệp là quá trình theo dõi các sự kiện xảy ra trên hệ thống máy tính hay hệ thống mạng, phân tích chúng để tìm ra các dấu hiệu xâm nhập bất hợp pháp. Xâm nhập bất hợp pháp được định nghĩa là sự cố gắng tìm mọi cách để xâm hại đến tính toàn vẹn, tính sẵn sàng,... hoặc việc vượt qua các cơ chế bảo mật của hệ thống máy tính hoặc mạng đó. Việc xâm nhập có thể xuất phát từ một kẻ tấn công trên mạng Internet hoặc cũng có thể từ người dùng được phép trong hệ thống muốn chiếm thêm các quyền.

1.2 Sự cần thiết của phát hiện xâm nhập mạng

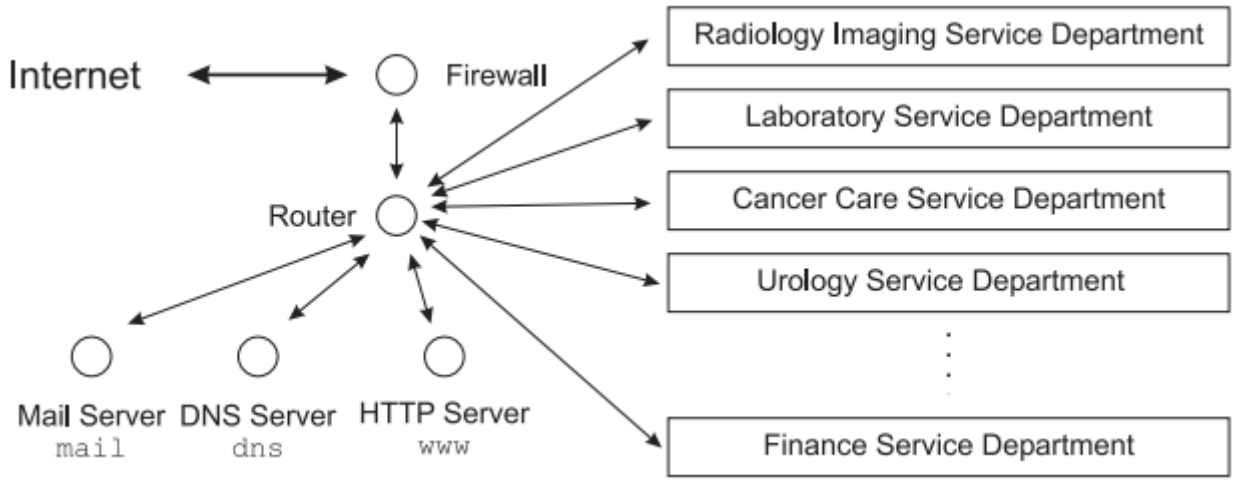
Bình thường, các máy tính chạy ở chế độ cho phép mở nhiều dịch vụ cùng lúc trên cùng một host. Những dịch vụ này cho phép kết nối giữa các host trong mạng với nhiều dạng kiến trúc, hệ điều hành cũng như các chức năng như FTPServer, Webserver,.... Điều này sẽ khiến hệ thống luôn tiềm ẩn các nguy cơ, lỗ hổng có thể khai thác như người dùng không được xác thực có thể truy cập vào hệ

thống, ăn cắp các thông tin hoặc thực thi các hành vi động hại. Đây là điều không mong muốn và cần được ngăn ngừa.

Như đã trình bày ở trên, tường lửa và mã hóa có thể ngăn ngừa việc khai thác. 2 kỹ thuật trên chỉ nâng cao khả năng bảo mật trong mạng nhưng nó không phải phương pháp có thể giải quyết tất cả mọi vấn đề. Một vài dịch vụ như HTTP service hay SMTP service luôn mở công khai (tường lửa cho phép mọi gói tin đi qua). Các yêu cầu nghiêm ngặt về thời gian thực thi cho các dịch vụ công khai thông dụng không phù hợp với việc sử dụng mã hóa, vì chúng làm cho việc truyền tải các thông điệp chậm hơn. Do đó, những dịch vụ kiểu này là mục tiêu phổ biến để tấn công.

Trong thực tế, các hệ thống thường có các dịch vụ chạy công khai ra ngoài như DNS Server, Mail Server, Webserver, ... Những dịch vụ như này sẽ phải đối mặt với các mối đe dọa đến từ hacker, những kẻ luôn muốn chiếm quyền truy cập trái phép vào hệ thống.

Giả sử với hệ thống mạng của một bệnh viện như sau: Bệnh viện muốn cung cấp thông tin của mình tới khách hàng qua Webserver www và cho phép người dùng gửi mail qua hệ thống Mail server. Hệ thống mạng của bệnh viện được bảo vệ bởi tường lửa, đảm bảo việc các kết nối từ Internet công cộng đều chỉ truy cập được vào mail, www và dns.



Hình 1.9: Hệ thống mạng bệnh viện

Hacker luôn muốn các cuộc tấn công phải được xuất phát từ nơi mà danh tính thực của chúng được che giấu, vì vậy hệ thống của công ty này là mục tiêu tốt đối với chúng. Trước khi có thể xâm nhập vào hệ thống, hacker thường thu thập các thông tin công khai có thể hữu ích cho việc tấn công nhiều nhất có thể. Do đó, kẻ tấn công thường sử dụng các nguồn công khai. Trong trường hợp của người viết luận, kẻ tấn công sử dụng các công cụ như nslookup, nmap để tìm các cổng và dịch vụ trong hệ thống. Bên cạnh đó, kẻ tấn công sẽ cố gắng thu thập các thêm các thông tin phụ như các dịch vụ đã biết ở trong các host (dns, mail, www) và thông tin liên lạc của quản trị viên. Giả sử hacker tìm được phiên bản của mail server, sau đó hacker sẽ tìm kiếm trên Internet các nguy cơ bảo mật của phiên bản này. Một lỗ hổng cho phép khai thác lỗ hổng tràn bộ đệm (buffer overflow) được tìm ra. Hacker thực hiện việc khai thác, chiếm quyền quản trị viên khi sử dụng command shell, giúp hacker có thể cài vào đó một backdoor. Với backdoor này, hacker có thể chèn mã độc, đánh cắp thông tin hoặc sử dụng các tài nguyên hệ thống vào các mục đích khác. Trong trường hợp này, tường lửa không giúp bảo vệ hệ thống khỏi kẻ tấn công ở bên ngoài mạng.

Ví dụ trên cho thấy các phương pháp bảo mật truyền thống không thể giúp ngăn ngừa hệ thống khỏi các cuộc tấn công từ bên ngoài. Điều này thậm chí có thể đe dọa đến cuộc sống khi hacker có

thể thay đổi hệ thống đèn giao thông, gây ra chết người. Hệ thống phát hiện xâm nhập mạng là một phương pháp bổ sung cho các khuyết điểm của các phương pháp bảo mật truyền thống. Các hệ thống này tìm cách phát hiện ra các sự cố, xâm nhập bằng cách phân tích các thông tin có thể thu thập được. Trái với phương pháp bảo mật một lớp đã giới thiệu trên, IDS là một phần của phương pháp Bảo mật theo chiều sâu (Defense -in-depth). Trong cách tiếp cận này, một loạt các kỹ thuật (với nhiều mức độ khác nhau) được sử dụng, khi một kỹ thuật thất bại, một kỹ thuật khác sẽ được sử dụng để ngăn cản cuộc tấn công. Bộ cảm biến (sensor) của IDS sẽ phân tích các dữ liệu thu thập được và xác định xem liệu có một cuộc tấn công diễn ra hay không.

Thậm chí khi kẻ tấn công có thể xâm nhập vào mạng, các hành động sau đó của hắn có thể được phát hiện với IDS. Việc quét cổng (scan port) của kẻ tấn công với mục đích tìm ra các host và port chứa lỗ hổng có thể bị phát hiện bởi các network-based IDS chỉ trong vài giây. Ngay ca khi kể tấn công có thể thực thi các phầm mềm bắt các gói tin với hi vọng tìm được account/password của nạn nhân thì việc này cũng có thể được cảnh báo. Rất nhiều loại cảnh báo khác nhau sinh ra bởi IDS được sử dụng để chống lại việc xâm nhập ngay từ những bước đầu tiên, người quản trị sẽ nhận được thông báo qua e-mail, SMS hoặc tương tự. Khi việc xâm nhập bị phát hiện, attacker có thể bị log out hoặc các tài khoản bị xâm nhập có thể được tắt ngay lập tức, các file quan trọng được khôi phục từ một hệ thống backup, các tiến trình không xác định sẽ được đóng lại. Tường lửa sẽ được tự động cấu hình lại, chỉ kết nối từ một số nguồn tin cậy. Hệ thống sẽ được chuyển về trạng thái an toàn khi người quản trị được cảnh báo kịp thời.

1.3 Phân loại phát hiện xâm nhập

1.3.1 Phân loại theo kỹ thuật phát hiện

Việc phát hiện xâm nhập được tiến hành nhờ vào quá trình giám sát các sự kiện xảy ra trong hệ thống máy tính hay mạng và phân tích xem có dấu hiệu của việc xâm nhậ hay không. Hệ thống phát hiện xâm nhập (IDS) có thể là hệ thống phần cứng hay phần mềm cho phép tự động hóa quá trình phát hiện hành vi xâm nhập.

Về cơ bản có ba phương pháp chính dựa theo các dấu hiệu/chữ ký; bất thường; và đặc tả, còn gọi là phân tích giao thức có trạng thái (stateful protocol analysis).

Các dấu hiệu thường là các mô hình hay chuỗi ký tự tương ứng với các vụ tấn công hay mối đe dọa đã biết. Để phát hiện IDS so sánh các mô hình với các sự kiện thu được để nhận biết việc xâm nhập. Phương pháp này còn được gọi là phương pháp dựa trên tri thức do sử dụng cơ sở tri thức về các hành vi xâm nhập trước đó.

Với phương pháp dựa trên bất thường thì sự bất thường được coi là sự khác biệt với hành vi đã biết bằng các lập hồ sơ các hành vi thông thường lập từ việc theo dõi các hoạt động thường xuyên, kết nối mạng, máy trạm hay người dùng qua một khoảng thời gian. Hệ thống phát hiện so sánh các hồ sơ với các sự kiện quan sát được để nhận biết các vụ tấn công nghiêm trọng

IDS phát hiện theo đặc tả nhận biết và theo dõi được trạng thái các giao thức (sự tương ứng giữa cặp yêu cầu/đáp ứng). Việc xây dựng đặc tả phụ thuộc vào nhà cung cấp giao thức.

Phần dưới đây giới thiệu hệ thống phân loại theo [7] bao gồm 5 cách tiếp cận chính như sau:

- Thống kê chủ yếu dựa trên thiết lập ngưỡng, giá trị trung bình, phương sai, và xác suất để xác định hành vi xâm nhập. Phương pháp này dựa trên độ đo khoảng cách, công thức Bayes, lý thuyết trò chơi [9] sử dụng nguồn dữ liệu từ hồ sơ người dùng, dữ liệu kiểm toán hay viêc sử dụng tài nguyên máy tính như bộ nhớ và bộ xử lý.

- Đối sánh mẫu chủ yếu dùng để phát hiện xâm nhập đã biết bằng cách đối sánh mẫu, giám sát bàn phím, mạng Petri như giới thiệu trong [10]. Nguồn dữ liệu sử dụng ngoài các dữ liệu kiểm toán cần thêm hồ sơ người dùng, bản ghi phím bấm và các mẫu luật từ hồ sơ người dùng và chính sách sử dụng.
- Luật bằng cách thuật toán dựa trên véc-tơ học máy SVM, khai phá dữ liệu như trong [2]. Phương pháp này cho phép tự động xây dựng và cập nhật mô hình phát hiện giúp cho hệ thống được linh hoạt và mềm dẻo hơn.
- Trạng thái dựa trên việc phân tích trạng thái bằng mô hình Markov, phân tích giao thức như trong [7].
- Kinh nghiệm sử dụng mạng nơ-ron, lô-gíc mờ, thuật toán gen như [11, 12]. Phương pháp này có thể sử dụng dữ liệu từ lưu lượng mạng, thông tin từ các vụ việc xâm nhập thành công trước đó.

1.3.2 Phân loại theo công nghệ

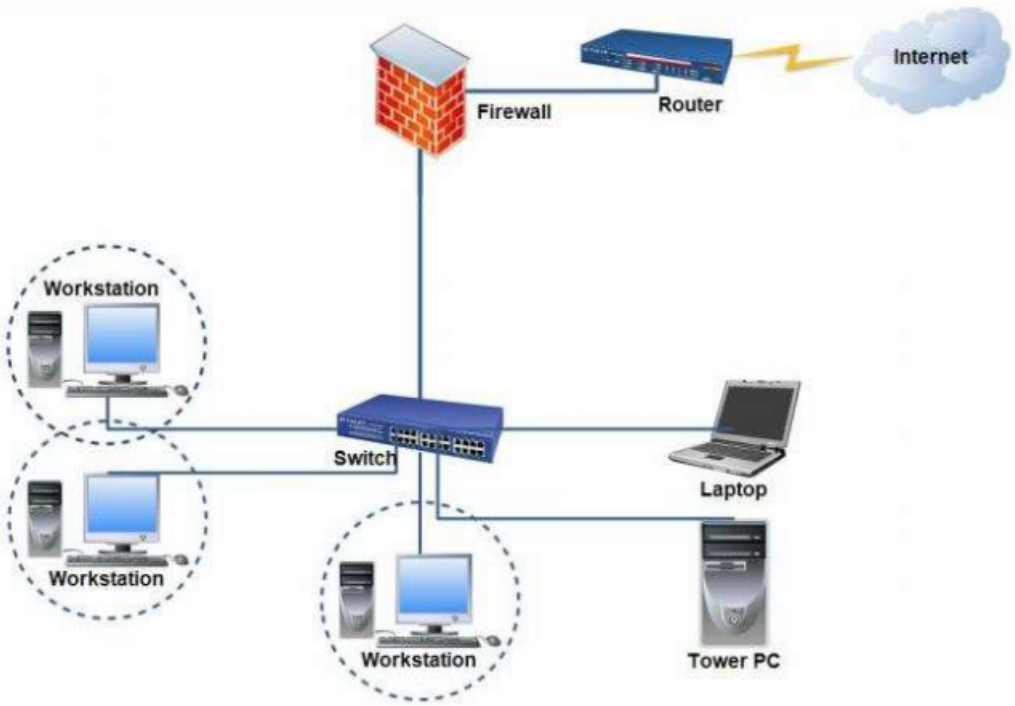
Xét về khía cạnh công nghệ để xây dựng và triển khai các hệ thống ngăn ngừa xâm có thể phân thành 4 loại cơ bản như dưới đây.

Phát hiện xâm nhập cho máy trạm HIDS

Mục tiêu của hệ thống này là giám sát và thu thập các đặc trưng về các máy trạm (host) chứa đựng các thông tin nhạy cảm, máy chủ và các hoạt động đáng ngờ.

Hệ thống có thể triển khai trên mạng thông thường hoặc có quản lý (an toàn) và thực hiện việc thu thập thông tin về các hoạt động của ứng dụng và lưu lượng mạng.

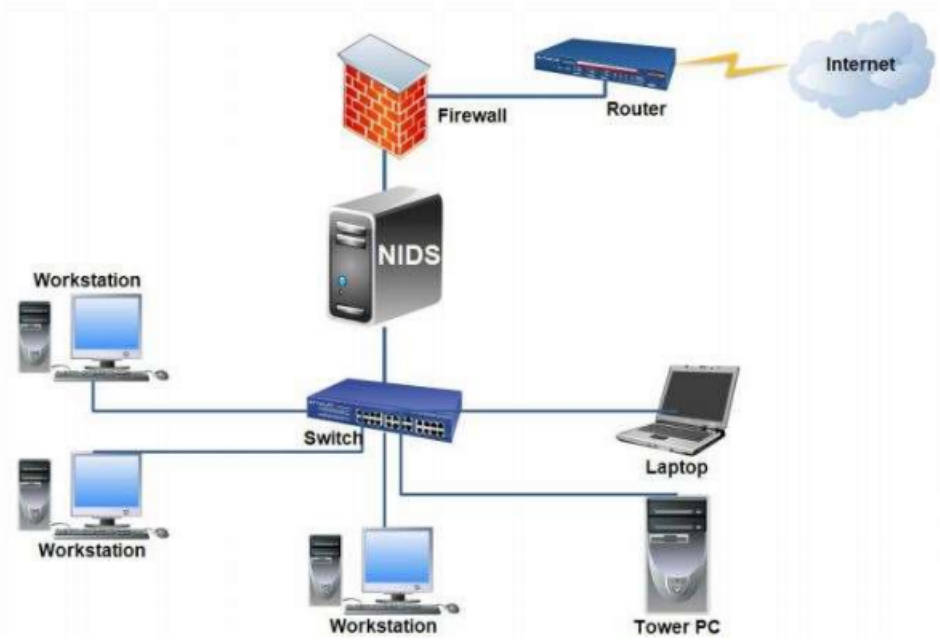
Hệ thống cung cấp các cảnh báo về các sự cố lớp ứng dụng, lớp vận chuyển và lớp mạng. Tuy nhiên độ chính xác của cảnh báo khó chuẩn xác do thiếu thông tin về ngữ cảnh mà máy tính và chương trình người dùng. Có thể xảy ra việc chậm trễ trong việc cảnh báo và hệ thống kiểu này phải sử dụng tài nguyên của máy trạm để hoạt động được. Trong một số tình huống có thể xung đột với các biện pháp kiểm soát hiện có như các chương trình quét vi-rút hay tường lửa.



Hình 1.10: HIDS

Phát hiện xâm nhập cho mạng NIDS

Hệ thống kiểu này thu thập các gói tin tại các phân đoạn mạng nhờ các cảm biến. Sau đó phân tích các hoạt động của ứng dụng và giao thức để phát hiện các hành vi đáng ngờ. Hệ thống thường được triển khai trong mạng có quản lý (an toàn) và thực hiện thu thập thông tin về các trạm, hệ điều hành, ứng dụng, và lưu lượng mạng. Hệ thống có thể cảnh báo việc tấn công, thăm dò lớp ứng dụng, lớp vận chuyển, lớp mạng, các dịch vụ ứng dụng bất thường, xung đột chính sách. Tuy nhiên, hệ thống không giám sát các giao thức mạng không dây WF. Về cơ bản có tỷ lệ nhầm FP-FN cao và không phát hiện được tấn công bên trong lưu lượng được mã hóa. Trong trường hợp hệ thống chịu tải nặng sẽ không thể phân tích đầy đủ.



Hình 1.11: NIDS

Phát hiện xâm nhập cho mạng không dây

Hệ thống phát hiện có tính năng giống như NIDS song tập trung vào lưu lượng mạng không dây: như mạng ad-hoc, mạng các cảm biến, mạng lưới. Hệ thống này có thể triển khai trên mạng có quản lý hoặc mạng bình thường và thực hiện việc thu thập thông tin về các thiết bị không dây WF. Hệ thống phát hiện thực hiện các cảnh báo các sự cố về các giao thức WF, thiết bị và mạng WF không an toàn, tấn công DOS, quét mạng, xâm phạm chính sách. Tuy nhiên, nhược điểm của hệ thống này là không thể giám sát các hoạt động của giao thức ở lớp trên và không tránh được kỹ thuật lẩn trốn. Mặt khác các cảm biến của hệ thống có thể bị tấn công chèn sóng vật lý và hệ thống không bù trừ được cho các giao thức không an toàn

Phát hiện xâm nhập qua phân tích hành vi mạng NBA

Việc phát hiện xâm nhập qua việc kiểm tra lưu lượng mạng để phát hiện tấn công khi có lưu lượng bất thường. Hệ thống này rất hữu ích trong việc chống thăm dò, dựng lại quá trình lây nhiễm của phần mềm độc hay tấn công DOS và có thể triển khai mạng có quản lý hay mạng bình thường. Hệ thống tiến hành thu thập thông tin về các máy trạm, hệ điều hành, các dịch vụ và đưa ra các cảnh báo các luồng lưu lượng bất thường, các dịch vụ ứng dụng bất thường, quét mạng và vi phạm chính sách [6].

1.3.3 Kết luận

Mỗi kỹ thuật phát hiện hành vi xâm nhập có điểm mạnh và yếu riêng. Hệ thống dựa trên dấu hiệu hiệu quả với việc phát hiện các tấn công biết trước. Trong khi đó hệ thống dựa trên luật gặp phải vấn đề cập nhật tri thức với các tấn công mới. Hệ thống dựa trên kinh nghiệm gặp khó khăn khi hoạt động ở chế độ thời gian thực do mất nhiều thời gian huấn luyện và đào tạo để hệ thống có khả năng ứng phó với các dạng tấn công mới. Việc có thông tin đầy đủ về các ưu và nhược điểm của mỗi phương pháp giúp lựa chọn, sử dụng hiệu quả các kỹ thuật ngăn ngừa và bố trí một cách phù hợp các hệ thống ngăn ngừa xâm nhập để tăng cường các thuộc tính an toàn của hệ thống.

1.4 Mục tiêu của đồ án

Trên cơ sở tìm hiểu về các kỹ thuật sử dụng trong phát hiện tấn công mạng, đồ án hướng đến xây dựng hệ thống phát hiện tấn công mạng sử dụng phương pháp học có giám sát dựa trên kỹ thuật machine learning mà cụ thể ở đây là thuật toán Boosted Tree .

Boosted Tree là một thuật toán học máy khá phổ biến trong việc giải quyết các bài toán phân lớp dữ liệu. Đây là một thuật toán mạnh mẽ, được áp dụng rất nhiều vào các bài toán thực tế. Đồng thời, Boosted Tree cũng là thuật toán được các nhà vô địch các cuộc thi về học máy và khoa học dữ liệu trên trang Kaggle [13].

Đối tượng áp dụng của nghiên cứu trong đồ án là tập dữ liệu UNSW-NB15. Đây là một trong những tập dữ liệu mới nhất dành cho việc nghiên cứu phát hiện xâm nhập mạng. Chi tiết của tập dữ liệu này sẽ được giới thiệu kỹ hơn tại mục 1 chương 3.

Chương 2

MÔ HÌNH BOOSTED TREE

Trong chương này, đồ án giới thiệu về phương pháp học máy có giám sát, tổng quan về phương pháp học Ensemble Learning và một mô hình cụ thể của phương pháp này là Boosted Tree.

2.1 Học có giám sát - Supervised Learning

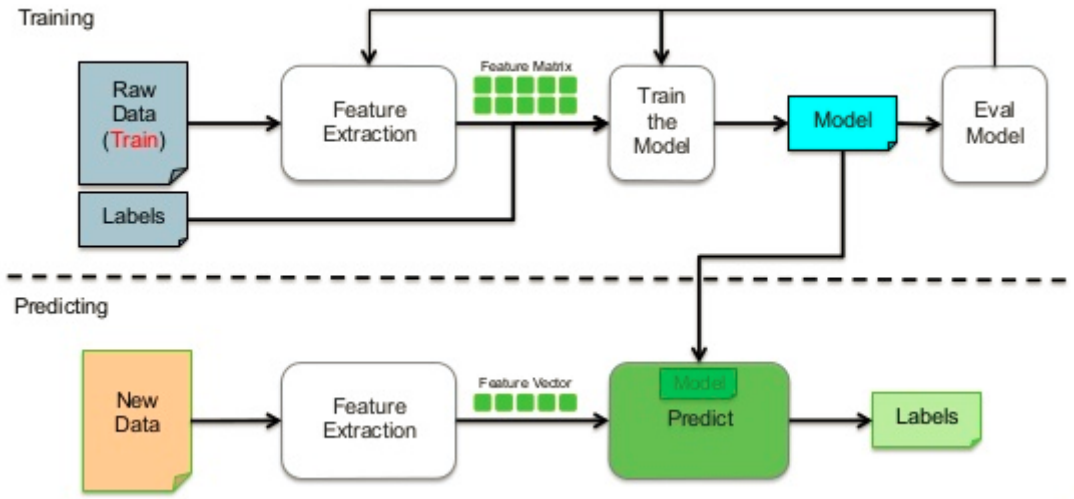
2.1.1 Giới thiệu

Học có giám sát (Supervised learning) là thuật toán dự đoán đầu ra (outcome) của một dữ liệu mới (new input) dựa trên các cặp (input, outcome) đã biết từ trước. Cặp dữ liệu này còn được gọi là (data, label), tức (dữ liệu, nhãn). Supervised learning là nhóm phổ biến nhất trong các thuật toán Machine Learning.

Một cách tổng quát, khi chúng ta có một tập hợp các biến đầu vào $\mathscr{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ và một tập hợp nhãn tương ứng $\mathscr{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$, trong đó $\mathbf{x}_i, \mathbf{y}_i$ là các vector. Các cặp dữ liệu biết trước $(\mathbf{x}_i, \mathbf{y}_i) \in \mathscr{X} \times \mathscr{Y}$ được gọi là tập training data (dữ liệu huấn luyện). Từ tập traing data này, chúng ta cần tạo ra một hàm số ánh xạ mỗi phần tử từ tập $\mathscr{X}$ sang một phần tử (xấp xỉ) tương ứng của tập $\mathscr{Y}$:

$$\mathbf{y}_i \approx f(\mathbf{x}_i), \quad \forall i = 1, 2, \ldots, N$$

Mục đích là xấp xỉ hàm số f thật tốt để khi có một dữ liệu $\mathbf{x}$ mới, chúng ta có thể tính được nhãn tương ứng của nó $\mathbf{y} = f(\mathbf{x})$.



Hình 2.1: Mô hình hoạt động của Học có giám sát

Bài toán Supervised learning còn được chia thành 2 bài toán chính:

- **Phân loại - Classification**: các label của input data được chia thành một số hữu hạn nhóm. Ví dụ: Dự đoán kiểu tấn công dựa vào một luồng traffic cho trước. Dự đoán một tin nhắn là spam hay không, ...

- **Hồi quy - Regression**: Các label không được chia thành nhóm mà có 1 giá trị cụ thể. Ví dụ: Ước lượng giá của một căn nhà dựa trên diện tích.

Với mô hình hồi quy tuyến tính, $\hat{y}$ được tính bằng:

$$\hat{y_i} = \sum_j \theta_j x_{ij} \tag{2.1}$$

Ta có $\hat{y_i}$ là giá trị dự đoán được dựa trên input x và trọng số θ - thành phần cần được học dựa trên bộ training data.

2.1.2 Hàm mục tiêu

Dựa vào các cách hiểu và sử dụng y_i khác nhau, các mô hình khác nhau được ra đời: phân loại, hồi quy, ... Khi huấn luyện một mô hình học có giám sát, tập dữ liệu (dataset) được chia làm 2 phần: *tập huấn luyện* (training set) và *tập kiểm thử* (testing set).

- **Tập huấn luyện**: được sử dụng để học - tức quá trình tối ưu hóa tham số Θ, xây dựng mô hình.

- **Tập kiểm thử**: được sử dụng để đánh giá độ tốt của mô hình. Dữ liệu test được giả sử là không được biết trước, và không được sử dụng để xây dựng các mô hình Machine Learning.

Huấn luyện mô hình là quá trình tìm một phương pháp huấn luyện trên tập huấn luyện sao cho mô hình dự đoán tốt trên tập kiểm thử. Người ta thường ít quan tâm đến độ tốt của mô hình trên tập huấn luyện bởi vì nó thường rất cao. Độ tốt trên tập huấn luyện chỉ thể hiện được **khả năng ghi nhớ** của mô hình về những gì đã nhìn thấy. Với một mô hình tốt thật sự, ta cần thêm **khả năng tổng quát hóa**, chính là việc dự đoán tốt trên dữ liệu chưa hề được nhìn thấy.

Để làm được 2 điều trên, hàm mục tiêu được sử dụng trong quá trình học. Hàm mục tiêu được định nghĩa bởi công thức sau:

$$Obj(\Theta) = L(\Theta) + \Omega(\Theta) \tag{2.2}$$

Với L được gọi là *hàm mất mát huấn luyện* (training loss function) và Ω được gọi là *hàm chuẩn tắc* (regualarization). Hàm mất mát phản ánh độ tốt của mô hình trên tập huấn luyện. L được tính bởi công thức:

$$L = \sum_{i=1}^{n} l(y_i, \hat{y}_i) \tag{2.3}$$

Hàm mất mát phổ biến thường được sử dụng là mean squared error:

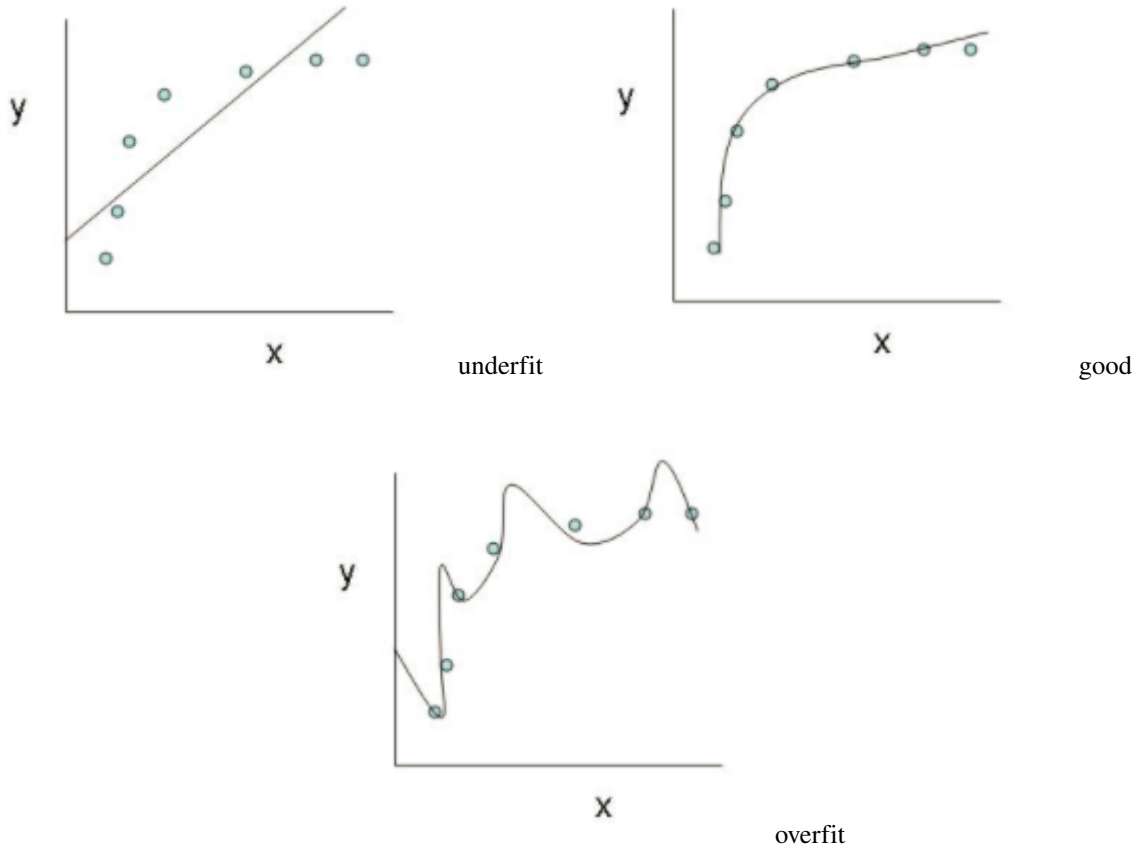
$$l(y_i, \hat{y}_i) = (y_i - \hat{y}_i)^2 \tag{2.4}$$

Một hàm mất mát phổ biến khác sử dụng trong phương pháp logistic regression là:

$$l(y_i, \hat{y}_i) = y_i \ln(1 + e^{-\hat{y}_i}) + (1 - y_i) \ln(1 + e^{\hat{y}_i}) \tag{2.5}$$

L càng nhỏ có nghĩa mô hình càng *khớp* với tập huấn luyện. Tuy nhiên, trong thực tế, việc một mô hình quá *khớp* với dữ liệu sẽ bị phản tác dụng. Việc quá khớp này có thể dẫn đến việc dự đoán nhầm nhiễu, và chất lượng mô hình không còn tốt trên dữ liệu kiểm thử nữa. Mô hình trở nên quá phức tạp để mô phỏng dữ liệu huấn luyện. Điều này đặc biệt xảy ra khi lượng dữ liệu huấn luyện quá nhỏ trong khi độ phức tạp của mô hình quá cao. Hiện tượng này gọi là *quá khớp* (overfitting).

Hàm regularization được sinh ra để giải quyết vấn đề này. Regularization, một cách cơ bản, là thay đổi mô hình để tránh quá khớp trong khi vẫn giữ được tính tổng quát của nó (tính tổng quát là tính mô tả được nhiều dữ liệu, trong cả tập huấn luyện và kiểm thử). Điều này làm cho mô hình *đơn giản hơn* mặc dù giá trị của hàm mất mát có tăng lên. Tuy nhiên nếu regularization quá lớn sẽ làm mô hình xảy ra hiện tượng *không kxhớp* (underfitting). Ngược lại với quá khớp, underfitting sẽ khiến khả năng dự đoán của mô hình trên tập huấn luyện giảm.



Hình 2.2: Ví dụ minh họa về overfit và underfit

Như vậy, hàm mục tiêu sẽ giúp mô hình đáp ứng được tính dự đoán cao và tính đơn giản. Việc làm này còn được gọi là bias-variance tradeoff[29].

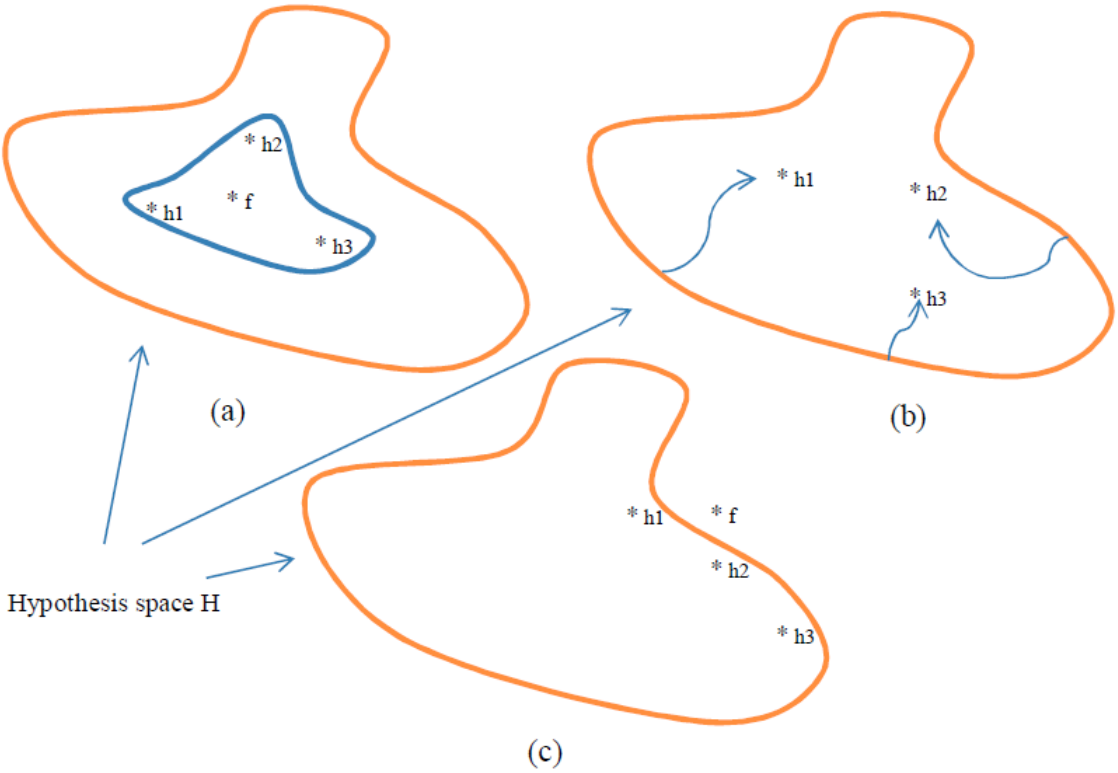
2.2 Ensemble Learning

2.2.1 Giới thiệu

Trong thực tế, dữ liệu thu thập được từ nhiều nguồn có thể khác nhau về bản chất. Một phân loại được tạo ra từ một thuật toán học có thể đạt được độ chính xác cao với một số bộ dữ liệu này nhưng lại có tỉ lệ sai cao hơn trong một số bộ dữ liệu khác. Cụ thể, việc áp dụng các thuật toán học khác nhau vào một tập dữ liệu có thể tạo ra kết quả phân loại khác nhau. Không có thuật toán học đơn lẻ nào thực hiện tốt trên tất cả các bộ dữ liệu. Thực nghiệm cho thấy các thuật toán đơn giản như K-neighbour neighbor (kNN) trong một số trường hợp có thể có độ chính xác cao hơn so với các phương pháp phức tạp hơn như Decision Tree hoặc Support Vector Machine (SVM).
Một cách tiếp cận khác để thu được hiệu suất cao trong việc phân loại là kết hợp nhiều thuật toán học với nhau để có được độ chính xác cao hơn so với một thuật toán duy nhất. Nhìn chung, rất khó để biết một thuật toán học nào phù hợp cho một tập dữ liệu cụ thể. Phương pháp Ensemble kết hợp các mô hình khác nhau với mục tiêu đạt được tỷ lệ lỗi phân loại thấp hơn so với sử dụng một mô hình duy nhất. Khái niệm "mô hình" trong các phương pháp kết hợp được hiểu theo nghĩa rộng, bao gồm không chỉ việc thực hiện các thuật toán học khác nhau, hoặc tạo ra nhiều tập huấn luyện cho cùng một thuật toán học, mà còn là sinh ra các bộ phân loại chung kết hợp với nhau để nâng cao độ chính xác phân loại.

2.2.2 Cách thức hoạt động

Giả sử có N quan sát. Một thuật toán học có đầu ra là một bộ phân loại là một hàm hypothesis thể hiện quan hệ f giữa các quan sát và các nhãn. Nhãn của quan sát x sẽ được dự đoán dựa trên hypothesis. Một hệ thống gồm K thuật toán học sẽ cho ra K hàm hypothesis, ký hiệu bởi $h_1, h_2, ..., h_k$. Dietterich[7] đã cho thấy 3 lý do vì sao một phương pháp ensemble lại tốt hơn một bộ phân loại đơn lẻ. (Hình 2.3)



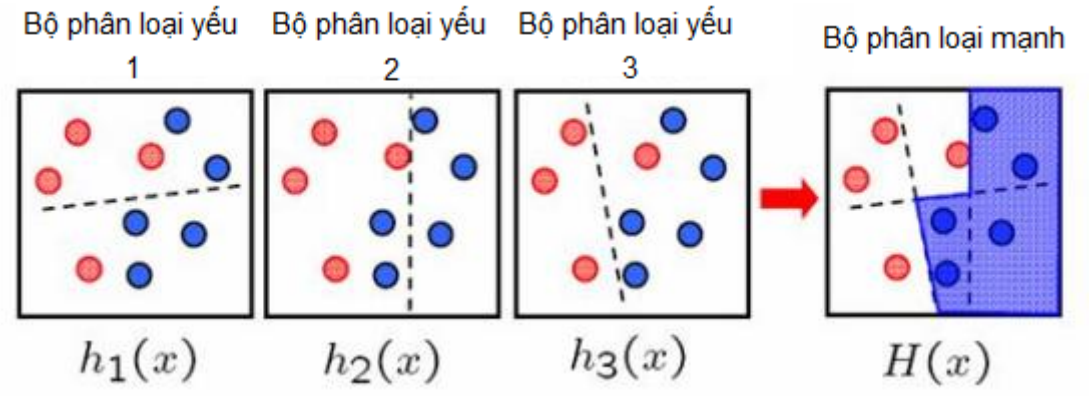
Hình 2.3: Ba lý do phương pháp Ensemble tốt hơn

(a) Tính thống kê (Statistical): Trong một số trường hợp, số lượng quan sát trong một tập huấn luyện là không đủ so với kích thước của không gian hypothesis H bao gồm tất cả các hypothesis được tạo ra bởi một thuật toán học. Do đó, thuật toán sẽ tìm kiếm trên nhiều hypothesis có cùng tỷ lệ lỗi. Bằng cách sử dụng phương pháp emsemble, chúng ta có thể thực hiện bình chọn bình chọn trong số tất cả các thuật toán học.

(b) Khả năng tính toán (Computational): Rất nhiều thuật toán sử dụng phương pháp tìm kiếm cục bộ để thu được giải pháp tối ưu cục bộ. Trong phương pháp ensemble, bằng cách thay đổi điểm bắt đầu (starting point) của các thuật toán học, chúng ta có thể thu được một hàm xấp xỉ tốt hơn thể hiện quan hệ f giữa vector đặc trưng x và nhãn y_x so với một thuật toán đơn lẻ.

(c) Tính đại diện (Representational): Quan hệ f_x giữa x và y_x trong một vài trường hợp không thể mô hình hóa bởi một hypothesis đơn. Với phương pháp ensemble, điều này có thể giải quyết bằng cách kết hợp nhiều hypothesis lại.

2.3 Boosting



Hình 2.4: Boosting

Phương pháp ensemble learning nổi tiêng nhất là Boosting, phương pháp mà Breiman [14] cho rằng có quan trọng nhất trong việc phân loại trong thế kỷ 20.

Ý tưởng của cách tiếp cận này là kết hợp các giải thuật học yếu (weak learner), có độ chính xác lớn hơn 50%, tức là lớn hơn đoán ngẫu nhiên, nhằm đạt được tỷ lệ lỗi thấp hơn các thuật toán học yếu chạy trên cùng tập huấn luyện (xem Algorithm 1).

Algorithm 1 Thủ tục chung cho các phương pháp Boosting

Input: Dataset D

Thuật toán học K

Số lần lặp M

1: Khởi tạo $D_1 = D$

2: **for** $m = 1 \ldots M$ **do**

3: Học hypothesis $h_m = Learn(K, D_m)$;

4: Tính error rate $e_m = P_{x\ D_m}(h_m(x) \neq y_x)$

$D_{t+1} = Adjust\_distribution(D_m, e_m)$

5: **end for**

Output: $H = Combine\_outputs(h_1, h_2, ..., h_M)$

2.4 Boosted Tree

Là một phương pháp Boosting, Boosted Tree xuất phát từ ý tưởng kết hợp các mô hình cây phân loại để tạo ra một mô hình mới tốt hơn.

2.4.1 Mô hình cây - Tree model

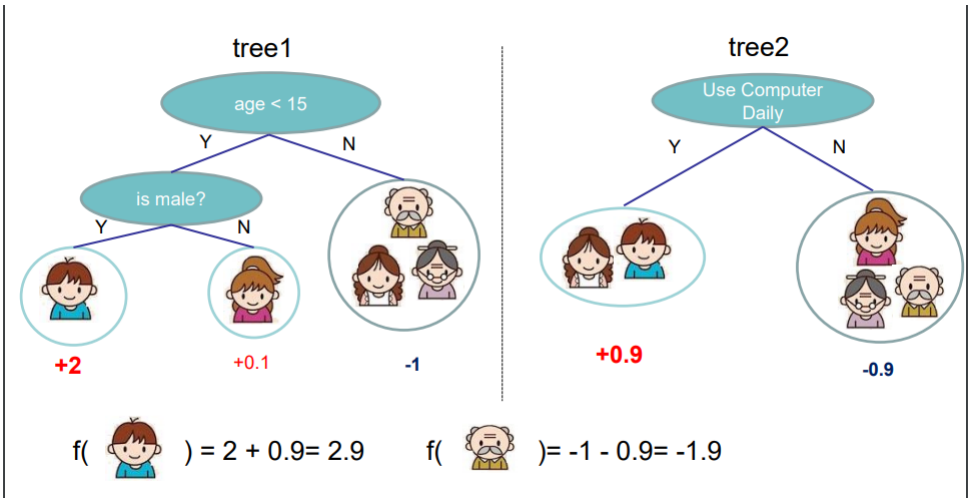
Một mô hình cây là một cấu trúc dữ liệu gồm các nút (node) liên kết với nhau. Nút ở trên cùng của cây gọi là gốc (root). Mỗi nút lại gồm nhiều các nhánh con. Nút ở dưới cùng của cây gọi là lá (leaf). Các mô hình cây thường có khả năng dự đoán hạn chế. Nhưng khi kết hợp các mô hình cây lại với nhau như trong Bagged trees (1996), Random Forest (2001) của Breiman [14], hoặc trong thuật toán Boosted Tree để tạo thành một bộ phân loại mới có khả năng dự đoán tốt hơn. Có nhiều cây *yếu* như Decision Tree (Cây quyết định) , CART (classification and regression trees - Cây phân loại và hồi quy),...

Phương pháp Boosted Tree sử dụng các cây CART làm các cây cơ sở. CART có một chút khác biệt

so với cây quyết định bình thường, mỗi lá của cây là một giá trị thực. Giá trị này cho chúng ta nhiều diễn giải phong phú hơn so với việc phân loại thông thường. Điều này sẽ làm cho việc tối ưu hóa đơn giản hơn.

CART được tạo nên bằng một giải thuật tham lam theo cách tiếp cận từ trên xuống (top-down) sử dụng phân chia nhị phân. Từ gốc, các cách chia nhánh con được đưa ra để chọn lựa, cách chia nhánh nào tối thiểu hóa được hàm mục tiêu sẽ được chọn. Thủ tục này được thực hiện đệ quy cho đến khi thỏa mãn một số tiêu chí dừng.

Ví dụ dưới đây sử dụng 2 cây để phân loại các thành viên gia đình vào các lá khác nhau và gán các giá trị cho mỗi lá. Điểm của mỗi thành viên trong gia đình được tính bằng tổng điểm thu được ở mỗi cây.



Hình 2.5: Ví dụ minh họa về Boosted Tree

Việc tính điểm có thể tổng quát hóa thành:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in \mathcal{F} \tag{2.6}$$

với K là số lượng cây, f là một hàm thuộc không gian hàm $\mathcal{F}$, và $\mathcal{F}$ là một tập hợp các CART. Các hàm mục tiêu cần được tối ưu được ở (2.2) viết lại thành:

$$\text{obj}(\theta) = \sum_{i}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{2.7}$$

2.4.2 Học bổ sung - Additive training

Các hàm f là cấu trúc cây chứ không phải các vector số học. Vì vậy để huấn luyện, không thể sử dụng các phương pháp truyền thống với gradient như gradient descent. Không thể huấn luyện tất cả các cây đồng thời, vì vậy, giải pháp được đưa ra cho việc tìm các f là sử dụng chiến thuật: mỗi lần

học đưa thêm 1 cây mới vào. Giá trị dự đoán được tại cây thứ t được tính bởi:

$$\begin{aligned}
\hat{y}_i^{(0)} &= 0 \\
\hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\
\hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\
&\cdots \\
\hat{y}_i^{(t)} &= \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)
\end{aligned} \tag{2.8}$$

Tại bước thứ t, hàm mục tiêu được tính bởi:

$$\begin{aligned}
\text{obj}^{(t)} &= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^{t} \Omega(f_i) \\
&= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + constant
\end{aligned} \tag{2.9}$$

Nhiệm vụ tại lần học thứ t là tìm được cây $f_{(t)}$ để tối thiểu hóa hàm mục tiêu $\text{obj}^{(t)}$.
Từ khai triển Taylor[4]:

$$f(x+\Delta x) \simeq f(x) + f^{'}(x)\Delta x + \frac{1}{2} f^{''}(x)\Delta x^2 \tag{2.10}$$

Hàm mục tiêu được viết lại thành:

$$\text{obj}^{(t)} = \sum_{i=1}^{n} [l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) + constant \tag{2.11}$$

với h_i và g_i được định nghĩa:

$$\begin{aligned}
g_i &= \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) \\
h_i &= \partial^2_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})
\end{aligned} \tag{2.12}$$

Sau khi bỏ qua hằng số, hàm mục tiêu tại bước thứ t trở thành:

$$\sum_{i=1}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) \tag{2.13}$$

Hàm (2.13) là mục tiêu mới cần được tối ưu cho cây tại bước thứ t. Cách viết hàm mục tiêu kiểu này có một vài lợi ích sau:

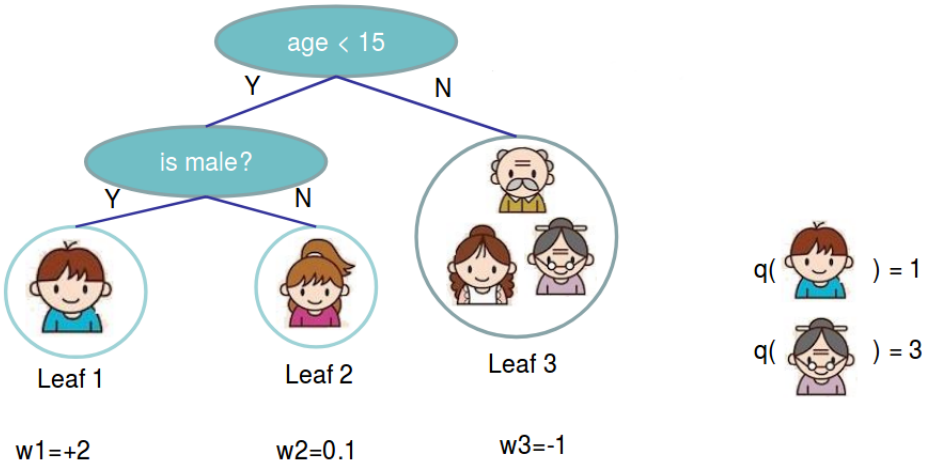
- Biết được các thành phần cần học để hội tụ.
- g_i và h_i sẽ định nghĩa hàm mất mát
- Quá trình học chỉ phục thuộc vào g_i và h_i
- Có thể tùy biến việc sử dụng các hàm mất mát khác nhau như: Logistic loss, square loss

2.4.3 Hàm mục tiêu

Cây sẽ được định nghĩa bởi 2 thành phần: một vector trọng số tại các lá và một hàm ánh xạ từ thực thể ra chỉ số lá.

$$f_t(x) = w_{q(x)}, w \in R^T, q : R^d \to \{1, 2, \cdots, T\}. \tag{2.14}$$

Với w là vector trọng số trên các lá, q là một hàm ánh xạ mỗi điểm dữ liệu (data point) tới lá đại diện, và T là số lượng lá.



Hình 2.6: Ví dụ về cây theo cách định nghĩa mới

Trong một số công cụ nổi tiếng như XGBoost[3], hàm regularization được định nghĩa như sau:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{2.15}$$

Như vậy, sau khi tái định nghĩa cây, hàm mục tiêu tại cây thứ t được tính bởi:

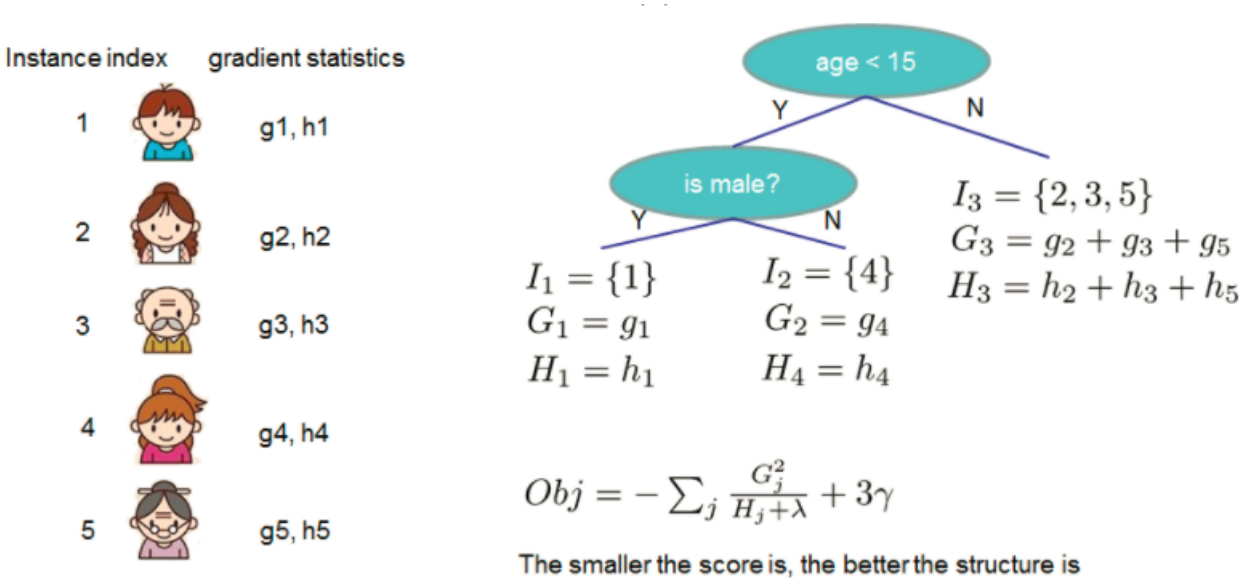
$$\begin{aligned} Obj^{(t)} &\approx \sum_{i=1}^{n} [g_i w_{q(x_i)} + \frac{1}{2} h_i w_{q(x_i)}^2] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \\ &= \sum_{j=1}^{T} [(\sum_{i \in I_j} g_i) w_j + \frac{1}{2} (\sum_{i \in I_j} h_i + \lambda) w_j^2] + \gamma T \end{aligned} \tag{2.16}$$

với $I_j = \{i | q(x_i) = j\}$ là tập hợp các chỉ số của các data point được gán cho lá thứ j. Đặt $G_j = \sum_{i \in I_j} g_i$ và $H_j = \sum_{i \in I_j} h_i$. (2.16) trở thành:

$$\text{Obj}^{(t)} = \sum_{j=1}^{T} [G_j w_j + \frac{1}{2}(H_j + \lambda) w_j^2] + \gamma T \tag{2.17}$$

Theo phương trình (2.17), ta có $G_j w_j + \frac{1}{2}(H_j + \lambda) w_j^2$ là một hàm bậc 2 đơn biến nên w_j tốt nhất với các $q(x)$ cho trước và nhỏ nhất thu được là:

$$\begin{aligned} w_j^* &= -\frac{G_j}{H_j + \lambda} \\ \text{obj}^* &= -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T \end{aligned} \tag{2.18}$$



Hình 2.7: Ví dụ cách tính hàm mục tiêu

2.4.4 Học cấu trúc cây

Để tìm cây tốt nhất tại mỗi lần lặp, phương án lý tưởng là duyệt tất cả các cấu trúc cây có thể. Nhưng điều này là bất khả thi vì số lượng cây rất lớn. Để giải quyết vấn đề này, tại mỗi mức, cây sẽ được tối ưu. Việc phân cây được thực hiện dựa trên cách tính *Gain* mà mỗi lần tách lá thu được:

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] - \gamma \tag{2.19}$$

Phương trình (2.19) bao gồm 4 thành phần: Điểm tại lá trái, điểm tại lá phải, điểm tại lá ban đầu và regularization khi thêm lá mới. Nếu gain nhỏ hơn γ, việc phân nhánh là không cần thiết. Cách làm này được gọi là phương pháp cắt tỉa trên các mô hình cây. Với các dữ liệu thực, phương án tìm kiếm tối ưu đó là sắp xếp các thuộc tính tăng dầnvà thực hiện tìm kiếm để quyết định giá trị nào là tốt nhất để phân tách.

Algorithm 2 Thuật toán Tree Boosting

Input: Dataset D

Hàm mất mát L

Số lượng cây M

Số lượng lá T

Learning rate η

Khởi tạo $\hat{f}^{(0)}(x) = \hat{f}_0(x) = \hat{\theta}_0 = argmin_\theta \sum_{i=1}^{n} L(y_i, \theta)$;

for $m = 1 \dots M$ **do**

$\hat{g}_m(x_i) = [\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}]_{f(x)=\hat{f}^{(m-1)}}$;

$\hat{h}_m(x_i) = [\frac{\partial^2 L(y_i, f(x_i))}{\partial f(x_i)}]_{f(x)=\hat{f}^{(m-1)}}$;

Xác định $\{\hat{R}_{jm}\}_{j=1}^{T}$ bằng cách tìm cách chia sao cho cực đại hóa $Gain = \frac{1}{2}[\frac{G_L^2}{H_L} + \frac{G_R^2}{H_R} - \frac{G_{jm}^2}{H_{jm}}]$;

Xác định trọng số lá $\{\hat{w}_{jm}\}_{j=1}^{T}$ cho cấu trúc học được bằng cách $\hat{w}_{jm}^{T} = -\frac{G_{jm}}{H_{jm}}$

$\hat{f}_m(x) = \eta \sum_{j=1}^{T} \hat{w}_{jm} I(x \in \hat{R}_{jm})$;

$\hat{f}^{(m)}(x) = \hat{f}^{(m-1)}(x) + \hat{f}_m(x)$;

end for

Output: $\hat{f}(x) \equiv \hat{f}^{(M)}(x) = \sum_{m=0}^{M} \hat{f}_m(x)$

2.4.5 Xử lý dữ liệu thiếu

Nhiều giải thuật gặp vấn đề khi xử lý dữ liệu thiếu. Thường những điểm dữ liệu (data point) bị thiếu dữ liệu sẽ bị bỏ qua hoặc sẽ được tìm cách thêm giá trị tại bước tiền xử lý dữ liệu.

Boosted Tree xử lý dữ liệu thiếu bằng cách học các hướng mặc định (default directions). Tại mỗi nút, sẽ có 2 hướng, trái hoặc phải. Trong quá trình dự đoàn, khi gặp dữ liệu bị thiếu, hướng mặc định sẽ được chọn. Khi gặp thiếu dữ liệu trong quá trình huấn luyện, hướng mặc định sẽ được học theo hướng làm tối thiểu hóa hàm mục tiêu. [18]

Chương 3

MÔ HÌNH BOOSTED TREE CHO PHÁT HIỆN XÂM NHẬP MẠNG

Chương này mô tả cách thiết lập thực nghiệm, đưa ra các mô hình thực nghiệm, giới thiệu các công cụ được sử dụng trong bài toán, kết quả thực nghiệm và phân tích đánh giá.

3.1 Giới thiệu bộ dữ liệu UNSW-NB15

Một hệ thống phát hiện xâm nhập dựa trên mạng (NIDS)giám sát các luồng dữ liệu để xác định có xuất hiện xâm nhập hay không. Như đã đề cập, NIDS được phân thành 2 loại, signature based - giám sát dựa trên chữ ký và anomaly based - giám sát truy nhập dựa trên bất thường. Signature based sẽ so khớp mỗi traffic flow với một loại tấn công đã biết để phát hiện ra xâm nhập. Ngược lại, với phương pháp Anomaly based, một hồ sơ được tạo ra với các hành vi "bình thường", tất cả các hành vi lệch với hồ sơ đều được coi là tấn công. Các Signature based NIDS sẽ không phát hiện được các loại tấn công chưa biết, do đó việc sử dụng Anomaly based được khuyến nghị.

Để đánh giá hiệu quả của NIDS, chúng ta cần một bộ dữ liệu gồm các hành vi bình thường và bất thường. Các bộ dữ liệu như KDDCUP 99 [19] và NSLKDD [20] được sử dụng rộng rãi để đánh giá hiệu năng của NIDS. Tuy nhiên, qua một vài nghiên cứu [21][22][23][24], việc đánh giá NIDS qua các bộ dữ liệu này không phản ánh kết quả thực tế vì một số lý do. Thứ nhất, bộ dữ liệu KDD 99 chứa một số lượng lớn các bản ghi thừa trong tập huấn luyện. Các bản ghi dư thừa này ảnh hưởng lớn đến kết quả phát hiện. Thứ hai, trong bộ dữ liệu này thiếu nhiều các bản ghi, điều này làm thay đổi bản chất của dữ liệu. Thứ ba, tuy bộ NSLKDD, một phiên bản cải tiến của KDDCUP 99, đã giải quyết một số vấn đề về mất cân bằng dữ liệu giữa các bản ghi bình thường/bất thường hoặc là các giá trị bị thiếu. Tuy nhiên, bộ dữ liệu này vẫn không phải là một bộ dữ liệu giúp đánh giá toàn diện NIDS trong môi trường tấn công hiện đại.

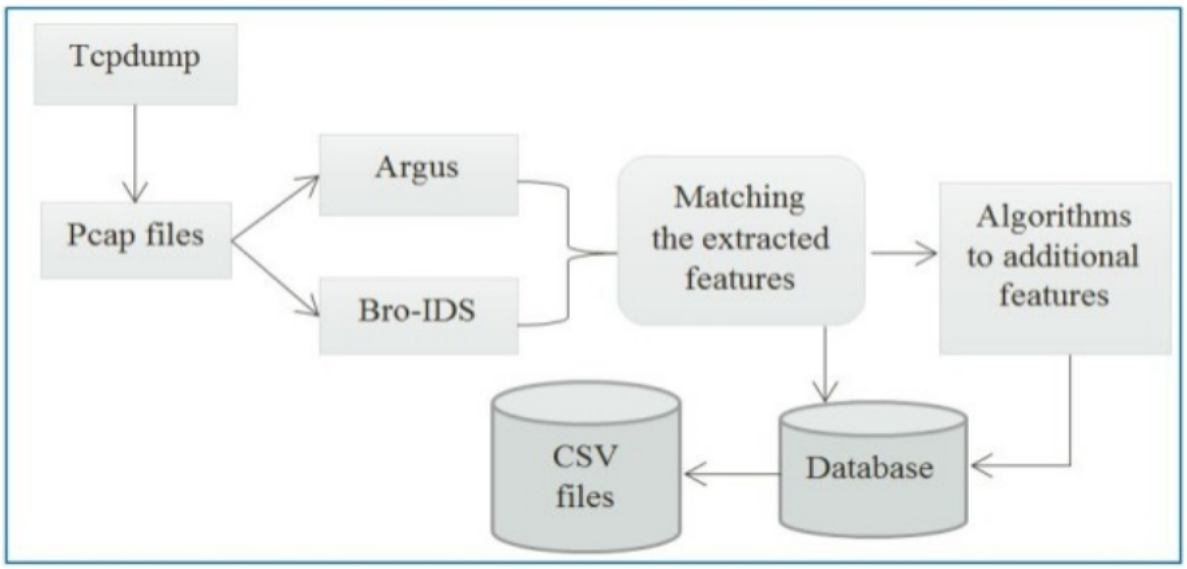
Chính vì các lý do trên, một nhóm nghiên cứu thuộc viện nghiên cứu của Trung tâm An ninh mạng Úc (Australian Centre of Cyber Security - ACCC) và các nhà nghiên cứu về lĩnh vực này trên toàn cầu đã cho ra mắt bộ dữ liệu UNSW-NB15 nhằm giúp ích trong việc đánh giá một NIDS.

3.1.1 Phương pháp thu thập dữ liệu

Công cụ IXIA PerfectStorm[25] được sử dụng để tạo ra các traffic bình thường và bất thường. Dữ liệu bất thường đi qua IXIA mô phỏng 9 loại tấn công mô tả trong Bảng 3.1. Công cụ IXIA bao gồm tất cả thông tin về các loại tấn công mới được cập nhật liên tục tại trang CVE[26], một bộ từ điển công khai các lỗ hổng bảo mật.

Bắt các lưu lượng mạng dưới dạng gói tin bằng cách sử dụng công cụ tcmdump. Việc mô phỏng kéo

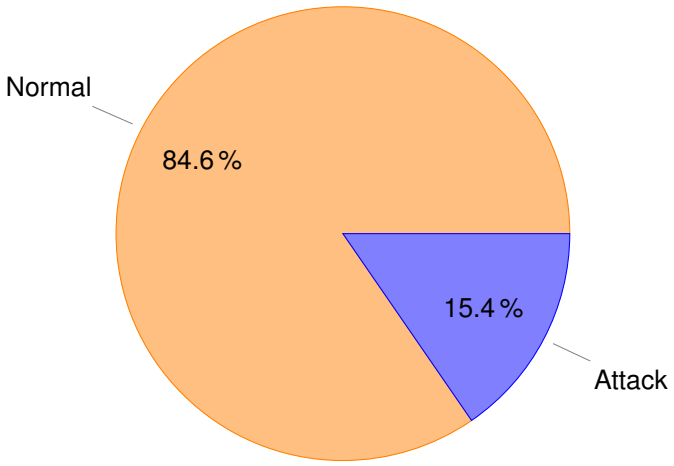
dài 16 giờ ngày 22/01/2015 và 15 giờ ngày 17/02/2015, thu được 100Gb dữ liệu. Mỗi Pcap file được chia nhỏ mỗi file 1000MB bằng tcpdump. Argus và Bro-IDS được sử dụng để tạo ra các đặc trưng tin cậy. Thêm vào đó, 12 thuật toán được phát triển sử dụng ngôn ngữ C# để phân tích sâu các gói tin. Bộ dữ liệu được gán nhãn bao gồm tất cả các loại tấn công được mô phỏng.



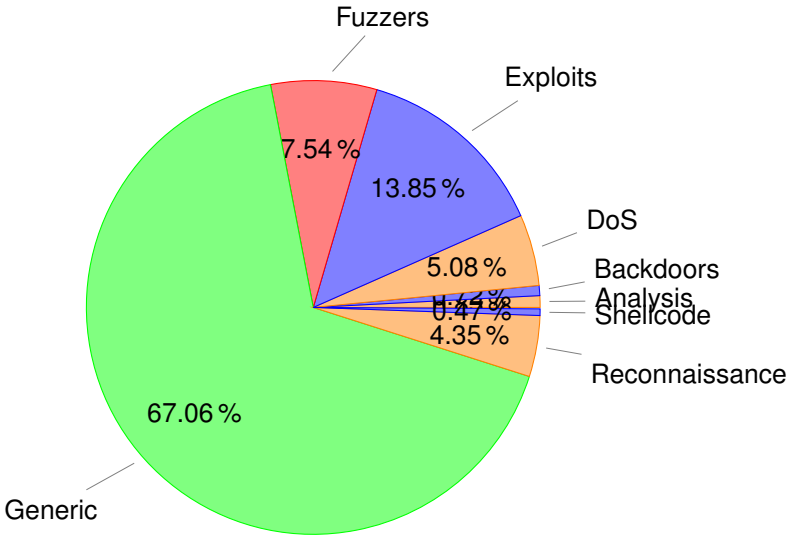
Hình 3.1: Kiến trúc mô hình sinh bộ dữ liệu UNSW-NB15

Loại	Số lượng bản ghi	Tỉ lệ	Mô tả
Normal	1,765,693	84.6%	Dữ liệu tương tác bình thường
Fuzzers	24,246	1.16%	Cố gắng khiến một chương trình hoặc hệ thống mạng bị đình chỉ bằng cách sinh dữ liệu ngẫu nhiên
Analysis	2,677	0.12%	Bao gồm nhiều loại tấn công khác nhau: quét cổng, spam và xâm nhập các file html
Backdoors	2,329	0.11%	Một kỹ thuật vượt qua hệ thống xác thực để truy nhập vào máy tính hoặc dữ liệu
DoS	16,353	0.78%	Tấn công từ chối dịch vụ
Exploits	44,525	2.13%	Hacker khai thác lỗ hổng bảo mật đã biết trước của hệ thống
Generic	215,481	10.32%	Kỹ thuật chống lại mã hóa khối.
Reconnaissance	13,987	0.67%	Bao gồm tất cả Strikes mô phỏng lại việc thu thập thông tin
Shellcode	1,511	0.07%	Một đoạn code ngắn được sử dụng như payload trong khai thác lỗ hổng phần mềm
Worms	174	0.008%	Các tập tin độc hại tự nhân bản để lan sang các máy tính khác.

Bảng 3.1: Phân bố bản ghi của bộ dữ liệu



Hình 3.2: Tỉ lệ nhãn bình thường và nhãn tấn công

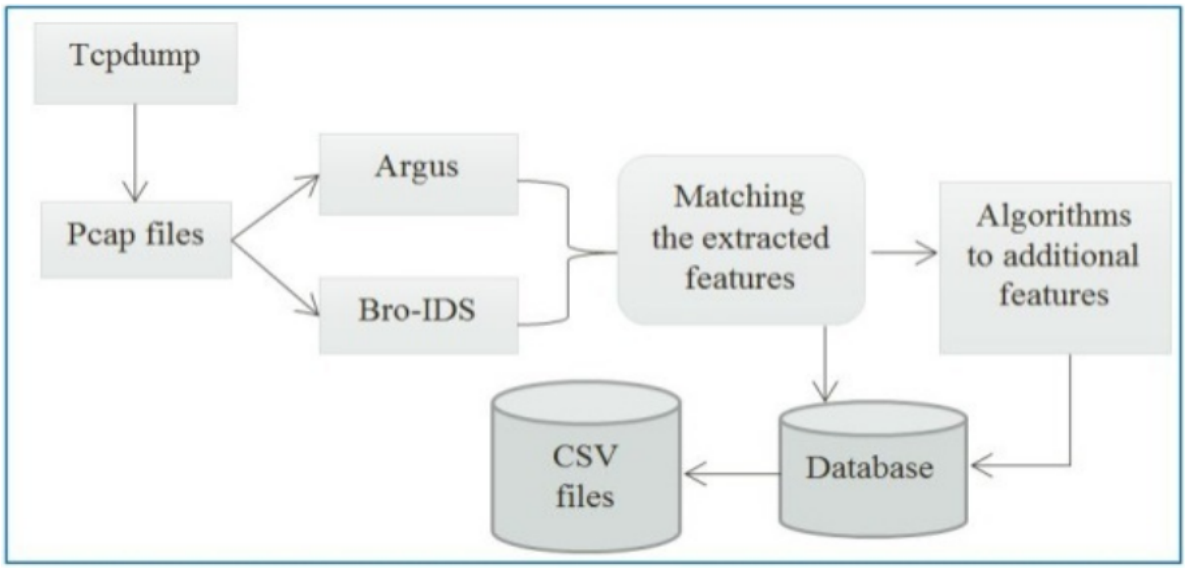


Hình 3.3: Tỉ lệ nhãn từng nhãn tấn công trong tổng các số nhãn tấn công

3.1.2 Cấu hình cho IXIA

Theo hình 3.1, bộ sinh lưu lượng IXIA được cấu hình với 3 server ảo. Server 1 và 3 được cấu hình để sinh ra các lưu lượng mạng bình thường còn server được dùng để sinh các lưu lượng mạng bất thường. Thiết lập kết nối giữa các server để thu được các lưu lượng riêng tư và công khai, có 2 virtual interface với IP 10.40.85.30 và 10.40.184.30. Các server được kết nối tới các host qua các router. Router 1 có IP 10.40.184.30 và 10.40.85.30, router 2 có IP 10.40.184.1 và 10.40.183.1. Các router kết nối tới tường lửa đã được cấu hình để cho tất cả các lưu lượng mạng đi qua. Công cụ tcpdump được cài đặt tại router 1 để bắt các Pcap file trong khi mô phỏng. Ý đồ chính của việc này nhằm để bắt các lưu lượng bình thường và bất thường, bắt nguồn từ công cụ IXIA và phân tán trong các nút mạng. Điều quan trọng là công cụ IXIA được sử dụng như một bộ sinh lưu lượng tấn công với các hành vi từ CVE nhằm mô phỏng chính xác nhất môi trường tấn công hiện đại. Dựa vào tốc độ lưu lượng mạng và phương pháp khai thác của các kiểu tấn công hiện đại, công cụ IXIA sinh ra 1 cuộc tấn công mỗi giây trong thời gian mô phỏng, thu về 50 GB đầu. Với cách mô phỏng và cấu hình thứ 2 sinh ra được 10 cuộc tấn công mỗi giây, thu được 50 GB còn lại.

3.1.3 Danh sách các đặc trưng trong bộ dữ liệu



Hình 3.4: Minh họa mô hình tạo bộ dữ liệu UNSW-NB15

Bộ dữ liệu được tạo ra theo mô hình như Hình 3.2. Các đặc trưng trong các file csv được chia thành các nhóm chính như sau:

Các đặc trưng mô tả luồng kết nối:

#	Tên	T	Mô tả
1	*scrip*	N	Địa chỉ IP nguồn
2	*sport*	I	Port nguồn
3	*dstip*	N	Địa chỉ IP đích
4	*dstport*	I	Port đích
5	*proto*	N	Giao thức

Bảng 3.2: Đặc trưng luồng

Các đặc trưng thu được từ bộ công cụ Argus và Bro-IDS. Các đặc trưng này bao gồm các đặc trưng dựa trên gói tin (packet-based) và các đặc trưng dựa trên lưu lượng (flow-based). Các đặc trưng dựa trên gói tin hỗ trợ kiểm tra payload bên cạnh tiêu đề (header) của các gói. Ngược lại, để thu được đặc trưng dựa trên luồng và duy trì phân tích tính toán với chi phí thấp thay vì quan sát tất cả các gói tin đi qua mạng, chỉ các gói kết nối tới mạng được xem xét. Các đặc trưng được phân thành ba nhóm: Cơ bản, Nội dung và Thời gian được mô tả trong các Bảng 3.3, 3.4 và 3.5.

#	Tên	T	Mô tả
6	*state*	N	Trạng thái phụ thuộc vào giao thức
7	*dur*	F	Thời gian thu thập của bản ghi
8	*sbytes*	I	Số byte từ nguồn đến đích
9	*dbytes*	I	Số byte từ đích đến nguồn
10	*sttl*	I	Thời gian sống từ nguồn đến đích
11	*dttl*	I	Thời gian sống từ đích đến nguồn
12	*sloss*	I	Các gói nguồn được truyền lại hoặc bị rớt
13	*dloss*	I	Các gói đích được truyền lại hoặc bị rớt
14	*service*	N	http, ftp, ssh, dns ..,hoặc (-)
15	*sload*	F	Số bit nguồn trong một giây
16	*dload*	F	Số bit đích trong một giây
17	*spkts*	I	Số packet từ nguồn đến đích
18	*dpkts*	I	Số packet từ đích về nguồn

Bảng 3.3: Các đặc trưng cơ bản

#	Tên	T	Mô tả
19	*swin*	I	Cửa sổ TCP nguồn
20	*dwin*	I	Cửa sổ TCP đích
21	*stcpb*	I	Số thứ tự TCP nguồn
22	*dtcpb*	I	Số thứ tự TCP đích
23	*smeansz*	I	Kích thức trung bình của luồng truyền qua nguồn
24	*dmeansz*	I	Kích thước trung bình của luồng truyền qua đích
25	*trans_depth*	I	Độ sâu kết nối của http request/response
26	*res_bdy_len*	I	Kích thức nội dung truyền từ máy chủ http

Bảng 3.4: Các đặc trưng nội dung

#	Tên	T	Mô tả
27	*sjit*	F	jitter nguồn
28	*djit*	F	jitter đích
29	*stime*	T	Thời gian bắt đầu ghi
30	*ltime*	T	Thời gian kết thúc ghi
31	*sintpkt*	F	Thời gian đến gói tin giữa các gói tin nguồn(mSec)
32	*dintpkt*	F	Thời gian đến gói tin giữa các gói tin đích (mSec)
33	*tcprtt*	F	Tổng số 'synack' và 'ackdat' của TCP
34	*synack*	F	Thời gian giữa gói tin SYN và SYN_ACK
35	*ackdat*	F	Thời gian giữa gói tin SYN_ACK và ACK

Bảng 3.5: Đặc trưng thời gian

Bảng 3.6 mô tả về các đặc trưng bổ sung của bộ dữ liệu UNSW-NB15. Các đặc trưng từ 36-40 là các đặc trưng mục đích chung, các đặc trưng 41-47 là các đặc trưng của kết nối. Với đặc trưng mục đích chung, mỗi tính năng có mục đích riêng, theo quan điểm phòng thủ, trong khi đó các đặc trưng kết nối chỉ được tạo ra để cung cấp biện pháp phòng vệ trong trong các kịch bản kết nối. Những kẻ tấn công có thể quét máy chủ theo cách tự lập. Ví dụ, mỗi lần một phút hoặc một lần quét trên một giờ. Để xác định những kẻ tấn công này, các đặc trưng 36-47 của Bảng 3.6 được sắp xếp theo thứ

tự tương ứng với đặc trưng cuối cùng để thu thập các đặc tính tương tự của các bản ghi kết nối với 100 kết nối tuần tự.

#	Tên	T	Mô tả
36	*is_sm_ips_ports*	B	Nếu nguồn và đích cùng ip và port thì là 1, ngược lại là 0
37	*ct_state_ttl*	I	Số lượng mỗi trạng thái (6) theo phạm vi giá trị cụ thể của thời gian sống nguồn / đích
38	*ct_flw_http_mthd*	I	Số lượng luồng dùng phương thức POST và GET trong http
39	*is_ftp_login*	B	1 nếu dịch vụ ftp được truy cập bởi user, ngược lại là 0
40	*ct_ftp_cmd*	B	Số lượng luồng có 1 lệnh trong phiên ftp
41	*ct_srv_src*	I	Số lượng kết nối dùng cùng một dịch vụ và địa chỉ nguồn(1) trong 100 kết nối theo (26)
42	*ct_srv_dst*	I	Số lượng kết nối dùng cùng một dịch vụ và địa chỉ đích(1) trong 100 kết nối theo (26)
43	*ct_dst_ltm*	I	Số lượng kết nối có cùng đích trong 100 kết nối theo (26)
44	*ct_src_ ltm*	I	Số lượng kết nối có cùng địa chỉ nguồn trong 100 kết nối theo (26)
45	*ct_src_dport_ltm*	I	Số lượng kết nối có cùng địa chỉ nguồn (1) và cổng đích (4) trong 100 kết nối theo (26)
46	*ct_dst_sport_ltm*	I	Số lượng kết nối có cùng địa chỉ đích (3) và cổng nguồn (2) trong 100 kết nối theo (26)
47	*ct_dst_src_ltm*	I	Số lượng kết nối có cùng địa chỉ nguồn và đích trong 100 kết nối theo (26)

Bảng 3.6: Đặc trưng bổ sung

2 đặc trưng cuối cùng trong bộ dữ liệu là các nhãn.

#	Tên	T	Mô tả
48	*attack_cat*	N	Tên của các nhãn tấn công, bao gồm 9 nhãn
49	*Label*	B	1 là tấn công, 0 là bình thường

Bảng 3.7: Các nhãn

3.2 Phương pháp đánh giá

Để đánh giá hiệu quả của mô hình phân loại, đồ án sử dụng phương pháp Ma trận lỗi (Error Matrix hay Confusion matrix [27]). Mỗi hàng của ma trận đại diện cho một nhãn được dự đoán, trong khi mỗi cột của ma trận đại diện cho một nhãn chính xác.
Từ Ma trận lỗi, chúng ta có thể tính được các giá trị cho mỗi nhãn k trong bộ dữ liệu, bao gồm Độ chính xác(Precision), độ phủ (Recall) và F1 (Trung bình điều hòa):

Độ chính xác được định nghĩa là tổng số quan sát có nhãn c_i được dự đoán chính xác chia cho tổng số các quan sát được dự đoán có nhãn c_i.

$$P(c_i) = \frac{\text{Số quan sát được gán chính xác nhãn } c_i}{\text{Tổng số quan sát được gán nhãn } c_i} \tag{3.1}$$

Độ phủ được tính bằng tổng số các quan sát có nhãn c_i được dự đoán chính xác chia cho tổng số các quan sát có nhãn c_i trong tập kiểm thử.

$$R(c_i) = \frac{\text{Số quan sát được gán chính xác nhãn } c_i}{\text{Tổng số các quan sát có nhãn } c_i \text{ trong tập huấn luyện}} \tag{3.2}$$

F1 là một trung bình điều hòa của các tiêu chí P và R. F1 có các tính chất sau:

- F1 có xu hướng lấy giá trị gần với giá trị nào nhỏ hơn giữa 2 giá trị P và R.
- F1 có giá trị lớn nếu cả 2 giá trị P và R đều lớn.

$$P(c_i) = \frac{2 * P * R}{P + R} \tag{3.3}$$

3.3 Thực nghiệm

3.3.1 Hệ thống máy tính

Cấu hình phần cứng phục vụ cho quá trình huấn luyện:

- Vi xử lý: Intel Corporation Xeon E7 v3/Xeon E5 v3/Core i7 Power Control Unit
- RAM: 157.0 GB
- Hệ điều hành: Ubuntu Server 16.04
- Dung lượng ổ cứng 2TB

3.3.2 Các chương trình và thư viện phần mềm

Pandas

Pandas là thư viện mã nguồn mở, được cấp phép bởi BSD, cung cấp các cấu trúc dữ liệu hiệu năng cao, dễ sử dụng và các công cụ phân tích dữ liệu cho ngôn ngữ lập trình Python. Pandas bao gồm:

- Tập hợp các cấu trúc dữ liệu dạng mảng, phần chính là Series và DataFrame
- Các đối tượng chỉ mục cho phép lập chỉ mục trục đơn giản và lập chỉ mục trục đa cấp / phân cấp
- Bộ công cụ tích hợp để chuyển đổi và tập hợp dữ liệu
- Bộ sinh dữ liệu theo ngày
- Bộ công cụ I/O: Đọc dữ liệu dạng bảng ở các file (CSV, delimited, Excel 2003) và lưu các đối tượng pandas theo PyTables/HDF5 format.
- Các phiên lưu trữ dữ liệu thưa trên bộ nhớ hiệu quả, các cấu trúc dữ liệu tiêu chuẩn để lưu trữ dữ liệu thiếu hoặc hằng số (một số có giá trị cố định)
- Các cửa sổ thống kê

Scikit-learn

Scikit-learn là thư viện mã nguồn mở dùng cho ngôn ngữ lập trình Python. Scikit-learn bao gồm rất nhiều các thuật toán học máy phân cụm, hồi quy, phân loại,.. và được thiết kế để tương thích với các thư viện số học như Numpy [15] và SciPy [16]

Thư viện XGBoost

XGBoost[3] là thư viện mã nguồn mở, cài đặt thuật toán Boosted Tree đã mô tả ở chương 2, dùng để áp dụng vào các bài toán học máy. Đây là thư viện mạnh mẽ, đã thống trị học máy ứng dụng và các cuộc thi về Khoa học dữ liệu trên Kaggle[13]trong thời gian gần đây. XGBoost hỗ trợ rất nhiều các giao diện (interface) như:

- Giao diện dòng lệnh (CLI - Command line interface).
- C++
- Python interface cùng với mô hình theo chuẩn Scikit-learn
- R interface
- Julia
- Java và các ngôn ngữ sử dụng JVM như Scala, các platform như Hadoop

Đồ án sử dụng Python interface để huấn luyện mô hình.

3.4 Tiền xử lý dữ liệu

3.4.1 Loại bỏ các đặc trưng dư thừa

Như đã mô tả ở phần 3.1.2, UNSW-NB15 là một bộ dữ liệu được tạo ra từ phòng thí nghiệm với số lượng IP cố định. Dựa vào các cuộc xâm nhập mạng trong thực tế, các IP đến từ rất nhiều nguồn và port khác nhau. Việc sử dụng các đặc trưng này sẽ khiến mô hình học bị overfit, khi đưa áp dụng vào thực tế sẽ đưa ra dự đoán thiếu chính xác. Vì vậy, các đặc trưng *Srcip*, *Sport*, *Dstip*, *Dport* được loại bỏ khỏi tập dữ liệu.
Các đặc trưng *Stime* - thời gian bắt đầu ghi, *Ltime* - thời gian kết thúc ghi được thay thế bằng đặc trưng *duration* - độ dài thời gian ghi dữ liệu.
Các đặc trưng nhãn mô tả ở Bảng 3.7 được tách riêng ra. Bảng 3.8 và 3.9 là ví dụ về việc bỏ các thuộc tính thừa:

’59.166.0.4’ 17491 ’149.171.126.0’ ’53’ ’udp’ ’CON’ 0.001256 132.0 164.0 31.0 29.0 0.0 0.0 ’dns’ 420382.15629999997 522293.0 2.0 2.0 0.0 0.0 0.0 0.0 66.0 82.0 0.0 0.0 0.0 0.0 1421927424.0 1421927424.0 0.012 0.0080000000000000002 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 12.0 13.0 1.0 1.0 1.0 1.0 1.0 nan 0.0

Bảng 3.8: Bản ghi trước khi bỏ các đặc trưng thừa

’udp’ ’CON’ 0.001256 132.0 164.0 31.0 29.0 0.0 0.0 ’dns’ 420382.15629999997 522293.0 2.0 2.0 0.0 0.0 0.0 0.0 66.0 82.0 0.0 0.0 0.0 0.0 1421927424.0 1421927424.0 0.012 0.0080000000000000002 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 12.0 13.0 1.0 1.0 1.0 1.0 1.0

Bảng 3.9: Bản ghi sau khi bỏ các đặc trưng thừa

3.4.2 Xử lý các dữ liệu dạng ký hiệu

Với tập dữ liệu UNSW-NB15, các bản ghi được lưu dưới dạng các file csv. Như ở Bảng 3.3, 3.4, 3.5, 3.6, 3.7, dữ liệu gồm 4 dạng integer(I), float(F), binary(B) và nominal(N). Boosted Tree là một thuật toán chỉ hoạt động trên dữ liệu dạng số học (numeric) vì vậy dữ liệu trước khi được đưa vào huấn luyện cần phải chuyển đổi từ dạng ký hiệu (nominal) sang dạng số học. Các đặc trưng cần được chuyển đổi bao gồm:

STT	Tên	Mô tả
5	proto	text
6	state	text
14	service	text

Bảng 3.10: Các đặc trưng dạng ký hiệu

Số lượng các giá trị của một đặc trưng dạng nominal là hữu hạn. Có nhiều phương pháp tiếp cận để chuyển đổi các đặc trưng dạng nominal sang dạng numeric như:

- Label Encoding: Với cách tiếp cận này, mỗi giá trị đặc trưng sẽ được đánh một số. Ví dụ với đặc trưng service, mỗi giá trị sẽ được đánh một số như sau:

-	0
dns	1
http	2
ftp-data	3
smtp	4
ssh	5
ftp	6
pop3	7
dhcp	8
ssl	9
snmp	10
radius	11
irc	12

- One-hot Encoding: Label Encoding có lợi thế về sự đơn giản, trực quan nhưng điều này lại có nhược điểm: các giá trị số có thể bị giải thích sai bởi thuật toán. Với cách đánh nhãn như trên, thuật toán sẽ hiểu *snmp* có trọng số lớn hơn so với *ssl* hoặc *http* sẽ có trọng số lớn hơn so với *dns*. Đây là một thứ tự không mong muốn, vì đồ án muốn các giá trị này độc lập với nhau nên việc sử dụng one-hot encoding là điều cần thiết. Với một đặc trưng có k giá trị khác nhau, one-hot encoding sẽ biến một đặc trưng ban đầu thành k đặc trưng đại diện cho các giá trị khác nhau. Đặc trưng đại diện cho giá trị dạng nominal ban đầu sẽ được đặt giá trị numeric là 1, các đặc trưng còn lại được đặt giá trị 0. Đặc trưng service sau khi sử dụng one-hot encoding:

nan	dhcp	dns	ftp	ftp-data	http	irc	pop3	radius	smtp	snmp	ssh	ssl
1	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0

Bảng 3.11: Đặc trưng service biểu diễn dưới dạng One-hot Encoding

0.001256 132.0 164.0 31.0 29.0 0.0 0.0 420382.15629999997 522293.0 2.0 2.0 0.0 0.0 0.0 0.0 66.0 82.0 0.0 0.0 0.0 0.0 1421927424.0 1421927424.0 0.012 0.0080000000000000002 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 12.0 13.0 1.0 1.0 1.0 1.0 1.0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0

Bảng 3.12: Caption

3.5 Các mô hình thực nghiệm

Đồ án tiến hành thực nghiệm với phương pháp đã giới thiệu tại Chương II trên bộ dữ liệu UNSW-NB15 với 2 mô hình phân loại:

- Bài toán phân lớp 2 nhãn
- Bài toán phân lớp 10 nhãn

3.5.1 Phân lớp 2 nhãn

Bộ dữ liệu được chia thành 2 phân tập huấn luyện (training set) và tập kiểm thử (testing set). Tập huấn luyện chiếm 80% bộ dữ liệu ban đầu. Tập huấn luyện gồm 1669580 bản ghi, trong đó 1412554 nhãn bình thường (mã hóa là 0), 257026 nhãn tấn công(mã hóa là 1 Tập kiểm thử gồm 417396 bản ghi với 353437 bản ghi có nhãn bình thường, 63959 nhãn tấn công.

Thiết lập cấu hình cho mô hình

Để tìm tham số cho mô hình, đồ án đã thực hiện chạy Grid Search với hàm scoring = 'f1_weighted' danh sách các tham số:

- **max_depth**: 3, 6, 9, 12
- **n_estimators**: 50,100,150,200
- **learning_rate**: 0.01, 0.05, 0.1

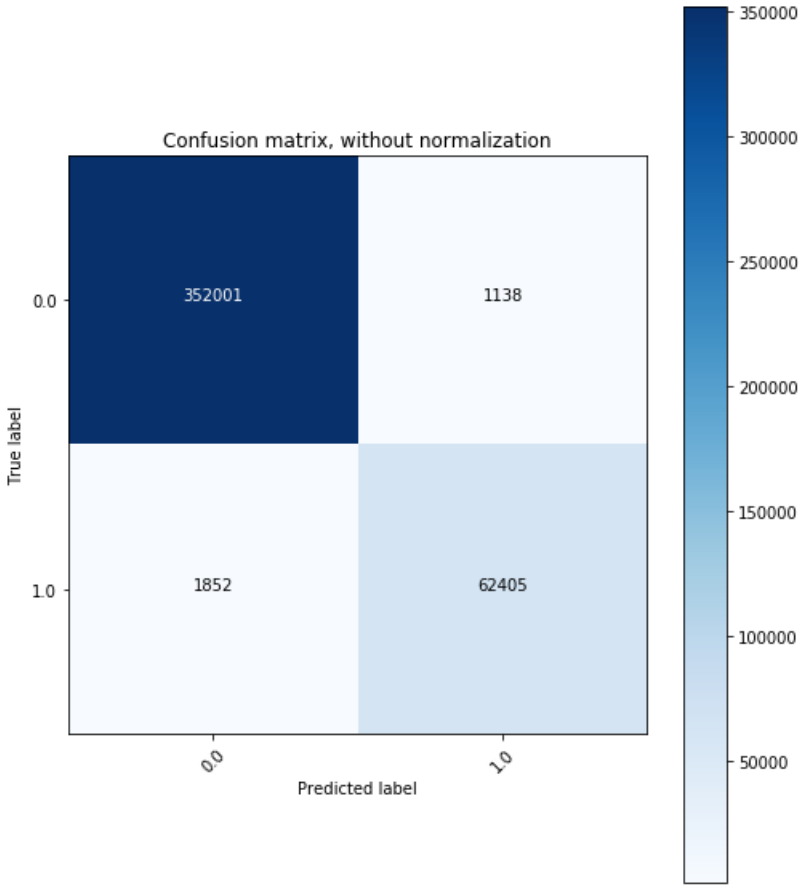
```
#grid search
alg = XGBClassifier()
clf = GridSearchCV(alg,{'max_depth': [3,6,9,12],
                        'n_estimators': [50,100,150,200],
                            'learning_rate': [0.01, 0.05, 0.1],
                           },
                        verbose=2,
                        scoring='f1_weighted', n_jobs = -1)

clf.fit(X_sample,y_sample)
clf.best_score_, clf.best_params_
```

Hình 3.5: Cấu hình GirdSearch cho bài toán phân lớp 2 nhãn

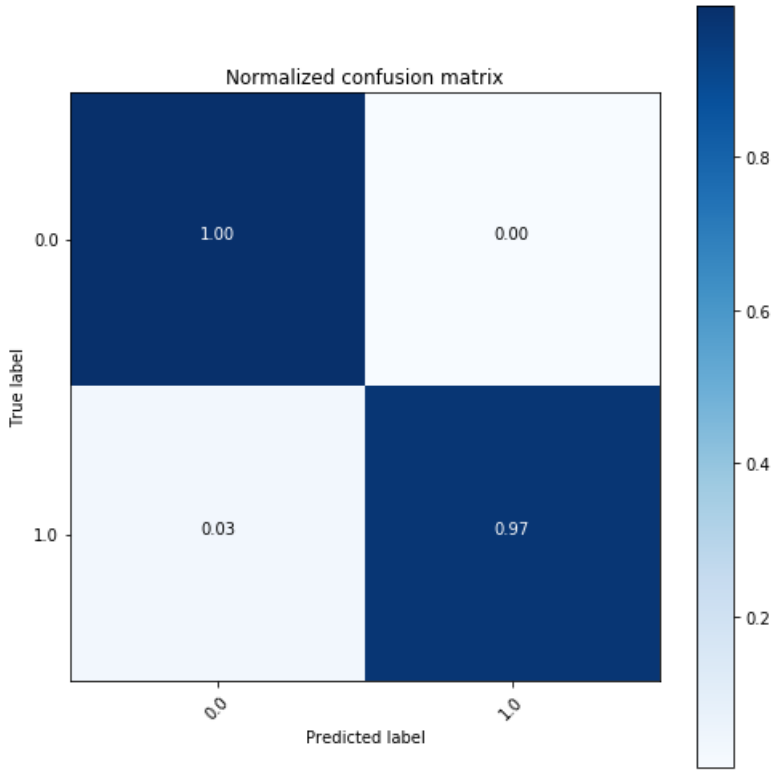
Tham số tốt nhất tìm được là:
$learning\_rate = 0.1$
$max\_depth = 9$
$n\_estamators = 200$

Mô hình sau khi huấn luyện được đánh giá với tập kiểm thử, chi tiết tại ma trận lỗi (confusion matrix) dưới đây:

*(**Ghi chú**: Trong cả 2 bài toán 2 nhãn và 10 nhãn, đồ án sử dụng 2 loại dạng ma trận lỗi. Với ma trận lỗi chưa chuẩn hóa, các phần tử nằm trên đường chéo chính đại diện cho số lượng các quan sát được dự đoán chính xác so với thực tế, các phần tử nằm ngoài đường chéo chính là những nhãn được dự đoán sai bởi bộ phân loại. Với ma trận lỗi chuẩn hóa, các phần tử được biểu diễn bởi tỉ lệ dự đoán)*



Hình 3.6: Ma trận lỗi chưa chuẩn hóa phân lớp 2 nhãn với Boosted Tree



Hình 3.7: Ma trận lỗi chuẩn hóa phân lớp 2 nhãn với Boosted Tree

So sánh kết quả

Mô hình sử dụng Boosted Tree được so sánh với các mô hình khác. Với cùng thuật toán họ ensemble boosting, đồ án sử dụng Adaptive Boosting [28] để làm baseline. Bên cạnh Adaptive Boosting, mô hình được so sánh với một thuật toán khác là Naive Bayes[30]. Bảng so sánh đánh giá độ chính xác, độ phủ và trung bình điều hòa F1 của 3 phương pháp::

Nhãn	Adaptive Boosting			Naive Bayes			Boosted Tree		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
0.0	0.99	0.99	0.99	0.87	0.96	0.91	1.00	0.99	1.00
1.0	0.97	0.97	0.97	0.81	0.53	0.64	0.97	0.98	0.98

Bảng 3.13: So sánh kết quả phân lớp 2 nhãn của 3 phương pháp

3.5.2 Phân lớp 10 nhãn

Tương tự với bộ phân loại 2 lớp phát hiện bất thường, bộ dữ liệu cũng được chia thành 2 phần tập huấn luyện và tập kiểm thử. Tập huấn luyện gồm 1669580 bản ghi, tập kiểm thử gồm 417396 bản ghi. Thống kê mỗi nhãn xuất hiện trong mỗi tập ở bảng:

Nhãn	Tập huấn luyện	Tập kiểm thử
Analysis	2142	535
Backdoors	1863	466
DoS	13082	3271
Exploits	35620	8905
Fuzzers	19397	4849
Generic	172385	43096
Reconnaissance	11189	2798
Shellcode	1209	302
Worms	139	35
normal	1412554	353139

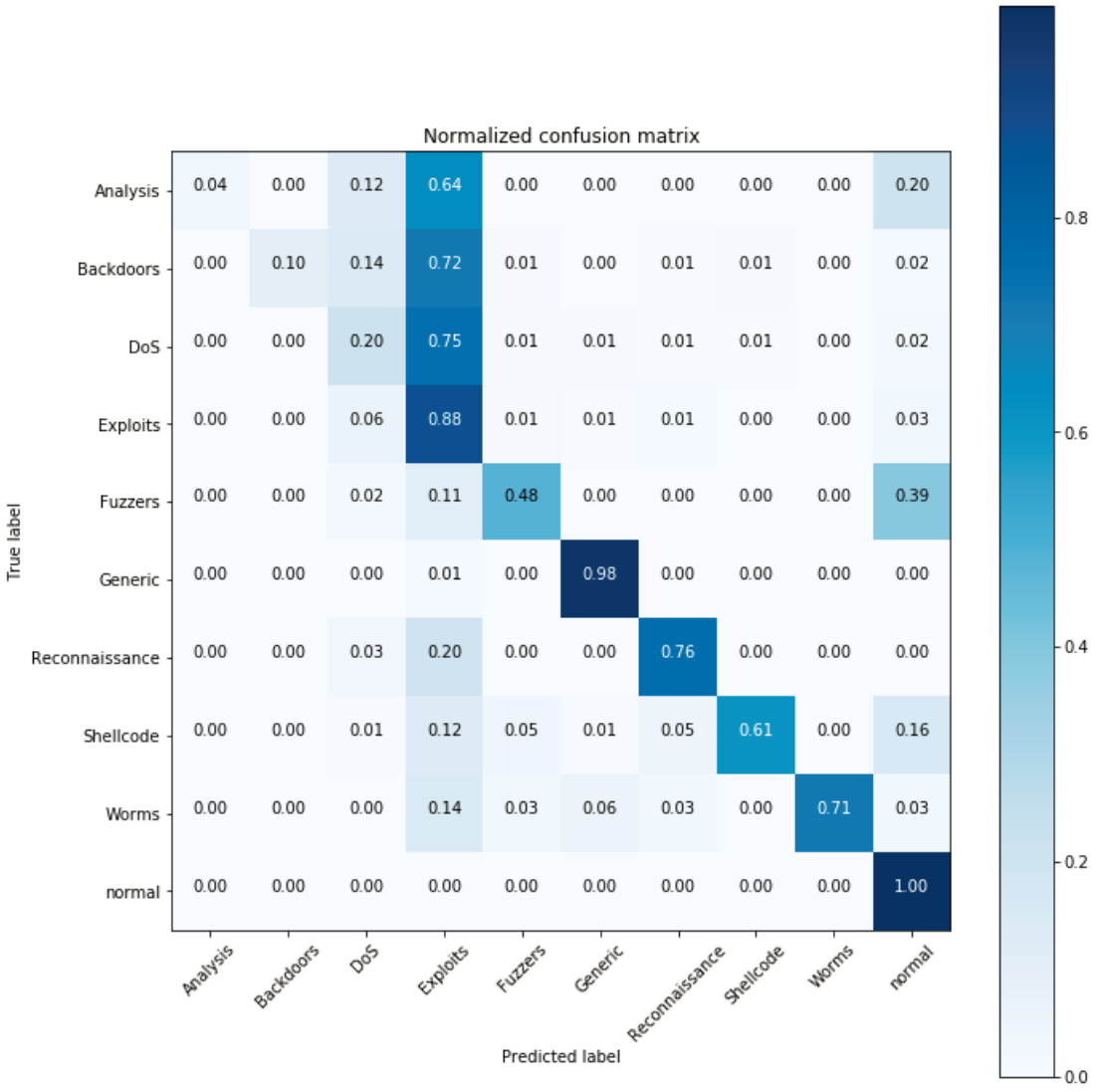
Bảng 3.14: Phân bố nhãn trong mỗi tập

Qua việc lựa chọn tham số mô hình qua grid-search với dãy tham số:

- **max_depth**: 3, 6, 9, 12
- **n_estimators**: 50,100,150,200
- **learning_rate**: 0.01, 0.05, 0.1

Tham số tốt nhất tìm được là:
$learning\_rate = 0.1$
$max\_depth = 6$
$n\_estamators = 100$
Mô hình sau khi huấn luyện được đánh giá với testing set, chi tiết ở ma trận lỗi (confusion matrix):

Hình 3.8: Ma trận lỗi chưa chuẩn hóa phân lớp 10 nhãn với Boosted Tree



Hình 3.9: Ma trận lỗi chuẩn hóa phân lớp 10 nhãn với Boosted Tree

So sánh kết quả

Tương tự với bộ phân lớp 2 nhãn, đồ án cũng sử dụng 1 bộ phân lớp cùng họ là Adaptive Boosting và 1 bộ phân lớp khác là Naive Bayes để so sánh. Bảng so sánh đánh giá độ chính xác, độ phủ và trung bình điều hòa F1 của 3 phương pháp:

Nhãn	Adaptive Boosting			Naive Bayes			Boosted Tree		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Analysis	0.20	0.00	0.01	0.00	0.00	0.00	0.04	0.77	0.07
Backdoors	0.03	0.03	0.03	0.59	0.04	0.07	0.10	0.90	0.18
DoS	0.04	0.15	0.06	0.12	0.16	0.14	0.20	0.42	0.27
Exploits	0.58	0.07	0.13	0.08	0.85	0.14	0.88	0.62	0.72
Fuzzers	0.42	0.43	0.42	0.00	0.00	0.00	0.48	0.72	0.58
Generic	0.98	0.96	0.97	0.98	0.47	0.63	0.98	1.00	0.99
Reconnaissance	0.57	0.63	0.60	0.00	0.00	0.00	0.76	0.92	0.84
Shellcode	0.16	0.28	0.20	0.00	0.00	0.00	0.61	0.62	0.62
Worms	0.74	0.81	0.01	0.01	0.01	0.01	0.71	0.81	0.76
normal	1.00	0.99	1.00	0.86	0.98	0.92	1.00	0.99	1.00

Bảng 3.15: So sánh kết quả phân lớp 10 nhãn của 3 phương pháp

3.6 Nhận xét, đánh giá

Từ bảng 3.17, 3.18, 3.19 có thể thấy, mô hình dễ dàng phát hiện được các quan sát dạng bình thường. Với các traffic có nhãn là tấn công, các nhãn Generic, Reconnaissance và Exploits có độ chính xác cao. Các quan sát có nhãn Backdoors, Generic, Reconnaissance, Worms có độ phủ cao. Điều này cho thấy có sự phân loại nhầm giữa các quan sát có nhãn tấn công. Nguyên nhân có thể do các nhãn tấn công có số lượng tập học ít, phân bố lệch nhau.

Có thể thấy Adaptive Boosting tuy độ chính xác, độ phủ và F1 tại mỗi nhãn kém hơn 1 chút so với Boosted Tree nhưng nhìn chung các bộ phân lớp Boosting có kết quả cao hơn so với các thuật toán khác trên bộ dữ liệu UNSW-NB15.

Các thuật toán Boosting có kết quả tốt hơn hẳn so với Naive Bayes.

Chương 4

TỔNG KẾT

Phát hiện xâm nhập mạng sử dụng học máy tuy không mới nhưng đây là một lĩnh vực nghiên cứu rất tiềm năng. Trong phạm vi của mình, đồ án đã đạt được những kết quả sau:

- Trình bày tổng quan về vấn đề xâm nhập mạng, các phương pháp về phát hiện xâm nhập mạng hiện nay.
- Nghiên cứu về giải thuật Học kết hợp (Ensemble Learning) cụ thể là thuật toán Boosted Tree. Đây là một thuật toán mạnh mẽ, có khả năng áp dụng cao vào thực tế.
- Áp dụng mô hình Boosted Tree vào phát hiện xâm nhập mạng. Từ đó rút ra được những điểm mạnh và điểm yếu của mô hình.

Bên cạnh những điều đã nghiên cứu được, đồ án vẫn còn một số hạn chế và định hướng phát triển tiếp theo:

- Cải thiện khả năng dự đoán các nhãn tấn công, tránh nhầm lẫn giữa các nhãn với nhau
- Thực hiện tiền xử lý dữ liệu. Hiện tại đồ án chưa có bước tiền xử lý dữ liệu từ gói tin thô sang bộ các đặc trưng.
- Từ mô hình học được, nhúng vào các hệ thống phát hiện xâm nhập mạng.

TÀI LIỆU THAM KHẢO

[1] Tood McGuiness. Defense In Depth. *http://www.sans.org/rr/securitybasics/defense.php*, November 2001.

[2] Patcha, Animesh, and Jung-Min Park. "An overview of anomaly detection techniques: Existing solutions and latest technological trends." *Computer networks* 51, no. 12 (2007): 3448-3470.

[3] Xgboost Documents. *https://github.com/dmlc/xgboost*

[4] Kline, M. (1990). Mathematical Thought from Ancient to Modern Times. New York: Oxford University Press. pp. 35–37. ISBN 0-19-506135-7.

[5] T. Dietterich, Ensemble methods in machine learning, in the first International Workshop on Multiple Classifier Systems, Springer, pp. 1-15, 2000.

[6] Stavroulakis P, Stamp M. Handbook of information and communication security.-New York: Springer-Verlag; 2010.

[7] Couture M. Real time intrusion prediction based on optimized alerts with hidden Markov model. Journal of Networks 2012;7:311–21.

[8] Fragkiadakis AG, Tragos EZ, Tryfonas T, Askoxylakis IG. Design and performance evaluation of a lightweight wireless early warning intrusion detection proto- type. EURASIP Journal on Wireless Communications and Networking 2012;73: 1–18.

[9] Kantzavelou I, Katsikas S. A game-based intrusion detection mechanism to confront internal attackers. Computers amp; Security 2010;29:859–74.

[10] Sabahi F, Movaghar A, Intrusion detection: a survey, In: Third international conference on system and network communication, Sliema, Malta, 2008, pp. 23–26.

[11] L Li, Zhang G, Nie J, Niu Y, Yao A, The application of genetic algorithm to intrusion detection in MP2P network. In: Third international conference on advances in swarm intelligence, Shenzhen, China, 2012, pp. 390–397.

[12] Li Y, Xia J, Zhang S, Yan J, Ai X, Dai K. An efficient intrusion detection system based on support vector machines and gradually feature removal method. Expert Systems with Applications 2012;39:424–30.

[13] Kaggle: Your Home for Data Science. *https://www.kaggle.com/*

[14] L. Breiman, Random Forests, Machine Learning. 45(1) (2001), 5-32.

[15] Numpy. *http://www.numpy.org/*

[16] SciPy.org. *https://www.scipy.org/*

[17] scikit-learn: machine learning in Python. *http://scikitlearn.org/*

[18] Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA. ACM.

[19] KDDCup1999.Available-on: http://kdd.ics.uci.edu/databases/kddcup99/KDDCUP99.html, 2007

[20] NSLKDD. Available on: http://nsl.cs.unb.ca/NSLKDD/, 2009

[21] P.Gogoi et al, "Packet and flow based network intrusion dataset."Contemporary Computing". Springer Berlin Heidelberg, 2012. P 322-334.

[22] McHugh, John, "Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory". ACM transactions on Information and system Security, 3, 2000, p 262-294.

[23] V.Mahoney, and K.Philip, "An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection."Recent Advances in Intrusion Detection". Springer Berlin Heidelberg, 2003.

[24] A.Vasudevan, E. Harshini, and S. Selvakumar, "SSENet-2011: a network intrusion detection system dataset and its comparison with KDD CUP 99 dataset", Internet (AH-ICI), 2011, Second Asian Himalayas International Conference on. IEEE.

[25] PerfectStorm | Ixia *http://www.ixiacom.com/products/perfectstorm*

[26] CVE - Common Vulnerabilities and Exposures (CVE) *https://cve.mitre.org/*

[27] Confusion matrix; scikit-learn 0.19.1 documentation *http://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html*

[28] Freund, Yoav; Schapire, Robert E (1997). "A decision-theoretic generalization of on-line learning and an application to boosting". Journal of Computer and System Sciences. 55: 119. CiteSeerX 10.1.1.32.8918Freely accessible. doi:10.1006/jcss.1997.1504: original paper of Yoav Freund and Robert E.Schapire where AdaBoost is first introduced.

[29] Bias–variance decomposition, In Encyclopedia of Machine Learning. Eds. Claude Sammut, Geoffrey I. Webb. Springer 2011. pp. 100-101

[30] Domingos, Pedro; Pazzani, Michael (1997). "On the optimality of the simple Bayesian classifier under zero-one loss". Machine Learning. 29: 103–137.