ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA

NGUYỄN PHẠM ANH KHOA

Đề tài:

HIỆN THỰC HỆ THỐNG PHÁT HIỆN TẤN CÔNG TRONG MẠNG INTERNET DỰA VÀO CÁC BẤT THƯỜNG LÊN FPGA

Chuyên ngành: Khoa học máy tính

LUẬN VĂN THẠC SĨ

TP Hồ Chí Minh, tháng 12 năm 2011

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI
TRƯỜNG ĐẠI HỌC BÁCH KHOA
ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

Cán bộ hướng dẫn khoa học: TS. Trần Ngọc Thịnh .....

TS. Đinh Đức Anh Vũ .....

Cán bộ chấm nhận xét 1: TS. Huỳnh Hữu Thuận .....

Cán bộ chấm nhận xét 2: TS. Vũ Đức Lung .....

Luận văn thạc sĩ được bảo vệ tại trường Đại học Bách Khoa, ĐHQG TP. HCM ngày 06 tháng 01 năm 2012.

Thành phần Hội đồng đánh giá luận văn thạc sĩ gồm:

1. Chủ tịch hội đồng: TS. Trần Văn Hoài
2. Thư ký hội đồng: TS. Trần Mạnh Hà
3. Ủy viên phản biện 1: TS. Huỳnh Hữu Thuận
4. Ủy viên phản biện 2: TS. Vũ Đức Lung
5. Ủy viên hội đồng: TS. Trần Ngọc Thịnh

Chủ tịch hội đồng đánh giá LV **Bộ môn quản lý chuyên ngành**

TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KH & KT MÁY TÍNH

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc Lập - Tự Do - Hạnh Phúc

TP. HCM, ngày 11 tháng 01 năm 2012

NHIỆM VỤ LUẬN VĂN THẠC SĨ

Họ tên học viên: Nguyễn Phạm Anh Khoa Phái: Nam

Ngày, tháng, năm sinh: 25/11/1983 Nơi sinh: Đồng Nai

Chuyên ngành: Khoa học Máy tính MSHV: 09070447

I- TÊN ĐỀ TÀI:HIỆN THỰC HỆ THỐNG PHÁT HIỆN TẤN CÔNG TRONG MẠNG INTERNET DỰA VÀO CÁC BẤT THƯỜNG LÊN FPGA

II- NHIỆM VỤ VÀ NỘI DUNG:

Đề xuất mô hình và hiện thực phần lõi của việc phát hiện tấn công trong mạng internet dựa trên các bất thường lên FPGA. Mục tiêu là có khả năng phát hiện đúng các loại tấn công với tỉ lệ >90% và tỉ lệ phát ra những cảnh báo sai <1%.

III- NGÀY GIAO NHIỆM VỤ: 14/02/2011

IV- NGÀY HOÀN THÀNH NHIỆM VỤ: 02/12/2011

V- CÁN BỘ HƯỚNG DẪN: TS. Trần Ngọc Thịnh

TS. Đinh Đức Anh Vũ

CÁN BỘ HƯỚNG DẪN	**CHỦ NHIỆM BỘ MÔN**	**KHOA QL CHUYÊN NGÀNH**
(Họ tên và chữ ký)	QUẢN LÝ CHUYÊN NGÀNH	(Họ tên và chữ ký)
	(Họ tên và chữ ký)	

LỜI CAM ĐOAN

Tôi cam đoan rằng, ngoại trừ các kết quả tham khảo từ các công trình khác như đã ghi rõ trong luận văn, các công việc trình bày trong luận văn này là do chính tôi thực hiện và chưa có phần nội dung nào của luận văn này được nộp để lấy một bằng cấp ở trường này hoặc trường khác.

Thành phố Hồ Chí Minh, tháng 12 năm 2011

Nguyễn Phạm Anh Khoa

Lời cám ơn

Đầu tiên, tôi xin gửi lời cảm ơn chân thành đến TS. Trần Ngọc Thịnh và TS. Đinh Đức Anh Vũđã cung cấp tài liệu cũng như tận tình hướng dẫn, hỗ trợ tôi trong suốt quá trình nghiên cứuđể hoàn thành luận văn tốt nghiệp. Các thầy là động lực lớn nhất giúp tôi có thể hoàn thành đề tài này.

Bên cạnh đó, tôi cũng xin cảm ơn sự giúp đỡ nhiệt tình của các bạn đồng nghiệp ở phòng thí nghiệm máy tính tại tòa nhà C5 đã luôn tạo điều kiện tốt nhất để tôi có thể kiểm thử đề tài trên các board mạch thực tế.

Cuối cùng, xin cảm ơn đến gia đình và bạn bè, những người đã luôn ủng hộ và giúp đỡ tôi trong cuộc sống để tôi có thể hoàn thành tốt luận văn.

Thành phố Hồ Chí Minh, tháng 12 năm 2011

Tóm tắt luận văn thạc sĩ

Luận văn sẽ trình bày các kiến thức về kiến trúc FPGA và hệ thống phát hiện tấn công trong mạng internet nhưng đi sâu vào các phương pháp dựa vào những dấu hiệu bất thường của mạng.

Bên cạnh đó thì luận văn cũng đề xuất 1 cách thức để có thể hiện thực hiệu quả việc dò tìm tấn công dựa trên các bất thường của mạng lên board NetFPGA của Xilinx là hình thành 1 nhóm các cây quyết định. Cụ thể là xây dựng 5 cây quyết định với ngõ nhập là 41 thuộc tính của từng kết nối mạng và dùng phương thức bầu cử để đưa ra quyết định cuối cùng. Tập dữ liệu được sử dụng để test và so sánh kết quả là KDD 99.

Mục lục nội dung

Mục lục hình

---o0o---

Mục lục bảng
---o0o---

Phần 1. GIỚI THIỆU ĐỀ TÀI

1.1 Tổng quan

- Cùng với sự lớn mạnh của mạng internet và các dịch vụ đi kèm với nó trong thời gian gần đây thì những vấn đề về an toàn trong mạng internet cũng ngày càng được quan tâm nhiều hơn. Firewall với tính năng như một bộ lọc các gói dữ liệu ở mức cơ bản thì không đủ đáp ứng yêu cầu ngày càng cao của mạng internet. Vấn đề này đã làm phát sinh ra các thiết bị hay ứng dụng có tính năng dò tìm và phát hiện các tấn công trong một mạng internet. Những hệ thống này hoạt động dựa trên 2 cơ chế cơ bản đó là : phân tích những gói dữ liệu trong mạng và dựa vào so trùng các mẫu dữ liệu để tìm kiếm các loại tấn công đã được biết trước đó hay phát hiện các bất thường để phát ra cảnh báo cho người quản trị mạng. Mỗi cách tiếp cận đều có những ưu điểm và nhược điểm riêng của nó. Như cách so trùng mẫu dữ liệu (hay gọi là signature- based) thì có khả năng phát hiện được các tấn công đã được lưu trong cơ sở dữ liệu ở mức cao và ít có những cảnh báo giả nhưng nó lại không có khả năng phát hiện được những loại tấn công mới nên đòi hỏi phải cập nhật cơ sở dữ liệu liên tục và thường xuyên. Trong khi phương pháp dựa vào những bất thường trong mạng internet thì có thể phát hiện ra các loại tấn công mới nhưng bù lại thì khả năng phát hiện chính xác các loại tấn công thì không cao và thường phát ra những tín hiệu cảnh báo giả.

- Dù dùng phương pháp nào thì để phát hiện các loại tấn công luôn đòi hỏi một khối lượng tính toán rất lớn và với sự phát triển về tốc độ của mạng internet ngày nay thì các phần mềm gần như không thể đáp ứng được với yêu cầu về thời gian thực. Do

đó người ta có khuynh hướng dùng các mạch phần cứng để xử lí và FPGA như một giải pháp hiệu quả cho vấn đề này. Ngoài ra với khả năng tái cấu hình thì FPGA còn có thuận lợi hơn trong việc bảo trì, cập nhật hay nâng cấp hệ thống.

1.2 Giới thiệu đề tài

- Từ những yêu cầu an ninh trong mạng máy tính thì đề tài "Hiện thực hệ thống phát hiện các tấn công trong mạng internet dựa vào các dấu hiệu bất thường trên FPGA" được xây dựng nhằm giúp phát hiện được những loại tấn công mới cũng như tận dụng được sức mạnh tính toán của phần cứng để đáp ứng được yêu cầu về tính thời gian thực. Tuy vậy, việc sử dụng FPGA cũng đưa ra một thử thách lớn cho đề tài đó là làm sao vận dụng được tài nguyên một cách tối ưu vì mỗi chip FPGA có số "logic element" rất giới hạn.

- Do tập dữ liệu để kiểm tra tính đúng đắn của một hệ thống dò tìm và phát hiện tấn công thì không có sẵn trên mạng ngoại trừ tập dữ liệu trong dự án DARPA98 (Defense Advanced Research Projects Agency) của MIT Lincoln Lab[12]. Tuy nhiên, vì để thuận lợi cho việc xây dựng mô hình cũng như so sánh kết quả có được thì tôi chọn tập dữ liệu KDD99xem như vừa là tập huấn luyện và vừa là tập kiểm tra. Trong tập dữ liệu KDD99 thì mỗi mẫu là 1 tập hợp 41 thuộc tính của từng kết nối mạng, mà những kết nối mạng này có được từ dự án DARPA 98.

1.3 Tính cấp thiết của đề tài

- Do ngày càng có nhiều hình thức tấn công khác nhau được triển khai, nên nếu các hệ thống phát hiện tấn công chỉ dựa vào các loại tấn công đã được phát hiện thì không thể biết được các loại tấn công mới này, nhất là các kiểu tấn công dựa vào "Zero-day" (loại tấn công dựa vào các lỗ hỗng mà những lỗi này thường được phát hiện sau khoảng 384 ngày).

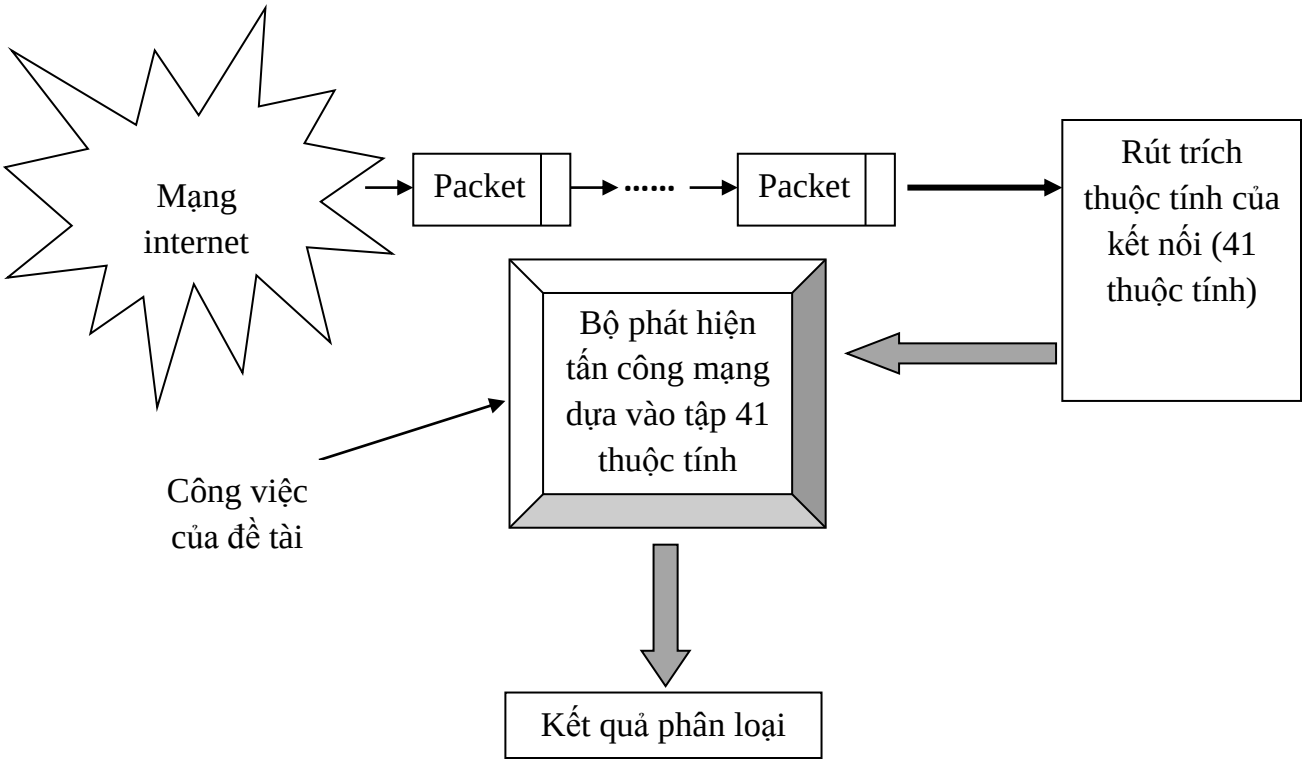
- Với những hệ thống phát hiện các loại tấn công mới có dựa vào sự bất thường trong mạng internet thì thời gian phát hiện thường chậm (khoảng vài ngày), chủ yếu do sự phức tạp của giải thuật cũng như khối lượng tính toán lớn. Nên với kiến trúc của FPGA thì ta sẽ có thể cải thiện được thời gian tính toán để phát hiện ra các loại tấn công.

- Nếu thành công thì đề tài sẽ giúp phát hiện các loại tấn công mới trong một thời gian ngắn. Và đồng thời khi được sử dụng cùng với các hệ thống phát hiện tấn công dựa trên sự so trùng mẫu thì sẽ càng tăng tính an toàn cho mạng máy tính.

1.4 Mục tiêu và giới hạn của đề tài

- Do giới hạn về thời gian nên mục tiêu của đề tài chỉ là xây dựng được phần lõi của hệ thống, dùng để phát hiện tấn công dựa vào tập 41 thuộc tính được rút trích từ mỗi kết nối mạng chứ không xây dựng hoàn chỉnh một hệ thống phát hiện tấn công trong mạng. Và hiện thực phần lõi đó xuống board NetFPGA của Xilinx.

- Phần công việc của đề tài có thể được mô tả rõ ràng trong hình bên dưới:



Hình 1 - Nhiệm vụ của đề tài

1.5 Tính khả thi của đề tài

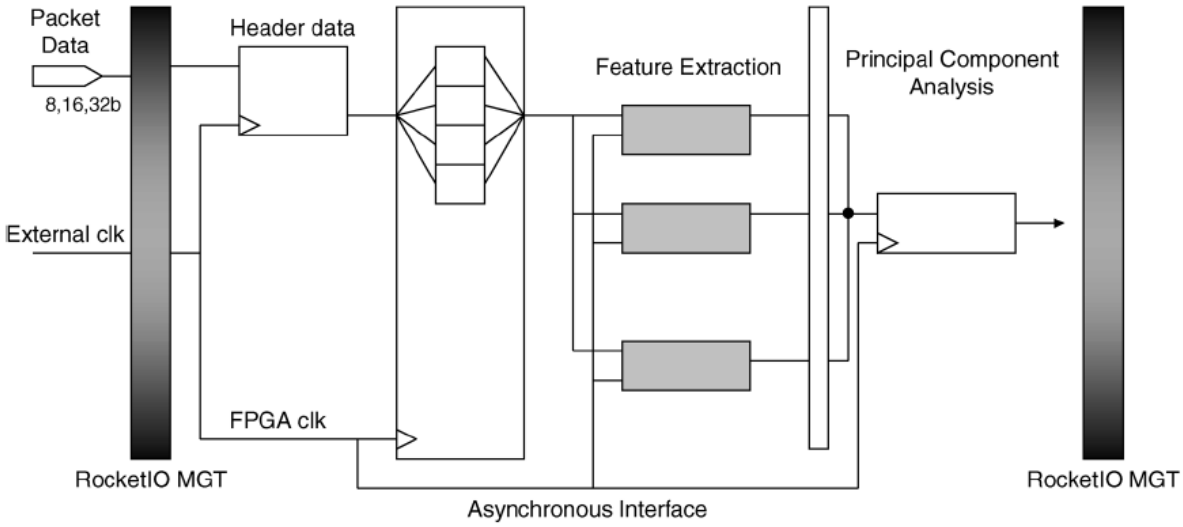
- Xuất phát trên yêu cầu cấp thiết của việc bảo đảm sự an ninh trong mạng internet và sự phát triển của công nghệ FPGA thì sự kết hợp 2 lĩnh vực này sẽ tạo ra một cách nhìn mới, một hướng giải quyết mới cho các vấn đề trong bảo mật mạng.

- Từ những năm 80 thì đã bắt đầu có những công trình nghiên cứu cũng như hệ thống hiện thực việc dò tìm các tấn công dựa trên sự bất thường của mạng internet. Dù hướng tiếp cận này trước đây không được quan tâm nhiều so với cách so trùng mẫu nhưng thời gian gần đây đã được tập trung nghiên cứu và có những thành công nhất định. Cộng với sự phát triển của nền tảng FPGA thì tôi tin tưởng đề tài sẽ thành công.

Phần 2. NHỮNG CÔNG TRÌNH LIÊN QUAN

2.1 Hệ thống phát hiện tấn công bằng phương pháp rút trích những thành phần cơ bản trên FPGA

- Hệ thống được hiện thực trên board FPGA gồm 2 module chính là module rút trích đặc trưng (FEM) và module phân tích các thành phần cơ bản (PCA)[1].



Hình 2 - Hệ thống phát hiện tấn công bằng phương pháp PCA

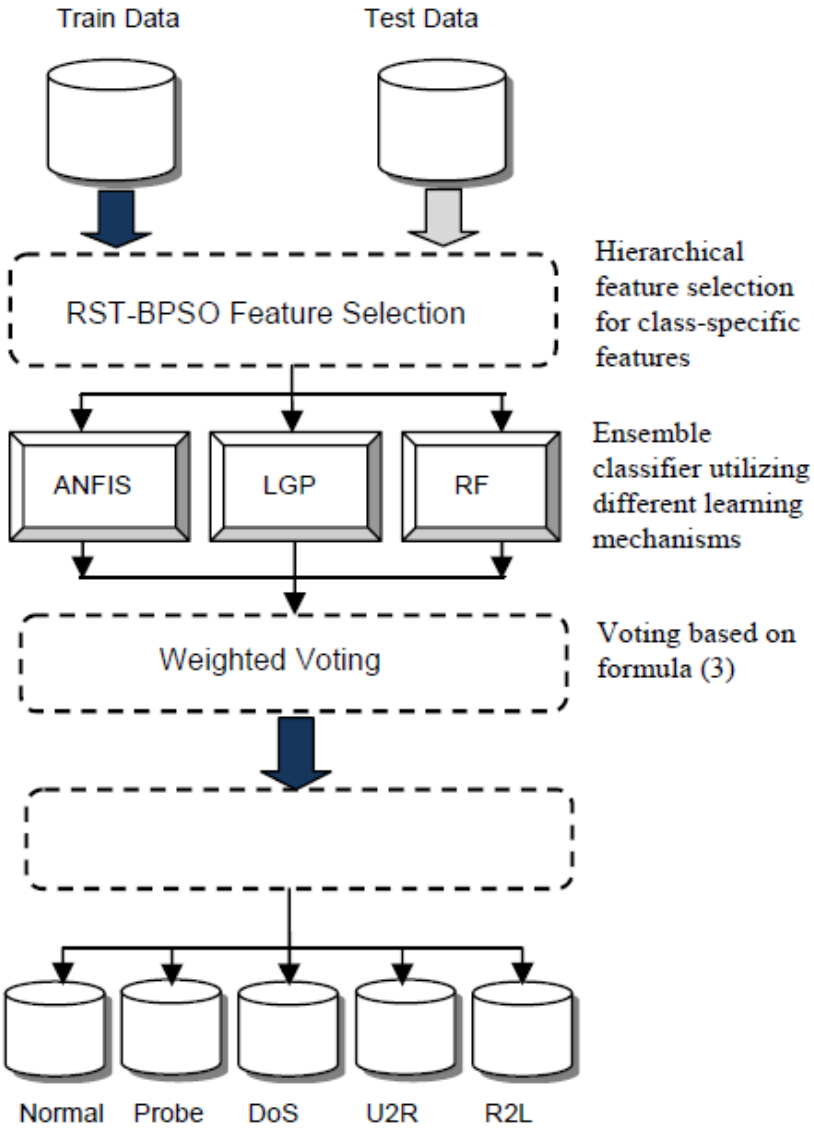
- Để rút trích những đặc trưng trong mạng internet thì tác giả chủ yếu dựa vào phần header của packet, nhưng ngoài ra vẫn có thêm thông tin như: thời gian duy trì connection, các cờ SYNC/FIN/RST. Sau đó thông tin từ module FEM sẽ được đưa vào PCA để rút gọn bớt số chiều và từ đây sẽ tính toán khoảng cách từ vector đặc trưng thu thập được với các vector có được khi mạng ở trạng thái bình thường. Nếu giá trị vượt quá một ngưỡng nào đó thì sẽ phát ra cảnh báo.

- Ưu điểm của cách làm này là giảm được số thuộc tính của từng kết nối mạng cần phải quan tâm, điều đó làm cho mạch trở nên đơn giản hơn và có tốc độ xử lí nhanh hơn. Tuy nhiên để đánh giá hiệu quả của hệ thống thì tác giả lại đánh giá tách rời giữa module rút trích đặc trưng mạng và module phân tích các thành phần cơ bản nên không phản ánh được tính chính xác của toàn hệ thống.

2.2 Kết hợp các phương pháp phân loại trong hệ thống phát hiện tấn công

- Mỗi phương pháp học đều có những điểm mạnh và yếu riêng nên tác giả đề nghị sự tổ hợp các phương pháp đó lại với nhau để tận dụng khả năng phân loại cũng như giảm thiểu những cảnh báo sai của từng giải thuật riêng lẻ. Những giải thuật mà tác giả chọn để kết hợp lại là: Linear Genetic Programming, Adaptive Neuro-Fuzzy Inference System, Random Forest[4].

- Tập 41 thuộc tính của từng kết nối mạng được rút gọn lại thành 15 thuộc tính và những thuộc tính này được sử dụng khác nhau cho từng loại kết nối mạng: Normal, Probe, DoS, User to Root (U2R), Remote to Local (R2L) trước khi được đưa đến các giải thuật riêng biệt để huấn luyện và kiểm tra. Mô hình hoạt động của hệ thống được đề xuất như sau:



Hình 3 - Hệ thống phát hiện tấn công bằng cách kết hợp nhiều giải thuật

- Với việc kết hợp nhiều giải thuật với nhau thì theo cách đánh giá của tác giả, hệ thống đạt được độ chính xác rất cao, khoảng trên 99%. Mặc dù tập dữ liệu được sử dụng để huấn luyện và test mà tác giả sử dụng được trích ra từ tập dữ liệu KDD99 nhưng lại có số lượng quá ít (khoảng 6000 kết nối) và không rõ ràng nên không thể so

sánh kết quả một cách chính xác, khách quan được. Tuy nhiên, thì việc kết hợp nhiều giải thuật phân loại khác nhau để nhằm tăng sự chính xác cũng là một gợi ý tốt.

2.3 Hệ thống phát hiện các bất thường trong mạng internet sử dụng giải thuật của lí thuyết thông tin, K-NN và KMC

- Tác giả sử dụng những kĩ thuật tính toán trong lí thuyết thông tin để xếp hạng mức độ quan trọng của các thuộc tính trong từng kết nối mạng và chỉ sử dụng những thuộc tính được cho là có mức độ quan trọng cao chứ không sử dụng hết tất cả 41 thuộc tính như tađã biết. Sau đó sử dụng phương pháp K-NN và KMC, cụ thể là xây dựng 2 giải thuật Selected feature 5NN (SF-5NN) và Suspect 5NN (SUS-5NN) để tính toán sự bất thường trong mạng internet bằng cách tính toán khoảng cách với các mẫu trong tập huấn luyện [5].

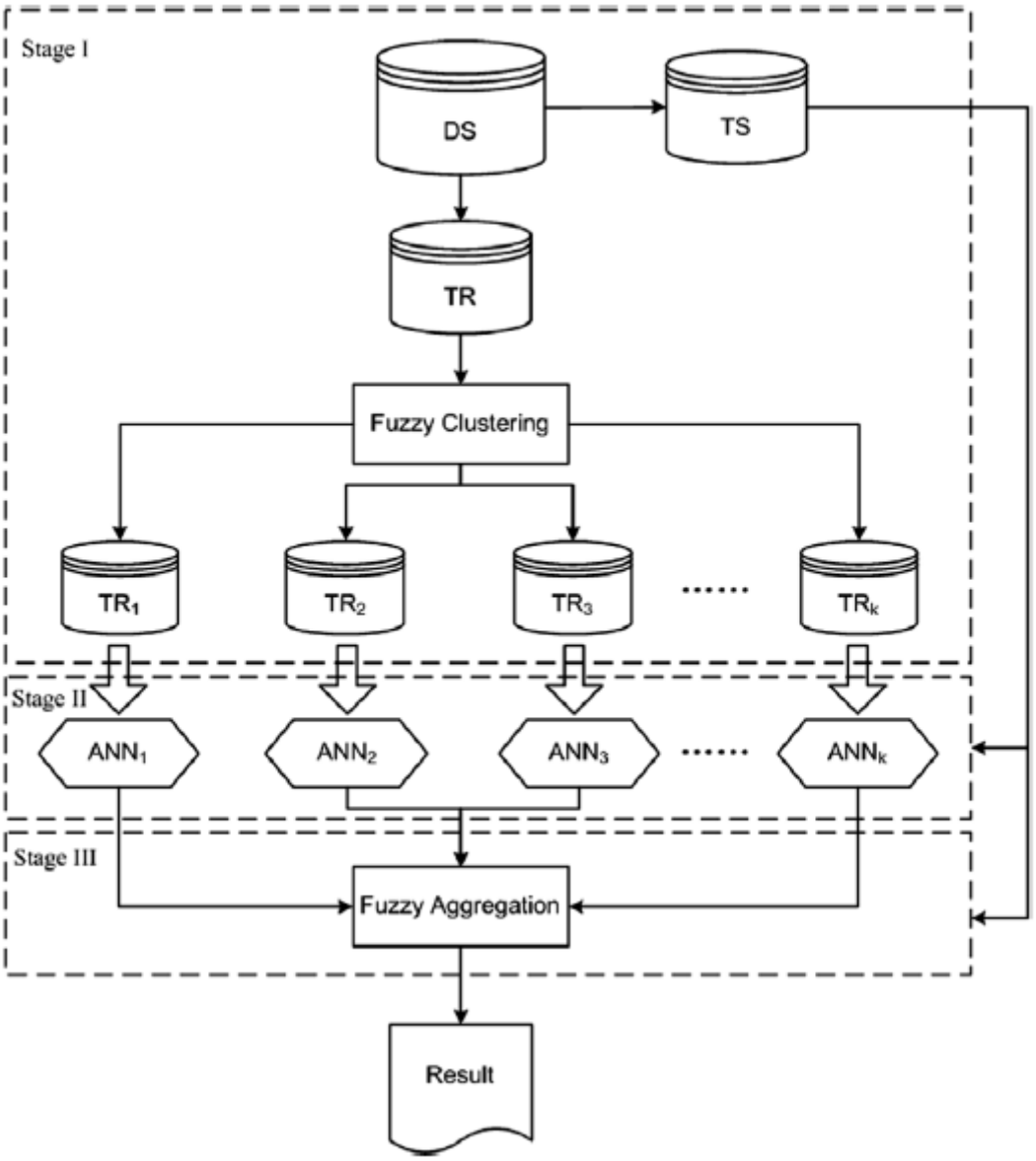
- Tuy độ chính xác tương đối cao (92.56% – 92.84%) nhưng khối lượng tính toán lớn và thời gian tính toán chậm nên phương pháp này chỉ thích hợp cho việc phân tích thông tin được lưu lại trên hệ thống hơn là dùng để phát hiện tấn công với thời gian thực.

2.4 Một cách tiếp cận mới cho hệ thống phát hiện tấn công bằng cách sử dụng mạng Neural và phương pháp phân cụm mờ

- Một số nghiên cứu đã chỉ ra sự hiệu quả của việc áp dụng mạng neural vào hệ thống phát hiện tấn công trong mạng internet nhưng dù được hiện thực dưới phương pháp nào đi nữa (học có giám sát, không giám sát hoặc lai ghép giữa 2 phương pháp học giữa giám sát và không giám sát) thì mạng neural cũng còn những mặt hạn chế

của nó như không có sự ổn định hay khả năng phát hiện kém các loại tấn công không thường xuyên. Phương pháp kết hợp giữa mạng neural và cách phân cụm mờ để cải thiện sự chính xác cho cách chỉ dùng duy nhất mạng neural [6].

- Cách hiện thực của phương pháp này gồm 3 bước:



Hình 4 - Mô hình của hệ thống FC-ANN

+ Dùng cách gom cụm C-Mean để phân chia tập huấn luyện thành 5 tập rời nhau (Normal, Probe, DoS, User to Root (U2R) and Remote to Local (R2L)). Vì cách gom cụm C-Mean là phương pháp phân chia mềm nên một mẫu dữ liệu để huấn luyện có thể được phân chia ở nhiều cụm khác nhau với tỉ lệ nhất định.

+ Xây dựng 5 mạng neural riêng biệt cho từng tập dữ liệu huấn luyện khác nhau. Mỗi mạng neural này có 3 tầng với 41 ngõ nhập cho tầng 1, tượng trưng cho 41 đặc tính của các mẫu dữ liệu để huấn luyện, tầng ẩn có 18 node và có 5 ngõ ra ở tầng cuối cùng.

+Bước tổng hợp cũng được hình thành nên 1 mạng neural 3 tầng với 5 ngõ nhập ở tầng 1 tương ứng với 5 ngõ xuất của bước 2, tầng ẩn có 13 node và cũng với 5 ngõ xuất.

- Với cách hiện thực này thì đã giảm bớt độ phức tạp trong việc xây dựng mạng neural và tăng độ chính xác cho hệ thống phát hiện tấn công. Tuy nhiên, phương pháp này thì chỉ hiệu quả trong việc phát hiện các tấn công có tần số thấp như U2R và R2L, còn về tổng thể thì khả năng phát hiện tấn công vẫn chưa cao.

Phần 3. CƠ SỞ LÍ THUYẾT

3.1 Kiến trúc tổng quát của một hệ thống phát hiện tấn công trong mạng internet

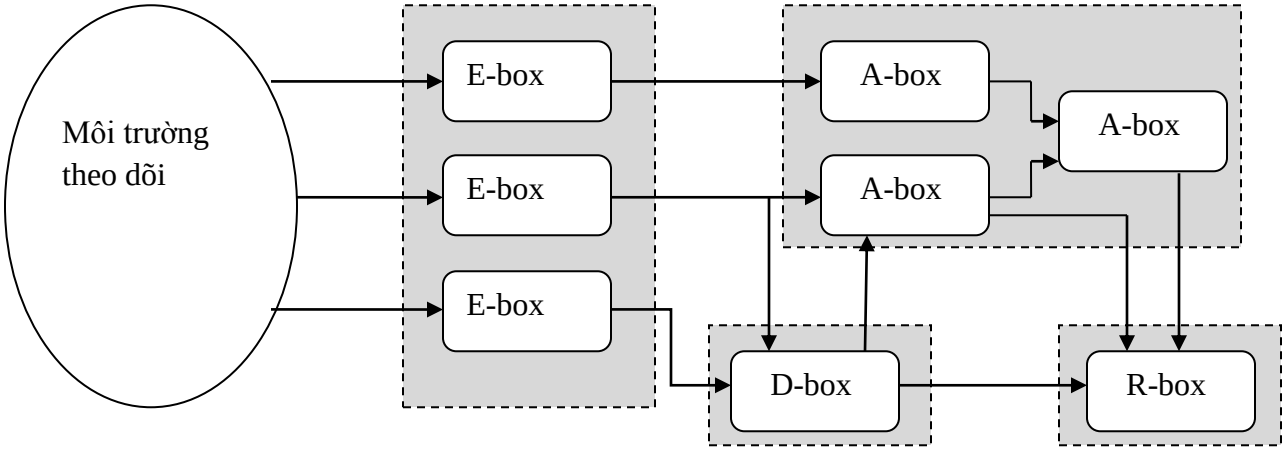
- Dự án DARPA (1998) đã đưa đến sự thành lập nhóm CIDF ("Common Intrusion Detection Framework") với nhiệm vụ chính là xây dựng nên 1 kiến trúc chung cho các hệ thống trong lĩnh vực phát hiện tấn công. Và đến năm 2000, thì nhóm này đã tích hợp với tổ chức IETF ("Internet Engineering Task Force") để thành lập nên nhóm IDWG ("Intrusion Detection Working Group") [10]. IDWG đã định ra một kiến trúc tổng quát cho các hệ thống phát hiện tấn công dựa vào 4 khối chức năng sau:

+ E-box (Event-boxes): Khối này được hình thành dựa trên 1 tập hợp các sensor dùng đển thu thập thông tin của hệ thống cần giám sát và những thông tin thu được sẽ được phân tích bởi các khối khác.

+ D-box (Database-boxes): Đây là những khối chức năng mà nhiệm vụ của nó là lưu lại những thông tin thu thập được từ những E-box để cho các khối A hoặc R xử lí.

+ A-box (Analysis-boxes): Khối này giữ nhiệm vụ xử lí, phân tích các thông tin từ E-box để phát hiện ra các tấn công hoặc phát ra cảnh báo.

+ R-box (Response-boxes): Chức năng chính của khối này chính là phát ra cảnh báo hoặc có những phản ứng thích hợp với những loại tấn công đã được phát hiện ra



Hình 5 - Kiến trúc tổng quát của một hệ thống phát hiện tấn công.

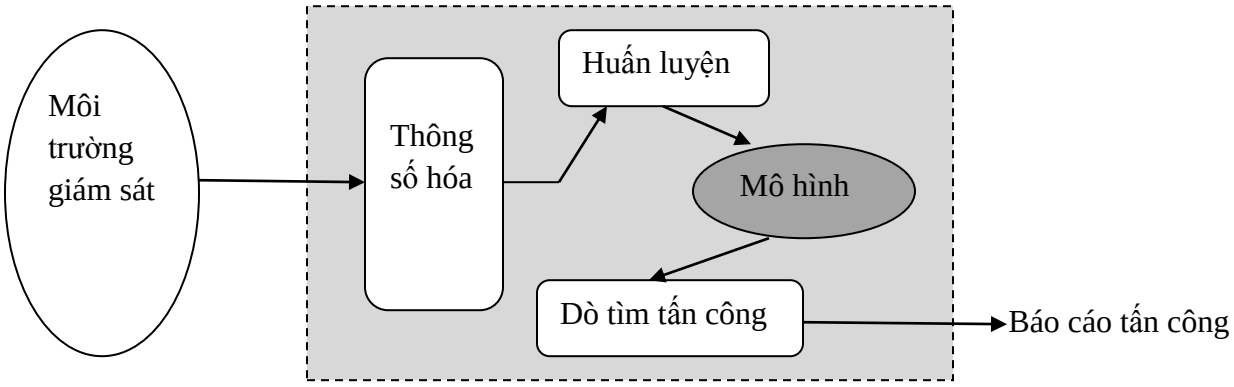
- Phụ thuộc vào nguồn thông tin thu thập được (E-box) thì ta phân chia hệ thống phát hiện tấn công thành 2 loại: dựa trên máy đơn hoặc trên 1 mạng máy tính.Trên 1 máy tính đơn thì thông tin dùng để phân tích tấn công là thông tin về các chương trình đang chạy và các lời gọi hàm hệ thống, thường có liên quan chính đến hệ điều hành. Còn trên 1 mạng máy tính thì thông tin đó lại là lưu lượng mạng, địa chỉ IP, các port dịch vụ, hoặc các giao thức trao đổi dữ liệu trên mạng đó

- Tùy vào cách thức phân tích, xử lí dữ liệu của khối A-box thì ta lại phân chia hệ thống phát hiện tấn công thành 2 kiểu đó là: dựa vào so trùng mẫu (signature-based hay misuse-based) hoặc dựa vào những dấu hiệu bất thường trong mạng

internet(anomaly-based). Luận văn này chúng ta chỉ tập trung xử lí và phát hiện tấn công dựa trên những dấu hiệu bất thường của mạng máy tính.

3.2 Các phương pháp được dùng trong hệ thống phát hiện tấn công dựa trên sự bất thường của mạng máy tính

- Mặc dù tồn tại nhiều cách tiếp cận khác nhau trong việc phát hiện tấn công trong hệ thống mạng dựa vào các dấu hiệu bất thường nhưng nhìn chung thì các phương pháp này đều có những khối chức năng cơ bản hoặc các bước giống nhau:



Hình 6- Các bước chính trong một hệ thống phát hiện tấn công dựa vào sự bất thường trong mạng internet

+ Thông số hóa: đây là bước mà ta dùng để đặc trưng hóa các trạng thái của hệ thống cần phải giám sát. Vì một hệ thống thường có rất nhiều yếu tố tác động lên nó mà nếu xử lí hết thì vừa không có tính khả thi mà vừa không hiệu quả nên ta phải rút gọn lại thành những thông số đặc trưng chính.

+ Huấn luyện: tập dữ liệu đặc trưng mô tả hành vi bình thường cũng như bất bình thường của hệ thống được dùng để huấn luyện và xây dựng nên mô hình dò

tìm, phát hiện tất công. Việc huấn luyện này có thể thực hiện một cách tự động hoặc thủ công tùy theo từng loại phương pháp.

+ Dò tìm tấn công: Một khi mô hình của hệ thống đã được xây dựng dựa vào bước huấn luyện thì ta sẽ so sánh với các thông số được đặc trưng của hệ thống. Nếu các thông số của hệ thống tại thời điểm quan sát cao hơn (hoặc thấp hơn) một giá trị ngưỡng mà ta qui định trước đó thì hệ thống sẽ phát ra cảnh báo.

+ Tùy thuộc vào kiểu xử lí liên quan đến mô hình được xây dựng cho hệ thống cần giám sát mà ta phân chia kĩ thuật phát hiện tấn công dựa trên sự bất thường thành 3 lớp chính: phương pháp thống kê, hệ tri thức và học máy.

- Phương pháp thống kê:

+ Trong phương pháp này thì các thông tin về hoạt động của mạng như lưu lượng dữ liệu được trao đổi, số packet cho từng loại giao thức, số lượng các kết nối hay địa chỉ IP khác nhau,…. sẽ được thu thập và tạo ra một bảng thông tin ngẫu nhiên đặc trưng cho mạng. Để thực hiện quá trình dò tìm và phát hiện tấn công thì phương pháp này cần 2 tập dữ liệu đó là dữ liệu cho trạng thái mạng hiện tại và một cho trạng thái được huấn luyện trong quá khứ để thực hiện quá trình so sánh và tính toán độ bất thường của hệ thống. Hệ thống sẽ phát ra cảnh báo khi độ bất thường vượt quá một ngưỡng đã được qui định trước.

+ Một trong những cách tiếp cận sớm nhất dựa trên phương pháp thống kê chính là mô hình đơn biến. Mô hình này thực hiện việc mô hình hóa các thông số dựa trên sự phân bố ngẫu nhiên và độc lập Gaussian, do đó mà khoảng giá trị chấp nhận được của mỗi biến đã được định ra trước. Nhưng trong các tập dữ liệu thực nghiệm

để kiểm tra và so sánh thì người ta thấy sẽ thuận lợi hơn trong việc phát hiện các tấn công bằng cách xác định mối quan hệ giữa các biến thay vì để các biến độc lập với nhau và đã dẫn tới việc xây dựng mô hình đa biến. Ngoài ra còn có một cách giải quyết khác chính là mô hình chuỗi thời gian. Với việc thực hiện quan sát các trạng thái của mạng cùng một bộ đếm sự kiện hay tài nguyên được sử dụng trong một khoảng thời gian nào đó thì hệ thống sẽ phát ra cảnh báo nếu một trạng thái được quan sát là có tỉ lệ xảy ra quá thấp so với trước đây.

+ Việc áp dụng phương pháp thống kê trong dò tìm và phát hiện tấn công cho mạng internet cũng có nhiều thuận lợi như không cần phải đặc tả trước thế nào là hoạt động bình thường của hệ thống mà phương pháp này có thể tự học được bằng cách quan sát và thời gian học càng dài thì hệ thống càng tăng tỉ lệ chính xác khi phát ra cảnh báo hơn. Tuy nhiên, phương pháp này cũng còn một số hạn chế. Trước hết thì đây là phương pháp rất nhạy cảm với lỗi vì nó dựa trên thông tin thống kê được, và việc thiết lập các giá trị ngưỡng là một công việc cực kì khó khăn bởi nó ảnh hưởng trực tiếp đến việc phát ra các cảnh báo giả hoặc bỏ qua các loại tấn công tới hệ thống. Ngoài ra hàm phân bố xác suất cho các biến cũng là một loại giả định mà không phải biến nào cũng tuân theo.

- Phương pháp sử dụng hệ cơ sở tri thức:

+ Hệ chuyên gia là một trong những phương pháp được sử dụng nhiều nhất dựa trên nền tảng hệ cơ sở tri thức trong việc dò tìm và phát hiện tấn công. Mục tiêu chính của hệ chuyên gia là dùng để phân loại các dòng dữ liệu cần xác thực dựa trên một tập luật và gồm có 3 bước. Đầu tiên là các thuộc tính và các lớp khác nhau của

dữ liệu sẽ được định ra dựa vào tập huấn luyện. Kế tiếp là một tập các luật dùng để phân loại cũng như các thông số và phương pháp sẽ được dẫn xuất ra. Cuối cùng là dòng dữ liệu cần được xác thực sẽ được phân vào các lớp tương ứng.

+ Hệ chuyên gia thì được xây dựng một cách thủ công bởi cách đặt tả các tập luật của các chuyên gia, nếu sự đặt tả này càng đầy đủ thì khả năng nhận biết các loại tấn công của hệ thống lại càng tăng cao. Ngoài ra nó cũng có tác dụng giảm thiểu các cảnh báo sai của hệ thống. Việc đặc tả các tập luật này có thể được phát triển bằng cách sử dụng một số công cụ hình thức như: máy trạng thái hữu hạn (FSM), ngôn ngữ mô tả (UML, N-grammars),…

+ Lợi điểm chính của việc phát triển hệ thống phát hiện tấn công theo hướng này chính là sự mạnh mẽ và tính uyển chuyển. Tuy nhiên khó khăn cũng chính là việc phát triển tập luật có chất lượng cao, đó là công việc rất khó và tốn kém thời gian.

- Phương pháp học máy:

+ Phương pháp học máy là kĩ thuật dựa trên việc thiết lập một kiểu mô hình tường minh hoặc không tường minh mà có khả năng phân loại các mẫu dữ liệu vào các lớp khác nhau. Một đặc trưng cho các cách thực hiện trong phương pháp này đó là cần một tập dữ liệu đã được gán nhãn trước để thực hiện quá trình huấn luyện cho mô hình được xây dựng. Mặc dù phương pháp học máy này có khả năng thay đổi chiến lược thực thi khi có yêu cầu mới về thông tin nhưng hạn chế của nó chính là cần quá nhiều tài nguyên cho quá trình tính toán. Một số cách thực hiện trong phương pháp học máy này

như: mạng Bayes, mô hình Markov, mạng neural [8], các kĩ thuật trong logic mờ, hay giải thuật di truyền [9] và các cách gom cụm [7] cùng với phát hiện các điểm bất thường.

+ Mạng Bayes: là một mô hình dùng để mã hóa mối quan hệ giữa các biến mà ta cần quan tâm. Nhìn chung thì kĩ thuật này thường được sử dụng kết hợp với phương pháp thống kê và nó đưa đến một số lợi ích như khả năng mã hóa mối quan hệ phụ thuộc giữa các biến, dự đoán được các sự kiện cũng như khả năng kết hợp với những dữ liệu và tri thức được biết trước. Tuy nhiên, mạng Bayes có một bất lợi rõ ràng chính là kết quả của nó thì thường tương tự với những gì được dẫn xuất từ hệ thống dựa trên ngưỡng nhưng lại đòi hỏi một khối lượng tính toán lớn. Mặc dù việc sử dụng mạng Bayes đã được chứng minh là hiệu quả trong một số trường hợp nhất định nhưng kết quả có được thì phụ thuộc quá nhiều vào việc giả định các hành vi của hệ thống.

+ Mô hình Markov: là tập hợp các trạng thái mà được kết nối với nhau thông qua những hàm chuyển trạng thái với xác suất nào đó và nó hình thành nên cấu trúc liên kết và khả năng của mô hình. Trong suốt giai đoạn xây dựng mô hình Markov thì ta phải xác định các giá trị xác suất gắn liền với những bước chuyển trạng thái và những giá trị xác suất này có được bằng cách quan sát hoạt động của hệ thống. Cách phát hiện ra các tấn công được thực hiện bằng cách tính toán sự bất thường (dựa vào các giá trị xác suất của các bước chuyển trạng thái) và so sánh với một giá trị ngưỡng đã được xác định trước. Mô hình Markov thì được sử dụng nhiều trong việc phát hiện tấn công với một máy đơn, chủ yếu dựa vào các lời gọi hàm hệ thống nhưng trong một số trường hợp nó cũng được áp dụng cho một mạng các máy tính. Với mô hình Markov thì trong hầu hết các trường hợp thì nó cũng cho một cách tiếp cận tốt khi được dẫn xuất từ

hệ thống cần giám sát, tuy nhiên cũng như mạng Bayes thì nó phụ thuộc quá lớn vào sự giả định ban đầu về hệ thống.

+ Mạng neural: với mục tiêu là xây dựng một mô hình mô phỏng theo hoạt động của bộ não con người, mạng neural đã được áp dụng vào trong lĩnh vực dò tìm và phát hiện tấn công dựa trên những bất thường của mạng bởi sự mềm dẻo và khả năng thích ứng với những thay đổi của môi trường giám sát. Tuy nhiên, một trong những đặc trưng chung của mạng neural đó là nó không cung cấp một mô hình rõ ràng với những thông số chi tiết để giải thích vì sao nó có thể đưa ra được những quyết định phát hiện tấn công của mạng.

+ Logic mờ: là kĩ thuật có nguồn gốc từ lí thuyết tập hợp mờ với suy luận dựa trên sự xấp xỉ chứ không mang tính chính xác, rõ ràng như trong logic cổ điển. Logic mờ được áp dụng vào trong lĩnh vực bảo mật mạng nhờ vào khả năng mờ hóa các biến đặc trưng cho môi trường mạng. Mặc dù logic mờ đã cho thấy sự hiệu quả của nó, đặc biệt trong phát hiện một số loại tấn công như dò tìm port hoặc thăm dò hệ thống mạng tuy nhiên nó lại có một nhược điểm lớn đó là sử dụng tài nguyên quá nhiều.

+ Giải thuật di truyền: là một kỹ thuật nhằm tìm kiếm giải pháp thích hợp cho các bài toán tối ưu tổ hợp và nó cũng là một phân ngành của giải thuật tiến hóa bằng cách vận dụng các nguyên lý của tiến hóa như di truyền, đột biến, chọn lọc tự nhiên và trao đổi chéo. Do đó mà giải thuật di truyền cũng được xem như một loại trong phương pháp học máy với khả năng dẫn xuất ra các luật để phân loại, lựa chọn các thuộc tính đặc trưng thích hợp và tối ưu hóa các thông số trong quá trình phát hiện các loại tấn

công. Điểm thuận lợi chính của giải thuật di truyền là ở khả năng mềm dẻo và phương pháp tìm kiếm lời giải tối ưu tốt, tuy nhiên nó cũng cần tiêu tốn tài nguyên quá nhiều.

+ Gom cụm và phát hiện các điểm bất thường: giải thuật gom cụm hoạt động bằng cách gom nhóm những đối tượng được cho trước vào các nhóm khác nhau dựa vào sự tương tự và dùng một phần tử đại diện cho mỗi nhóm. Khi có một phần tử mới thì sẽ thực hiện việc tính toán độ tương tự với các phần tử đại diện và sau đó xếp phần tử mới này vào một nhóm thích hợp. Nếu như phần tử này không thuộc vào bất kì nhóm nào thì đây chính là sự bất thường trong quá trình phát hiện các loại tấn công. Tuy nhiên phương pháp gom cụm chỉ phát hiện ra các loại tấn công đối với dòng dữ liệu xác thực thô.

- Ngoài 3 lớp giải thuật chính được liệt kê bên trên thì cũng còn có nhiều phương pháp khác được sử dụng trong việc phát hiện các tấn công như phương pháp phân tích các thành phần chính (Principal component analysis – PCA) hay phương pháp tìm kiếm các luật kết hợp (Association rule discovery).

3.3 Rút trích những thông tin đặc trưng trong mạng internet

- Với một hệ thống dò tìm tấn công dựa vào sự bất thường trong mạng internet thì phải xây dựng được các phương pháp rút trích những thông tin đặc trưng và xem đây là những tiêu chí để so sánh giữa các trạng thái của mạng

- Dựa vào các công trình nghiên cứu trước đây, mỗi kết nối trong mạng sẽ rút trích ra được 41 thuộc tính đặc trưng. Những thuộc tính này thì được chia thành 3 nhóm chính như sau:

+ Nhóm những thuộc tính cơ bản của một kết nối mạng, những thuộc tính này được rút ra từ phần header của packet:

Số thứ tự	Tên thuộc tính	Kiểu	Mô tả
1	duration	Integer	Thời gian duy trì của một kết nối
2	protocol_type	Nominal	Giao thức của kết nối mạng: TCP, UDP hay ICMP
3	service	Nominal	http, ftp, smtp, telnet,….
4	flag	Nominal	Trạng thái của kết nối, có thể có các trạng thái sau: S0, S1, S2, S3, OTH, REJ, RSTO, SRSOS0, SH, RSTRH, SHR
5	src_bytes	Integer	Số byte được gởi đi trong 1 kết nối
6	dst_bytes	Integer	Số byte nhận được trong 1 kết nối
7	land	Binary	Có giá trị là 1 nếu kết nối đến và đi từ cùng 1 địa chỉ IP/port, và có giá trị là 0 cho các trường hợp còn lại
8	wrong_fragment	Integer	Tổng số các packet có phần checksum bị lỗi

9	urgent	Integer	Tổng số packet có bit "urgent" được bật lên trong 1 kết nối.

Bảng 1- Những thuộc tính cơ bản của 1 kết nối

+ Nhóm những thuộc tính nội dung của kết nối mạng, những thuộc tính này được rút ra từ phần nội dung của packet trong 1 kết nối dựa trên kiến thức của những chuyên gia về mạng:

Số thứ tự	**Tên thuộc tính**	**Kiểu**	**Mô tả**
10	hot	Integer	Tổng số các hành động "hot" trong 1 kết nối như: truy cập vào thư mục hệ thống, tạo chương trình hay thực thi một chương trình nào đó
11	num_failed_logins	Integer	Số lần login không thành công
12	logged_in	Binary	Sẽ có giá trị là 1 nếu login thành công, còn ngược lại sẽ có giá trị là 0
13	num_compromised	Integer	Tổng số lần lỗi "not found" xuất hiện trong 1 kết nối
14	root_shell	Binary	Có giá trị là 1 nếu root mở 1 shell và là 0 nếu ngược lại

15	su_attempted	Binary	Có giá trị là 1 nếu lệnh "su" được sử dụng, nếu không thì có giá trị là 0
16	num_root	Integer	Tổng số các tác vụ được thực hiện với tài khoản root
17	num_file_creations	Integer	Tổng số các tác vụ tạo file được diễn ra
18	num_shells	Integer	Tổng số shell được tạo ra trong 1 kết nối
19	num_access_files	Integer	Tổng số thao tác điều khiển file
20	num_outbound_cmds	Integer	Tổng số lệnh "outbound" trong 1 ftp session
21	is_hot_login	Binary	Nếu user đang login vào hệ thống dưới vai trò là root hay admin
22	is_guest_login	Binary	Nếu user đang login vào hệ thống dưới vai trò là khách, người viếng thăm hay mạo danh

Bảng 2- Những thuộc tính nội dung của 1 kết nối

+ Nhóm những thuộc tính phản ánh lưu lượng trong mạng internet. Những thuộc tính này được tính toán dựa trên những kết nối trước đó và được chia thành 2 nhóm nhỏ hơn là : lưu lượng theo thời gian và lưu lượng theo máy (machine traffic attribute).

Sự khác nhau giữa 2 nhóm thuộc tính này là ở cách lựa chọn những kết nối trước đó trong mô hình tính toán.

- Lưu lượng theo thời gian: để tính toán những thuộc tính này thì ta dựa vào những kết nối trong khoảng thời gian 2 giây trước đó.

Số thứ tự	**Tên thuộc tính**	**Kiểu**	**Mô tả**
23	count	Integer	Tổng số các kết nối có cùng địa chỉ IP ở máy đích
24	srv_count	Integer	Tổng số các kết nối có cùng địa chỉ port ở máy đích
25	serror_rate	Real	Tỉ lệ các kết nối đã bật thuộc tính flag (4) lên thành S0, S1, S2, hay S3 trong tổng số kết nối được tính trong thuộc tính *count (23)*
26	srv_serror_rate	Real	Tỉ lệ các kết nối đã bật thuộc tính flag (4) lên thành S0, S1, S2, hay S3 trong tổng số kết nối được tính trong thuộc tính *srv_count (24)*

27	rerror_rate	Real	Tỉ lệ các kết nối đã bật thuộc tính flag (4) lên thành REJ trong tổng số kết nối được tính trong thuộc tính *count (23)*
28	srv_error_rate	Real	Tỉ lệ các kết nối đã bật thuộc tính flag (4) lên thành REJ trong tổng số kết nối được tính trong thuộc tính *srv_count (24)*
29	same_srv_rate	Real	Tỉ lệ các kết nối có cùng dịch vụ giữa những kết nối được tính trong thuộc tính *count (23)*
30	diff_srv_rate	Real	Tỉ lệ các kết nối có dịch vụ khác nhau giữa những kết nối được tính trong thuộc tính *count (23)*
31	srv_diff_host_rate	Real	Tỉ lệ các kết nối có địa chỉ máy đích khác nhau giữa những kết nối được tính trong thuộc tính *srv_count (24)*

Bảng 3- Những thuộc tính lưu lượng theo thời gian

- Lưu lượng theo máy: để tính toán những thuộc tính này thì ta dựa vào 100 kết nối trước đó.

Số thứ tự	Tên thuộc tính	Kiểu	Mô tả
32	dst_host_count	Integer	Tổng số các kết nối có cùng địa chỉ IP của máy đích
33	dst_host_srv_count	Integer	Tổng số các kết nối có cùng địa chỉ port ở máy đích
34	dst_host_same_srv_rate	Real	Tỉ lệ các kết nối có cùng dịch vụ giữa những kết nối được tính trong thuộc tính *dst_host_count (32)*
35	dst_host_diff_srv_rate	Real	Tỉ lệ các kết nối có các dịch vụ khác nhau giữa những kết nối được tính trong thuộc tính *dst_host_count (32)*
36	dst_host_same_src_port_rate	Real	Tỉ lệ các kết nối có cùng địa chỉ port ở máy nguồn trong tổng số các kết nối được tính trong thuộc tính *dst_host_srv_count(33)*
37	dst_host_srv_diff_host_rate	Real	Tỉ lệ các kết nối có địa chỉ máy đích khác nhau trong tổng số các kết nối được tính trong thuộc tính *dst_host_srv_count (33)*

38	dst_host_serror_rate	Real	Tỉ lệ các kết nối đã bật thuộc tính flag (4) lên thành S0, S1, S2 hay S3 trong tổng số kết nối được tính trong thuộc tính *dst_host_count (32)*
39	dst_host_srv_serror_rate	Real	Tỉ lệ các kết nối đã bật thuộc tính flag (4) lên thành S0, S1, S2 hay S3 trong tổng số kết nối được tính trong thuộc tính *dst_host_srv_count (33)*
40	dst_host_rerror_rate	Real	Tỉ lệ các kết nối đã bật thuộc tính flag (4) lên thành REJ trong tổng số kết nối được tính trong thuộc tính *dst_host_count (32)*
41	dst_host_srv_error_rate	Real	Tỉ lệ các kết nối đã bật thuộc tính flag (4) lên thành REJ trong tổng số kết nối được tính trong thuộc tính *dst_host_srv_count (33)*

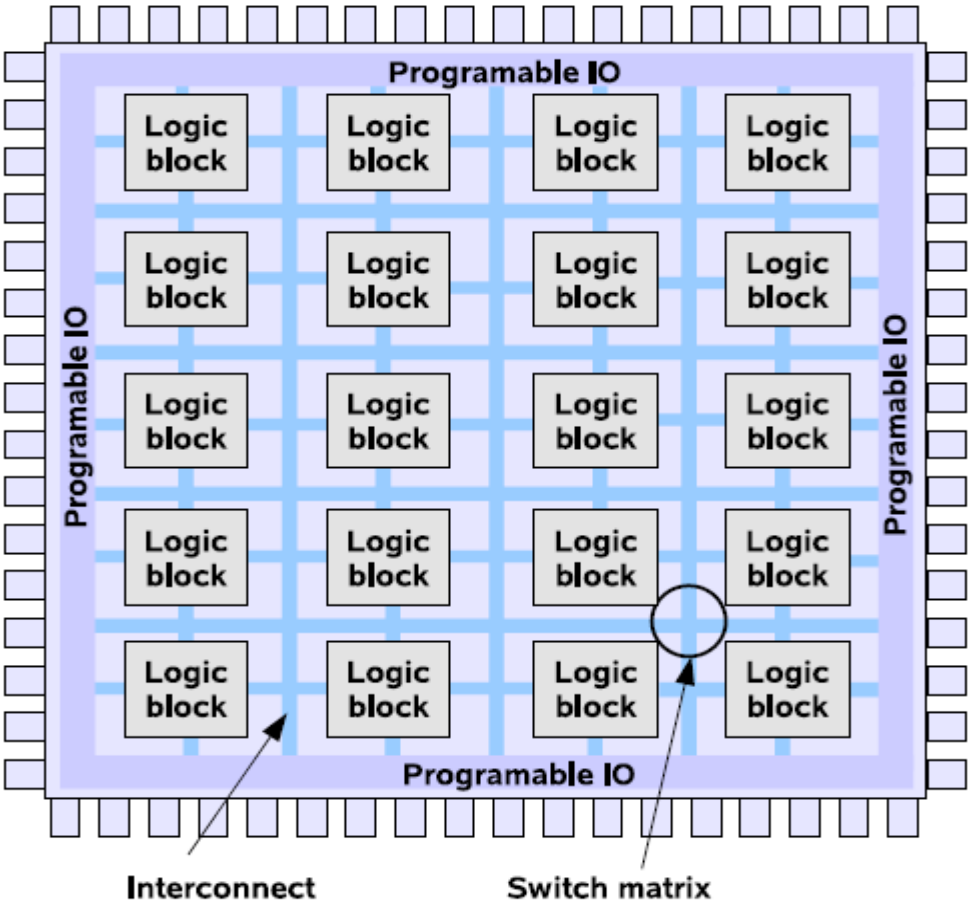
Bảng 4- Những thuộc tính lưu lượng theo máy

3.4Công nghệ Field Programmable Gate-Array và board NetFPGA

- Thiết bị FPGA là một thiết bị bán dẫn mà có thể được cấu hình bởi người lập trình sau khi thiết bị đã được sản xuất, chính vì vậy mà nó có tên là "field-programmable". FPGA được lập trình sử dụng bằng ngôn ngữ đặc tả phần cứng (HDL) [11].

- FPGA chứa các khối gọi là khối luận lý (Logic Blocks) và một kiến trúc có thể tái cấu hình các kết nối bên trong điều này cho phép các khối có thể kết nối được với nhau. Các khối có thể được cấu hình để thực hiện một chức năng phức tạp, hay chỉ đơn giản là sự kết hợp của các cổng luận lý. Trong hầu hết các FPGA thì các khối luận lý nó bao gồm các đơn vị bộ nhớ (các đơn vị bộ nhớ này có thể đơn giản chỉ là các flip-flops hay cũng có thể là các khối bộ nhớ hoàn chỉnh).

- Quy trình thiết kế FPGA bao gồm các bước: đặc tả chức năng, hiện thực sang ngôn ngữ đặc tả phần cứng (HDL), mô phỏng hành vi, tổng hợp, kết nối, phân tích ràng buộc thời gian, nạp lên FPGA. Hiện nay thì các nhà sản xuất FPGA chính hiện nay có thể kể đến là Xilinx, Altera hay Actel, Achronix… Trong đó thì Xilinx chiếm đến 50% và cung cấp luôn phần mềm thiết kế trên Windows và Linux, Altera hỗ trợ miễn phí các công cụ trên windows.

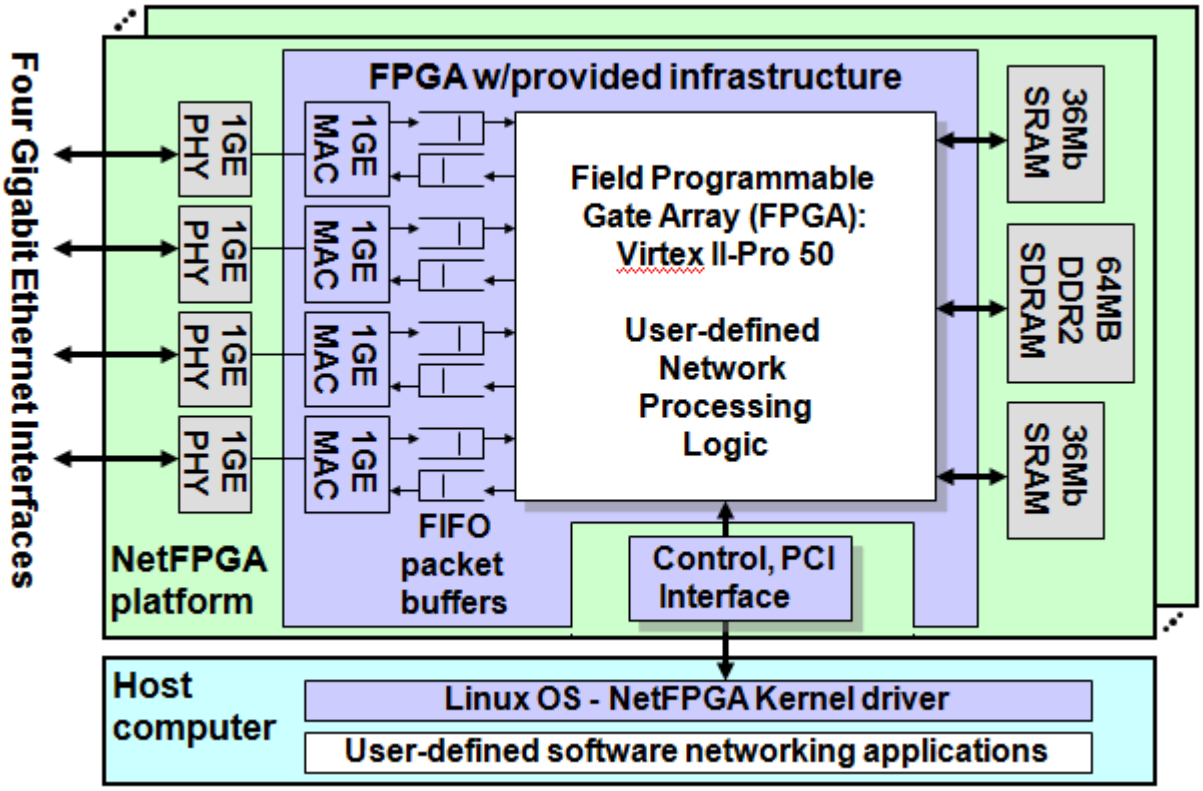


Hình 7 - Cấu trúc tổng quát của một chip FPGA[13]

- Kiến trúc board NetFPGA của Xilinx

Hình 8 - Board NetFPGA [16]

- FPGA chip: Xilinx Virtex II Pro50
  - ✓ 53,136 logic cells
  - ✓ 4,176 Kbit BlockRAM (232 18Kb-BlockRAMs)
  - ✓ 738 Kbit distributed RAM
  - ✓ 2 PowerPC cores
- 4 Gigabit Ethernet ports
- 4.5 Mbytes SRAM (2 x 2.25 Mbytes; 36 x 512k)
- 64 Mbytes SDRAM (32 x 16M)
- Standard PCI interface



Hình 9 - Kiến trúc bên trong của board NetFPGA [16]

Phần 4. HIỆN THỰC HỆ THỐNG

4.1 Tập dữ liệu KDD 99(Knowledge Discovery and Data mining)

- Dự án DARPA 1998 được chuẩn bị và quản lí bởi phòng thí nghiệm MIT Lincoln. Mục tiêu của dự án là để khảo sát và đánh giá các phương pháp trong việc dò tìm và phát hiện các loại tấn công trong mạng internet. Một tập dữ liệu chuẩn đã được hình thành, và trong đó bao gồm một loạt các sự mô phỏng cho việc xâm nhập vào mạng máy tính trong môi trường quân sự. Cuộc thi KDD 99 được tổ chức với chủ đề là dò tìm sự tấn công trong mạng máy tính đã sử dụng một phiên bản của tập dữ liệu này [15].

- Lincoln Labs đã thiết lập và vận hành một môi trường mạng LAN giả lập U.S AirForce LANnhưng thêm vào đó với nhiều loại tấn công để có được dữ liệu thô trong 9 tuần dưới dạng TCP dump.

- Lưu lượng trong mạng LAN của 7 tuần đầu được sử dụng như tập dữ liệu để huấn luyện với dung lượng khoảng 4GB và có 5 triệu kết nối. Tương tự như vậy thì tập dữ liệu có được trong 2 tuần cuối được sử dụng như dữ liệu để test với khoảng 2 triệu kết nối.

- Một kết nối là một chuỗi các TCP packet được gởi đi và nhận về từ những địa chỉ IP thông qua một giao thức mạng được định nghĩa rõ ràng trong 1 khoảng thời gian nhất định. Mỗi một kết nối thì được đánh dấu là bình thường hay là một loại tấn công với kiểu tấn công được ghi chú chính xác. Mỗi record kết nối thì có dung lượng khoảng 100 bytes. Ví dụ về 1 số record kết nối như sau:

0,tcp,http,SF,181,5450,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00,0.00,9,9,1.00,0.00,0.11,0.00,0.00,0.00,0.00,0.00,normal.

0,icmp,ecr_i,SF,1032,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,511,511,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,255,1.00,0.00,1.00,0.00,0.00,0.00,0.00,0.00,smurf.

- Các loại tất công thì được chia làm 4 loại chính:

- DOS (denial-of-service): dạng tấn công từ chối dịch vụ, ví dụ như syn flood, …
- R2L (remote to local) : dạng tấn công truy cập trái phép từ những máy từ xa nhằm chiếm quyền sử dụng với một tài khoản người dùng bình thường, ví dụ như ftp_write,….
- U2R (user to root) : dạng tấn công truy cập trái phép vào tài nguyên của hệ thống dưới quyền "root" từ các tài khoản người dùng bình thường, ví dụ như các loại tấn công khác nhau kiểu "buffer overflow"
- PROBE: đây là dạng tấn công kiểu dò tìm để biết tình trạng của mạng đang hoạt động và là bước chuẩn bị cho các cuộc tấn công mạng thật sự, ví dụ như: port scanning…

- Một lưu ý quan trọng là tập dữ liệu dùng để test thì có xác suất phân bố dữ liệu khác với tập dữ liệu dùng để huấn luyện và nó bao gồm 1 số loại tấn công mà không có trong tập huấn luyện. Điều này sẽ làm cho tập dữ liệu của cuộc thi trở nên gần với thực tế hơn. Một số chuyên gia mạng tin rằng hầu hết các loại tấn công mới chính là

những biến thể từ những loại tấn công đã được biết trước đó và những dấu hiệu cho các loại tấn công cũ cũng là có thể đủ để phát hiện ra các biến thể mới. Tập dữ liệu có chứa 24 loại tấn công cho quá trình huấn luyện và 14 loại tấn công để test. Tổng cộng có 38 loại tấn công và được phân chia vào các nhóm chính như bảng dưới đây:

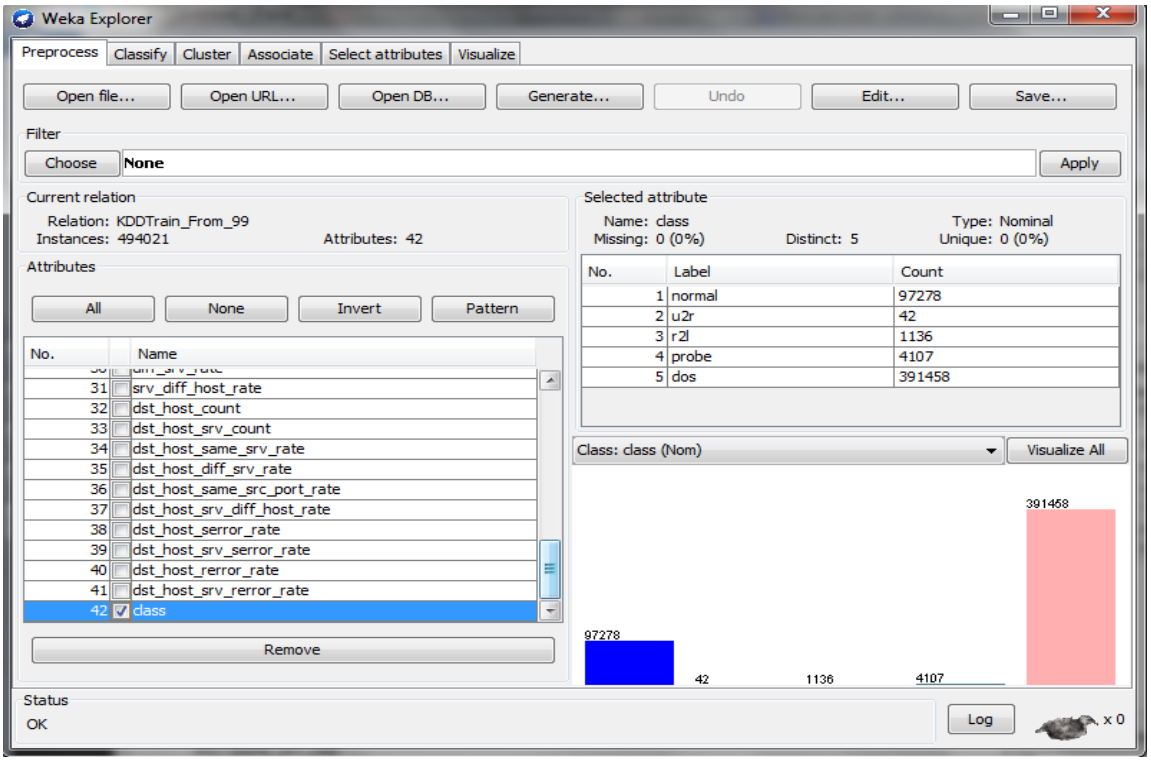
Số thứ tự	**Tên nhóm tấn công**	**Các loại tấn công được liệt kê trong nhóm**
1	U2R (7)	sqlattack, buffer_overflow, ps, loadmodule, perl, xterm, rootkit
2	R2L (15)	snmpguess,snmpgetattack,worm,warez(warezmaster, warezclient) , guess_passwd, ftp_write, phf, xlock, xsnoop, imap, named, sendmail, multihop, spy, httptunel
3	PROBE (6)	ipsweep, nmap, portsweep, satan, mscan, saint
4	DOS (10)	back, neptune, pod, smurf, land, apache2, mailbomb, processtable, udpstorm, teardrop

Bảng 5- Các loại tấn công trong tập dữ liệu KDD99

4.2 Công cụ hỗ trợ xây dựng và kiểm thử các phương pháp trong datamining

- WEKA [14] là một phần mềm mở về Data Mining, nó tập hợp nhiều giải thuật về học máy được viết trên nền tảng Java và phát triển bởi đại học Waikato. Những giải thuật này có thể được áp dụng trực tiếp trên tập dữ liệu hoặc được gọi từ Java code của người sử dụng. WEKA có những công cụ cho phép tiền xử lí dữ liệu, phân loại dữ liệu, gom cụm, rút trích luật kết hợp hay dùng để trực quan hóa dữ liệu. Nó cũng là 1 bộ công cụ tốt trong việc phát triển các giải thuật học máy mới.

- Phần mềm WEKA có 4 chức năng chính: Explorer, Experimenter, KnowledgeFlow, SimpleCLI. Nhưng ta chỉ sử dụng chủ yếu là chức năng Explorer. Với Explorer, ta đủ để xây dựng và kiểm thử tính đúng đắn của mô hình được sinh ra.

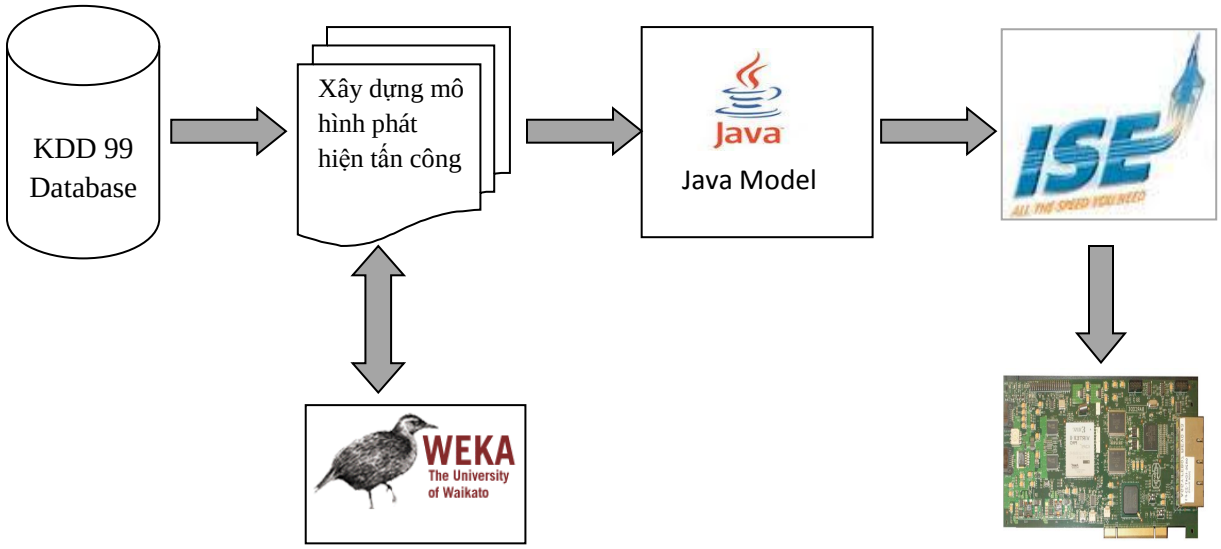


Hình 10 - Giao diện của phần mềm WEKA với tính năng Explorer

4.3 Qui trình phát triển và hiện thực hệ thống

- Để xây dựng và phát triển một hệ thống lên trên board FPGA thì ta cần tuân theo một qui trình phát triển rõ ràng để đảm bảo tính đúng đắn của hệ thống cũng như thuận lợi trong việc tìm ra lỗi khi chương trình hoạt động không đúng như đặc tả ban đầu.

- Qui trình phát triển mà tôi sử dụng để hiện thực hệ thống phát hiện tấn công trong mạng internet dựa vào các dấu hiệu bất thường của mạng như sau:



Hình 11- Qui trình phát triển và hiện thực hệ thống lên board NetFPGA

- Từ tập dữ liệu KDD99 ta sẽ xây dựng mô hình để thực hiện quá trình dò tìm và phát hiện tấn công trong mạng internet bằng cách dùng phần mềm Weka để thực hiện việc kiểm tra tính đúng đắn của mô hình. Sau khi có được mô hình với độ chính xác và tính phức tạp ở mức chấp nhận được như đặc tả ban đầu thì ta sẽ chuyển sang hiện

thực mô hình dưới dạng "Java model". Ở bước này thì thông thường sẽ được phát triển thành "C model" nhưng do những thuận lợi của ngôn ngữ Java trong việc xử lí chuỗi kí tự mà tôi chọn ngôn ngữ Java. Quá trình xây dựng "Java model" rất quan trọng vì nó là bước xác thực lại tính chính xác của mô hình cũng như những dữ liệu có được từ đây sẽ được xem như "golden data" cho bước phát triển mô hình bằng ngôn ngữ Verilog. Sau khi có được "Java model" thì ta phát triển hệ thống bằng ngôn ngữ Verilog trên môi trường ISE của Xilinx để thực hiện quá trình mô phỏng và dịch chương trình trước khi đưa file đã được tổng hợp xuống board NetFPGA.

4.4Xây dựng mô hình

- Như ta đã thấy trong phần cơ sở lí thuyết thì để xây dựng mô hình cho hệ thống phát hiện xâm nhập và tấn công trong mạng internet thì sẽ có nhiều cách làm nhưng do yêu cầu đặc thù của hệ thống là hiện thực trên board NetFPGA nên ta phải đảm bảo mô hình vừa đơn giản, dễ hiện thực và vừa có độ chính xác cao. Trong 3 phương pháp tổng quát là hệ tri thức, học máy và thống kê thì tôi nhận thấy phương pháp học máy là phù hợp trong trường hợp này hơn cả.

- Bảng sau đây thực hiện việc so sánh các khía cạnh khác nhau của những giải thuật thường được sử dụng trong phương pháp học máy có sự giám sát: trong đó **** thể hiện mức cao nhất và * cho mức thấp nhất.

	Cây quyết định	**Mạng neural**	**Mạng Bayes ngây thơ**	**kNN**	**Support Vector Machine**	**Rút trích qui luật**
Khả năng phân loại chính xác tổng quát	**	***	*	**	****	**
Tốc độ học tương ứng với số thuộc tính và độ lớn của dữ liệu	***	*	****	****	*	**
Tốc độ phân loại	****	****	****	*	****	****
Khả năng chống lại việc thiếu giá trị	***	*	****	*	**	**
Khả năng chống lại việc những thuộc tính không	***	*	**	**	****	**

liên quan nhau						
Khả năng xử lí việc dư thừa thuộc tính	**	***	*	*	***	**
Khả năng xử lí việc các thuộc tính có sự phụ thuộc cao lẫn nhau	**	***	*	*	***	**
Khả nẳng xử lí các kiểu dữ liệu của thuộc tính (nhị phân, rời rạc, liên tục)	****	*** (không có khả năng xử lí kiểu rời rạc)	*** (không có khả năng xử lí kiểu liên tục)	*** (không có khả năng xử lí kiểu rời rạc trực tiếp	** (không có khả năng xử lí kiểu rời rạc)	*** (không có khả năng xử lí kiểu liên tục trực tiếp)
Khả năng chống nhiễu	**	**	***	*	**	*
Khả năng xử lí	**	*	***	***	**	**

với việc "overfitting"						
Khả năng thích ứng với viêc học tăng cường	**	***	****	****	**	*
Khả năng giải thích rõ ràng của mô hình được sinh ra	****	*	****	**	*	****
Khả năng điều khiển các thông số của mô hình	***	*	****	***	*	***

Bảng 6- So sánh giữa các giải thuật trong phương pháp học máy[17]

- Dựa vào kết quả bảng so sánh bên trên thì tính về mặt trung bình ta sẽ thấy phương pháp cây quyết định sẽ tốt hơn cả vì vừa đơn giản, vừa có tính chính xác tương đối cũng như khả năng phân loại với nhiều kiểu khác nhau của thuộc tính. Tất nhiên ở đây ta sẽ tìm cách để tăng độ chính xác của phương pháp này lên, nhưng nếu đơn thuần chỉ hiện thực 1 cây quyết định riêng lẻ thì so với các phương pháp khác cây quyết

định cũng cho độ chính xác khá cao trong thực nghiệm (Ở đây tôi không chọn phương pháp kNN, mạng Neural hay SVM để khảo sát vì những phương pháp đó nếu cho kết quả tốt cũng rất phức tạp và khó khăn khi hiện thực lên board FPGA).

	Cây J48	**Random Forest**	**Cây AD**	**Mạng Bayes**	**Bảng quyết định**	**Rút trích thuộc tính để phân loại**
Tỉ lệ phát hiện tấn công(%)	90.4123	77.7679	89.7742	89.4262	84.2528	84.8397

Bảng 7- So sánh tỉ lệ phát hiện tấn công của các giải thuật học máy

- Từ bản so sánh bên trên, tôi tập trung vào cây J48 để tìm cách nâng cao độ chính xác cũng như giảm tỉ lệ phát hiện sai của giải thuật. Trước hết để giúp cho việc huấn luyện hiệu quả thì tôi không phân chia các mẫu thành 2 loại: anomaly và normal mà phân chia các mẫu thành 5 nhóm chính : normal, u2r, r2l, probe và dos. Sau đó áp dụng giải thuật Bagging[18] để cho ra kết quả là 5 cây quyết định khác nhau và dùng qui tắc bầu cử để chọn ra kết quả cuối cùng. Ý tưởng của giải thuật Bagging có được từ việc kết hợp nhiều phương pháp phân loại khác nhau thì hiếm khi cho ra kết quả tệ hơn việc dùng 1 phương pháp phân loại riêng lẻ. Và trong trường hợp này thì Bagging thực hiện việc

phân chia tập huấn luyện thành 5 tập nhỏ hơn có cùng kích thước bằng cách lựa chọn mẫu ngẫu nhiên. Tương ứng với mỗi tập dữ liệu con thì Bagging xây dựng 1 cây quyết định J48 và kết quả là 5 cây quyết định này khác nhau về cấu trúc. Ta mong muốn sự khác nhau đến từ các cây quyết định này sẽ giúp ta có thể bao phủ nhiều trường hợp trong tập dữ liệu huấn luyện.

- Trong các giải thuật về học máy thì sẽ có thể hiện tính thiên vị và ở đây thì tôi dựa vào nhận xét về tần suất xuất hiện của các mẫu trong tập huấn luyện màưu tiên thứ tự kết quả là : dos, normal, probe, r2l, u2r. Và bằng thực nghiệm đã cho thấy việc lựa chọn xây dựng hệ thống từ 5 cây quyết định là phù hợp với độ chính xác cũng như độ phức tạp khi hiện thực lên board FPGA. Dưới đây là bảng so sánh tính chính xác tương ứng với số lượng cây quyết định:

Số lượng các cây quyết định →	**5**	**10**	**20**	**50**
Khả năng phát hiện tấn công (%)	91.217317	91.1230813	91.0743663	91.2276989
Tỉ lệ phát ra cảnh báo sai (%)	0.5017081	0.5000743	0.5066759	0.4984239

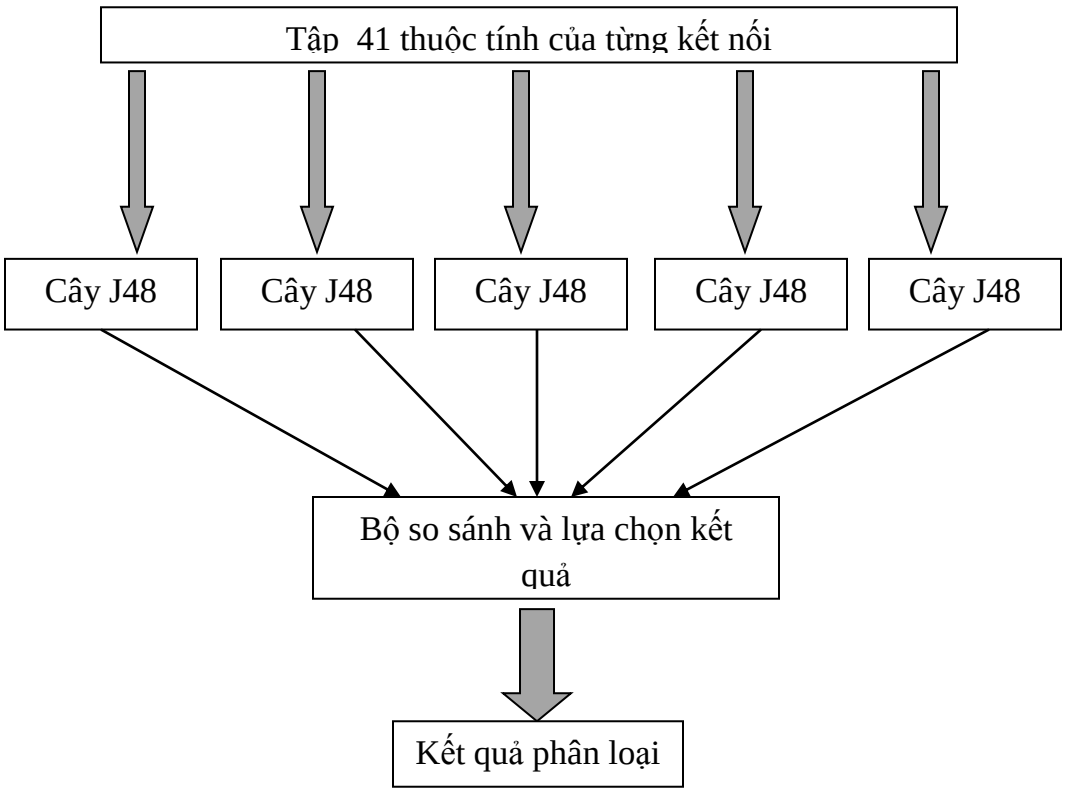
Bảng 8 - So sánh độ chính xác giữa nhóm các cây quyết định

- Các cây quyết định được sinh tra có cấu trúctổng quát như sau:

	Cây J48 thứ 1	**Cây J48 thứ 2**	**Cây J48 thứ 3**	**Cây J48 thứ 4**	**Cây J48 thứ 5**
Số node lá	603	679	536	679	968
Kích thước cây	697	772	629	780	1085

Bảng 9 - Cấu trúc các cây quyết định của mô hình được sinh ra

- Mô hình xây dựng được thể hiện qua hình dưới đây:

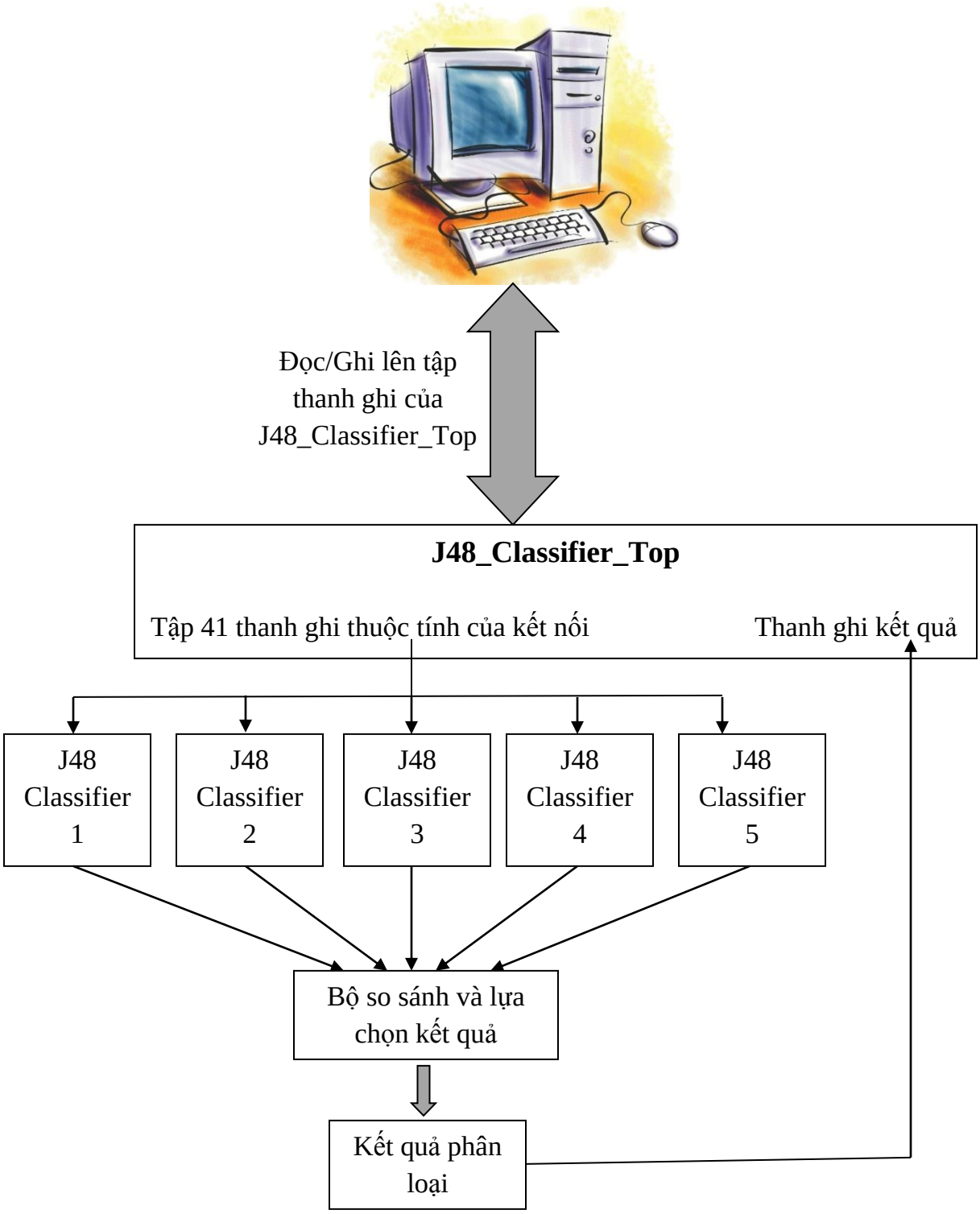


Hình 12 - Mô hình hệ thống phát hiện tấn công

4.5Hiện thực mô hình lên board NetFPGA

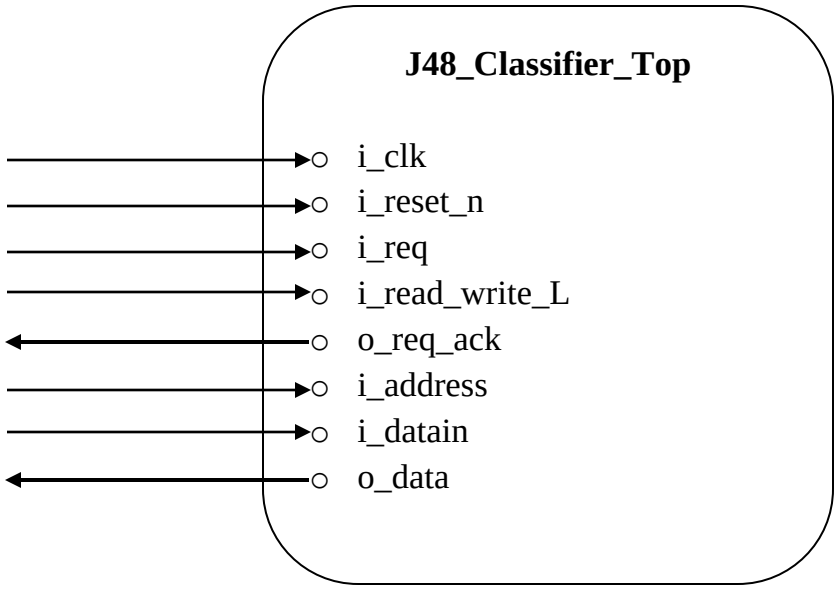
- Do đề tài chỉ tập trung vào xây dựng phần lõi của hệ thống với giả định là tập thuộc tính đã có được từ các kết nối trước đó nên khi hiện thực lên trên board NetFPGA thì tôi không hiện thực phần tương tác trực tiếp với card mạng mà tập 41 thuộc tính này sẽ được truyền từ máy tính host xuống board với mục đính chứng minh tính đúng của mô hình đã được xây dựng.

- Khi hiện thực lên board thì hệ thống được mô tả như hình dưới đây:



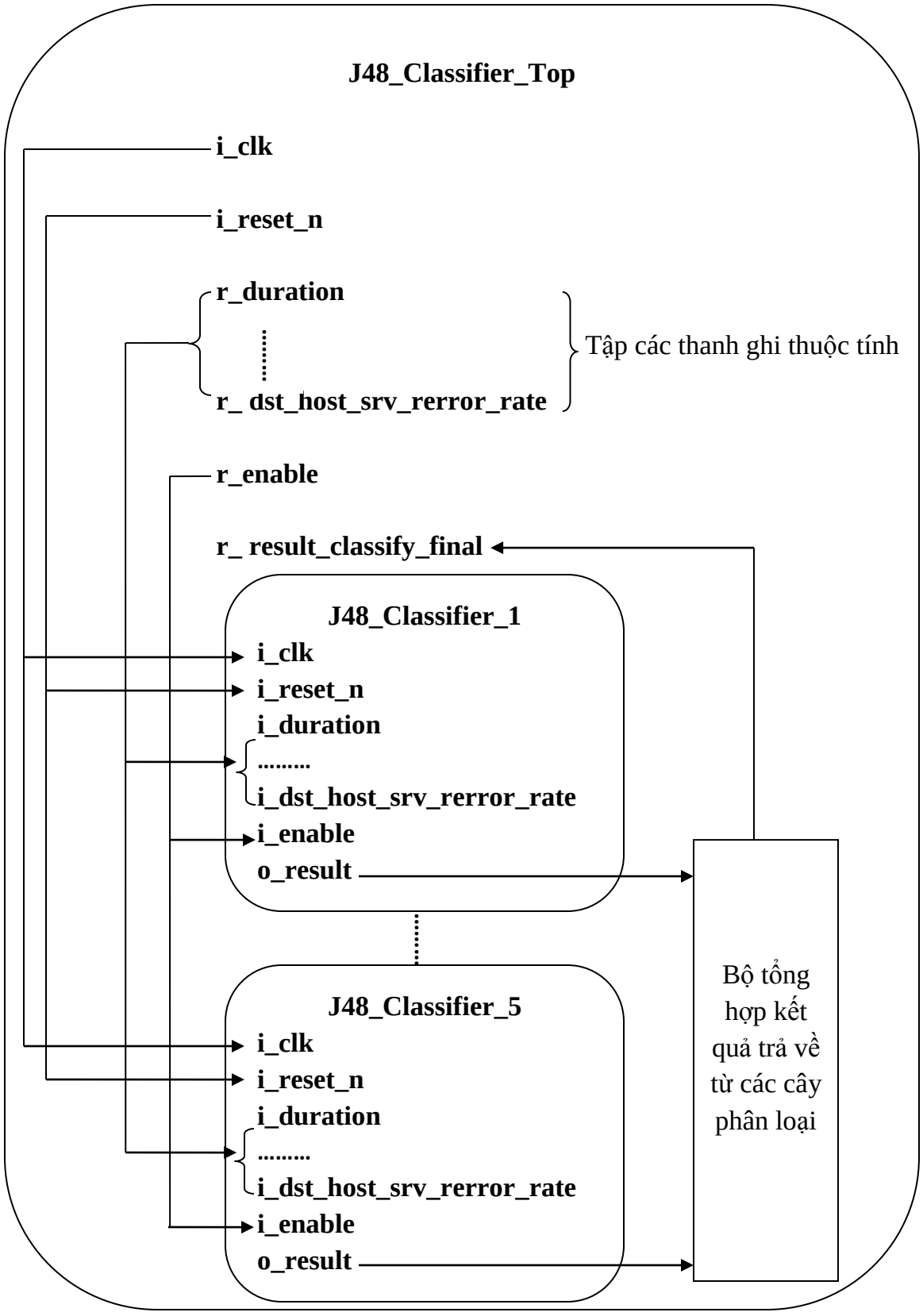
Hình 13 - Hiện thực hệ thống lên board NetFPGA

- Phần giao tiếp của module J48_Classifier_Top chính là những tín hiệu dùng để đọc ghi tập thanh ghi thuộc tính và kết quả trả về:



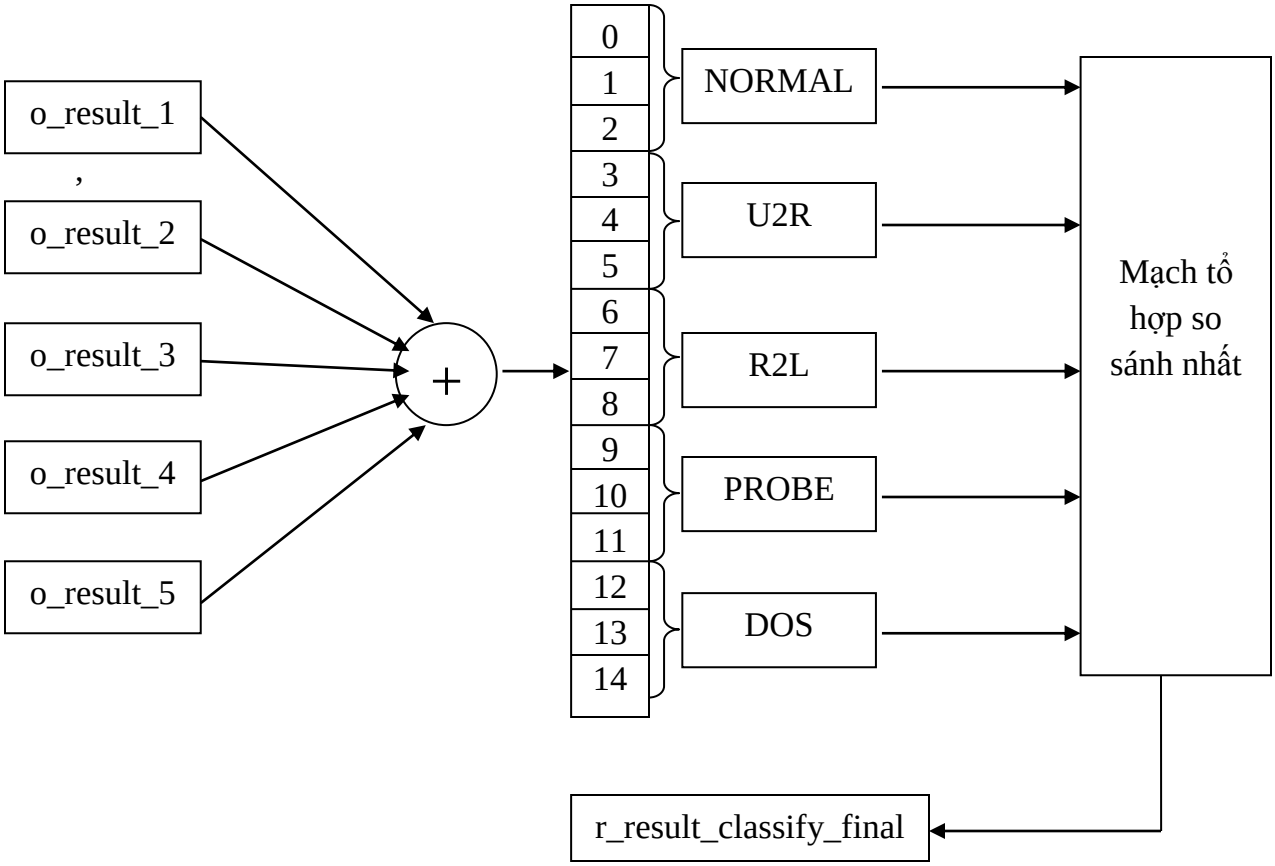
Hình 14 - Phần giao tiếp của module J48_Classifier_Top

-J48_Classifier_Top thực hiện instance 5 cây quyết định và truyền các thanh ghi thuộc tính như ngõ nhập cho các cây này. Dưới đây là mô hình chi tiết:



Hình 15- Cấu trúc chi tiết của module J48_Classifier_Top

- Phần tổng hợp các kết quả trả về từ các cây phân loại và chọn lựa kết quả cuối cùng được hiện thực như sau:



Hình 16 - Bộ tổng hợp kết quả từ các cây phân loại

Kết quả có được từ các cây quyết định là những thanh ghi 15bit, những thanh ghi này sau đó được đi qua một bộ cộng. Thanh ghi tổng chính là thanh ghi chứa tổng số kết quả phân loại giống nhau của từng cây quyết định với vị trí của từng loại như trên hình vẽ. Sau cùng thì những kết quả này được đi qua một mạch tổ hợp để thực hiện việc so sánh nhất với ngõ ra là kết quả phân loại cuối cùng.

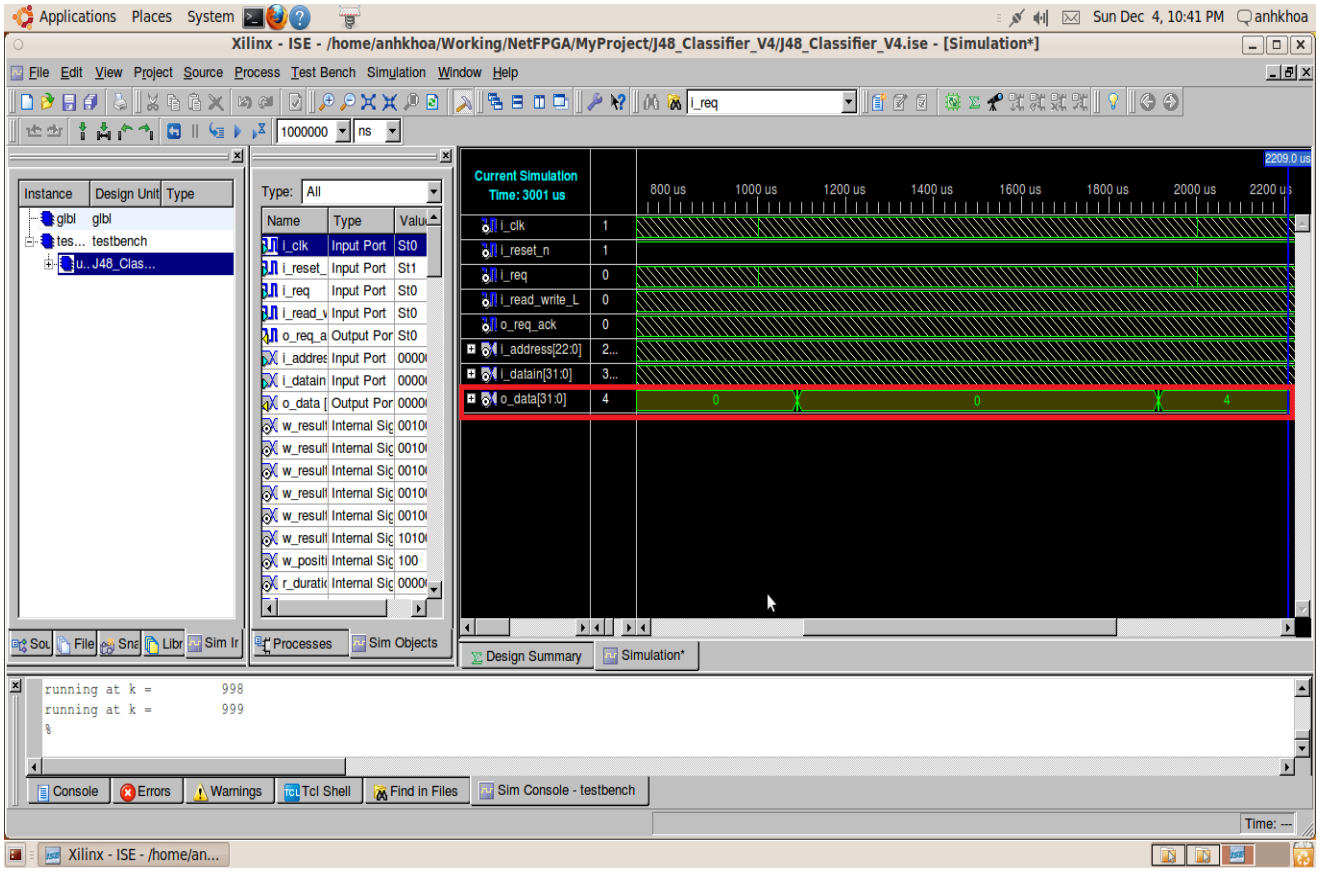
- Với mỗi cây quyết định thì vì là dạng cây nên ta có thể chỉ cần dùng cấu trúc"*if ... else if ... else ...*" là có thể hiện thực được. Mặc dù số node trên mỗi cây là

tương đối lớn và phải dùng đến khoảng 500 dòng code cho từng cây quyết định nhưng ta có thể dễ dàng chuyển code từ Java Model sang Verilog với những điều chỉnh rất ít.Đây cũng là ưu điểm của cây quyết định so với các giải thuật khác khi chọn hiện thực lên FPGA.

4.6 Kết quả thực nghiệm

- Do tuân theo 1 qui trình phát triển rõ ràng cùng với việc dữ liệu kết quả của từng bước được sử dụng để so sánh với nhau và tận dụng lại code của bước trước mà kết quả có được sau khi mô phỏng hệ thống trên môi trường ISE và hiện thực hệ thống lên board NetFPGA đã đạt được độ chính xác như những gì có được trên Java Model. Với tỉ lệ phát hiện tấn công là *91.217317%* và tỉ lệ cảnh báo sai là *0.5017081%*.

- Sau đây là hình ảnh mô phỏng mạch phát hiện tấn công trên môi trường ISE với 1000 mẫu đầu tiên (do giới hạn về bộ nhớ nên tôi chỉ chọn mô phỏng 1000 mẫu đầu tiên để so sánh với kết quả có được trên Java Model)



Hình 17- Kết quả mô phỏng trên ISE với 1000 kết nối đầu tiên

- Tập dữ liệu được sử dụng để kiểm tra khả năng phát hiện của hệ thống chính là tập dữ liệu kiểm tra của cuộc thi KDD 99. Sau đây là bảng so sánh kết quả mà hệ thống tôi xây dựng so với người chiến thắng ở cuộc thi KDD 99:

Phân loại →	**NORMAL**		**U2R**		**R2L**		**PROBE**		**DOS**	
NORMAL	60262	60289	4	2	6	2	243	224	78	76
U2R	168	208	30	5	10	13	20	1	0	1
R2L	14527	14878	8	1	1360	968	294	340	0	2

PROBE	511	622	4	0	6	0	3471	3199	78	345
DOS	5299	6287	0	0	0	0	1328	24	223226	223542

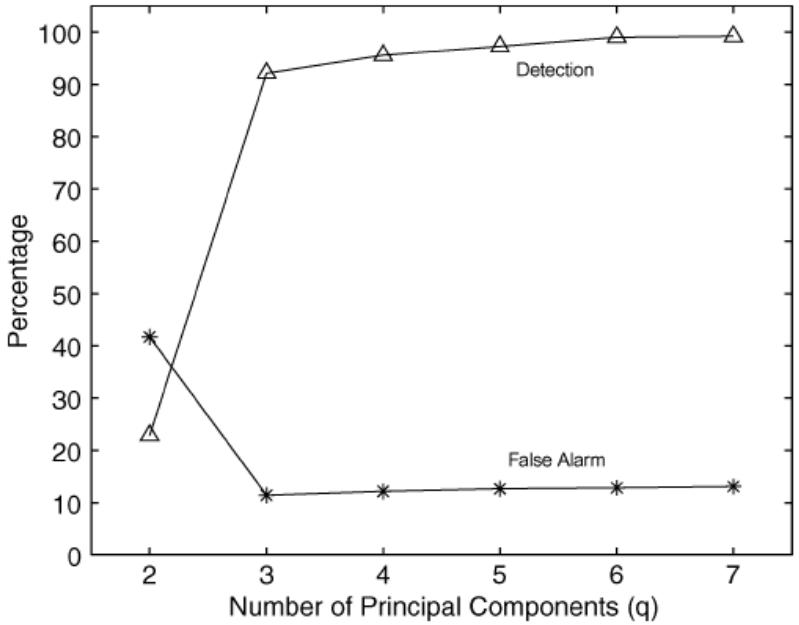
: Kết quả của người chiến thắng cuộc thi KDD 99

quả của hệ thống mà tôi hiện thực

Bảng 10- So sánh kết quả phân loại giữa hệ thống và người chiến thắng KDD99[19]

- Nhìn vào bảng so sánh, ta có thể thấy được khả năng phát hiện tấn công mà hệ thống do tôi xây dựng gần xấp xỉ với người chiến thắng cuộc thi KDD99 (***91.217317%*** *-91.8122794%)* nhưng bù lại thì tỉ lệ phát ra cảnh báo sai mà hệ thống tôi có thì tốt hơn (***0.5017081%*** *- 0.5462677%).*

- Nếu so sánh với kết quả có được từ phương pháp PCA [1] như hình bên dưới:



Hình 18- Kết quả phân loại và tỉ lệ phát ra cảnh báo sai của phương pháp PCA

Thì ta thấy rằng mặc dù với tỉ lệ phát hiện tấn công khoảng 92.2% nhưng tỉ lệ phát ra cảnh báo sai của phương pháp này là quá lớn (trên 10%) nên sẽ không hiệu quả khi áp dụng vào thực tế.

- So với tỉ lệ phát hiện tấn công trung bình khoảng 83.3% của phương pháp kết hợp phân cụm mờ và mạng neural [5], cũng như tỉ lệ phát hiện tấn công khoảng 92.5% nhưng có tỉ lệ phát ra cảnh báo sai khoảng 2-4% của phương pháp SF-KNN[4] thì hệ thống mà tôi hiện thực có được khả năng phát hiện tấn công cao hơn hoặc xấp xỉ cùng với tỉ lệ phát ra cảnh báo sai thấp hơn nhiều.

- Hệ thống có khả năng hoạt động ở tần số tối đa là 103.327MHz trên board NetFPGA của Xilinx.

Phần 5. KẾT LUẬN

5.1 Tổng kết

- Luận văn đã trình bày nhiều phương pháp cũng như có sự đánh giá, so sánh những giải thuật trong phương pháp học máy để lựa chọn và xây dựng phần lõi của hệ thống phát hiện tấn công trong mạng internet dựa vào những dấu hiệu bất thường lên board NetFPGA.

- Cùng với khả năng phát hiện những loại tấn công cao hơn hoặc xấp xỉ với những phương pháp được nghiên cứu trước đây cũng như tỉ lệ phát ra cảnh báo sai thấp của hệ thống thì việc hệ thống đã hoạt động ổn định với tần số 100Mhz trên board NetFPGA đã hoàn thành phần nhiệm vụ được đặt ra cho luận văn.

5.2 Những đóng góp của đề tài

- Trước đây, khi ứng dụng những giải thuật trong Data Mining để phát hiện tấn công trong mạng internet thì chủ yếu hệ thống được hiện thực bằng phần mềm và nếu được hiện thực lên FPGA thì gần như đó là những giải thuật cho việc so trùng chuỗi. Thành công trong việc kết hợp giữa 2 lĩnh vực Data Mining và FPGA, đề tài đã góp phần mở ra 1 hướng đi mới trong việc phát hiện tấn công mạng.

- Đề tài cũng cho thấy việc kết hợp nhiều cây quyết định J48 sẽ có được tỉ lệ phát hiện tấn công trong mạng tốt hơn cũng như khả năng hiện thực dễ dàng lên board NetFPGA nhằm tăng tốc độ xử lí, đáp ứng yêu cầu về thời gian trong thực tế.

5.3 Hướng phát triển

- Do khối lượng công việc nhiều và thời gian hạn chế nên tôi chỉ phát triển phần lõi của hệ thống, nhưng để hệ thống hoàn chỉnh và có thể hoạt động trong thực tế thì sắp tới tôi sẽ xây dựng thêm bộ rút trích các thuộc tính cho các kết nối mạng. Cũng như chúng ta đã thấy thì mặc dù hệ thống phát hiện tấn công dựa vào những bất thường trong mạng có thể phát hiện được những loại tấn công mới nhưng khả năng phát hiện được các loại tấn công chỉ ở khoảng 91.2% nên đây chỉ nên được xem như là 1 phần mở rộng của hệ thống chống xâm nhập thì sẽ hiệu quả hơn.

- Ngoài việc có thể tăng tần số hoạt động của mạch bằng cách phân chia lại kiến trúc hệ thống thành dạng pipeline thì chúng ta cũng có thể dựa vào những phương pháp Entropy hay xác suất Bayes để rút ngắn lại danh sách những thuộc tính này để giảm độ phức tạp của mạch.

Phần 6. TÀI LIỆU THAM KHẢO

[1] Abhishek Das, David Nguyen, Joseph Zambreno, Gokhan Memik, and Alok Choudhary, "An FPGA-based network intrusion detection architecture", *IEEE Transactions on information forensics and security, Vol.3, No. 1*, March 2008

[2] Faisal M. Cheema, Adeel Akram, Zeshan Iqbal, "Comparative Evaluation of Header vs. Payload based Network Anomaly Detectors", *Proceedings of the World Congress on Engineering 2009 Vol I*, July 1 - 3, 2009, London, U.K.

[3] Prasanta Gogoi, B Borah and D K Bhattacharyya, "Anomaly Detection Analysis of Intrusion Data using Supervised & Unsupervised Approach", *Journal of Convergence Information Technology Volume 5, Number 1*, February 2010

[4] Anazida Zainal, Mohd Aizaini Maarof and Siti Mariyam Shamsuddin, "Ensemble Classifiers for Network Intrusion Detection System", *Journal of Information Assurance and Security 4*, pp.217-225, 2009

[5] Hossein M. Shirazi, Malek-Ashtar, "Anomaly Intrusion Detection System Using Information Theory, K-NN and KMC Algorithms", *Australian Journal of Basic and Applied Sciences* , 3(3): pp.2581-2597, 2009

[6] Gang Wang, Jinxing Hao, Jian Ma, Lihua Huang, "A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering", Exper Systems wit Applications, Vol.37, pp.6225-6232, 2010

[7] Mrutyunjaya Panda and Manas Ranjan Patra, "A Novel Classification via Clustering Method for Anomaly Based Network Intrusion Detection System*",* *International Journal of Recent Trends in Engineering*, Vol 2, No. 1, November 2009

[8] Vivek A. Patole, V. K. Pachghare, Parag Kulkarni, "Self Organizing Maps to Build Intrusion Detection System", *International Journal of Computer Applications*, Volume 1 – No. 8, 2010

[9] Hua Jiang, Junhu Ruan, "The Application of Genetic Neural Network in Network Intrusion Detection", *Journal of computer*, Vol.4,No. 12, December 2009

[10] P.Garcia-Teodoro, J.Diaz-Verdejo, G.Macia-Fernandez, E.Vazquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges", *Computer & Security*, Vol.28, pp.18-28, 2009

[11] Field-Programmable Gate Array, http://en.wikipedia.org/wiki/Field-programmable_gate_array

[12] DARPA, http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/index.html

[13] Christophe Bodda, "Introduction to Reconfigurable Computing", Springer, 2007

[14] WEKA, http://www.cs.waikato.ac.nz/ml/weka/

[15] KDD99 , http://kdd.ics.uci.edu/databases/kddcup99/task.html

[16] NetFPGA Forum,http://netfpga.org/

[17] S.B.Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques", *Informatica,* Vol.31**,** pp.249-268, 2007.

[18] Ian H.Witten, Eibe Frank, Mark A.Hall, "Data Mining: Practical Machine Learning Tools and Techniques", Third Edition, Elsevier, pp.352-356, 2011.

[19] Charles Elkan, "Results of the KDD'99 Classifier Learning", SIGKDD Explorations, Volume 1, Issue 2, pp.63–64, 2000.