

BÁO CÁO THỰC HÀNH

Môn học: Hệ thống tìm kiếm, phát hiện và ngăn ngừa xâm nhập

Tên chủ đề: Học máy trong IDS

GVHD: Trương Thị Hoàng Hảo

Nhóm: 07

1. THÔNG TIN CHUNG:

Lớp: NT204.P21.ANTT.2

STT	Họ và tên	MSSV	Email
1	Nguyễn Khánh Linh	22520769	22520769@gm.uit.edu.vn
2	Nguyễn Phúc Nhi	22521041	22521041@gm.uit.edu.vn
3	Phạm Thị Cẩm Tiên	22521473	22521473@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

STT	Nội dung	Tình trạng	Trang
1	Yêu cầu 1.1	100%	2
2	Yêu cầu 2.1	100%	2 – 7
3	Yêu cầu 2.2	100%	7 – 12
4	Yêu cầu 3.1	100%	12 – 21

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

¹ Ghi nội dung công việc, các kịch bản trong bài Thực hành

BÁO CÁO CHI TIẾT

Yêu cầu 1.1 Sinh viên tìm hiểu về tập dữ liệu KDD Cup 1999 và điền các kết quả tìm hiểu được vào form bên dưới.

1. Số nhóm tấn công: 4

Kể tên các nhóm tấn công: DoS, R2L, U2R, Probe

2. Số kiểu tấn công: 22

Kể tên các kiểu tấn công được gán nhãn:

- DoS: back, land, neptune, pod, smufl, teardrop,
- R2L: ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster,
- U2R: buffer_overflow, loadmodule, perl, rootkit
- Probe: ipsweep, nmap, portsweep, satan

3. Mỗi instance trong tập dữ liệu KDD Cup 1999 bao gồm 41 thuộc tính, cụ thể gồm các thuộc tính: duration, protocol_type, service, flag, src_bytes, dst_bytes, land, wrong_fragment, urgent, hot, num_failed_logins, logged_in, num_compromised, root_shell, su_attempted, num_root, num_file_creations, num_shells, num_access_files, num_outbound_cmds, is_host_login, is_guest_login, count, srv_count, serror_rate, srv_serror_rate, rerror_rate, srv_rerror_rate, same_srv_rate, diff_srv_rate, srv_diff_host_rate, dst_host_count, dst_host_srv_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst_host_serror_rate, dst_host_srv_serror_rate, dst_host_rerror_rate, dst_host_srv_rerror_rate.

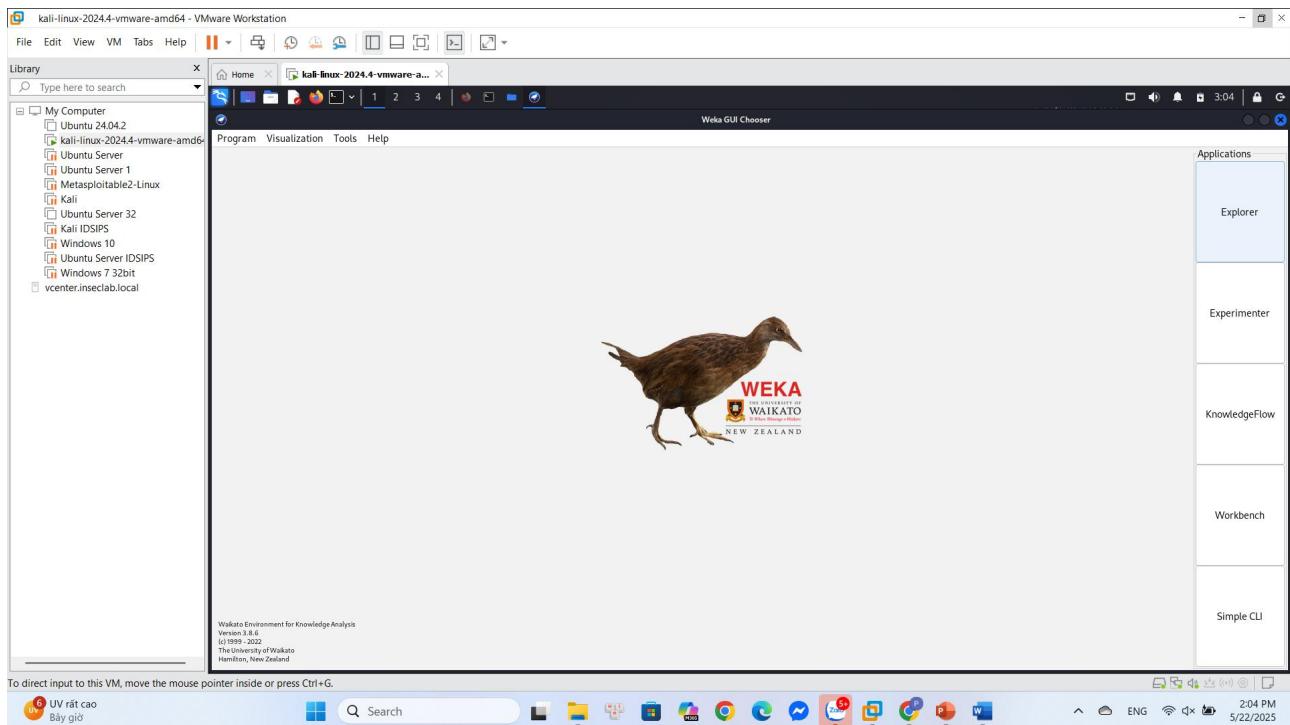
Yêu cầu 2.1 Sinh viên cài đặt WEKA, tìm hiểu và load một tập dữ liệu có định dạng .arff đơn giản có sẵn của WEKA

Bước 1: Cài đặt WEKA

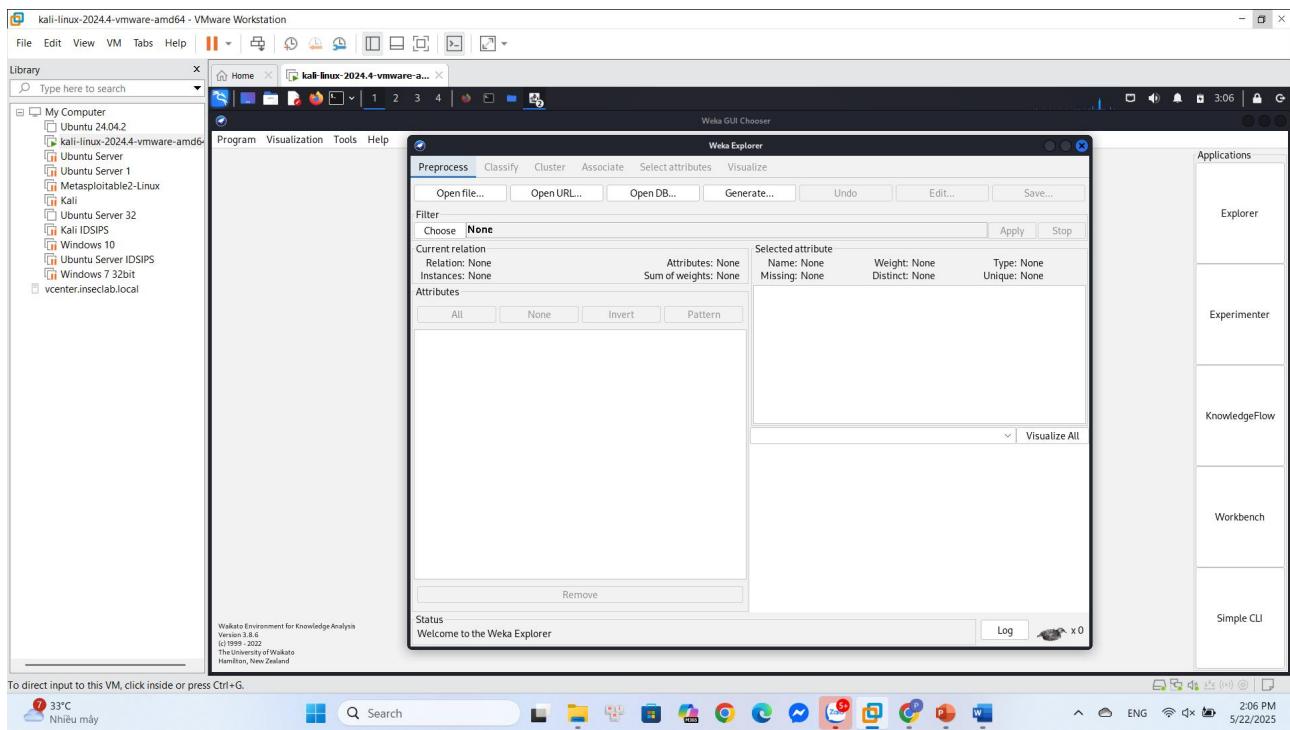
Tải công cụ WEKA tại đường dẫn <https://waikato.github.io/weka-wiki/> và tham khảo các hướng dẫn cài đặt trên hệ điều hành Kali Linux.

Bước 2: Chạy WEKA

Trên terminal trong folder vừa unzip từ file .zip vừa tải ở bước trên, nhập lệnh `./weka.sh` để chạy giao diện WEKA.

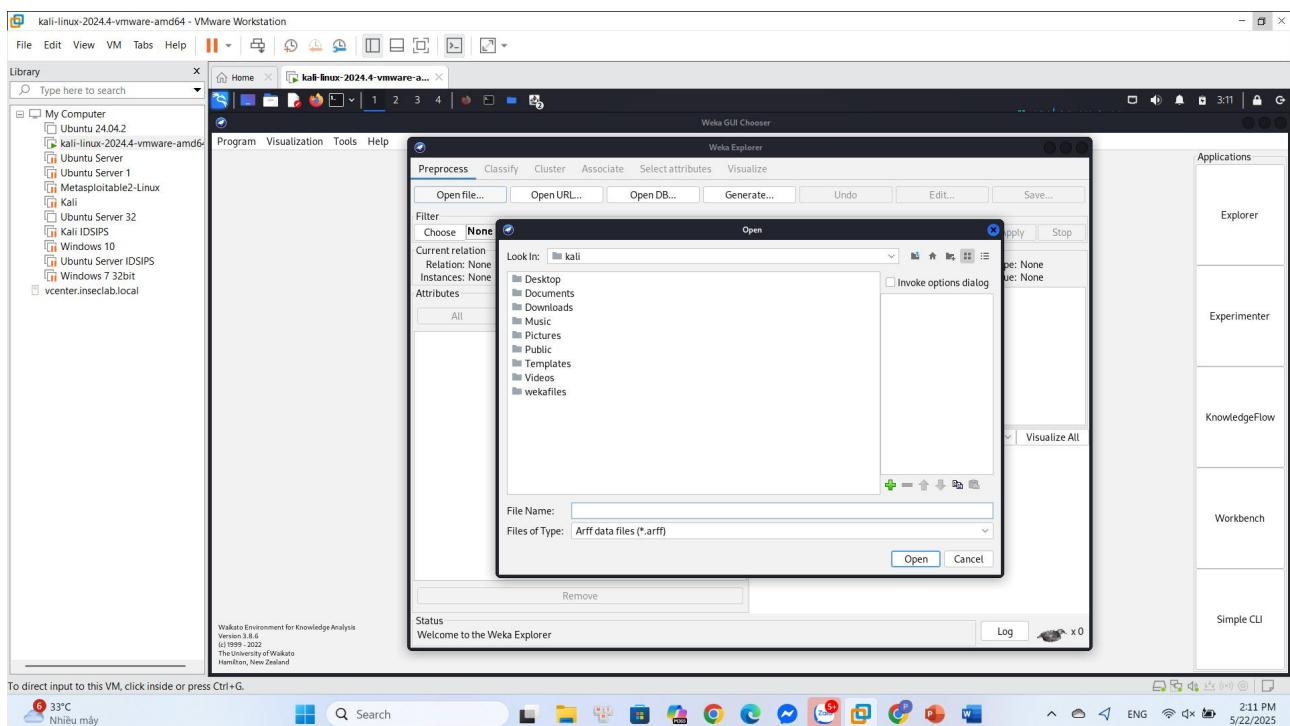


Bước 3: Trong cửa sổ GUI Chooser, chọn Explorer

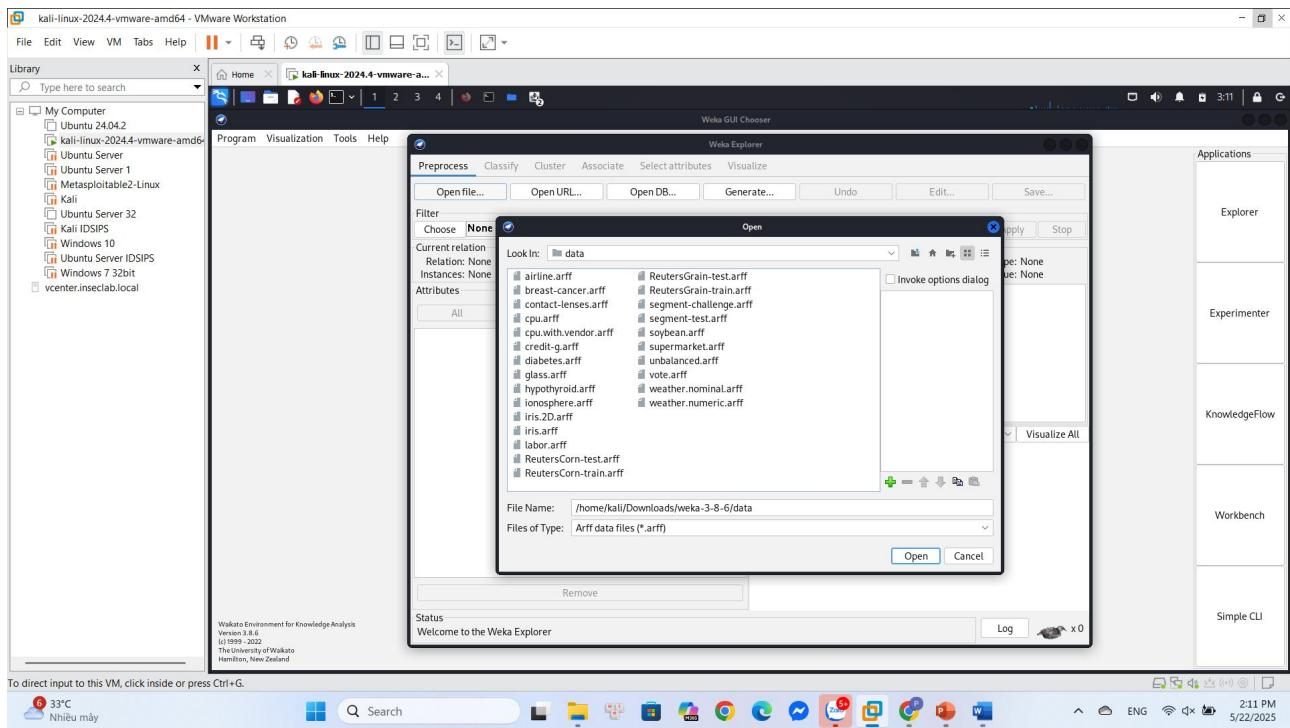


Bước 4: Load dữ liệu vào WEKA

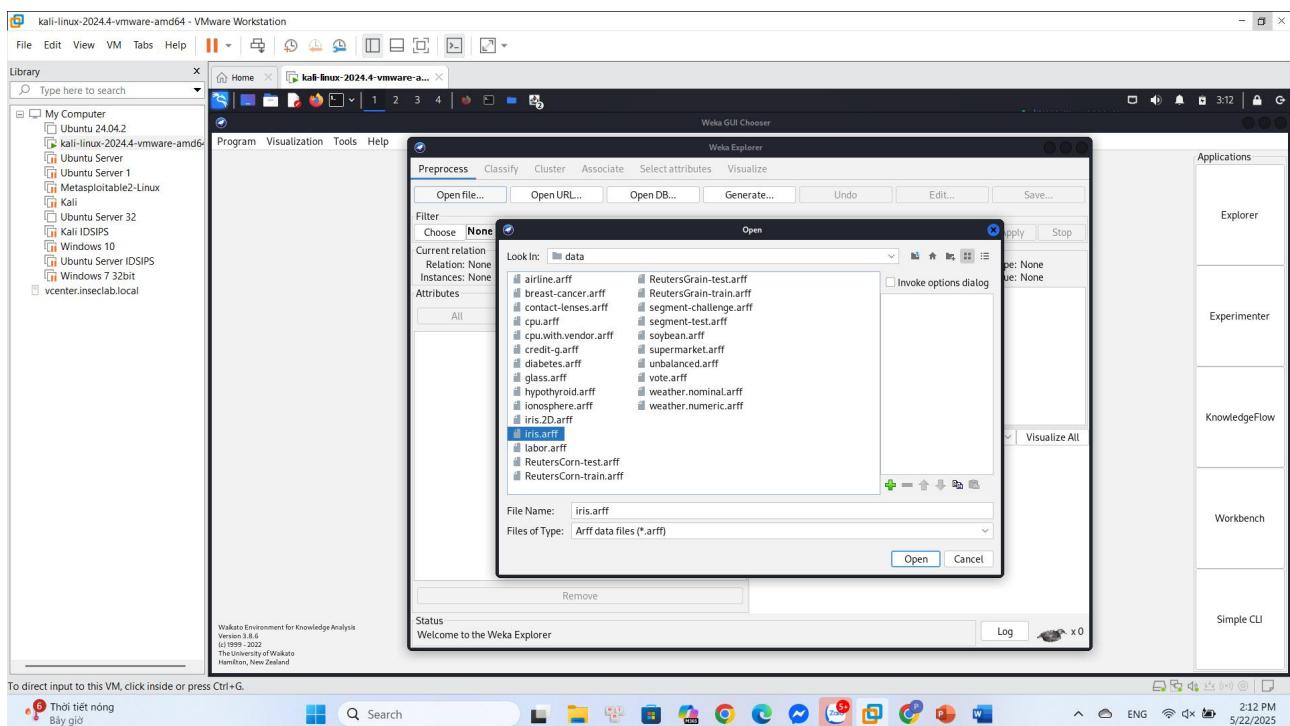
Ở tab Preprocess trong cửa sổ WEKA Explorer, click chọn Open files... để chọn file dữ liệu cần đưa vào.



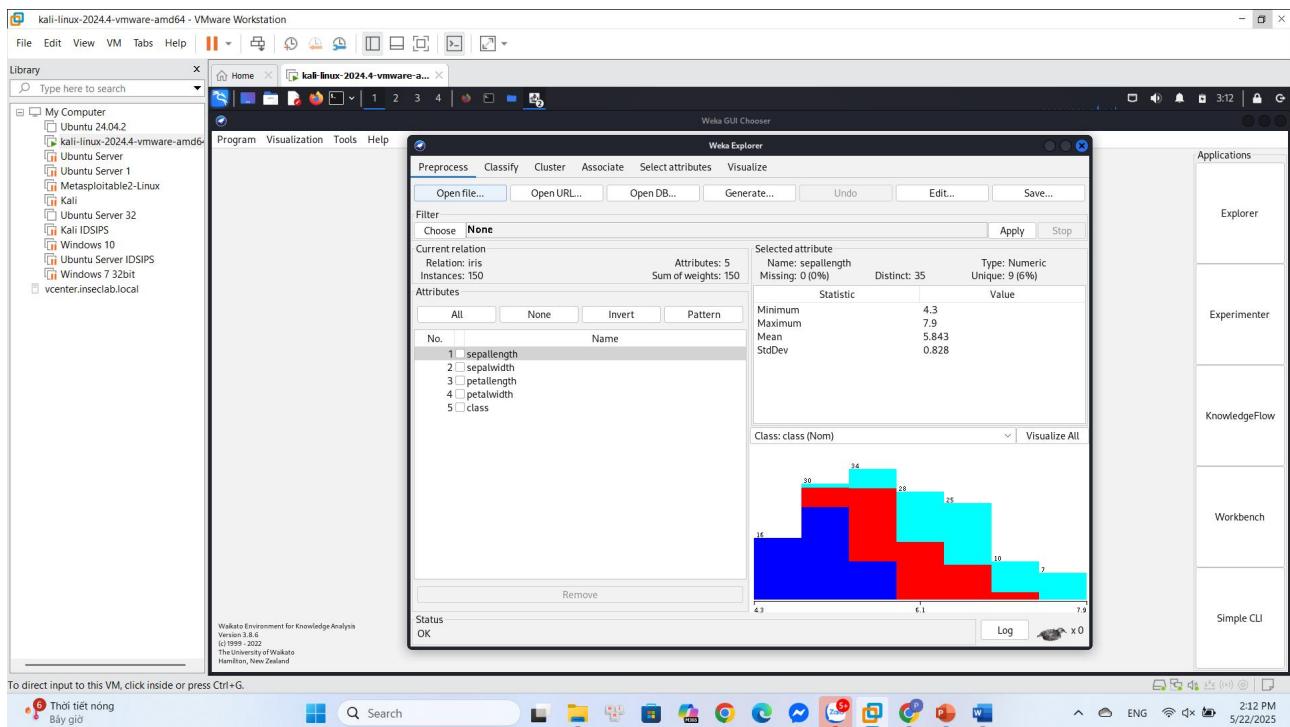
WEKA khi cài đặt có cung cấp sẵn một số tập dữ liệu đơn giản ở thư mục data/ trong thư mục cài đặt của WEKA. Trong thư mục này bao gồm nhiều file có định dạng .arff, là các dataset có thể đưa vào sử dụng ngay trong WEKA.



Ta sử dụng file iris.arff, là một dataset chứa dữ liệu thu thập được về đặc điểm của một số loại hoa diên vĩ (iris) khác nhau như độ dài và độ rộng của cánh hoa và thuộc tính phân lớp là loại hoa diên vĩ nào.



Bước 5: Quan sát và giải thích các giá trị trong tab Preprocess.



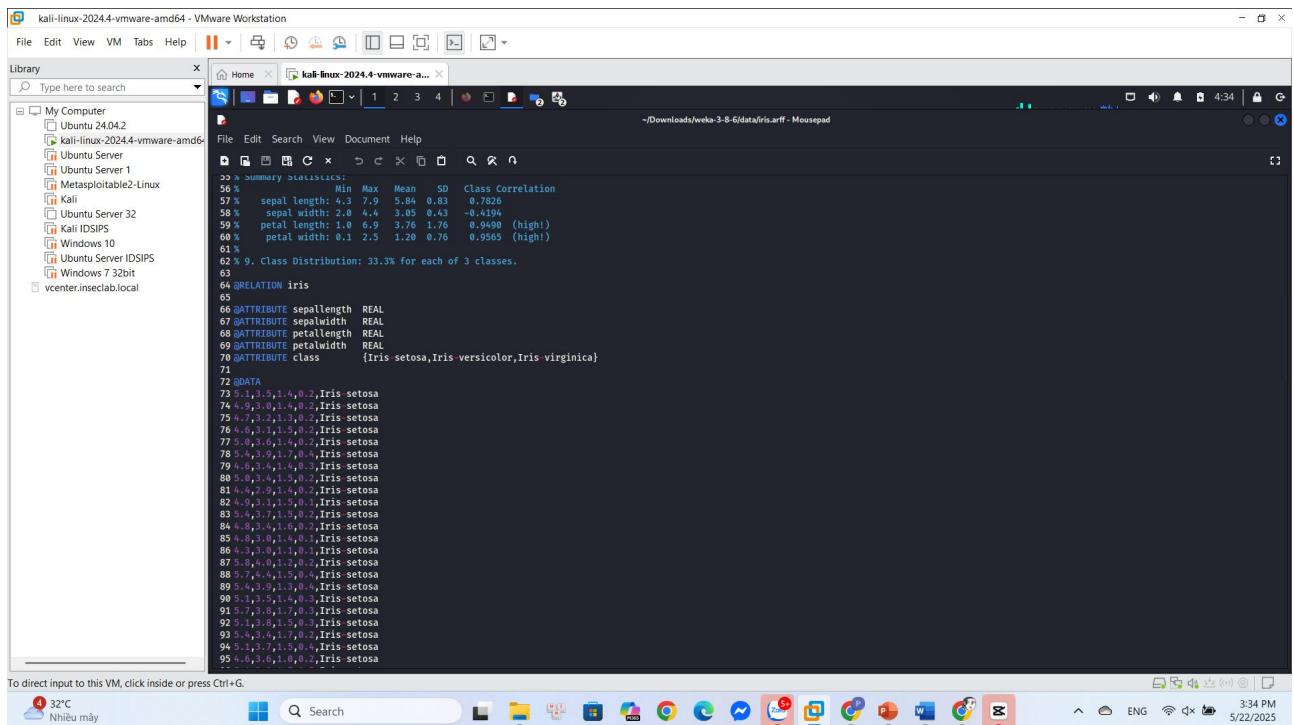
Giải thích các giá trị:

- Current relation – Thông tin của tập dữ liệu hiện tại:
 - + Relation: iris – Tên tập dữ liệu đang được tải.
 - + Instances: 150 – Số lượng mẫu hay dòng dữ liệu trong tập dữ liệu.
 - + Attributes: 5 – Số thuộc tính hay cột dữ liệu, bao gồm cả thuộc tính phân lớp.
- Attributes – Danh sách các thuộc tính:
 - + sepallength – độ dài đài hoa

- + sepalwidth – độ rộng đài hoa
- + petallength – độ dài cánh hoa
- + petalwidth – độ rộng cánh hoa
- + class – phân loại hoa diên vĩ (có thể là Iris-setosa, Iris-versicolor, Iris-virginica)
- Selected attribute – Danh sách các thông số và giá trị của thuộc tính được chọn:
 - + Name: sepallength – Tên của thuộc tính đang chọn.
 - + Distinct: 35 – số lượng giá trị khác nhau duy nhất trong một thuộc tính.
 - + Missing: 0 – Số dữ liệu bị thiếu.
 - + Unique: 9 (6%) – Số lượng giá trị không lặp lại và phần trăm của nó so với tổng số lượng mẫu, chỉ có nếu thuộc tính kiểu Numeric hoặc String.
 - + Type: Numeric – Kiểu dữ liệu số.
 - Numeric – số: Kiểu dữ liệu của các thuộc tính như sepallength.
 - Nominal – tên: Kiểu dữ liệu của các thuộc tính như class.
 - String – chuỗi ký tự
 - Date – ngày tháng
- Statistic và Value của thuộc tính kiểu Numeric (ở đây là sepallength):
 - + Minimum: 4.3 – Giá trị nhỏ nhất.
 - + Maximum: 7.9 – Giá trị lớn nhất.
 - + Mean: 5.843 – Giá trị trung bình.
 - + StdDev: 0.828 – Độ lệch chuẩn.
- Label, Count và Weight của thuộc tính kiểu Nominal hay String (thường là class):
 - + Label – Giá trị cụ thể của thuộc tính.
 - + Count – Số lượng mẫu của giá trị đó.
 - + Weight – Trọng số của các mẫu đó (thường là 1).
- Biểu đồ trong hình trên thể hiện phân bố của giá trị thuộc tính được chọn tương ứng với từng lớp, trong hình đang thể hiện phân bố của giá trị thuộc tính sepallength. Mỗi màu thể hiện một lớp phân loại, trong hình mỗi màu đang thể hiện một loại hoa. Dựa vào biểu đồ, ta có thể thấy sự phân bố giá trị của thuộc tính đang chọn giữa các loại hoa.
- Status: Hiển thị tình trạng của file. Chủ yếu có 2 trạng thái là đang xử lý (load) hoặc OK.

Định nghĩa và cấu trúc của file .arff:

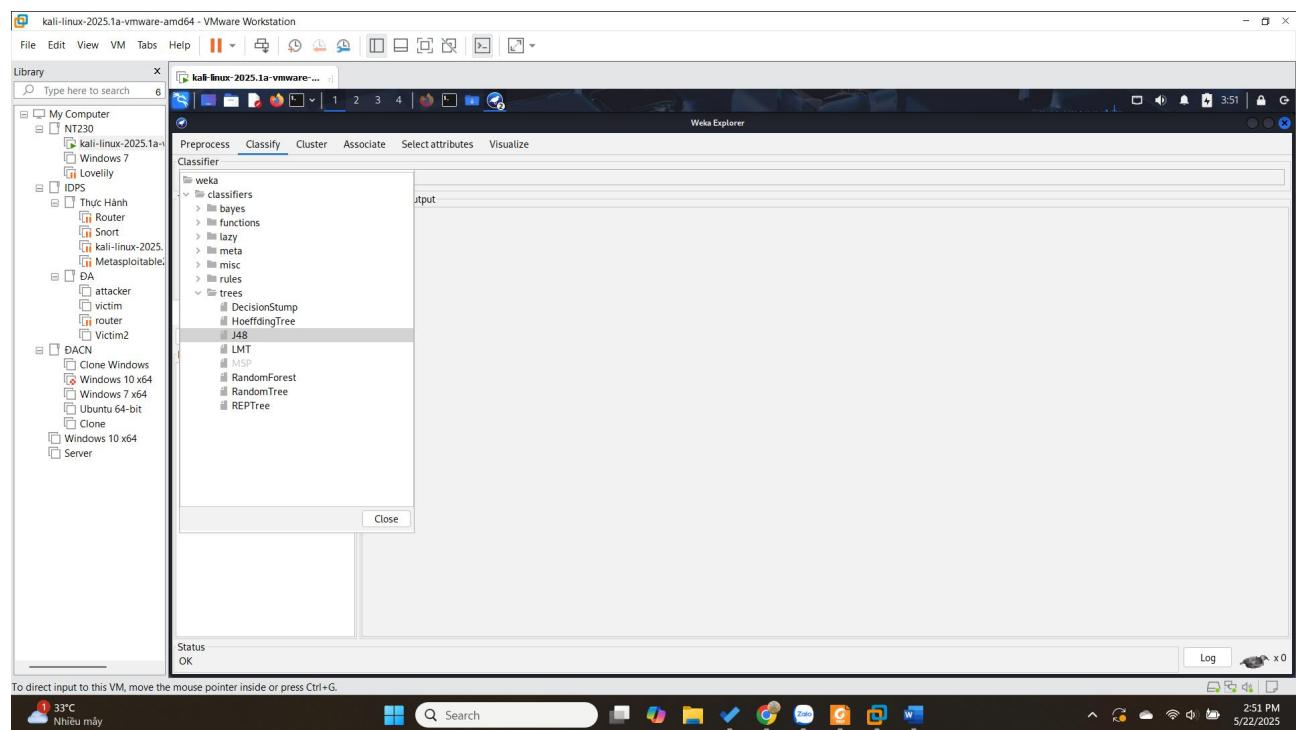
- Định nghĩa: File .arff hay Attribute-Relation File Format là định dạng tập tin được sử dụng chủ yếu bởi phần mềm WEKA để mô tả dữ liệu dùng cho các thuật toán học máy. Cấu trúc của file .arff gồm hai phần chính: phần khai báo header và phần dữ liệu data.
- Header: Phần này định nghĩa:
 - + @RELATION: Tên tập dữ liệu
 - + @ATTRIBUTE: Định nghĩa thuộc tính (mỗi thuộc tính có tên và kiểu dữ liệu), xác định thuộc tính mục tiêu (nếu có).
- Data: Bắt đầu bằng từ khóa @DATA và liệt kê từng instance, mỗi dòng tương ứng một data point.
- Ngoài ra, trong file .arff, dấu % được dùng để viết comment.



Yêu cầu 2.2 Sinh viên lựa chọn 01 bộ phân lớp (classifier) bất kỳ và thực hiện khai thác trên tập dữ liệu đã chọn ở trên. Trình bày và giải thích kết quả.

Bước 1: Lựa chọn bộ phân lớp cho tập dữ liệu

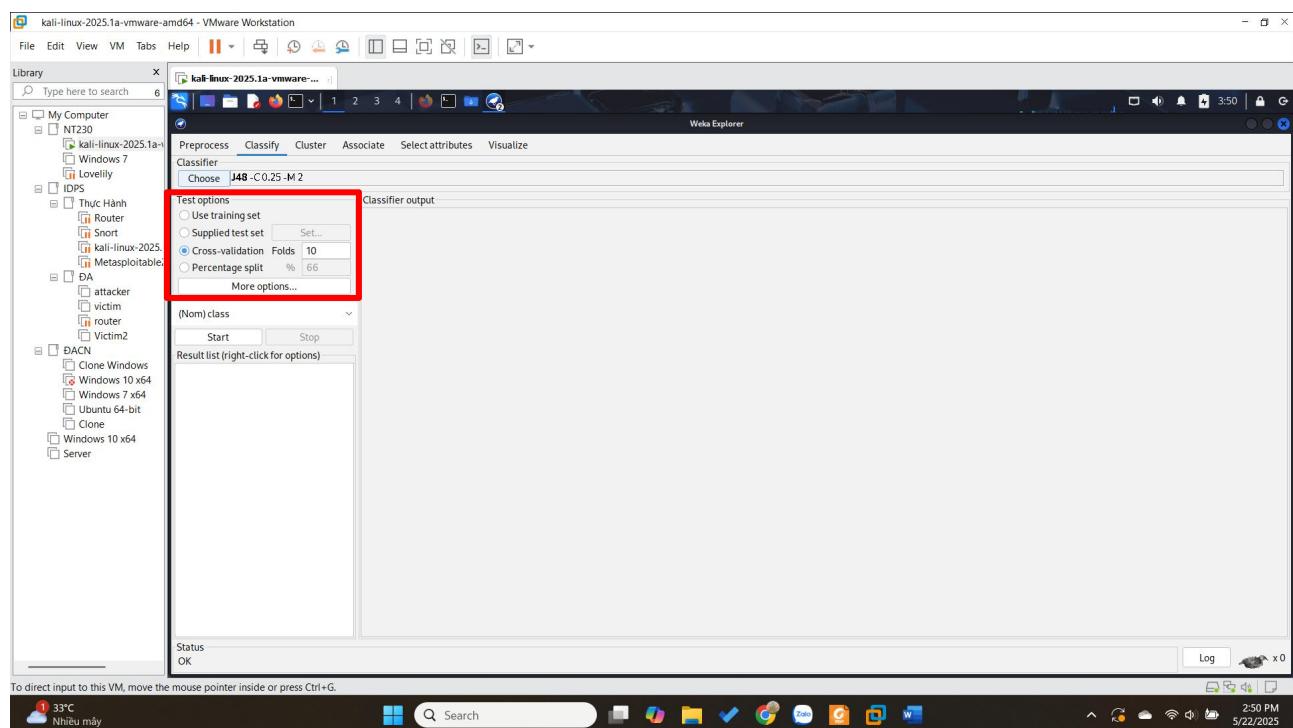
Trong cửa sổ WEKA Explorer, chọn tab **Classify** → **Choose** → J48





Bước 2: Lựa chọn Test options

Đây là các tùy chọn được WEKA hỗ trợ để định nghĩa 2 tập dữ liệu huấn luyện và kiểm tra. Ở đây, nhóm chọn option **Cross-validation (10-fold)**

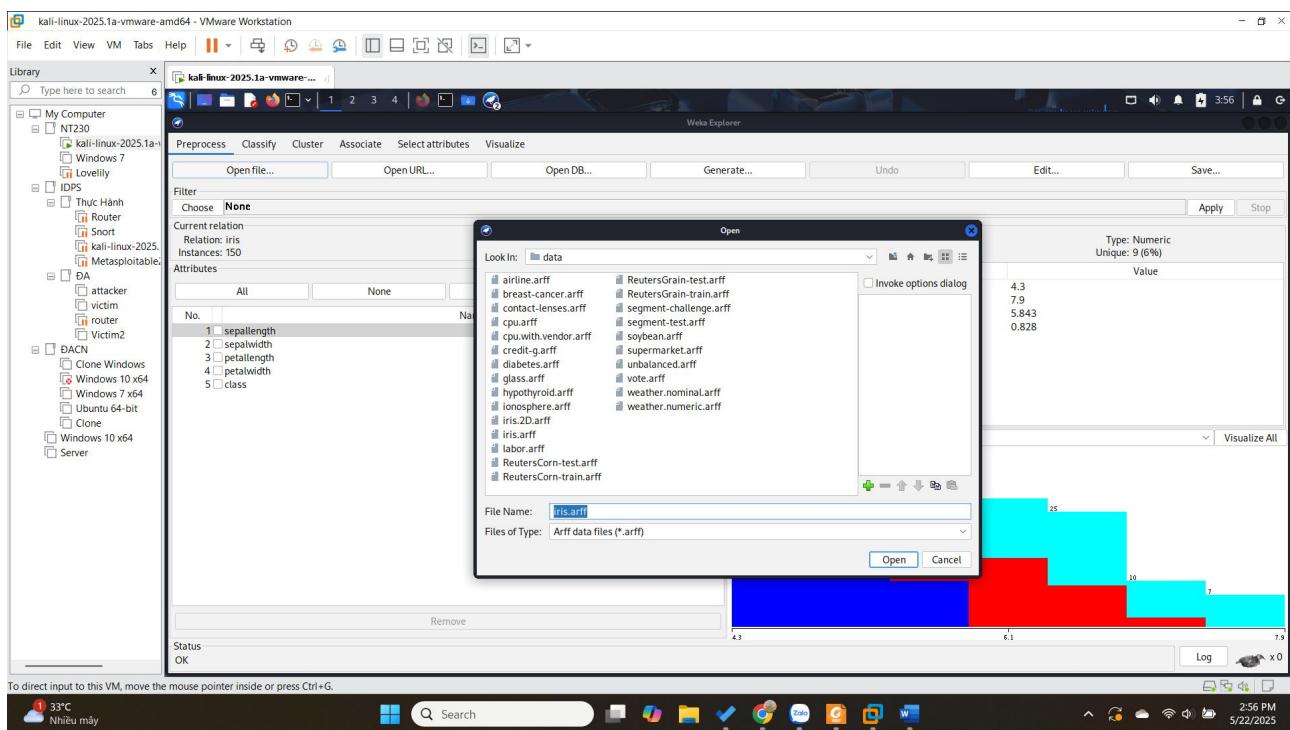


Giải thích Test Option:

- **Cross-validation** là tùy chọn sẽ chia dữ liệu thành k phần (ở đây k = 10). Mỗi lần lấy 1 phần làm test, phần còn lại làm train. Kết quả cuối cùng là trung bình của 10 lần chạy.
- Sử dụng tùy chọn này sẽ đảm bảo đánh giá tổng thể, và giúp giảm độ thiên lệch do ngẫu nhiên.

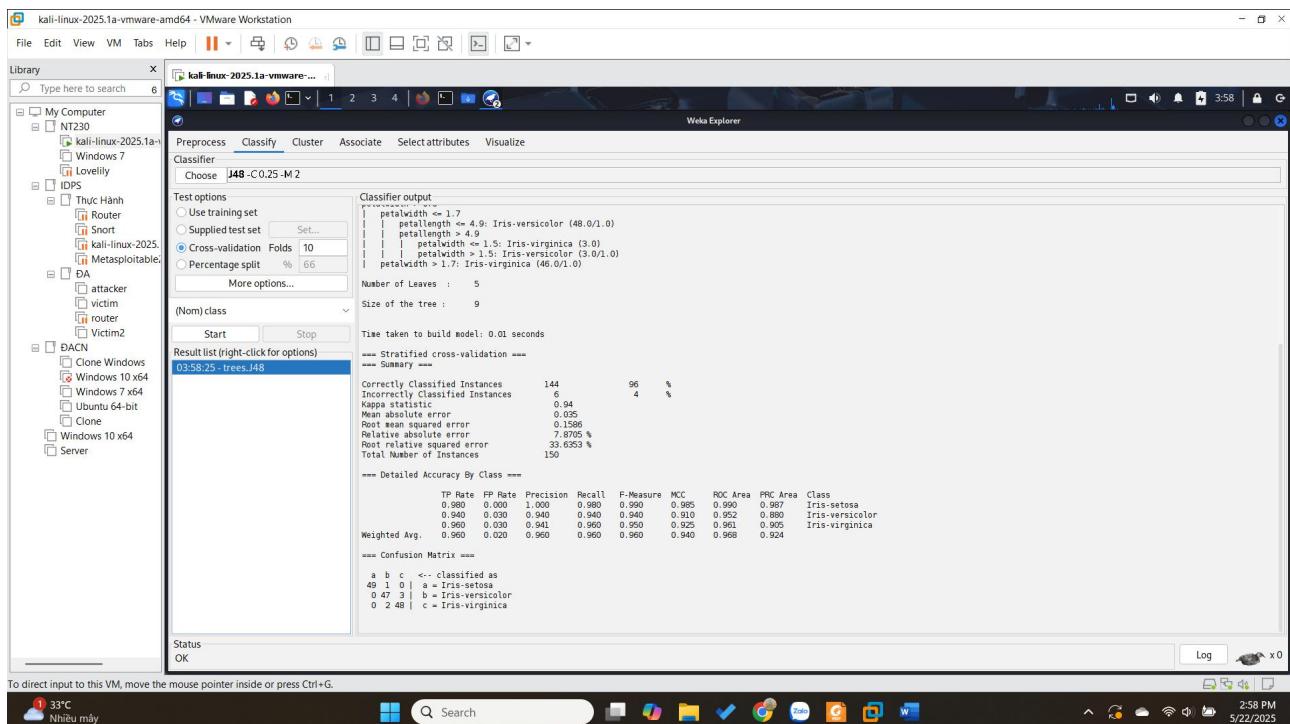
Bước 3: Lựa chọn tập dữ liệu

Ở đây, nhóm lựa chọn tập dữ liệu **iris.arff**



Bước 4: Chạy bộ phân lớp và quan sát kết quả

Nhấn chọn Start để bắt đầu chạy bộ phân lớp.



Giải thích kết quả:

```
==== Run information ===

Scheme:      weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:    iris
Instances:   150
Attributes:  5
              sepallength
              sepalwidth
              petallength
              petalwidth
              class
Test mode:   10-fold cross-validation
```

- Đây là nơi chứa thông tin mô hình:
 - Scheme: J48** là bộ phân lớp dựa trên thuật toán decision tree tương đương với thuật toán C4.5 được sử dụng rộng rãi trong học máy. Nó tạo ra các cây quyết định dựa trên các đặc trưng của dữ liệu để phân loại các mẫu. Trong trường hợp này, ta sử dụng bộ phân lớp J48 với các tham số -C 0.25 (confidence factor dùng để cắt tía cây, giá trị nhỏ → cây đơn giản hơn), -M 2 (số lượng mẫu tối thiểu để chia node).
 - Bộ phân lớp J48 được áp dụng cho dataset iris với 150 thực thể và 5 thuộc tính, test option là cross-validation 10-fold.

```
==== Classifier model (full training set) ===

J48 pruned tree
-----
petalwidth <= 0.6: Iris-setosa (50.0)
petalwidth > 0.6
|   petalwidth <= 1.7
|   |   petallength <= 4.9: Iris-versicolor (48.0/1.0)
|   |   petallength > 4.9
|   |   |   petalwidth <= 1.5: Iris-virginica (3.0)
|   |   |   petalwidth > 1.5: Iris-versicolor (3.0/1.0)
|   petalwidth > 1.7: Iris-virginica (46.0/1.0)

Number of Leaves :      5
Size of the tree :     9

Time taken to build model: 0.01 seconds
```

- Đây là decision trên đã cắt tía
 - Cây phân loại dựa trên petalwidth và pentallength – hai thuộc tính quan trọng nhất trong phân biệt các loại.
- Cấu trúc cây:
 - Số lá: 5
 - Kích thước cây: 9
 - Thời gian huấn luyện: 0.01 giây

Correctly Classified Instances	144	96	%
Incorrectly Classified Instances	6	4	%
Kappa statistic	0.94		
Mean absolute error	0.035		
Root mean squared error	0.1586		
Relative absolute error	7.8705 %		
Root relative squared error	33.6353 %		
Total Number of Instances	150		

- Các thông số:

- o **Correctly Classified Instances:** Tỉ lệ và số lượng instance được phân lớp đúng: 96% (144/150 mẫu)
- o **Incorrectly Classified Instances:** Tỉ lệ và số lượng instance bị phân lớp sai: 4% (4/150 mẫu)
- o **Kappa statistic:** Đo độ chính xác so với đoán ngẫu nhiên. Giá trị từ -1 đến 1: 0.94 → tốt
- o **Mean absolute error:** Sai số tuyệt đối trung bình (sai số giữa dự đoán và thực tế): 0.035
- o **Root mean squared error:** Sai số bình phương trung bình: 0.1586
- o **Relative absolute error:** Sai số tuyệt đối tương đối: 7.8705%
- o **Root relative squared error:** Sai số bình phương tương đối: 33.6353%
- o **Total Number of Instances:** Tổng số thực thể là 150

==== Detailed Accuracy By Class ====

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.980	0.000	1.000	0.980	0.990	0.985	0.990	0.987	Iris-setosa
	0.940	0.030	0.940	0.940	0.940	0.910	0.952	0.880	Iris-versicolor
	0.960	0.030	0.941	0.960	0.950	0.925	0.961	0.905	Iris-virginica
Weighted Avg.	0.960	0.020	0.960	0.960	0.960	0.940	0.968	0.924	

- Bảng phân tích độ chính xác theo lớp:

- o **TP Rate:** Tỷ lệ mẫu thuộc lớp đó và được mô hình gán đúng. (True Positive)
- o **FP Rate:** Tỷ lệ mẫu không thuộc lớp đó nhưng bị mô hình gán nhầm thành của lớp đó (False Positive)
- o **Precision:** Tỷ lệ mẫu được dự đoán đúng trong tất cả các mẫu được gán nhãn.
- o **Recall:** Tỷ lệ mẫu thuộc lớp đó mà mô hình đã dự đoán đúng.
- o **F-Measure:** Trung bình giữa Precision và Recall.
- o **MCC:** Chỉ số đánh giá độ chính xác tổng quát giữa dự đoán và thực tế.
- o **ROC Area:** Diện tích dưới đường cong ROC – đánh giá khả năng phân biệt giữa các lớp.
- o **PRC Area:** Diện tích dưới đường cong Precision-Recall – nhạy hơn ROC nếu dữ liệu mất cân bằng.
- o **Class:** Các lớp phân loại: Iris-setosa, Iris-versicolor, Iris-virginica.
- o **Weighted Avg.:** trung bình có trọng số.

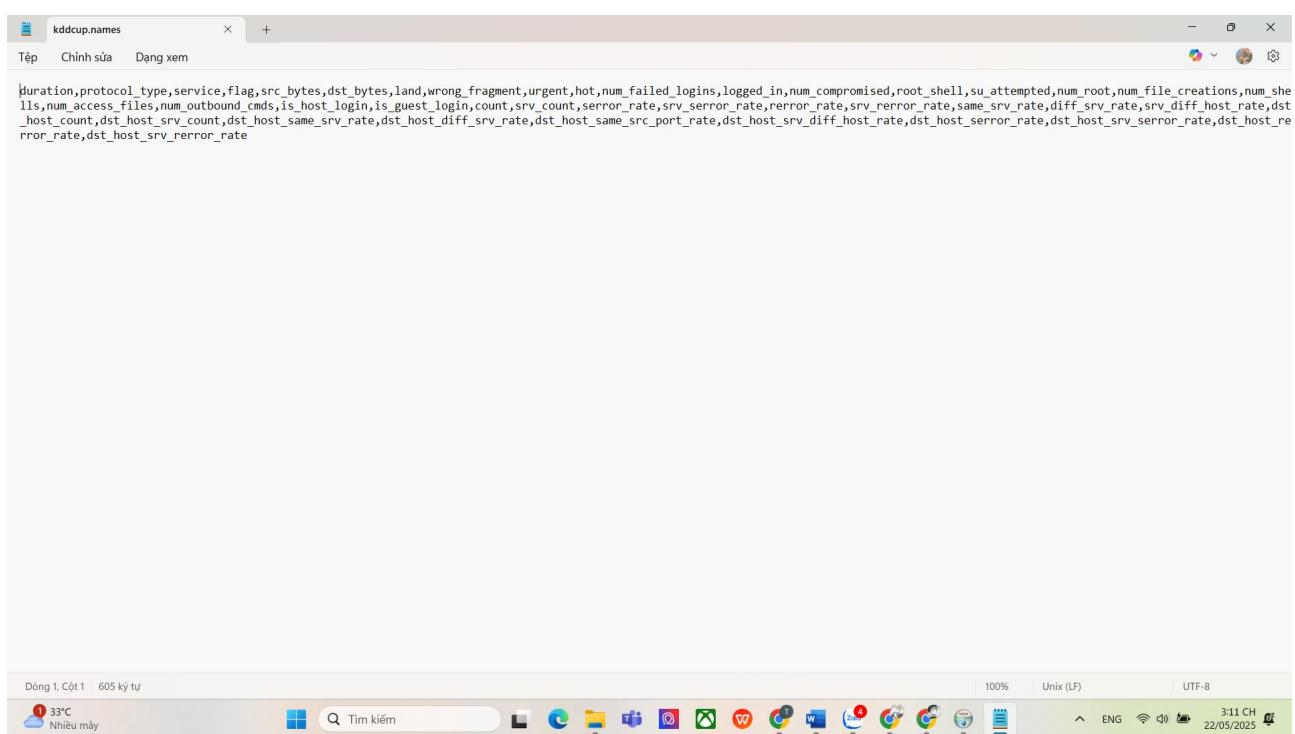
==== Confusion Matrix ====			
a	b	c	<-- classified as
49	1	0	a = Iris-setosa
0	47	3	b = Iris-versicolor
0	2	48	c = Iris-virginica

- **Confusion Matrix:** ma trận đánh giá hiệu suất của mô hình phân loại. Trong ma trận này:
 - o Các hàng đại diện cho các lớp thực tế.
 - o Các cột đại diện cho các lớp dự đoán bởi mô hình.
- ⇒ **Đánh giá kết quả:** Độ chính xác của mô hình là 96% và chỉ số Kappa = 0.94 gần bằng 1 → Cho thấy mô hình phân lớp tốt hơn so với việc dự đoán ngẫu nhiên.

Yêu cầu 3.1 Sinh viên lựa chọn 01 bộ phân lớp bất kỳ và thực hiện khai thác trên tập dữ liệu KDD Cup 1999. Giải thích và đánh kết quả.

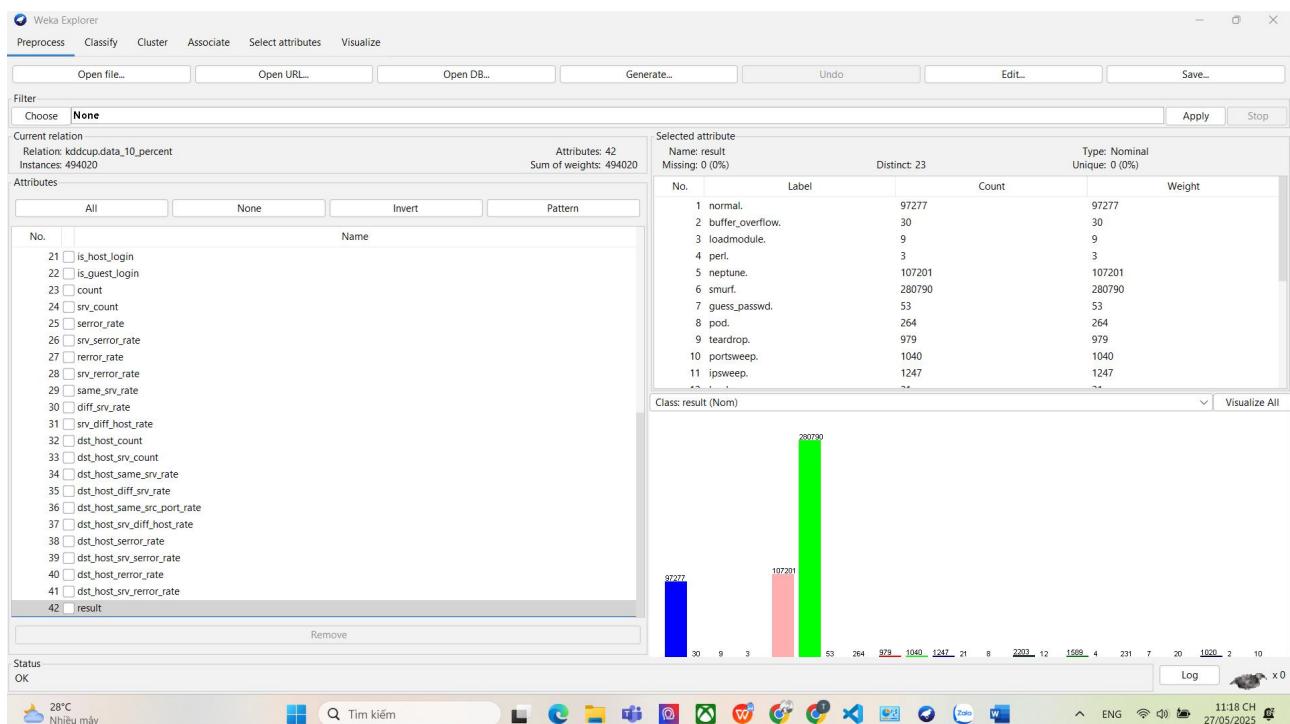
Chuẩn bị dataset:

- Tải các file cần thiết: kddcup.names và kddcup.data_10_percent.zip. Lấy danh sách các thuộc tính trong file kddcup.names và chuyển chúng thành một dòng, phân cách bằng dấu phẩy rồi chèn vào đầu file kddcup.data_10_percent.

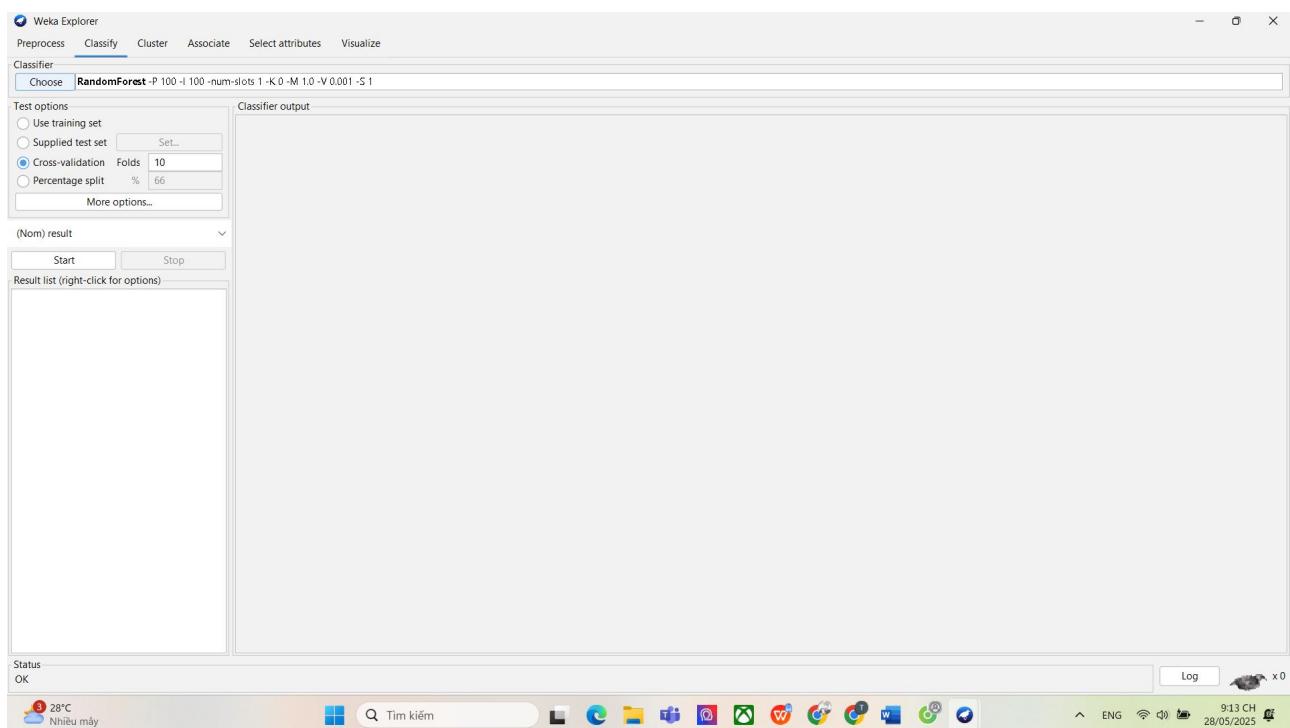


- Sử dụng Excel để mở tập dữ liệu, thao tác và lưu tại dưới định dạng file .csv.

- Thực hiện load file vào Weka.



- Trong tab Classify, nhấn Choose -> Chọn RandomForest để chạy bộ dữ liệu KDD Cup 1999.



*Nguyên lý hoạt động của bộ phân lớp RandomForest:

- RandomForest hoạt động bằng cách tạo ra một tập hợp lớn các cây quyết định.
- Hoạt động dựa trên các kỹ thuật chính:
 - o **Bootstrap Aggregating (Bagging):** Kỹ thuật lấy mẫu ngẫu nhiên với hoàn lại từ tập huấn luyện ban đầu để tạo ra nhiều tập con khác nhau. Mỗi cây quyết định sẽ được huấn luyện độc lập trên các tập con đó.

- **Lựa chọn đặc trưng ngẫu nhiên:** Tại mỗi nút của cây, thay vì xem xét toàn bộ các đặc trưng để tìm điểm chia tối ưu, chỉ một tập con ngẫu nhiên các đặc trưng được chọn để quyết định chia. Điều này giúp các cây trở nên khác biệt và giảm sự tương quan giữa chúng.
- **Tổng Hợp Kết Quả:** Khi phân loại một điểm dữ liệu mới, RandomForest kết hợp dự đoán từ tất cả các cây quyết định bằng cách sử dụng đa số phiếu (cho phân loại) hoặc trung bình (cho hồi quy).

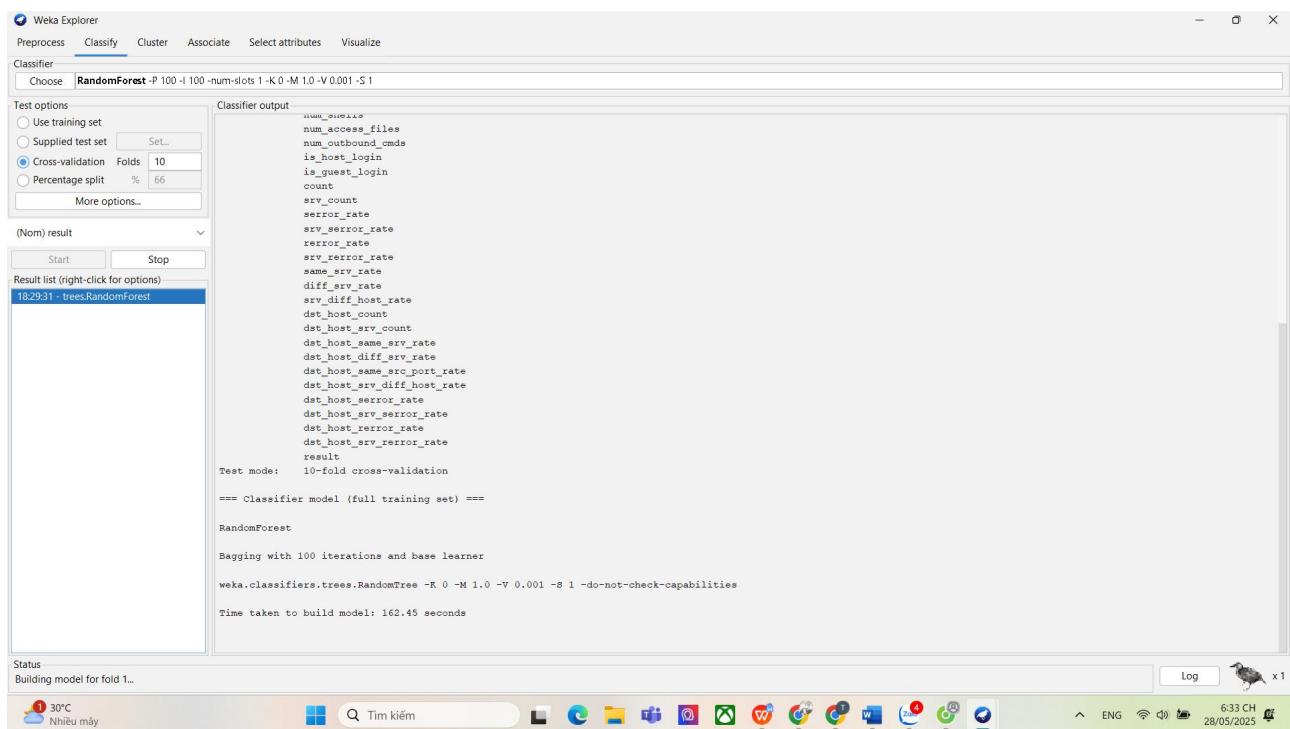
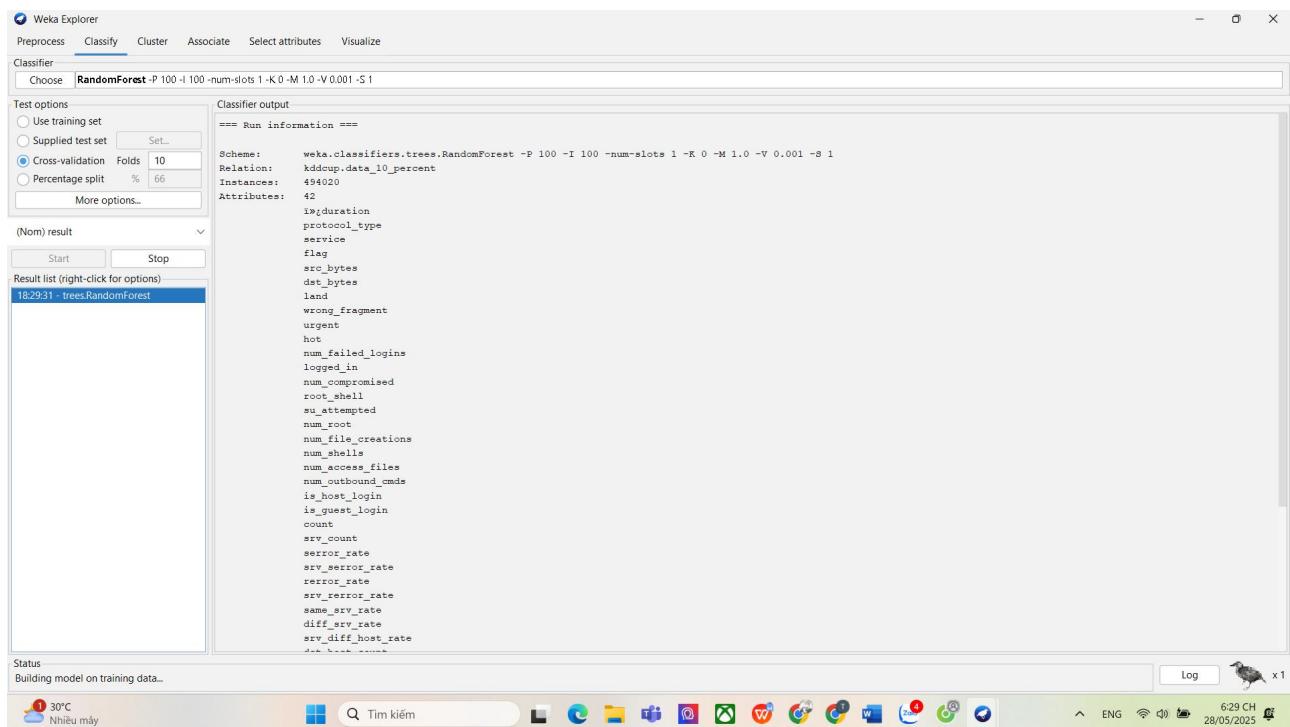
***Lý do chọn bộ phân lớp này:**

- RandomForest có khả năng xử lý tốt dữ liệu có nhiều đặc trưng, nhiều lớp và có khả năng tổng hợp nhiều cây giúp cải thiện độ chính xác, giảm sai số.
- Nhờ lựa chọn ngẫu nhiên mẫu và thuộc tính, RandomForest không dễ bị học thuộc dữ liệu như một cây quyết định duy nhất, chống overfitting.
- Tập dữ liệu KDD chứa các lớp không cân bằng mà RandomForest thường xử lý mất cân bằng tốt hơn so với các mô hình đơn lẻ.
- RandomForest có thể tự động đánh giá mức độ quan trọng của thuộc tính để hiểu được đặc trưng nào quan trọng trong phát hiện tấn công.
- RandomForest không yêu cầu chuẩn hóa hay chuyển đổi dữ liệu - giảm thời gian tiền xử lý.

***Giải thích test option:**

- **Use training set:** Đánh giá mô hình trên chính dữ liệu huấn luyện. Điều này giúp kiểm tra hiệu suất và thời gian chạy của thuật toán.
- **Supplied test set:** Cung cấp một tập dữ liệu kiểm tra riêng biệt, giúp đánh giá mô hình trên dữ liệu chưa từng được thấy trong quá trình huấn luyện.
- **Cross-validation:** Folds là số dataset con được tạo ra; với folds = x, dataset sẽ được chia thành x phần, WEKA sẽ train trên x-1 dataset con và thực hiện test trên tập còn lại; thực hiện công đoạn này cho đến khi mỗi tập dataset con đều được lấy làm test set, điểm trung bình được ghi lại là thước đo của hiệu suất mô hình.
- **Percentage split:** Tập dữ liệu được chia thành 2 phần; một phần để huấn luyện và phần còn lại để kiểm tra.

- Nhấn Start để bắt đầu chạy bộ phân lớp và kết quả:



Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier Choose **RandomForest-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1**

Test options

- Use training set
- Supplied test set Set...
- Cross-validation Folds 10
- Percentage split % 66
- More options...

(Nom) result

Start Stop

Result list (right-click for options) 18:29:31 - treesRandomForest

Classifier output

```
== Stratified cross-validation ==
== Summary ==
Correctly Classified Instances 493923 59.9804 %
Incorrectly Classified Instances 97 0.0196 %
Kappa statistic 0.9957
Mean absolute error 0.0001
Root mean squared error 0.004
Relative absolute error 0.1056 %
Root relative squared error 2.5217 %
Total Number of Instances 494020
```

== Detailed Accuracy By Class ==

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1,000	0,000	0,999	1,000	1,000	1,000	1,000	1,000	1,000	normal.
0,867	0,000	0,788	0,867	0,825	0,826	1,000	0,930	1,000	buffer_overflow.
0,222	0,000	1,000	0,222	0,364	0,471	1,000	0,691	1,000	loadmodule.
0,667	0,000	1,000	0,667	0,800	0,816	1,000	1,000	1,000	perl.
1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	neptune.
1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	smurf.
0,962	0,000	1,000	0,962	0,981	0,981	1,000	0,996	1,000	guess_passwd.
0,996	0,000	0,996	0,996	0,996	0,996	1,000	1,000	1,000	pod.
1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	teardrop.
0,996	0,000	0,998	0,996	0,997	0,997	1,000	0,999	1,000	portsweep.
0,992	0,000	0,996	0,992	0,994	0,994	1,000	1,000	1,000	ipsweep.
0,905	0,000	0,950	0,905	0,927	0,927	1,000	0,979	1,000	land.
0,500	0,000	1,000	0,500	0,667	0,707	1,000	0,604	1,000	ftp_write.
1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	back.
0,917	0,000	1,000	0,917	0,957	0,957	1,000	1,000	1,000	imap.
0,991	0,000	1,000	0,991	0,995	0,995	1,000	0,999	1,000	satan.
0,750	0,000	1,000	0,750	0,857	0,866	1,000	1,000	1,000	phf.
0,974	0,000	1,000	0,974	0,987	0,987	1,000	0,996	1,000	nmap.
0,429	0,000	0,600	0,429	0,500	0,507	1,000	0,602	1,000	microsoft_dfs.
0,800	0,000	0,889	0,800	0,842	0,843	0,975	0,899	1,000	warezmaster.
0,993	0,000	0,993	0,993	0,993	0,993	1,000	1,000	1,000	warezclient.
0,000	0,000	?	0,000	?	?	1,000	0,450	1,000	rootkit.

Status OK Log x 0

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier Choose **RandomForest-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1**

Test options

- Use training set
- Supplied test set Set...
- Cross-validation Folds 10
- Percentage split % 66
- More options...

(Nom) result

Start Stop

Result list (right-click for options) 18:29:31 - treesRandomForest

Classifier output

```
0,750 0,000 1,000 0,750 0,857 0,866 1,000 1,000 phf.
0,974 0,000 1,000 0,974 0,987 0,987 1,000 0,996 nmap.
0,429 0,000 0,600 0,429 0,500 0,507 1,000 0,602 multihop.
0,800 0,000 0,889 0,800 0,842 0,843 0,975 0,899 warezmaster.
0,993 0,000 0,993 0,993 0,993 0,993 1,000 1,000 warezclient.
0,000 0,000 ? 0,000 ? ? 1,000 0,450 spy.

Weighted Avg. 1,000 0,000 ? 1,000 ? ? 1,000 1,000 rootkit.

== Confusion Matrix ==
```

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	<-- classifier
97266	2	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	5	0	1		a = nor	
4	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		b = buf
4	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		c = loa
1	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		d = per
1	0	0	0	107199	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0		e = nep
0	0	0	0	0	280790	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		f = emu
2	0	0	0	0	0	51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		g = que
1	0	0	0	0	0	0	263	0	0	0	0	0	0	0	0	0	0	0	0	0	0		h = pod
0	0	0	0	0	0	0	579	0	0	0	0	0	0	0	0	0	0	0	0	0	0		i = tea
2	0	0	0	1	0	0	0	0	1036	1	0	0	0	0	0	0	0	0	0	0	0		j = por
9	0	0	0	0	0	0	0	0	1	1237	0	0	0	0	0	0	0	0	0	0	0		k = ips
0	0	0	0	2	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0		l = lan
2	1	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	1	0	0		m = ftp
0	0	0	0	0	0	0	0	0	0	0	0	2203	0	0	0	0	0	0	0	0	0		n = bac
1	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0		o = ima
13	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1574	0	0	0	0	0	0		p = sat
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0		q = phf
2	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	225	0	0	0	0	0		r = rma
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	2	1	0	0		s = mul
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	16	0	0	0	0	0		t = war
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1013	0	0		u = war
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		v = spy
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		w = roo

Status OK Log x 0

* Giải thích và đánh giá các kết quả thu được:

```
== Run information ==

Scheme:      weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1
Relation:    kddcup.data_10_percent
Instances:   494020
Attributes:  42
  i»;duration
  protocol_type
  service
  flag
  src_bytes
  dst_bytes
  land
  wrong_fragment
  urgent
  hot
  num_failed_logins
  logged_in
  num_compromised
  root_shell
  su_attempted
  num_root
  num_file_creations
  num_shells
  num_access_files
  num_outbound_cmds
  is_host_login
  is_guest_login
  count
  srv_count
  serror_rate
  srv_serror_rate
  rerror_rate
  srv_rerror_rate
  same_srv_rate
  diff_srv_rate
  srv_diff_host_rate
```

- Đây là thông tin mô hình và các thuộc tính trong tập dữ liệu. Mô hình RandomForest đã được huấn luyện thành công với 100 cây trên 494020 mẫu và sử dụng 42 thuộc tính.

```
Test mode:    10-fold cross-validation

== Classifier model (full training set) ==

RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 162.45 seconds
```

- Chế độ kiểm tra mô hình là 10-fold cross-validation.
- RandomForest thực chất là Bagging (tập hợp mô hình) sử dụng 100 cây quyết định. Mỗi cây là một mô hình RandomTree độc lập, được huấn luyện trên một mẫu ngẫu nhiên có lặp của dữ liệu gốc.
- RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1
 - o -P 100: Thực hiện phân lớp với 100% dữ liệu huấn luyện tại mỗi lần lặp.
 - o -I 100: Số lượng cây quyết định trong rừng là 100.
 - o -num-slots 1: Sử dụng một luồng đơn để huấn luyện mô hình.
 - o -K 0: Số lượng thuộc tính được chọn ngẫu nhiên tại mỗi nút phân chia là căn bậc hai của tổng số thuộc tính.

- -M 1.0: Số lượng mẫu nhỏ nhất để phân chia một nút.
- -V 0.001: Giới hạn chi nhánh mới sẽ không được tạo nếu sự tăng thông tin đạt được nhỏ hơn giá trị này.
- -S 1: Hạt giống ngẫu nhiên.
- Mô hình mất 162.45 giây để hoàn tất toàn bộ quá trình huấn luyện với cross-validation 10-fold.
- ⇒ Thời gian huấn luyện là hợp lý với quy mô dữ liệu lớn (494020 mẫu).

```
==== Stratified cross-validation ====
==== Summary ====

Correctly Classified Instances      493923           99.9804 %
Incorrectly Classified Instances    97              0.0196 %
Kappa statistic                   0.9997
Mean absolute error               0.0001
Root mean squared error          0.004
Relative absolute error          0.1056 %
Root relative squared error     2.5217 %
Total Number of Instances        494020
```

- Số mẫu được phân lớp đúng là 493,923 / 494,020 (99.9804%).
 - Chỉ có 97 mẫu bị phân lớp sai – rất nhỏ.
 - Chỉ số Kappa statistic đánh giá độ chính xác mô hình so với phân loại ngẫu nhiên. Giá trị này là 0.9997 (gần bằng 1) có nghĩa là mô hình có độ chính xác cao.
 - Mean absolute error (MAE) là sai số trung bình tuyệt đối. Ở đây chỉ số này có giá trị 0.0001 (gần bằng 0).
 - Chỉ số Root mean squared error (RMSE) là căn bậc hai sai số bình phương – càng gần 0 càng tốt.
 - Relative absolute error là MAE so với sai số trung bình của dữ liệu. Ở đây chỉ số đó rất nhỏ.
 - Chỉ số Root relative squared error là RMSE so với độ biến thiên của dữ liệu – rất thấp, chứng tỏ mô hình mạnh.
 - Total Number of Instances là tổng số mẫu trong tập dữ liệu kiểm tra (494020 mẫu).
- ⇒ Mô hình RandomForest hoạt động cực kỳ hiệu quả với độ chính xác gần như tuyệt đối (99.98%) và các chỉ số như Kappa, MAE, RMSE đều không cao thể hiện khả năng sai số thấp.

==== Detailed Accuracy By Class ====										
TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class		
1,000	0,000	0,999	1,000	1,000	1,000	1,000	1,000	normal.		
0,867	0,000	0,788	0,867	0,825	0,826	1,000	0,930	buffer_overflow.		
0,222	0,000	1,000	0,222	0,364	0,471	1,000	0,691	loadmodule.		
0,667	0,000	1,000	0,667	0,800	0,816	1,000	1,000	perl.		
1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	neptune.		
1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	smurf.		
0,962	0,000	1,000	0,962	0,981	0,981	1,000	0,996	guess_passwd.		
0,996	0,000	0,996	0,996	0,996	0,996	1,000	1,000	pod.		
1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	teardrop.		
0,996	0,000	0,998	0,996	0,997	0,997	1,000	0,999	portsweep.		
0,992	0,000	0,996	0,992	0,994	0,994	1,000	1,000	ipsweep.		
0,905	0,000	0,950	0,905	0,927	0,927	1,000	0,979	land.		
0,500	0,000	1,000	0,500	0,667	0,707	1,000	0,604	ftp_write.		
1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	back.		
0,917	0,000	1,000	0,917	0,957	0,957	1,000	1,000	imap.		
0,991	0,000	1,000	0,991	0,995	0,995	1,000	0,999	satan.		
0,750	0,000	1,000	0,750	0,857	0,866	1,000	1,000	phf.		
0,974	0,000	1,000	0,974	0,987	0,987	1,000	0,996	rmap.		
0,429	0,000	0,600	0,429	0,500	0,507	1,000	0,602	multihop.		
0,800	0,000	0,889	0,800	0,842	0,843	0,975	0,899	warezmaster.		
0,993	0,000	0,993	0,993	0,993	0,993	1,000	1,000	warezclient.		
0,000	0,000	?	0,000	?	?	1,000	0,450	spy.		
0,100	0,000	0,500	0,100	0,167	0,224	1,000	0,222	rootkit.		
Weighted Avg.	1,000	0,000	?	1,000	?	1,000	1,000			

Đây là đánh giá chi tiết theo lớp:

- TP Rate (True Positive Rate): Đo lường tỷ lệ các mẫu dương tính thực sự được dự đoán đúng.
 - FP Rate (False Positive Rate): Đo lường tỷ lệ các mẫu âm tính thực sự bị dự đoán sai.
 - Precision: Đo lường tỷ lệ dự đoán đúng trong tổng số dự đoán dương tính.
 - Recall: Đo lường tỷ lệ dự đoán đúng trong tổng số mẫu dương tính thực sự.
 - F-Measure: Trung bình hài hòa giữa Precision và Recall.
 - MCC (Matthews Correlation Coefficient): Đo lường mối tương quan giữa các giá trị thực và dự đoán.
 - ROC Area: Diện tích dưới đường cong ROC, đo lường khả năng phân biệt giữa các lớp.
 - PRC Area: Diện tích dưới đường cong Precision-Recall.
- ⇒ Các lớp có hiệu suất cao là normal., neptune., smurf., teardrop., back., iimap., phf.
-> Các lớp này có TP Rate, Precision, Recall, F-Measure, MCC đều bằng 1.0 ->
Phân lớp tuyệt đối chính xác.
- ⇒ Các lớp có hiệu suất thấp là loadmodule., rootkit., spy. -> Các lớp có tỷ lệ Recall thấp -> Mô hình bỏ sót nhiều mẫu hoặc khó học được đặc trưng của lớp này.
- ⇒ Mô hình học rất tốt các lớp phổ biến; chỉ số FP Rate bằng 0 thể hiện sự đáng tin cậy trong bối cảnh an ninh mạng và mô hình giữ được sự nhất quán và cân bằng tốt. Nhưng một số lớp hiếm cần cân bằng lại dữ liệu.

Confusion matrix:

Confusion matrix hiển thị số lượng các mẫu từ từng lớp được dự đoán thành từng lớp khác. Hàng đại diện cho lớp thực và cột đại diện cho lớp dự đoán.

== Confusion Matrix ==																								
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	<-- classified as	
97266	2	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	5	0	1	a = normal.		
4	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	b = buffer_overflow.		
4	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	c = loadmodule.		
1	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	d = perl.		
1	0	0	0	107199	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	e = neptune.		
0	0	0	0	0	280790	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	f = smurf.		
2	0	0	0	0	0	51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	g = guess_passwd.		
1	0	0	0	0	0	0	263	0	0	0	0	0	0	0	0	0	0	0	0	0	0	h = pod.		
0	0	0	0	0	0	0	0	979	0	0	0	0	0	0	0	0	0	0	0	0	0	i = teardrop.		
2	0	0	0	1	0	0	0	0	1036	1	0	0	0	0	0	0	0	0	0	0	0	j = portsweep.		
9	0	0	0	0	0	0	0	0	1	1237	0	0	0	0	0	0	0	0	0	0	0	k = ipsweep.		
0	0	0	0	2	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0	l = land.		
2	1	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	1	0	m = ftp_write.		
0	0	0	0	0	0	0	0	0	0	0	0	2203	0	0	0	0	0	0	0	0	0	n = back.		
1	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	o = imap.		
13	0	0	0	2	0	0	0	0	0	0	0	0	1574	0	0	0	0	0	0	0	0	p = satan.		
1	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	q = pnf.		
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	225	0	0	0	0	0	0	r = namap.		
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	2	1	0	0	0	s = multihop.		
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	16	0	0	0	0	t = warezmaster.		
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	u = warezclient.		
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	v = spy.		
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	w = rootkit.		