BÁO CÁO THỰC HÀNH

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

Lab 5: Học máy trong IDS

GVHD: Đỗ Hoàng Hiển

1. **THÔNG TIN CHUNG:**

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT204.O21.ATCL.2- Nhóm 3

STT	Họ và tên	MSSV	Email
1	Huỳnh Minh Tân Tiến	21521520	21521520@gm.uit.edu.vn
2	Lê Hoàng Oanh	21521253	21521253@gm.uit.edu.vn

2. **NỘI DUNG THỰC HIỆN:**

STT	Công việc	Kết quả tự đánh giá
1	**Yêu cầu 1**	100%
2	**Yêu cầu 2**	100%
3	**Yêu cầu 3**	100%

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

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.

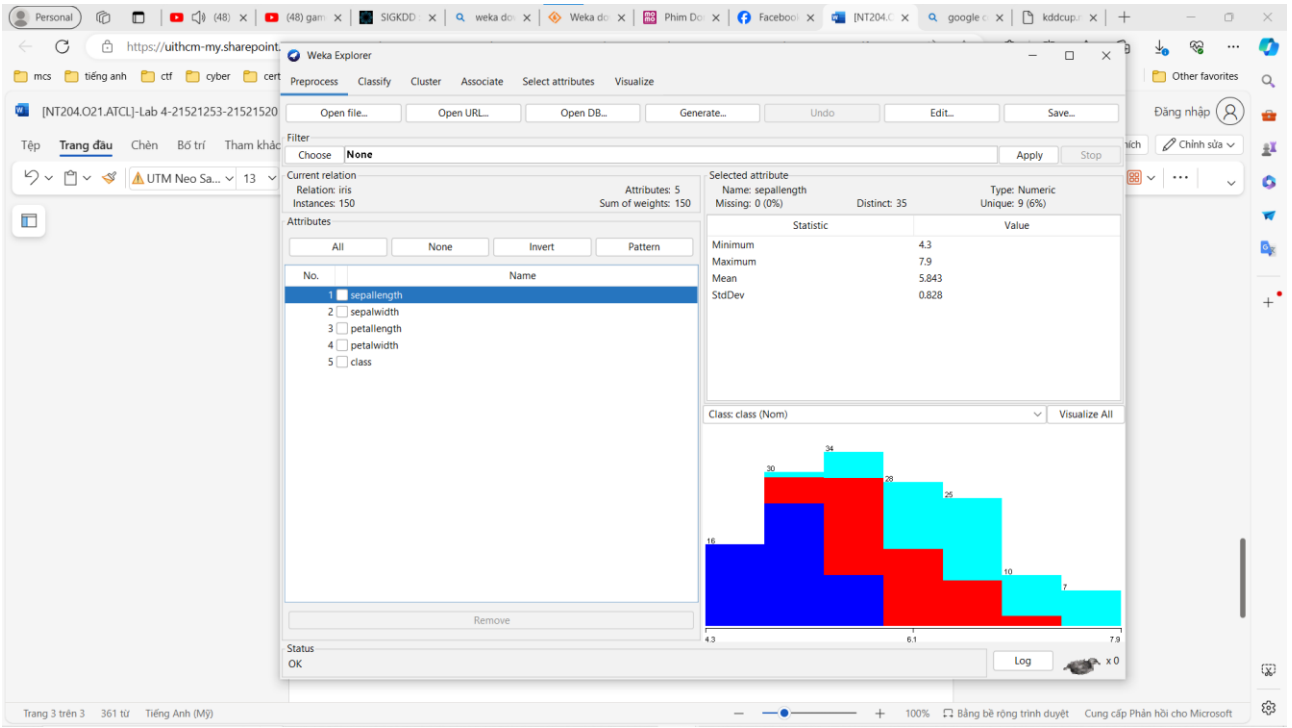
TÌM HIỂU VỀ TẬP DỮ LIỆU KDD CUP 1999 Dữ liệu trong bộ dữ liệu KDD Cup 1999 là lưu lượng mạng đã được thu thập, phân tích, xử lý để lấy các thuộc tính và từ đó gán nhãn tương ứng với loại tấn công hoặc dữ liệu bình thường. Sinh viên tìm hiểu các phần sau

1. Số nhóm tấn công: 4
   Tên các nhóm tấn công:
   - DOS (Denial of Service): Tấn công từ chối dịch vụ
   - R2L (Remote to Local): Tấn công từ xa đến cục bộ
   - U2R (User to Root): Tấn công từ người dùng lên quyền root
   - Probe: Tấn công thăm dò

2. Số kiểu tấn công: 23
   Kể tên các kiểu được gán nhãn : 'normal.' 'buffer_overflow.' 'loadmodule.' 'perl.' 'neptune.' 'smurf.' 'guess_passwd.' 'pod.' 'teardrop.' 'portsweep.' 'ipsweep.' 'land.' 'ftp_write.' 'back.' 'imap.' 'satan.' 'phf.' 'nmap.' 'multihop.' 'warezmaster.' 'warezclient.' 'spy.' 'rootkit.'

3. Mỗi instance trong tập dữ liệu KĐ Cup 1999 bao gồm 42 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','outcome'

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.



- Relation: Định nghĩa tên của tập dữ liệu. Ví dụ: @relation iris.
- Attribute: định nghĩa các thuộc tính đặc trưng của dữ liệu. Ví dụ: trong bảng attribute của hình trên, có 5 attribute: sepallength, sepalwidth, petal length và petalwidth .
- Instances: Số lượng các instance (mẫu dữ liệu), ví dụ: 150
- Class: Thuộc tính mục tiêu (class attribute), ví dụ: class.
- Name (Tên): Tên của thuộc tính. Ví dụ: sepallength.
- Type (Kiểu): Mô tả: Kiểu dữ liệu của thuộc tính, chẳng hạn như Numeric (số), Nominal (danh mục). Ví dụ: Numeric.
- Missing (Thiếu): Số lượng giá trị thiếu trong thuộc tính. Giá trị "Missing" cho biết có bao nhiêu mẫu trong tập dữ liệu mà thuộc tính này không có giá trị được gán. Ví dụ: 0 (không có giá trị thiếu).
- Distinct (Khác biệt): Số lượng giá trị khác nhau có trong thuộc tính. Ví dụ: 35 (có 35 giá trị khác nhau trong thuộc tính sepallength).
- Unique (Duy nhất): Số lượng giá trị duy nhất (unique values) trong thuộc tính. Giá trị "Unique" cho biết có bao nhiêu giá trị khác nhau xuất hiện trong thuộc tính, là một chỉ số về sự đa dạng của thuộc tính. Ví dụ: 35 (tất cả các giá trị khác nhau đều là duy nhất).
- Weight (Trọng số): Tổng trọng số của các instance trong tập dữ liệu, thường bằng tổng số lượng instance nếu không có trọng số cụ thể được gán. Ví dụ: 150.0.
- Minimum (Giá trị tối thiểu): Giá trị nhỏ nhất trong tập dữ liệu. Đại diện cho giá trị thấp nhất có thể có trong tập dữ liệu. Ví dụ: 4.3.

- Maximum (Giá trị tối đa): Giá trị lớn nhất trong tập dữ liệu. Đại diện cho giá trị cao nhất có thể có trong tập dữ liệu. Ví dụ: 7.9.
- Mean (Trung bình): Giá trị trung bình của tất cả các giá trị trong thuộc tính. Để tính trung bình, cộng tổng tất cả các giá trị và chia cho số lượng các giá trị. Ví dụ: 5.8433.
- StdDev (Độ lệch chuẩn): Mô tả mức độ phân tán của dữ liệu xung quanh giá trị trung bình. Công thức tính độ lệch chuẩn:

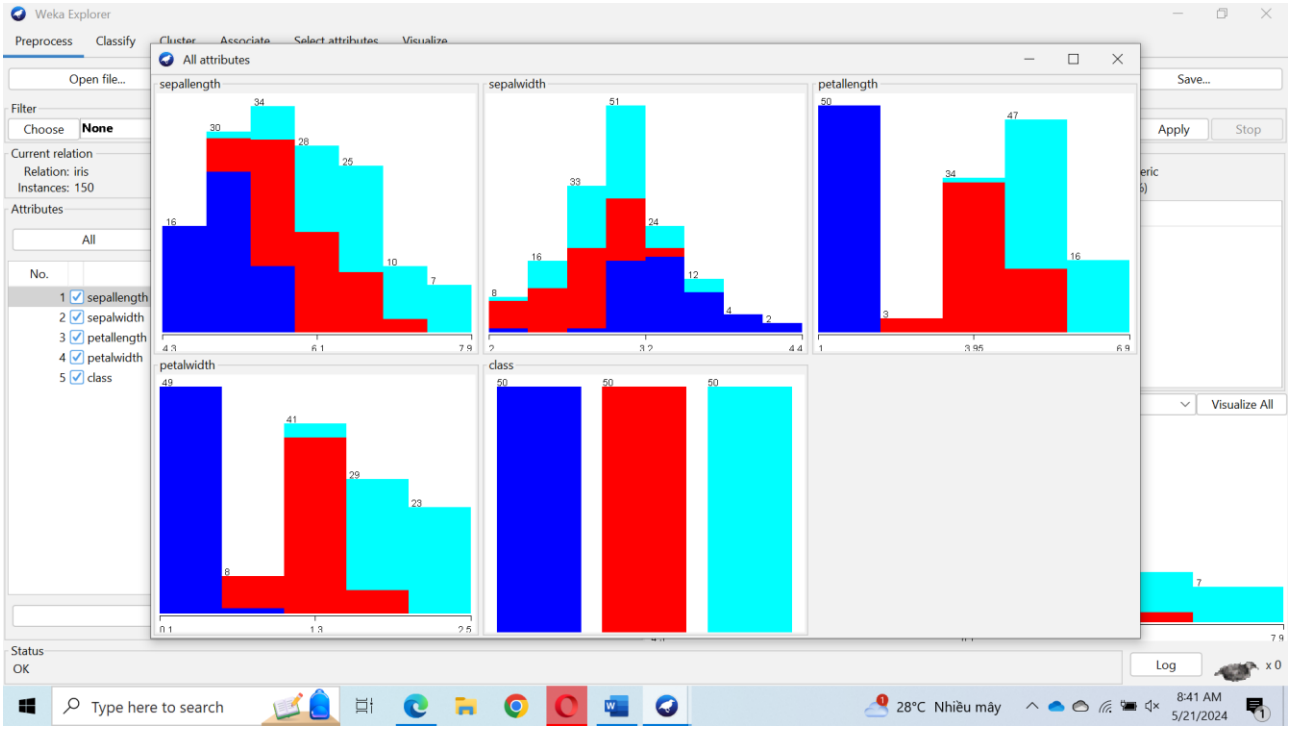
  $$\text{Độ lệch chuẩn} = \sqrt{[(\Sigma(xi - mean)^2) / N]}$$

  Trong đó:

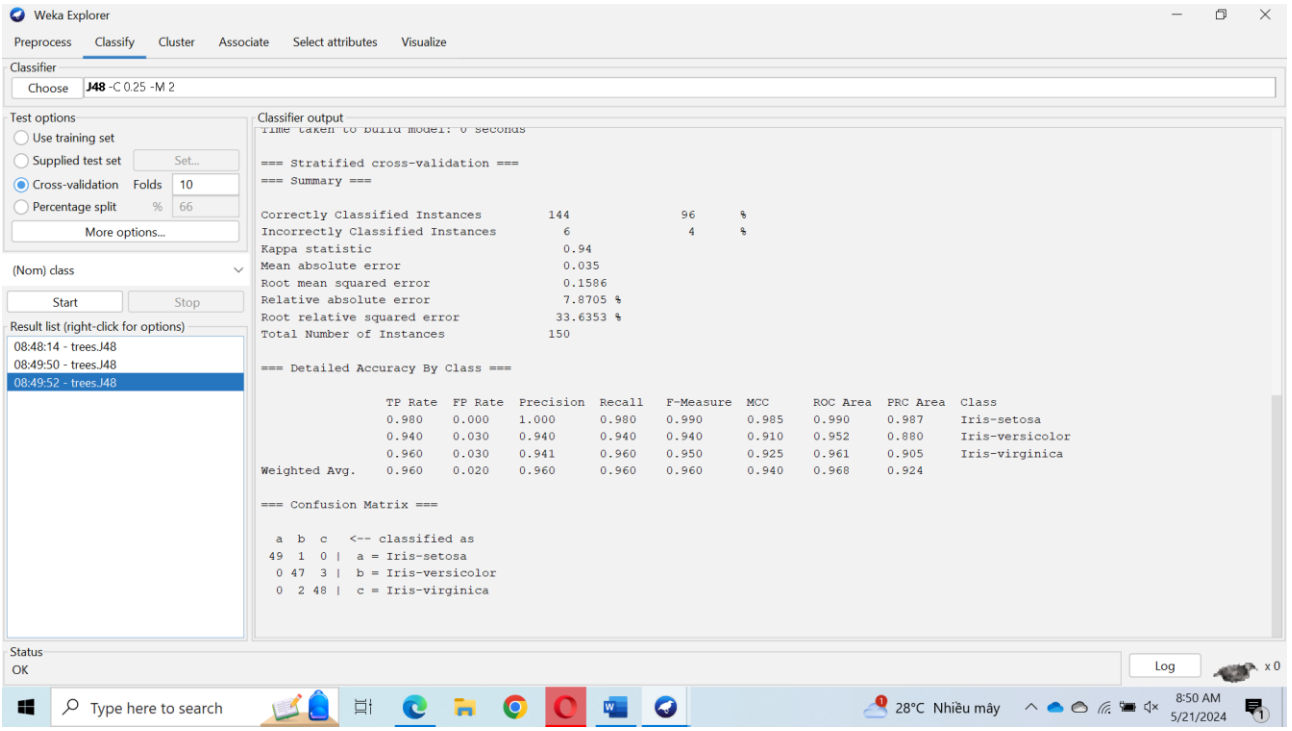
  x i là giá trị của từng điểm dữ liệu.
  mean là giá trị trung bình của tập dữ liệu.
  ∑ là ký hiệu tổng của tất cả các giá trị.
  N là số lượng các điểm dữ liệu trong tập.

  Ví dụ: 0.8281.

Đây là hình thể hiển tất cả các attribute của iris



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.



- Giới thiệu về bộ phân lớp J48

Bộ phân lớp J48 là một thuật toán cây quyết định (decision tree) dựa trên 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, chúng ta sử dụng bộ phân lớp J48 với các tham số -C 0.25 (confidence factor) và -M 2 (minimum number of instances per leaf).

- Tập dữ liệu Iris

Tập dữ liệu Iris là một tập dữ liệu nổi tiếng trong học máy, bao gồm 150 mẫu, mỗi mẫu có 4 đặc trưng (sepallength, sepalwidth, petallength, petalwidth) và 1 nhãn lớp (class), với ba loại hoa Iris (Iris-setosa, Iris-versicolor, Iris-virginica).

- Các tham số và tùy chọn test
  - Tham số -C 0.25: Đây là giá trị của hệ số tin cậy dùng để cắt tỉa cây (pruning). Cắt tỉa cây giúp giảm độ phức tạp của cây và ngăn chặn overfitting. Giá trị 0.25 có nghĩa là có 25% xác suất một nhánh được giữ lại trong cây.
  - Tham số -M 2: Đây là số lượng mẫu tối thiểu trong một node lá. Nếu một node có ít hơn 2 mẫu, nó sẽ không được chia thêm.
  - Test mode: 10-fold cross-validation: Phương pháp kiểm tra chéo này chia tập dữ liệu thành 10 phần. Mỗi phần lần lượt được dùng làm tập kiểm tra (test set), và 9 phần còn lại được dùng làm tập huấn luyện (training set). Kết quả cuối cùng là trung bình của 10 lần chạy.
- Kết quả của bộ phân lớp

Cây quyết định J48 được xây dựng như sau:

```
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)
```

Số lá: 5

Kích thước của cây: 9

Thời gian xây dựng mô hình: 0 giây

- Kết quả kiểm tra chéo (cross-validation)
  - Tỷ lệ chính xác: 96% (144 mẫu được phân loại chính xác trong tổng số 150 mẫu)
  - Kappa statistic: 0.94 (đo lường độ tin cậy của mô hình so với một mô hình ngẫu nhiên)
  - Mean absolute error: 0.035 (độ lệch trung bình tuyệt đối của các dự đoán so với giá trị thực)
  - Root mean squared error: 0.1586 (độ lệch trung bình bình phương căn bậc hai của các dự đoán)
  - Relative absolute error: 7.8705%
  - Root relative squared error: 33.6353%
- Phân tích độ chính xác theo lớp

```
=== 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
```

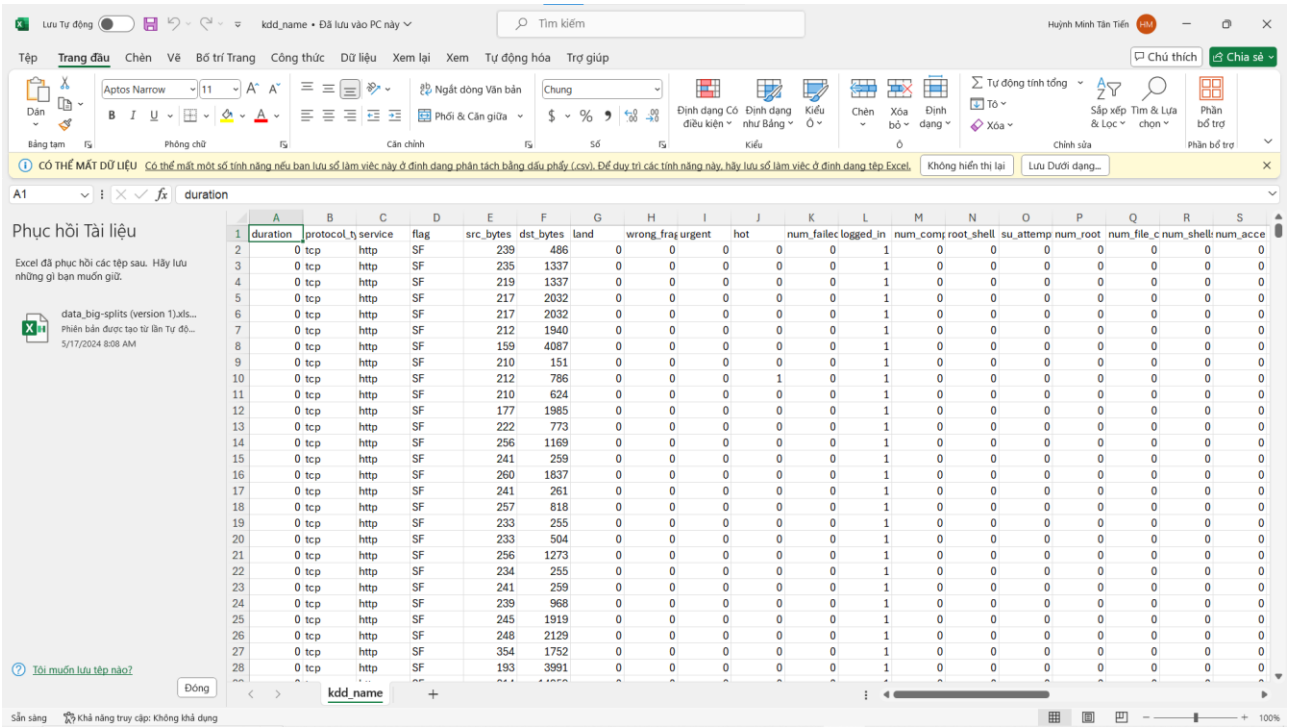
- Confusion Matrix

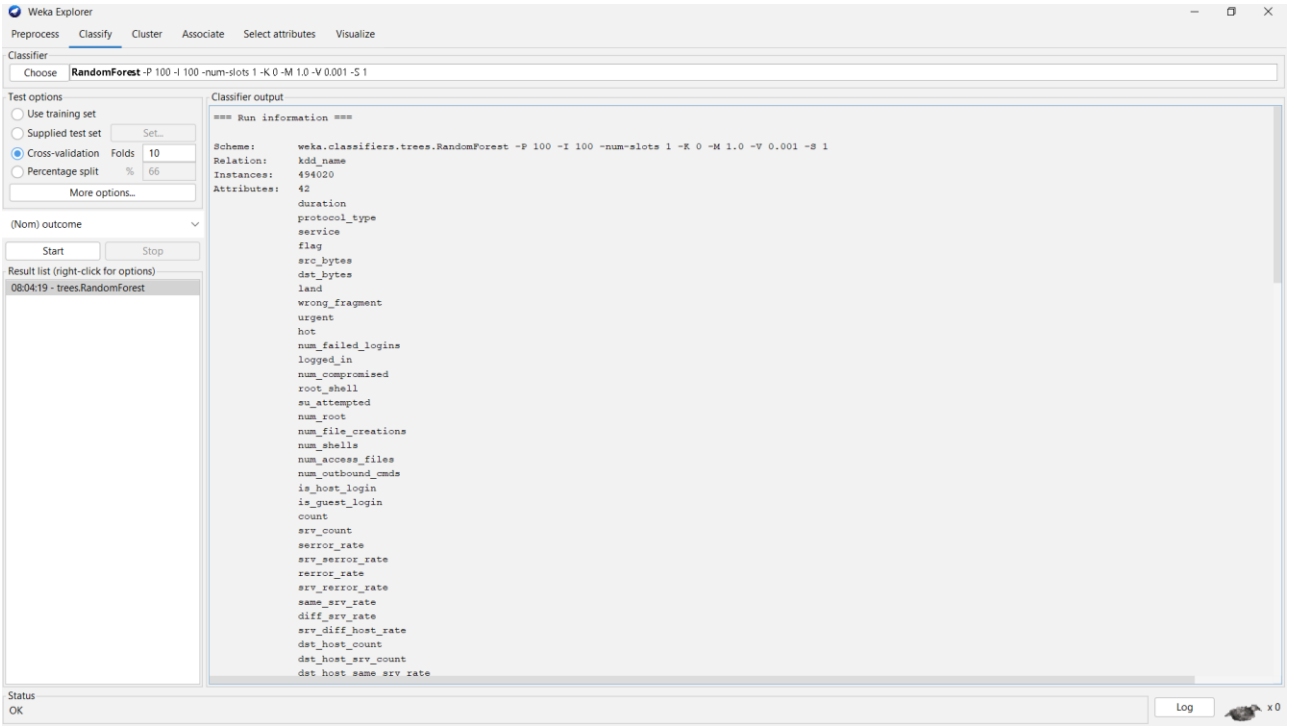
```
=== Confusion Matrix ===

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

- Giải thích kết quả
  - TP Rate (True Positive Rate): Tỷ lệ mẫu của mỗi lớp được dự đoán chính xác. Ví dụ, 98% của lớp Iris-setosa được dự đoán chính xác.

- FP Rate (False Positive Rate): Tỷ lệ mẫu của các lớp khác được dự đoán sai là lớp hiện tại. Ví dụ, không có mẫu nào của các lớp khác được dự đoán sai là Iris-setosa.
- Precision: Độ chính xác của các dự đoán dương tính (positive predictions). Ví dụ, tất cả các dự đoán Iris-setosa đều đúng.
- Recall: Khả năng nhận diện đúng các mẫu của lớp. Ví dụ, 98% của mẫu Iris-setosa được nhận diện đúng.
- F-Measure: Trung bình điều hòa của Precision và Recall. Ví dụ, F-Measure của Iris-setosa là 0.990.
- MCC (Matthews Correlation Coefficient): Một thước đo chất lượng phân loại binary. Giá trị càng gần 1 càng tốt.
- ROC Area và PRC Area: Đo lường khả năng phân loại của mô hình. Giá trị càng gần 1 càng tốt.

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.





Nguyên lý hoạt động của RandomForest

RandomForest là một mô hình học máy sử dụng nhiều cây quyết định để phân loại hoặc hồi quy dữ liệu. Nguyên lý cơ bản của RandomForest bao gồm:

1. **Bootstrap Aggregating (Bagging)**: RandomForest tạo nhiều tập dữ liệu con từ tập dữ liệu gốc bằng cách chọn ngẫu nhiên với phép lặp lại. Mỗi cây quyết định trong rừng được huấn luyện trên một tập dữ liệu con khác nhau.
2. **Tạo Cây Quyết Định Ngẫu Nhiên**: Mỗi cây quyết định trong RandomForest được xây dựng dựa trên một tập hợp ngẫu nhiên của các thuộc tính. Điều này giúp giảm sự tương quan giữa các cây, làm tăng tính đa dạng và khả năng tổng quát hóa của mô hình.
3. **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).

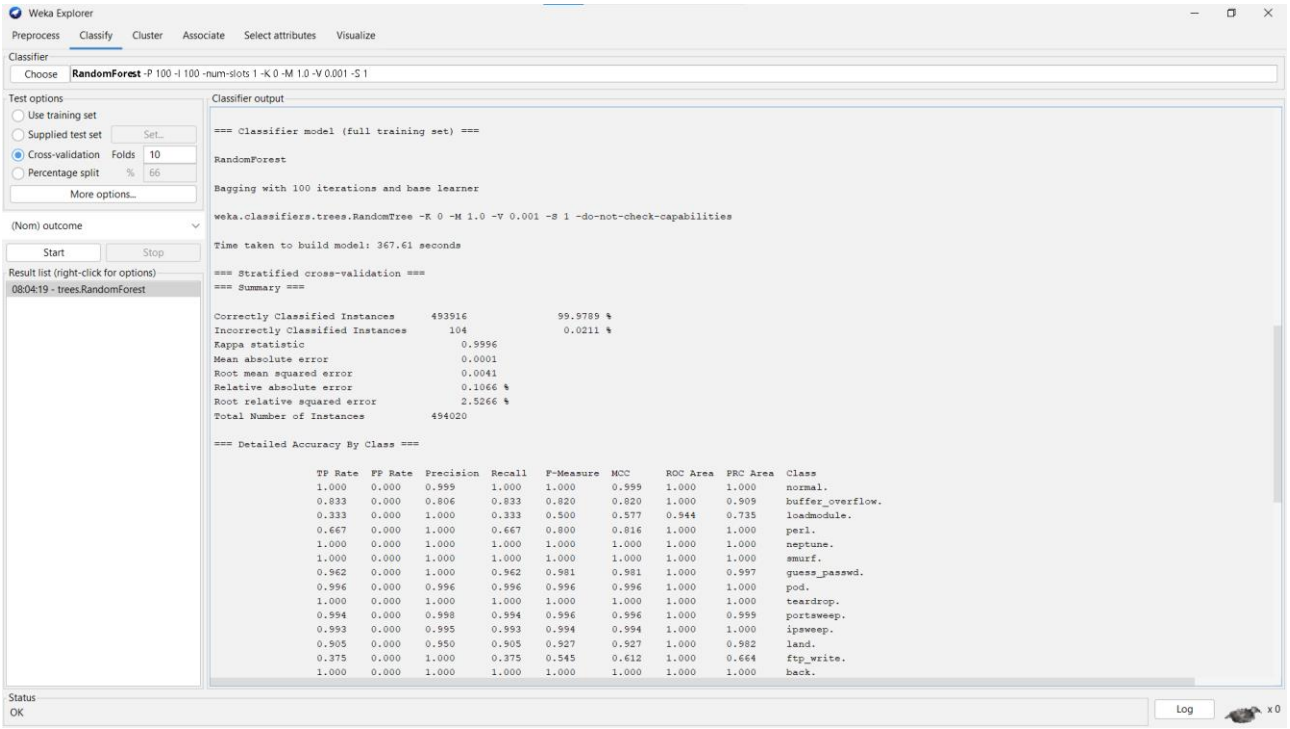
Giải thích lý do chọn bộ phân lớp này:

- **Hiệu suất Cao**: RandomForest thường cho hiệu suất cao trên nhiều loại dữ liệu và vấn đề khác nhau vì nó giảm thiểu overfitting thông qua việc trung bình hóa nhiều cây quyết định.
- **Đa dạng và Ổn định**: Do sự kết hợp của nhiều cây quyết định, RandomForest có khả năng đối phó tốt với dữ liệu có nhiều biến và không đồng nhất.
- **Khả năng Giải thích**: RandomForest cung cấp thông tin về tầm quan trọng của các thuộc tính, giúp hiểu rõ hơn về mô hình và dữ liệu.

a) Giải thích các test option:

- **-P 100**: Thực hiện phân lớp với 100% dữ liệu huấn luyện tại mỗi lần lặp.
- **-I 100**: Số lượng cây quyết định trong rừng là 100.
- **-num-slots 1**: Sử dụng một luồng đơn để huấn luyện mô hình.
- **-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 nhỏ nhất các mẫu tại một nút để có thể tiếp tục phân chia.
- **-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 cho việc chọn ngẫu nhiên.

b) Giải thích và đánh giá các kết quả thu được:



Kết quả chung:

- **Correctly Classified Instances**: 99.9789% - Tỷ lệ phân loại đúng rất cao, chứng tỏ mô hình hoạt động tốt trên tập dữ liệu này.
- **Incorrectly Classified Instances**: 0.0211% - Tỷ lệ phân loại sai rất thấp.
- **Kappa statistic**: 0.9996 - Giá trị Kappa cao cho thấy sự đồng thuận cao giữa mô hình và dữ liệu thực.
- **Mean absolute error**: 0.0001 - Sai số tuyệt đối trung bình rất thấp, chứng tỏ độ chính xác của mô hình.
- **Root mean squared error**: 0.0041 - Sai số trung bình bình phương gốc thấp, cho thấy độ chính xác cao.
- **Relative absolute error**: 0.1066% - Sai số tuyệt đối tương đối thấp.

- **Root relative squared error**: 2.5266% - Sai số trung bình bình phương gốc tương đối thấp.

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.

Ví dụ:

- **normal**: TP Rate và FP Rate đều gần như hoàn hảo, chứng tỏ mô hình dự đoán rất chính xác cho lớp này.
- **buffer_overflow**: TP Rate là 0.833 và Precision là 0.806, cho thấy mô hình có một số lỗi trong việc nhận diện lớp này.
- **loadmodule**: TP Rate là 0.333 nhưng Precision là 1.000, cho thấy mô hình có thể dự đoán đúng khi nó thực sự dự đoán nhưng bỏ lỡ nhiều mẫu.

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. Ví dụ:

- 97263 mẫu của lớp "normal" được dự đoán chính xác là "normal".
- 5 mẫu của lớp "buffer_overflow" được dự đoán chính xác, nhưng có 25 mẫu khác của lớp này bị nhầm lẫn.

```
on Matrix ===

  b      c      d      e      f      g      h      i      j      k      l      m      n      o      p      q      r      s      t      u      v      w   <-- classified as
  2      0      0      1      0      0      1      0      0      1      1      0      0      0      1      0      0      0      0      7      0      0 |      a = normal.
 25      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0 |      b = buffer_over
  2      3      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0 |      c = loadmodule.
  0      0      2      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0 |      d = perl.
  0      0      0 107200      0      0      0      0      1      0      0      0      0      0      0      0      0      0      0      0      0      0 |      e = neptune.
  0      0      0      0 280790      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0 |      f = smurf.
  0      0      0      0      0     51      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0 |      g = guess_passw
  0      0      0      0      0      0    263      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0 |      h = pod.
  0      0      0      0      0      0      0    979      0      0      0      0      0      0      0      0      0      0      0      0      0      0 |      i = teardrop.
  0      0      0      1      0      0      0      0   1034      1      0      0      0      0      1      0      0      0      0      0      0      0 |      j = portsweep.
  0      0      0      0      0      0      0      0      1   1238      0      0      0      0      0      0      0      0      0      0      0      0 |      k = ipsweep.
  0      0      0      2      0      0      0      0      0      0     19      0      0      0      0      0      0      0      0      0      0      0 |      l = land.
  1      0      0      0      0      0      0      0      0      0      0      3      0      0      0      0      0      0      0      1      0      0 |      m = ftp_write.
  0      0      0      0      0      0      0      0      0      0      0      0   2202      0      0      0      0      0      0      0      0      0 |      n = back.
  0      0      0      0      0      0      0      0      0      0      0      0      0     10      0      0      0      0      0      0      0      0 |      o = imap.
  0      0      0      3      0      0      0      0      0      0      0      0      0      0   1573      0      0      0      0      0      0      0 |      p = satan.
  0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      3      0      0      0      0      0      0 |      q = phf.
  0      0      0      0      0      0      0      0      0      4      0      0      0      0      0      0    226      0      0      0      0      0 |      r = nmap.
  1      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      3      2      0      0      0 |      s = multihop.
  0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      3     15      0      0      0 |      t = warezmaster
  0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0   1014      0      0 |      u = warezclient
  0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0 |      v = spy.
  0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0 |      w = rootkit.
```