TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÁO CÁO MÔN:

GRADUATION RESEARCH II

PHÂN LOẠI XÂM NHẬP MẠNG VỚI CÁC MÔ HÌNH HỌC MÁY

Sinh viên thực hiện : Vũ Hồng Quang

Lớp : **ICT02 – K65**

Giáo viên hướng dẫn : TS. Trần Hải Anh

Hà Nội, tháng 1 năm 2024

MỤC LỤC

Table of Contents

LỜI NÓI ĐẦU

Việc ứng dụng máy tính và Tin học trong quản lý thông tin đã trở thành một xu hướng không thể phủ nhận, đặc biệt là trong thời đại hiện nay khi sự phát triển của công nghệ thông tin đang định hình và thay đổi mọi khía cạnh của xã hội. Từ những nước tiên tiến đến những quốc gia đang phát triển, nhu cầu áp dụng Tin học để xử lý thông tin trong các lĩnh vực quản lý ngày càng trở nên quan trọng và bức thiết.

Chính vì vậy, em đã quyết định chọn đề tài "Phân loại xâm nhập mạng với các mô hình học máy" với mục đích nghiên cứu và áp dụng các phương pháp học máy để cải thiện khả năng phát hiện và ngăn chặn xâm nhập mạng. Đề tài này không chỉ là một bước tiến quan trọng trong lĩnh vực an ninh mạng mà còn đặt ra tầm quan trọng trong việc bảo vệ thông tin và dữ liệu quan trọng trước những mối đe dọa ngày càng phức tạp.

Quá trình nghiên cứu của em xoay quanh việc phân tích và ứng dụng các thuật toán học máy như Gaussian Naive Bayes, Decision Tree, Random Forest và Logistic Regression để phân loại xâm nhập mạng trên ba tập dữ liệu chính: KDD1999, NSL-KDD và CICIDS2018.

Em đã bắt đầu với quá trình tiền xử lý dữ liệu, đảm bảo rằng dữ liệu là đủ đa dạng và sẵn sàng cho việc huấn luyện mô hình. Sau đó, em triển khai mỗi thuật toán học máy trên từng tập dữ liệu và đánh giá hiệu suất của chúng.

Trong môi trường mạng ngày nay, việc bảo vệ hệ thống khỏi các mối đe dọa xâm nhập đang trở nên ngày càng quan trọng. Các tổ chức và doanh nghiệp không chỉ phải đối mặt với những cuộc tấn công ngày càng tinh vi mà còn cần có khả năng đánh giá và ứng phó nhanh chóng. Đề tài này không chỉ mang tính nghiên cứu mà còn hướng tới ứng dụng thực tế, giúp cải thiện khả năng phòng thủ an ninh mạng và bảo vệ thông tin quan trọng.

Trong quá trình nghiên cứu và triển khai, em hy vọng có thể đóng góp một phần nhỏ vào sự phát triển của lĩnh vực an ninh mạng và học máy. Chân thành cảm ơn sự hỗ trợ và đồng hành của Thầy Trần Hải Anh, giáo viên hướng dẫn, cũng như sự nhiệt tình và hỗ trợ của các cán bộ và thầy cô giáo tại Trường Đại học Bách Khoa Hà Nội.

Hà Nội, tháng 1 năm 2024

Vũ Hồng Quang

Quang

CHƯƠNG 1. MÔ TẢ VỀ CÁC TẬP DỮ LIỆU

1.1. KDD1999

1.1.1. Mục Đích

Tập dữ liệu KDD1999 được tạo ra nhằm hỗ trợ cuộc thi "Knowledge Discovery in Databases Cup 1999" (KDD Cup 1999). Cuộc thi này hướng đến việc thách thức các đội tham gia trong việc phát hiện xâm nhập trong môi trường mạng. Được công bố từ những năm 90, KDD1999 đã trở thành một trong những bộ dữ liệu nổi tiếng trong lĩnh vực an ninh mạng.

1.1.2. Đặc Điểm

- Sự Đa Dạng: Bao gồm nhiều loại tấn công như DoS, R2L, U2R và probing, tạo ra một môi trường đa dạng để kiểm thử mô hình.

- Kích Thước: KDD1999 lớn với khoảng 4 triệu mẫu dữ liệu và 41 thuộc tính cho mỗi mẫu, cung cấp đủ lượng dữ liệu cho quá trình huấn luyện mô hình.

1.2. NSL-KDD (NSL-KDD Cup 99)

1.2.1. Mục đích

NSL-KDD là một phiên bản cải tiến của KDD1999, được tạo ra để khắc phục một số hạn chế của tập dữ liệu gốc. Cuộc thi NSL-KDD Cup 99 đặt ra thách thức nghiên cứu về phân loại xâm nhập mạng, với sự cải tiến về đa dạng và thực tế.

1.2.2. Đặc Điểm

- Cải Tiến Đa Dạng: NSL-KDD giữ lại các loại tấn công từ KDD1999 nhưng cải tiến cách mô phỏng chúng, tạo nên một tập dữ liệu thực tế hơn.

- Kích Thước: Bao gồm khoảng 125.000 mẫu dữ liệu huấn luyện và 22.544 mẫu kiểm thử, làm cho nó lớn hơn và đáng tin cậy hơn so với tập dữ liệu gốc.

1.3. CICIDS2018

1.3.1. Mục Đích

Tập dữ liệu CICIDS2018 được phát triển bởi Canadian Institute for Cybersecurity nhằm mục đích nghiên cứu về phân loại xâm nhập mạng và phát hiện tấn công trong môi trường thực tế.

1.3.2. Đặc Điểm

- Đa Dạng và Thực Tế: Chứa nhiều loại tấn công cụ thể và hoạt động mạng thực tế, tạo nên một tập dữ liệu đa dạng và đủ lớn để đánh giá hiệu suất của các mô hình.

- Kích Thước: Với hơn 16 triệu ghi chú, CICIDS2018 là một trong những tập dữ liệu lớn, hỗ trợ nghiên cứu với quy mô lớn.

CHƯƠNG 2. Tiền xử lý dữ liệu

2.1. KDD1999

2.1.1. Đọc dữ liệu từ file và Chọn Cột

Đầu tiên, em đã đọc danh sách các đặc trưng từ file kddcup.names, sau đó tạo danh sách tên cột dựa trên nội dung của file này. Sau khi đọc dữ liệu từ file kddcup.data, thêm cột 'Attack Type' dựa trên nhãn 'target'.

2.1.2. Xử lý Dữ liệu Thiếu

Kiểm tra và loại bỏ các cột chứa giá trị NaN từ dataframe.

```
df = df.dropna(axis='columns')# drop columns with NaN
```

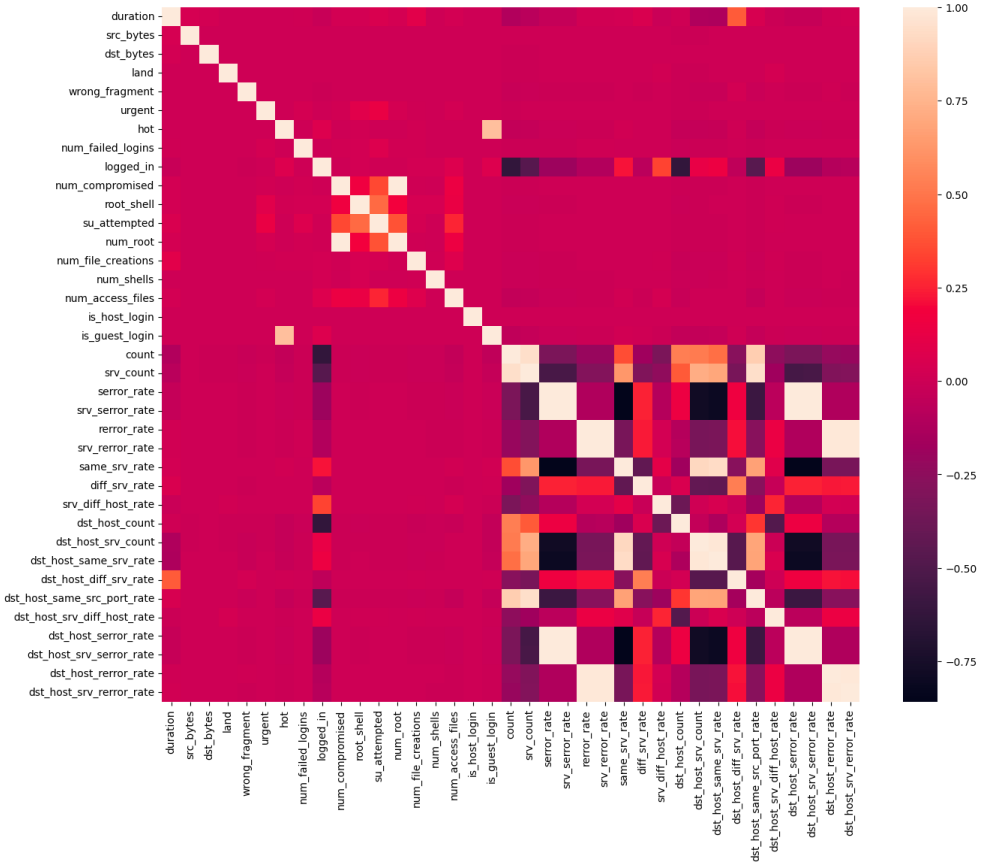
2.1.3. Chọn Các Cột Đa dạng

Em chỉ giữ lại các cột có nhiều hơn 1 giá trị duy nhất để giảm chiều dữ liệu và tăng tính đa dạng của dữ liệu.

```
df = df[[col for col in df if df[col].nunique() > 1]]
```

2.1.4. Kiểm Tra Tương Quan

Để hiểu mức độ tương quan giữa các đặc trưng số, em đã tính toán và vẽ biểu đồ heatmap cho ma trận tương quan.



2.1.5. Ánh Xạ Giá Trị

Em đã ánh xạ giá trị của cột 'protocol_type' và 'flag' sang dạng số để chuẩn bị cho mô hình học máy.

```
# flag feature mapping
fmap = {'SF':0, 'S0':1, 'REJ':2, 'RSTR':3, 'RSTO':4, 'SH':5, 'S1':6, 'S2':7, 'RSTOS0':8, 'S3':9, 'OTH':10}
df['flag'] = df['flag'].map(fmap)
```

2.1.6. Loại Bỏ Cột Irrelevant

Cuối cùng, quyết định loại bỏ cột 'service', vì nó không có ảnh hưởng đáng kể đến mô hình.

```
# Remove irrelevant features such as 'service' before modelling
df.drop('service', axis = 1, inplace = True)
```

2.2. NSL-KDD

Do tập dữ liệu này có cấu trúc tương tự KDD1999 nên được tiền xử lí theo như bên trên.

2.3. CICIDS2018

2.3.1. Đọc và Chuyển Đổi Kiểu Dữ Liệu

Với tập dữ liệu CICIDS2018, em đã thực hiện việc đọc dữ liệu từ file và chuyển đổi kiểu dữ liệu của các cột cần thiết như 'Dst Port', 'Protocol', 'Flow Duration', 'Tot Fwd Pkts', v.v.

```
df_dataset = pd.read_csv("/Users/user/Documents/Study/2023-1/gr2/cicids2018/03-02-2018.csv", low_memory=False)
```

```
df_dataset['Dst Port'] = df_dataset['Dst Port'].astype(int)
df_dataset['Protocol'] = df_dataset['Protocol'].astype(int)
df_dataset['Flow Duration'] = df_dataset['Flow Duration'].astype(int)
df_dataset['Tot Fwd Pkts'] = df_dataset['Tot Fwd Pkts'].astype(int)
df_dataset['Tot Bwd Pkts'] = df_dataset['Tot Bwd Pkts'].astype(int)
df_dataset['TotLen Fwd Pkts'] = df_dataset['TotLen Fwd Pkts'].astype(int)
```

2.3.2. Loại Bỏ Giá Trị Vô Hạn và Thay Thế Nhãn

Thay thế giá trị vô hạn và thay thế nhãn 'Bot' bằng 'Malicious'.

```
# replace +ve and -ve infinity with NaN
df_dataset.replace([np.inf, -np.inf], np.nan, inplace=True)
sampled_data.replace(to_replace=["Bot"], value="Malicious", inplace=True)
```

2.3.3. Loại Bỏ Hàng Trùng Lặp và Chuẩn Hóa Dữ Liệu

Loại bỏ các hàng trùng lặp và tiến hành chuẩn hóa các cột số trong tập huấn luyện và tập kiểm thử.

```
sampled_data.drop_duplicates(inplace = True)
```

```
min_max_scaler = MinMaxScaler().fit(train[numerical_columns])
train[numerical_columns] = min_max_scaler.transform(train[numerical_columns])
test[numerical_columns] = min_max_scaler.transform(test[numerical_columns])
```

2.3.4. Undersampling

Trong quá trình tiền xử lý dữ liệu tập CICIDS2018, để giải quyết vấn đề mất cân bằng giữa lớp Benign và Malicious, chúng tôi đã áp dụng kỹ thuật undersampling. Dữ liệu lớp Benign có một số lượng lớn (760,806 mẫu) so với lớp Malicious (282,310 mẫu). Để đảm bảo sự cân bằng trong việc huấn luyện mô hình, chúng tôi đã chọn ngẫu nhiên 282,310 mẫu từ lớp Benign và kết hợp chúng với toàn bộ lớp Malicious, tạo thành tập dữ liệu mới có sự cân bằng giữa hai lớp.

```
df1 = df[df["Label"] == "Benign"][:282310]
df2 = df[df["Label"] == "Malicious"][:282310]
df_equal = pd.concat([ df1,df2], axis =0)
```

Kỹ thuật này giúp mô hình học được từ đủ số lượng mẫu của cả hai lớp, đồng thời giảm thiểu ảnh hưởng của sự mất cân bằng đến quá trình huấn luyện. Điều này làm tăng khả năng mô hình phân loại hiệu quả giữa các lớp và cải thiện độ chính xác của mô hình trên dữ liệu không cân bằng.

2.3.5. Loại Bỏ Cột 'Timestamp'

Cuối cùng, loại bỏ cột 'Timestamp' không cần thiết.

```
train.drop(['Timestamp'], axis=1,inplace=True)
test.drop(['Timestamp'],axis=1,inplace=True)
```

Các bước tiền xử lý dữ liệu đã được thực hiện đều nhằm chuẩn bị dữ liệu cho quá trình huấn luyện mô hình và giúp cải thiện hiệu suất của mô hình trên các tập dữ liệu KDD1999, NSL-KDD và CICIDS2018.

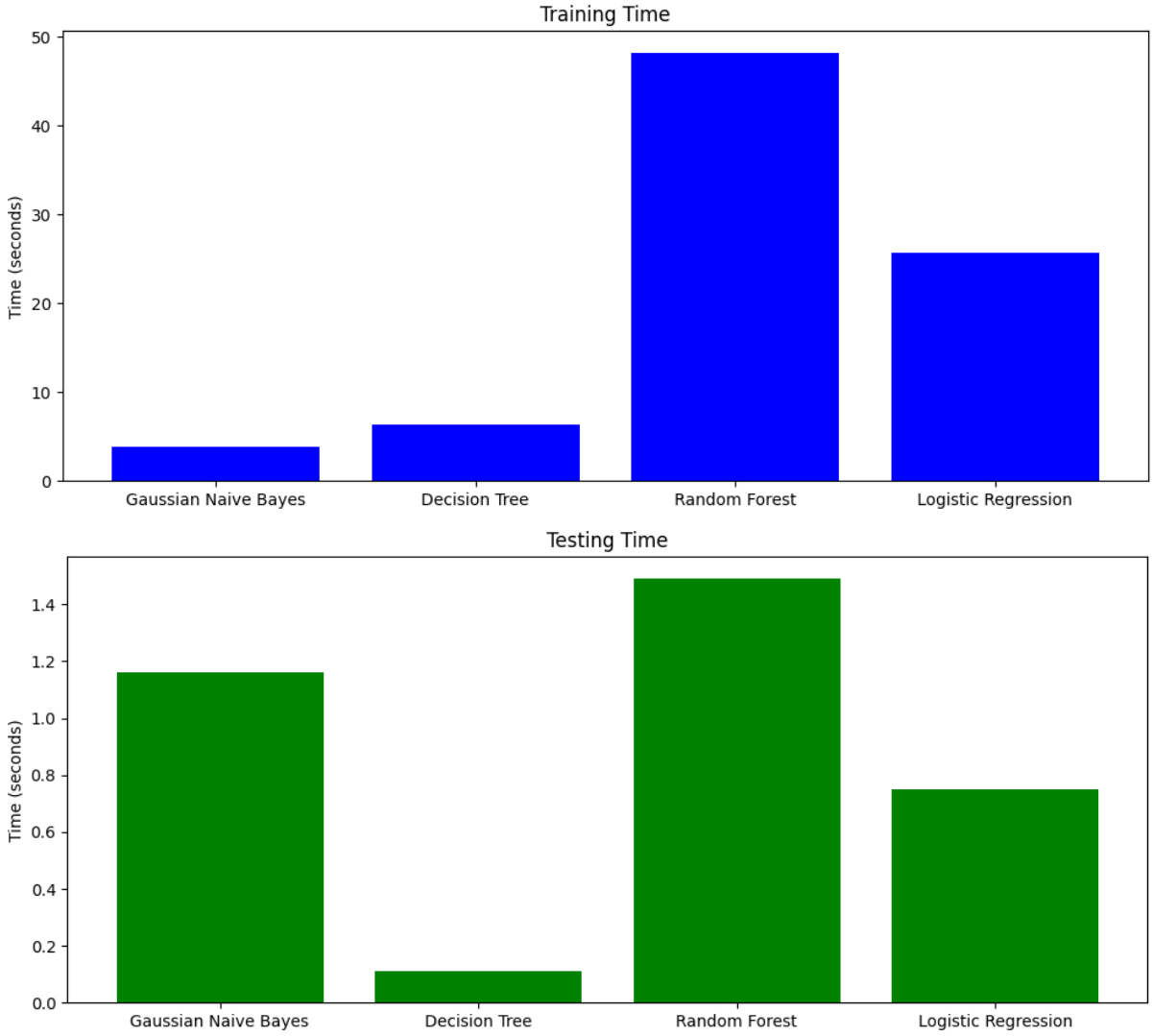
Ở đây, cả 3 tập dữ liệu được chia ra theo tỉ lệ train_test_split = 0.33

CHƯƠNG 3. Xây dựng mô hình và đánh giá

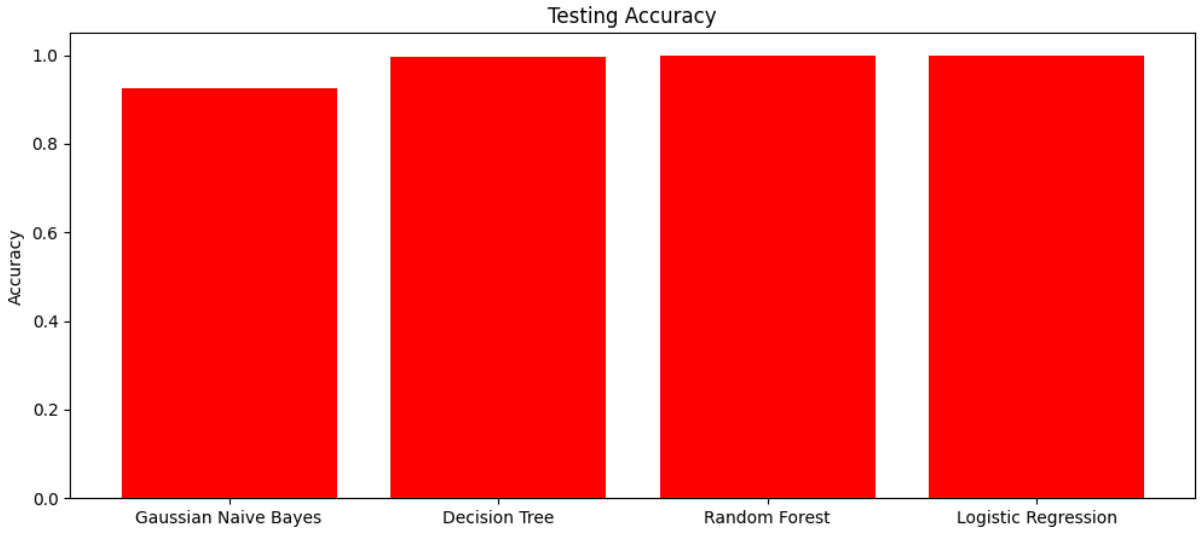
3.1. KDD-1999

3.1.1. Kết quả

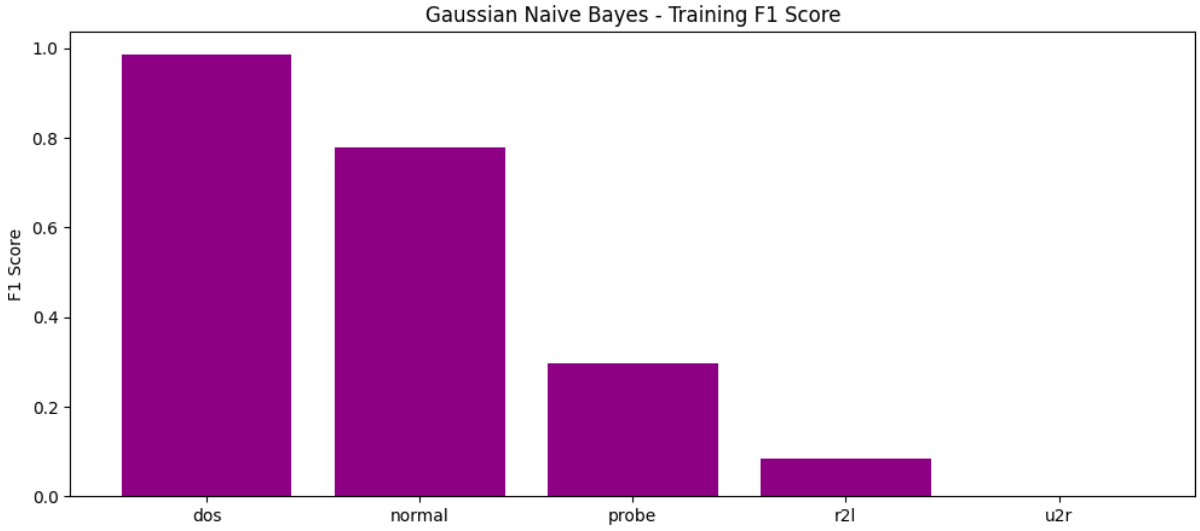
- Thời gian training và testing

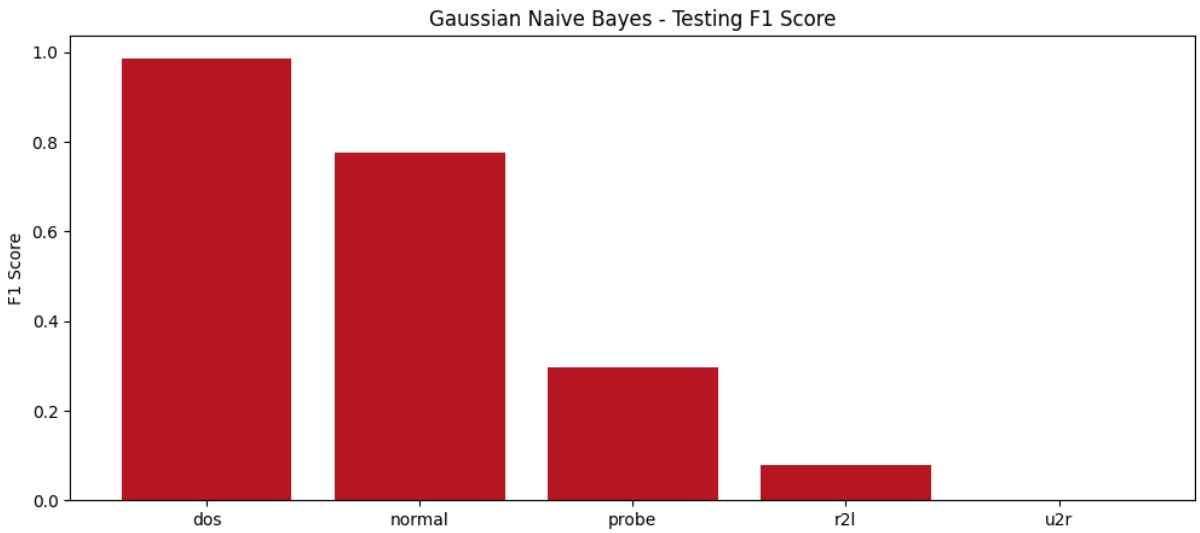


- Training và testing accuracy

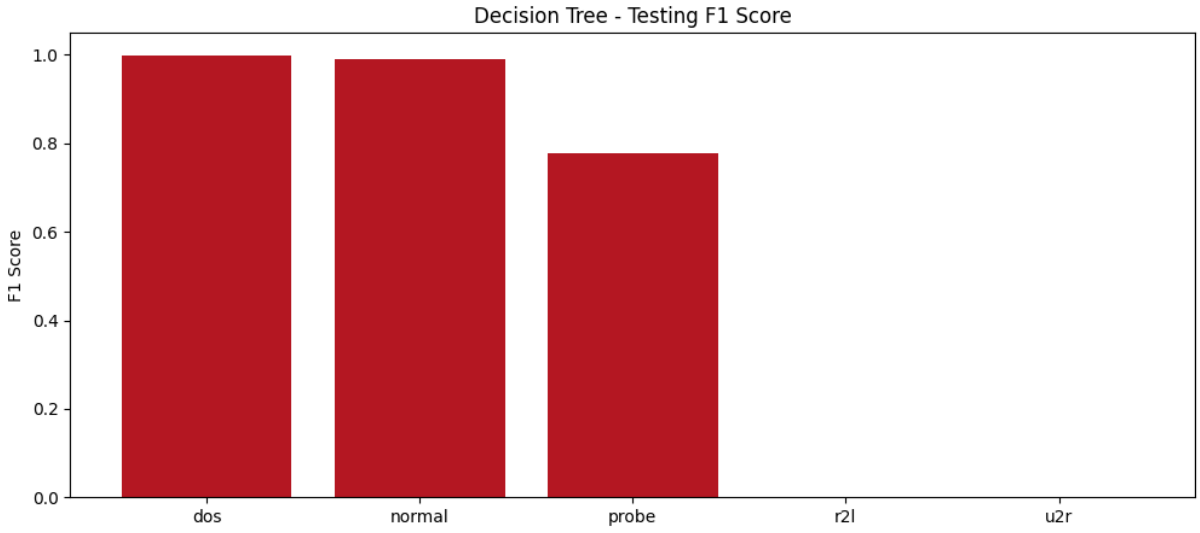


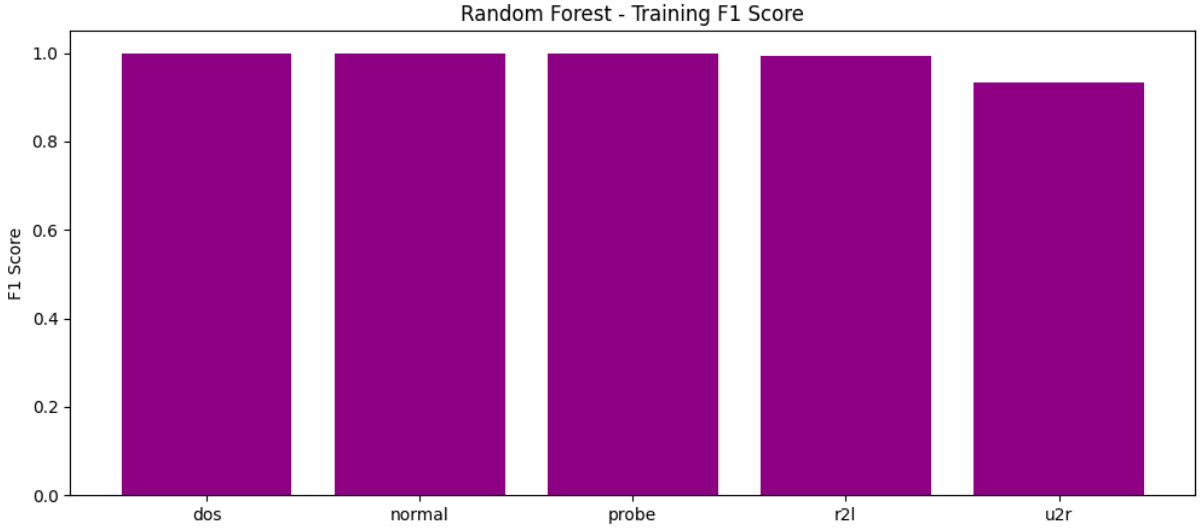
- F1 score:

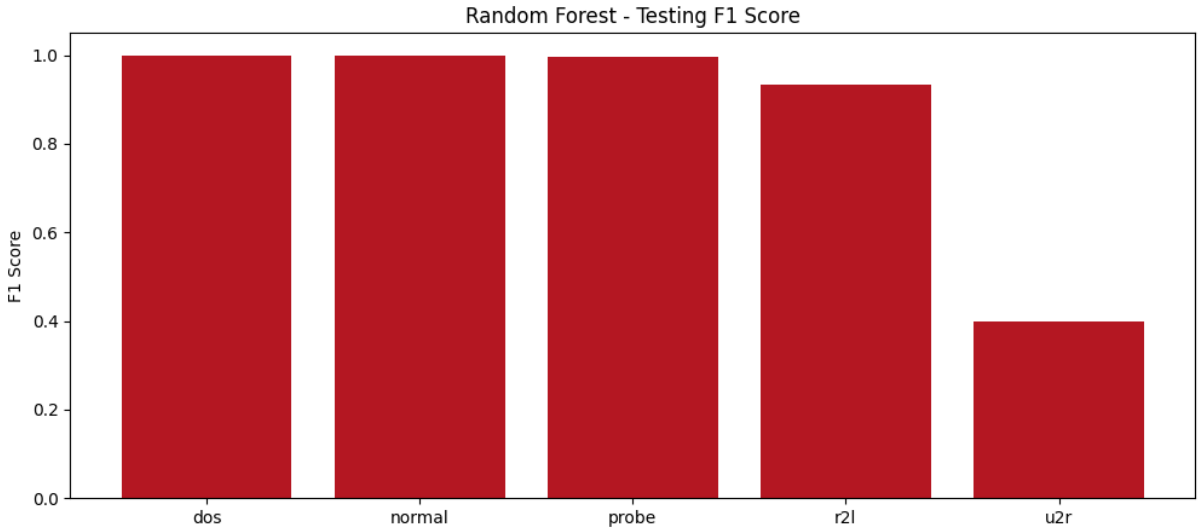


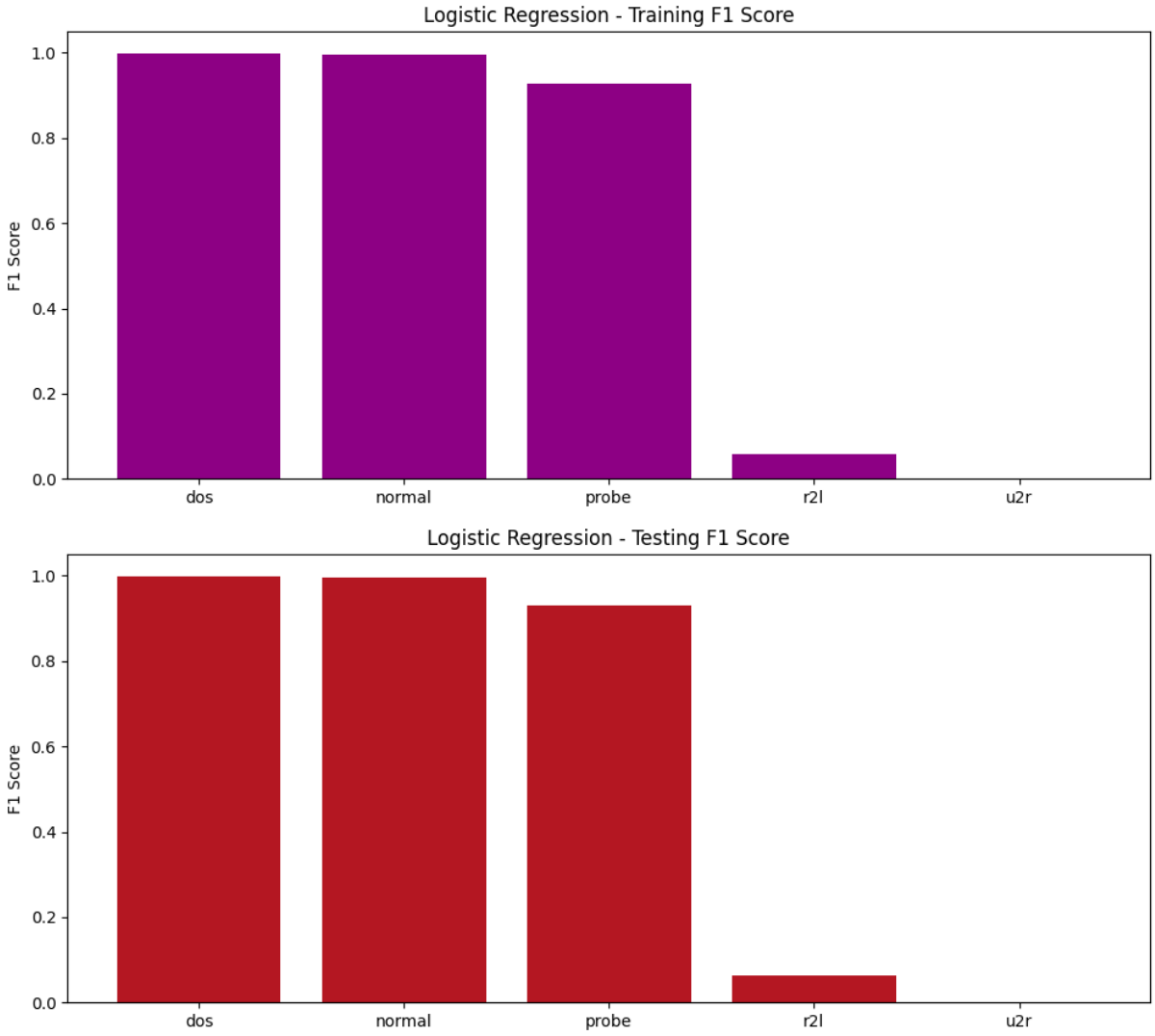

Gaussian Naive Bayes - Testing F1 Score
F1 Score
1.0
0.8
0.6
0.4
0.2
0.0
dos
normal
probe
r2l
u2r


Decision Tree - Training F1 Score
F1 Score
1.0
0.8
0.6
0.4
0.2
0.0
dos
normal
probe
r2l
u2r


Decision Tree - Testing F1 Score
F1 Score
1.0
0.8
0.6
0.4
0.2
0.0
dos
normal
probe
r2l
u2r


Random Forest - Training F1 Score
F1 Score
1.0
0.8
0.6
0.4
0.2
0.0
dos
normal
probe
r2l
u2r


Random Forest - Testing F1 Score
F1 Score
1.0
0.8
0.6
0.4
0.2
0.0
dos
normal
probe
r2l
u2r



3.1.2. Nhận xét

- Decision Tree và Random Forest:

- Ưu điểm: Đạt được độ chính xác cao trên cả tập huấn luyện và tập kiểm thử. F1 Score đối với mỗi lớp đều cao, đặc biệt là lớp "normal," cho thấy khả năng phân loại tốt cả hai loại kết quả (có và không xâm nhập).
- Nhược điểm: Thời gian huấn luyện và thử nghiệm lâu, đặc biệt là trong trường hợp của Random Forest.

- Logistic Regression:

- Ưu điểm: Độ chính xác cao và thời gian huấn luyện và thử nghiệm tương đối hợp lý.F1 Score cho lớp "normal" khá cao.
- Nhược điểm: F1 Score cho một số lớp minor (các loại tấn công) thấp, có thể cần cải thiện đối với các trường hợp xâm nhập.

- Gaussian Naive Bayes:

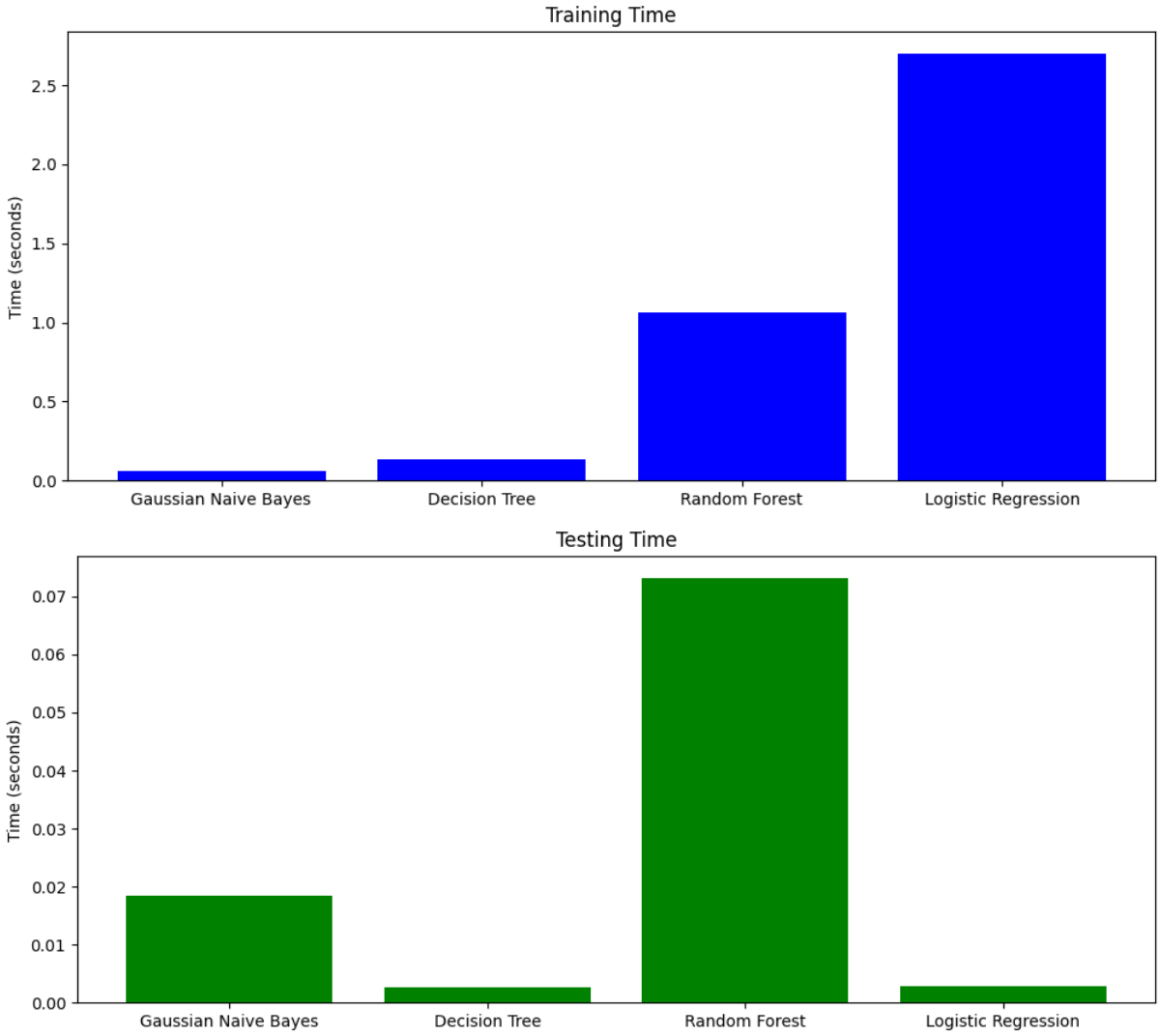
- Ưu điểm: Thời gian huấn luyện và thử nghiệm nhanh. Độ chính xác và F1 Score khá ổn định.
- Nhược điểm: F1 Score đối với một số loại tấn công thấp, đặc biệt là lớp "dos."

Tổng quát, các mô hình đều cho thấy hiệu suất tích cực, nhưng còn nhiều cơ hội để cải thiện, đặc biệt là đối với việc xử lý các loại tấn công ít phổ biến.

3.2. NSL-KDD

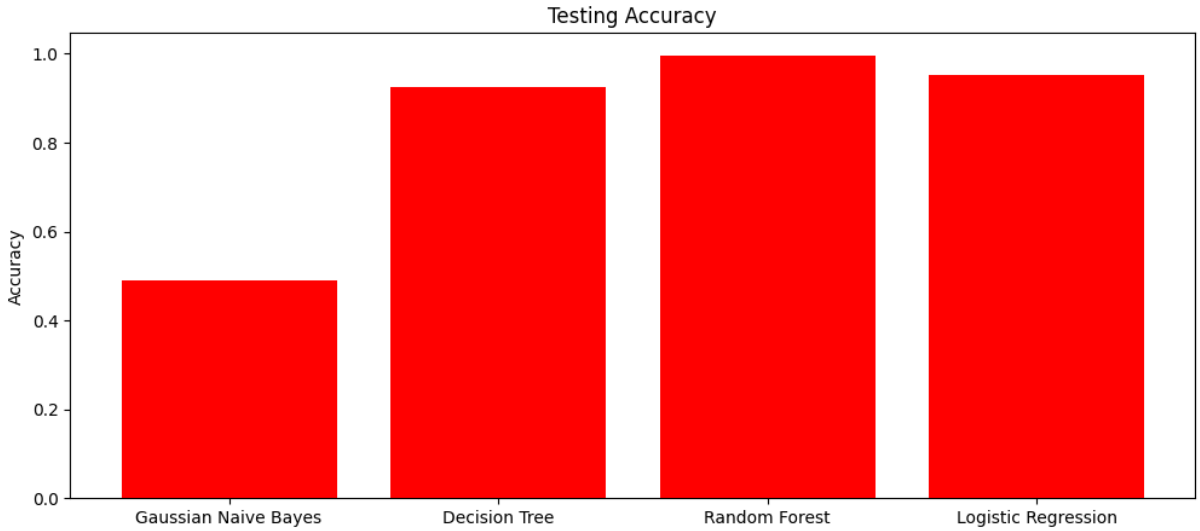
3.2.1. Kết quả

- Thời gian training và testing:

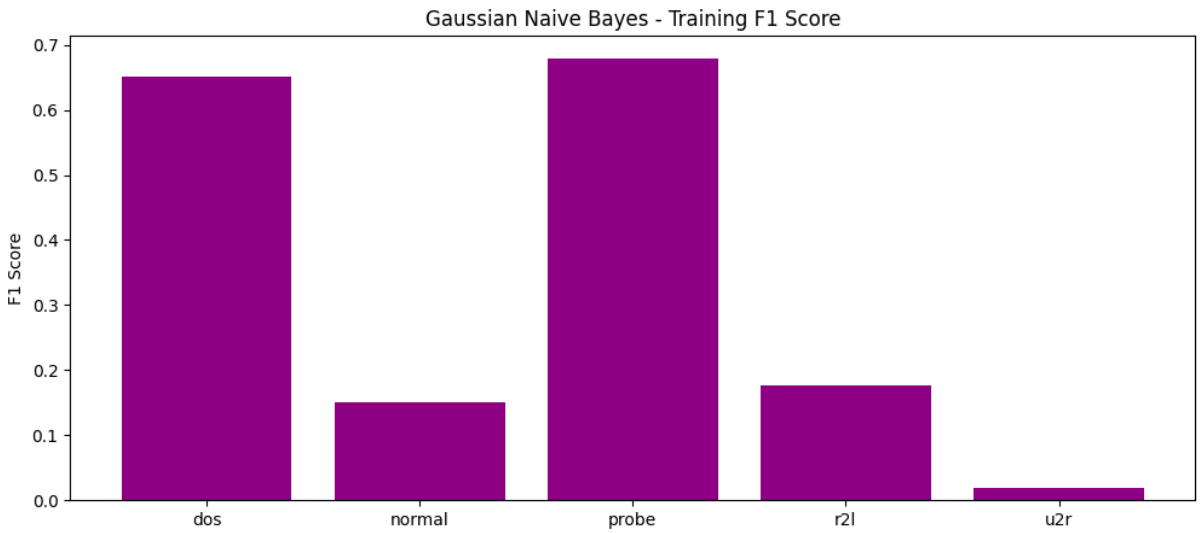


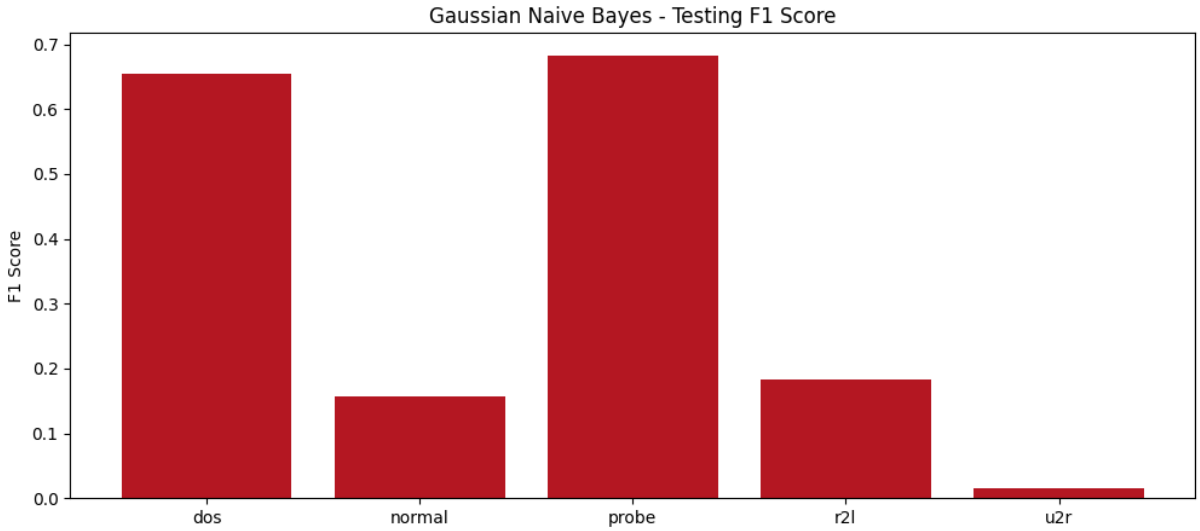
- Độ chính xác:


Training Accuracy
Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
Gaussian Naive Bayes
Decision Tree
Random Forest
Logistic Regression

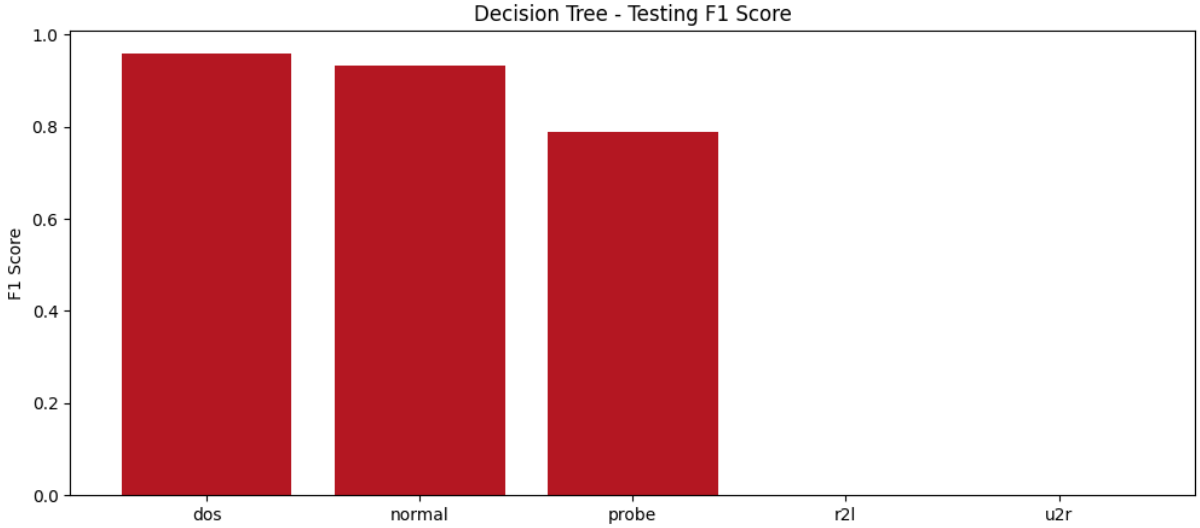

Testing Accuracy
Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
Gaussian Naive Bayes
Decision Tree
Random Forest
Logistic Regression

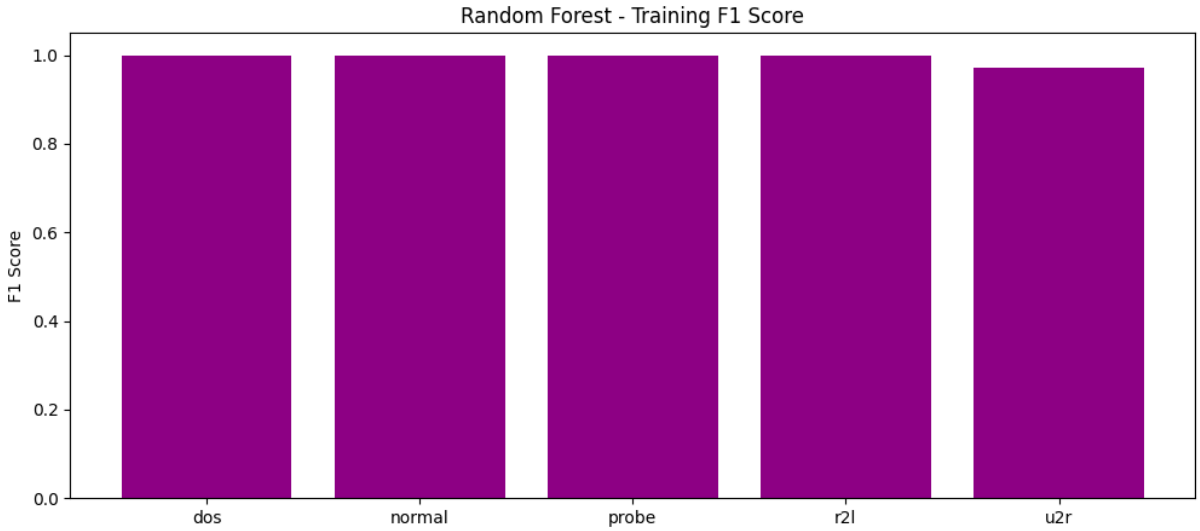
- F1 score:

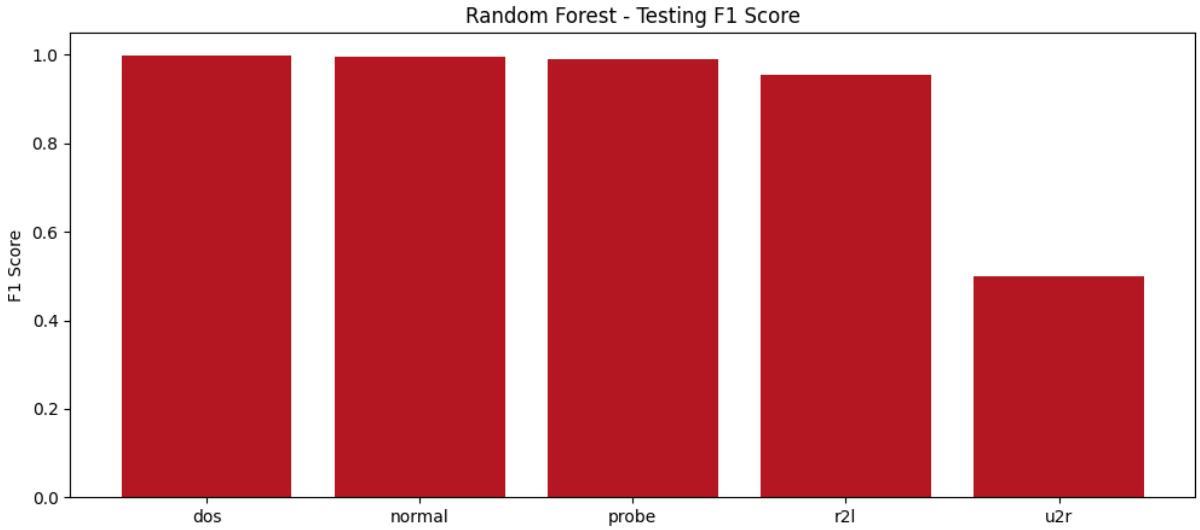

Gaussian Naive Bayes - Training F1 Score
F1 Score
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
dos
normal
probe
r2l
u2r

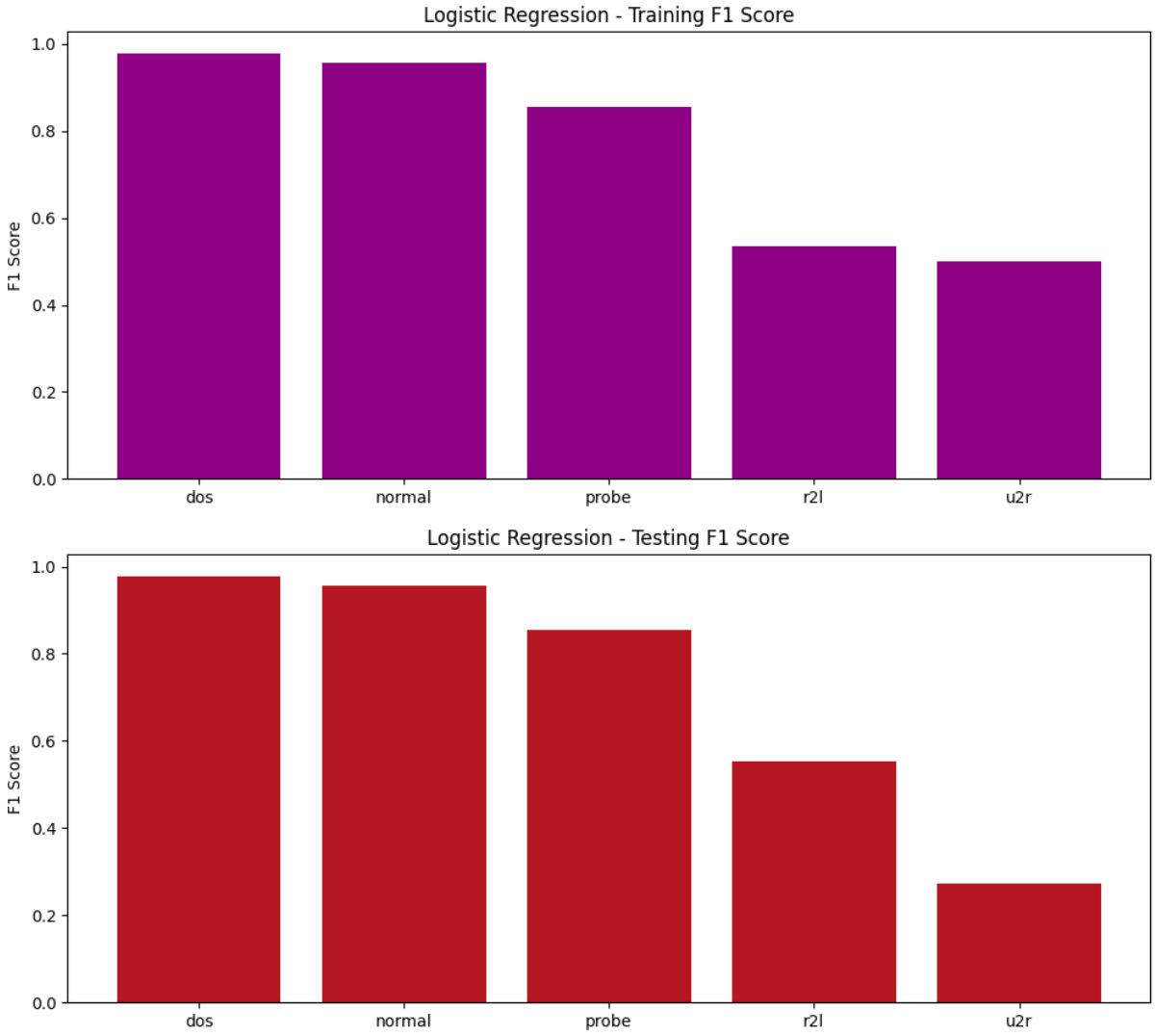

Gaussian Naive Bayes - Testing F1 Score
F1 Score
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
dos
normal
probe
r2l
u2r


Decision Tree - Training F1 Score
F1 Score
0.0
0.2
0.4
0.6
0.8
1.0
dos
normal
probe
r2l
u2r


Decision Tree - Testing F1 Score
F1 Score
1.0
0.8
0.6
0.4
0.2
0.0
dos
normal
probe
r2l
u2r


Random Forest - Training F1 Score
F1 Score
1.0
0.8
0.6
0.4
0.2
0.0
dos
normal
probe
r2l
u2r


Random Forest - Testing F1 Score
F1 Score
1.0
0.8
0.6
0.4
0.2
0.0
dos
normal
probe
r2l
u2r



3.2.2. Nhận xét

- Gaussian Naive Bayes:

- Ưu điểm: Thời gian huấn luyện và thử nghiệm rất nhanh. F1 Score khá đồng đều cho mỗi lớp, nhưng đối với lớp "normal" có F1 Score cao hơn so với các lớp tấn công.
- Nhược điểm: Độ chính xác thấp, đặc biệt là trên tập kiểm thử.

- Decision Tree:

- Ưu điểm: Thời gian huấn luyện và thử nghiệm nhanh. Độ chính xác và F1 Score đối với lớp "normal" khá cao.
- Nhược điểm: F1 Score thấp hoặc bằng 0 đối với một số lớp tấn công, có thể là do mô hình quá đơn giản.

- Random Forest:

- Ưu điểm: Đạt được độ chính xác và F1 Score rất cao trên cả tập huấn luyện và tập kiểm thử.
- Nhược điểm: Thời gian huấn luyện tăng so với các mô hình khác, nhưng vẫn chấp nhận được.

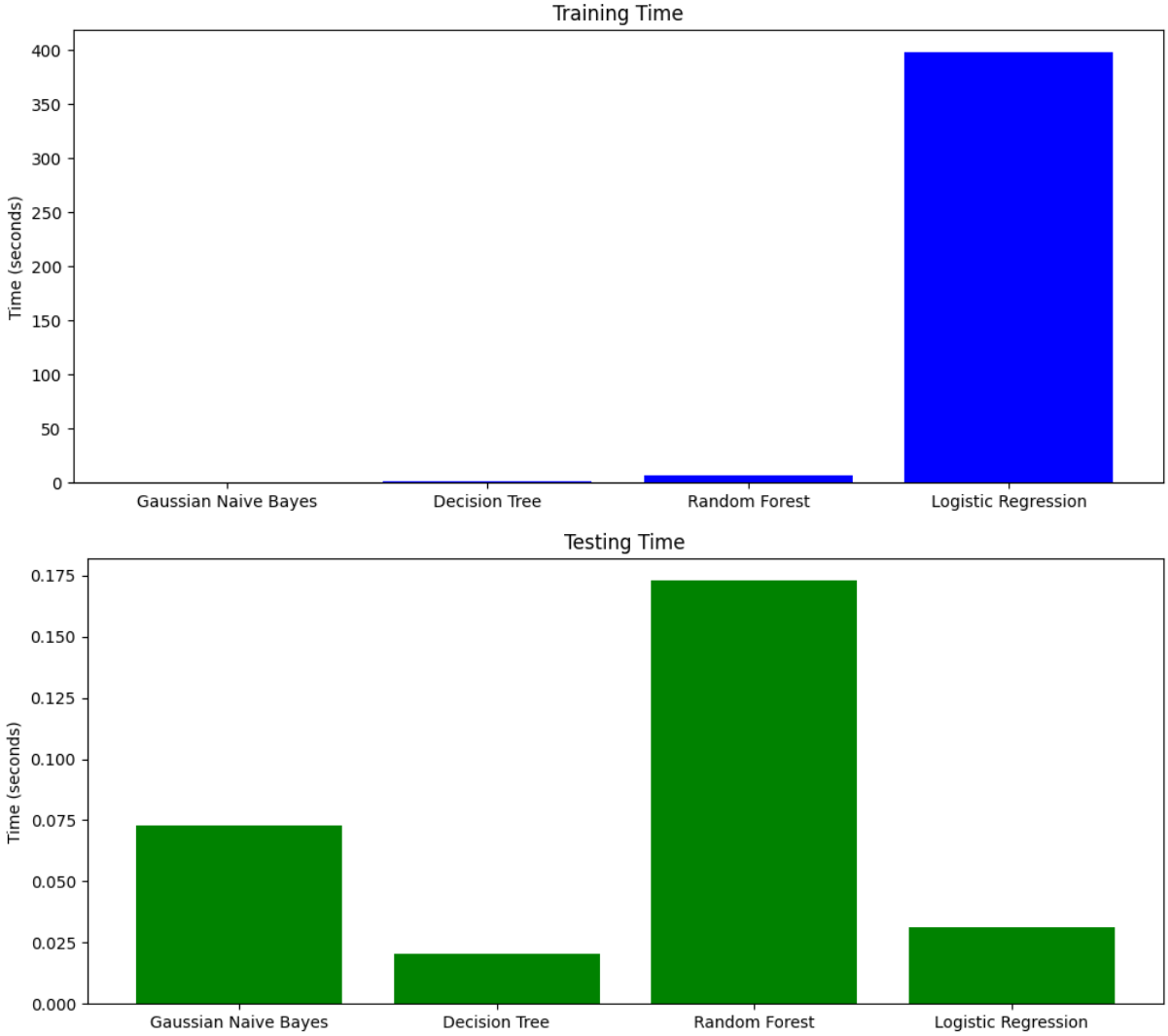
- Logistic Regression:
- Ưu điểm: Thời gian huấn luyện và thử nghiệm tương đối nhanh. Độ chính xác và F1 Score đối với lớp "normal" khá cao.
- Nhược điểm: F1 Score cho một số lớp tấn công thấp, có thể cần cải thiện.

Tổng quát, Random Forest là mô hình hiệu quả nhất với độ chính xác và F1 Score cao.

3.3. CICIDS2018

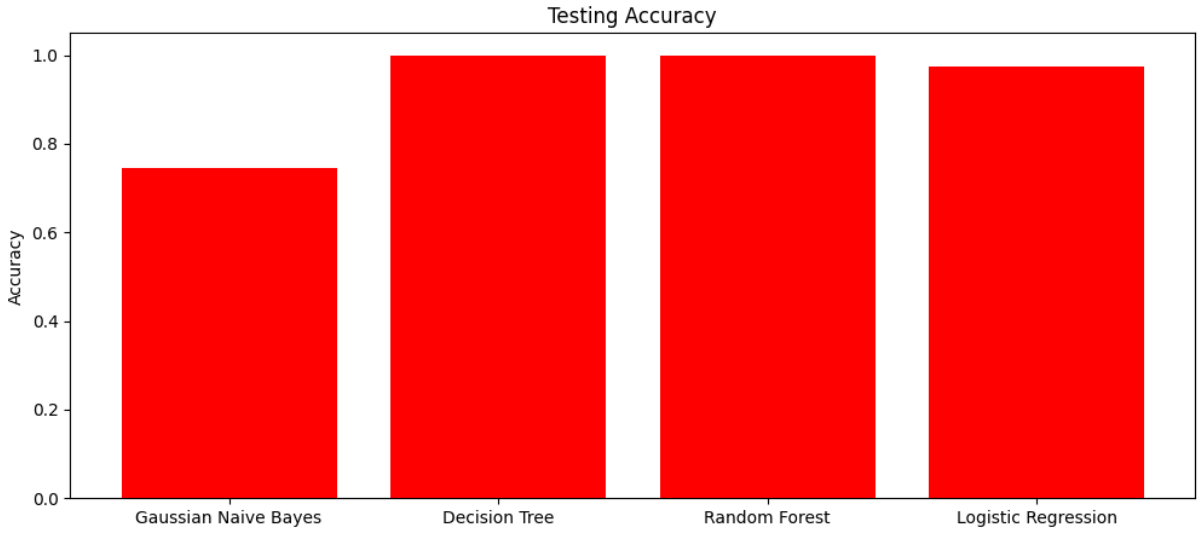
3.3.1. Kết quả

- Thời gian training và testing:

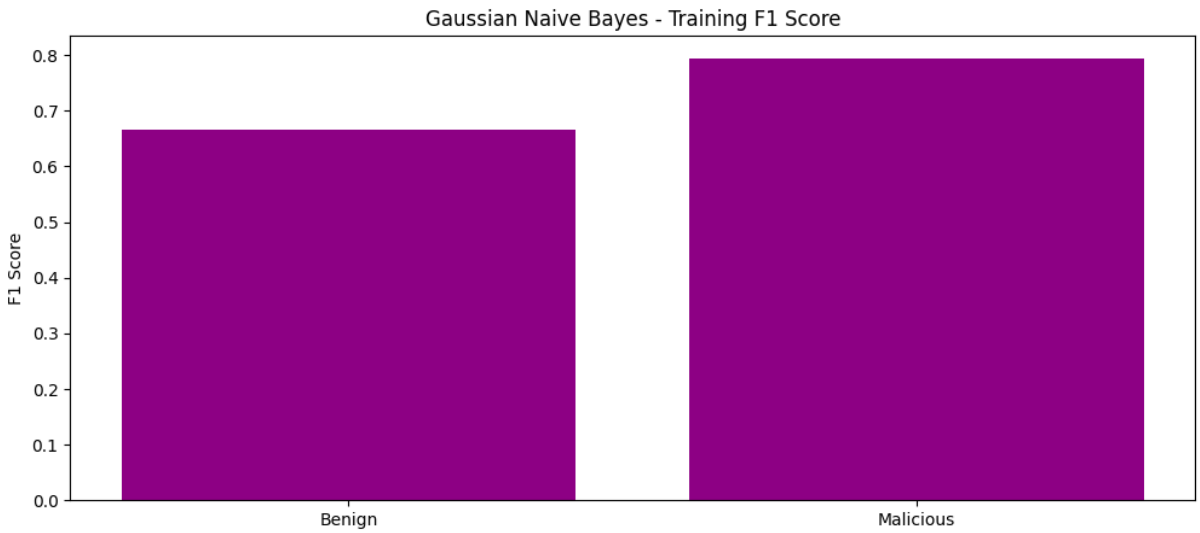


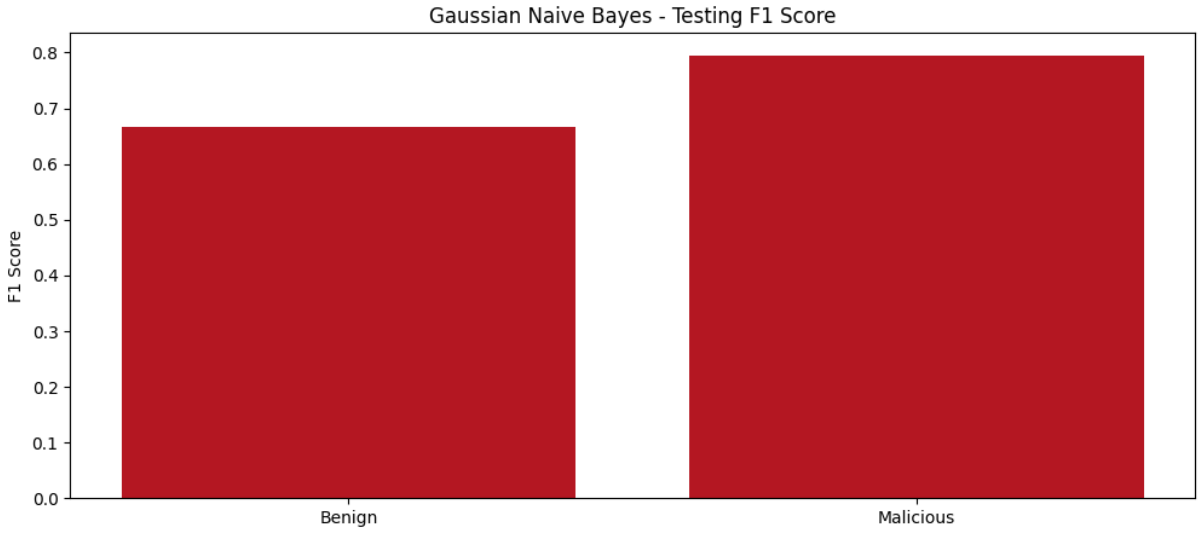
- Độ chính xác:


Training Accuracy
Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
Gaussian Naive Bayes
Decision Tree
Random Forest
Logistic Regression

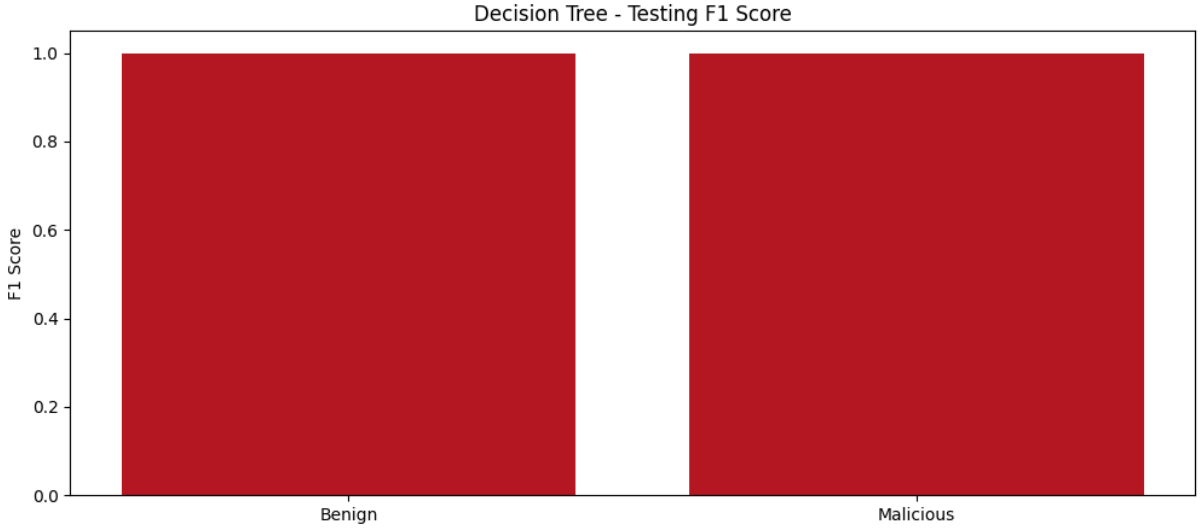

Testing Accuracy
Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
Gaussian Naive Bayes
Decision Tree
Random Forest
Logistic Regression

- F1 score:

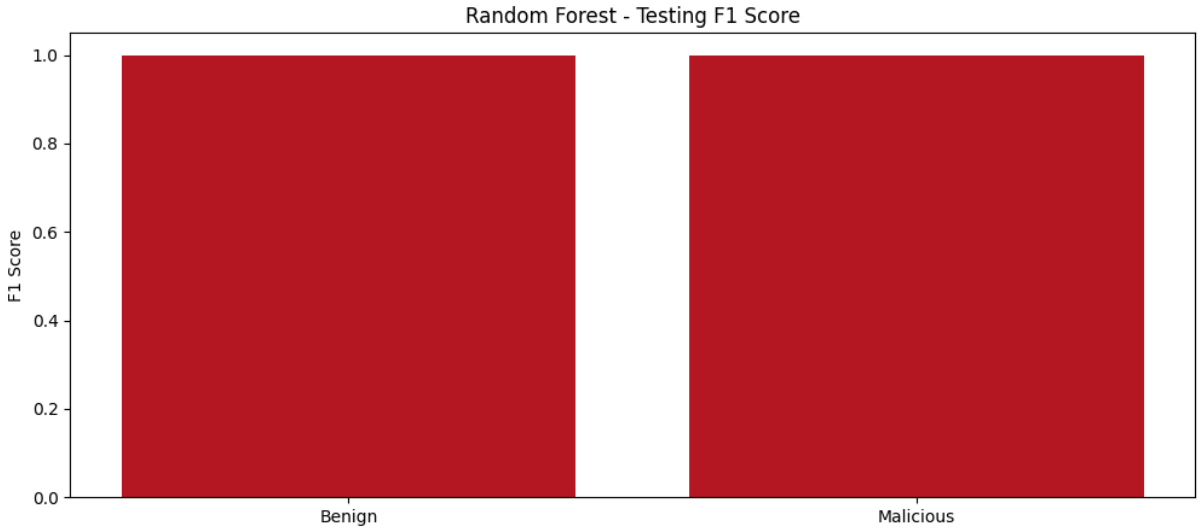

Gaussian Naive Bayes - Training F1 Score
F1 Score
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Benign
Malicious

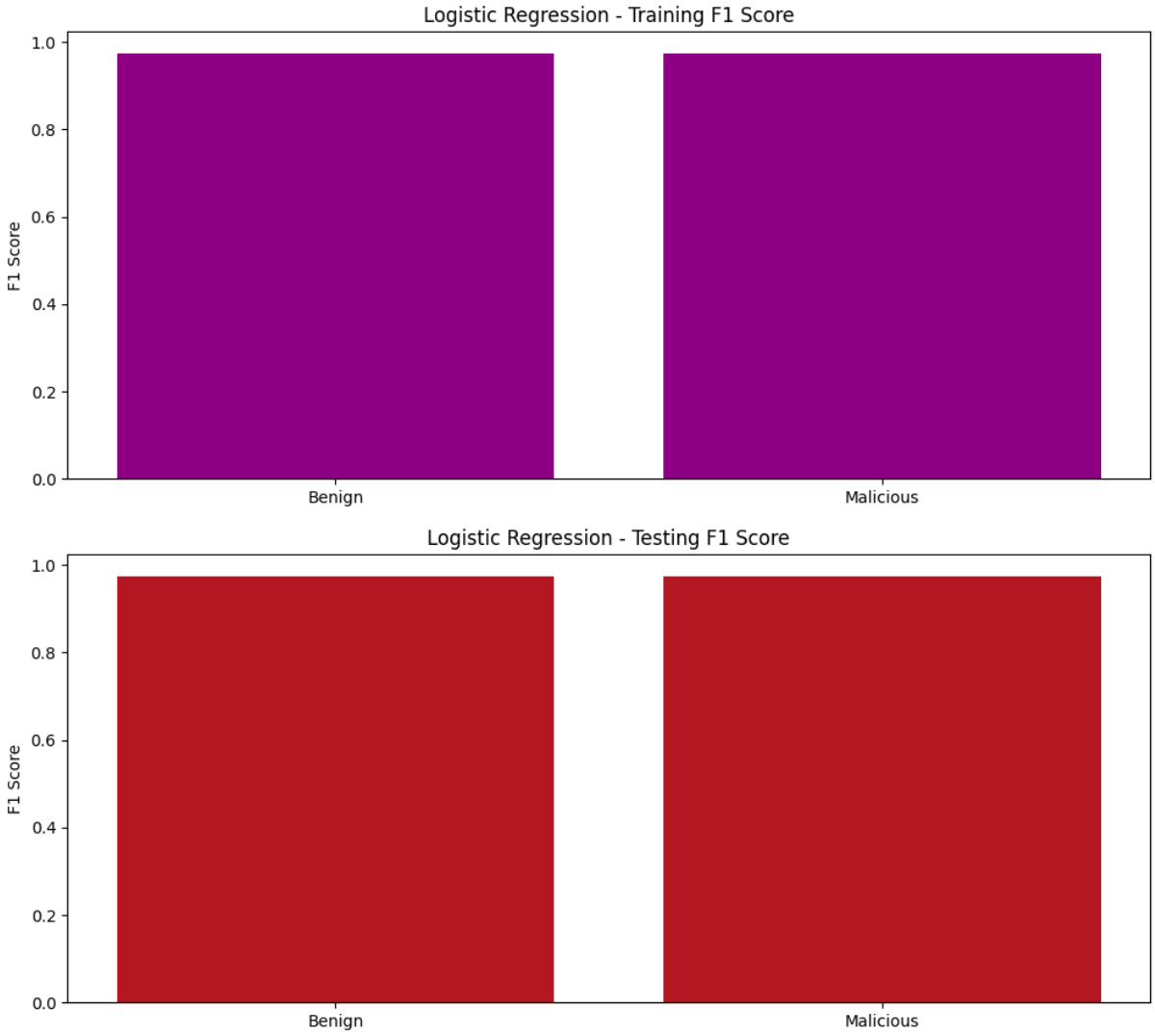

Gaussian Naive Bayes - Testing F1 Score
F1 Score
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Benign
Malicious


Decision Tree - Training F1 Score
F1 Score
0.0
0.2
0.4
0.6
0.8
1.0
Benign
Malicious


Decision Tree - Testing F1 Score
F1 Score
1.0
0.8
0.6
0.4
0.2
0.0
Benign
Malicious


Random Forest - Training F1 Score
F1 Score
1.0
0.8
0.6
0.4
0.2
0.0
Benign
Malicious


Random Forest - Testing F1 Score
F1 Score
1.0
0.8
0.6
0.4
0.2
0.0
Benign
Malicious



3.3.2. Nhận xét

- Gaussian Naive Bayes:
- Ưu điểm: Thời gian huấn luyện và thử nghiệm khá nhanh. Độ chính xác và F1 Score đối với cả hai lớp "Benign" và "Malicious" đều ổn định.
- Nhược điểm: Độ chính xác và F1 Score có vẻ thấp so với các mô hình phức tạp hơn.

- Decision Tree:
- Ưu điểm: Đạt được độ chính xác và F1 Score rất cao trên cả tập huấn luyện và tập kiểm thử.
- Nhược điểm: Có khả năng bị overfitting vì độ chính xác trên tập kiểm thử gần như hoàn hảo.

- Random Forest:
- Ưu điểm: Thời gian huấn luyện có vẻ lớn, nhưng đạt được độ chính xác và F1 Score rất cao trên cả tập huấn luyện và tập kiểm thử.
- Nhược điểm: Thời gian huấn luyện tăng so với các mô hình đơn giản hơn, nhưng vẫn chấp nhận được.

- Logistic Regression:
- Ưu điểm: Độ chính xác và F1 Score đối với cả hai lớp đều khá cao.
- Nhược điểm: Thời gian huấn luyện lớn hơn so với các mô hình đơn giản, có thể không phù hợp cho việc xử lý dữ liệu lớn.

Tổng quan: Cả ba mô hình Decision Tree, Random Forest và Logistic Regression đều có hiệu suất rất tốt trên tập kiểm thử sau khi áp dụng kỹ thuật Undersampling.
Mô hình Gaussian Naive Bayes có hiệu suất thấp hơn so với các mô hình khác, có thể do giả sử về sự độc lập giữa các đặc trưng không được đáp ứng tốt với dữ liệu thực tế.
Việc sử dụng F1 Score sẽ là hữu ích khi quan tâm đến cả Precision và Recall, đặc biệt khi có sự mất cân bằng giữa các lớp.

CHƯƠNG 4. Kết luận và định hướng tương lai

Trong quá trình nghiên cứu và xây dựng mô hình cho các tập dữ liệu bảo mật, em đã thực hiện nhiều bước tiền xử lý dữ liệu và xây dựng các mô hình học máy để phân loại các dạng tấn công và thông thường. Dưới đây là các điểm chính được kết luận từ nghiên cứu này:

- **Tiền xử lí dữ liệu:**

- Tập dữ liệu KDDCup 1999: Sử dụng phương pháp undersampling để giảm số lượng mẫu của lớp đa số và cân bằng dữ liệu. Điều này giúp mô hình học máy có khả năng học tốt hơn trên cả hai lớp.

- Tập dữ liệu NSL-KDD và CICIDS2018: Thực hiện chuẩn hóa dữ liệu và xử lý giá trị thiếu để đảm bảo rằng dữ liệu đầu vào cho mô hình là chất lượng.

- **Xây dựng mô hình:**

- Mô hình học máy: Em đã triển khai và đánh giá nhiều mô hình học máy khác nhau như Naive Bayes, Decision Tree, Random Forest và Logistic Regression.

- Đánh giá mô hình: Đối với mỗi mô hình, em đã đánh giá hiệu suất sử dụng các chỉ số như độ chính xác (accuracy) và F1-score trên cả tập huấn luyện và tập kiểm thử.

- **Kết quả:**

- Tập dữ liệu KDDCup 1999: Mô hình đạt được hiệu suất cao trên tập kiểm thử sau khi áp dụng kỹ thuật undersampling, đặc biệt là Decision Tree và Random Forest.

- Tập dữ liệu NSL-KDD và CICIDS2018: Mô hình cũng đạt được độ chính xác cao, nhưng F1-score cung cấp cái nhìn toàn diện hơn về hiệu suất, đặc biệt là đối với lớp thiểu số.

- **Hướng phát triển tương lai:**

- Tối ưu hóa thêm: Có thể tiếp tục tối ưu hóa các tham số của mô hình để cải thiện hiệu suất dự đoán.

- Sử dụng mô hình học sâu: Thử nghiệm với các mô hình học sâu như mạng nơ-ron để xem liệu chúng có thể cung cấp hiệu suất tốt hơn không.

- Mở rộng đối tượng nghiên cứu: Nghiên cứu và triển khai mô hình trên các tập dữ liệu bảo mật khác để xem xét khả năng tổng quát hóa của mô hình.

TÀI LIỆU THAM KHẢO

[1] https://archive.ics.uci.edu/dataset/130/kdd+cup+1999+data
[2] https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[3] https://www.kaggle.com/datasets/galaxyh/kdd-cup-1999-data
[4] https://www.unb.ca/cic/datasets/nsl.html
[5] https://inseclab.uit.edu.vn/nsl-kdd-goc-nhin-chi-tiet-ve-tap-du-lieu-huan-luyen-cho-cac-ids/
[6] https://www.kaggle.com/datasets/hassan06/nslkdd/code
[7] https://www.unb.ca/cic/datasets/ids-2018.html
[8] https://www.tensorflow.org/