BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CMC

BÁO CÁO BÀI TẬP LỚN

HỌC PHẦN: TRÍ TUỆ NHÂN TẠO

Tên đề tài: **Ứng dụng học máy phân loại và phát hiện xâm nhập mạng doanh nghiệp**

Nhóm sinh viên: **Bùi Khắc Khánh – BIT220084**

Phạm Quốc An – BIT220006

Lớp: **22IT3**

Hà Nội, tháng 5 - 2024

MỤC LỤC

MỞ ĐẦU

Trí tuệ nhân tạo đang là một trong những lĩnh vực được quan tâm mạnh mẽ trong nhiều ngành công nghiệp khác nhau. Trong lĩnh vực An ninh mạng, một trong những thách thức lớn mà các doanh nghiệp phải đối mặt là phát hiện và ngăn chặn các cuộc tấn công mạng, từ các cuộc tấn công DDoS đến các cuộc tấn công tinh vi như xâm nhập vào hệ thống, trộm cắp dữ liệu.

Để đối phó với những mối đe dọa này, các hệ thống IDS (Intrusion Detection System) giúp phát hiện sớm các hành vi xâm nhập vào hệ thống mạng phát hiện những hành vi này trước khi chúng gây ra hậu quả nghiêm trọng.

Nội dung chính của báo cáo gồm các mục sau đây:

- Phần 1: Tổng quan về vấn đề: Giới thiệu về mục tiêu của hệ thống IDS trong bảo vệ mạng doanh nghiệp, ngữ cảnh ứng dụng của IDS trong thực tế. tổng quan về tập dữ liệu, bao gồm các dạng tấn công phổ biến và các nguồn tài liệu tham khảo.
- Phần 2: Phương pháp học máy và mô hình đề xuất: Giới thiệu về các phương pháp học máy phổ biến được sử dụng trong phát hiện xâm nhập, bao gồm các thuật toán như Support Vector Machine (SVM) , Naive Bayes đề xuất sử dụng một mô hình cụ thể, có thể là một biến thể của một thuật toán học máy hiệu quả, để giải quyết bài toán phân loại và phát hiện xâm nhập mạng.
- Phần 3. Thực nghiệm và kết quả: Thực hiện thử nghiệm với các tập dữ liệu mô phỏng hoặc thực tế để đánh giá hiệu suất của hệ thống IDS, Phân tích kết quả, đưa ra nhận xét và đánh giá về hiệu suất của mô hình đề xuất so với các phương pháp khác.

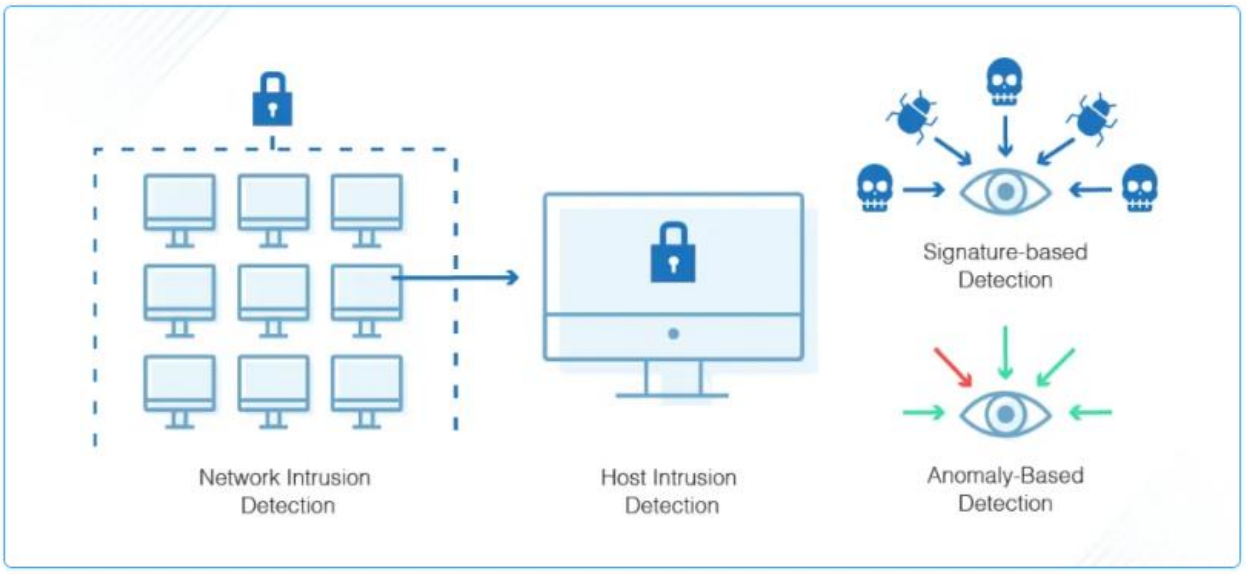
Báo cáo sẽ kết thúc bằng một phần kết luận, trong đó sẽ đánh giá các công việc đã thực hiện được, tổng kết về kết quả đạt được, và đề xuất một số hướng tìm hiểu và nghiên cứu tiếp theo trong lĩnh vực phát hiện và phân loại xâm nhập mạng doanh nghiệp.

PHẦN I. TỔNG QUAN

1.1. Giới thiệu bài toán

Trong môi trường kinh doanh ngày nay, việc bảo vệ mạng máy tính của doanh nghiệp trở nên cực kỳ quan trọng để đảm bảo an toàn thông tin và dữ liệu của khách hàng. Khía cạnh quan trọng trong việc này là Hệ thống Phát hiện Xâm nhập (IDS) công nghệ được thiết kế để phát hiện các hoạt động xâm nhập mạng không mong muốn.

IDS, được xem như là "mắt" của hệ thống mạng, chịu trách nhiệm theo dõi và phát hiện các mẫu hoạt động không bình thường hoặc có hại trong dữ liệu mạng. Một IDS hiệu quả sẽ sử dụng các phương pháp phân tích dữ liệu mạng để nhận biết các hành vi xâm nhập, bao gồm phân tích gói tin, dòng lưu lượng, và hành vi người dùng.



Hình 1: Hệ thống IDS

Các tính năng quan trọng nhất của IDS bao gồm:

- Về giám sát traffic mạng và các hoạt động khả nghi: IDS theo dõi và phân tích lưu lượng mạng, xác định các hoạt động không bình thường và có thể là mối đe dọa đến hệ thống mạng. Bằng trí tuệ nhân tạo có thể học từ dữ liệu lưu lượng mạng lịch sử để nhận diện các dấu hiệu tiềm ẩn của tấn công mạng, bao gồm các cuộc tấn công phân tán, tấn công từ chối dịch vụ (DoS), và phần mềm độc hại.

- Về học từ kinh nghiệm và cải thiện liên tục: Quan trọng nhất, hệ thống IDS được cải thiện bằng trí tuệ nhân tạo có thể học từ kinh nghiệm và cải thiện liên tục qua thời gian. Bằng cách thu thập dữ liệu về các mối đe dọa mạng và kết quả của các biện pháp phòng

chống, AI có thể cung cấp các thông tin quý báu để nâng cao hiệu suất và hiệu quả của hệ thống.

Sự phát triển của trí tuệ nhân tạo, đặc biệt là các phương pháp học máy và học sâu, đã tạo ra cơ hội mới để cải thiện hiệu suất của IDS. Trí tuệ nhân tạo có thể được sử dụng để phát hiện các mẫu hoạt động xâm nhập mạng phức tạp hơn và tinh vi hơn, đồng thời cung cấp các giải pháp ngăn chặn tự động và linh hoạt hơn.

Với các phân tích ở trên, nhóm thực hiện lựa chọn chủ đề Nghiên cứu về học máy cho xử lý phát hiện và phân loại xâm nhập mạng ứng dụng trong doanh nghiệp .

1.2. Dữ liệu

Trong bài toán này, nhóm thực hiện sử dụng các tập dữ liệu KDD Cup 1999:

Kích thước: Tập dữ liệu gốc KDD Cup 1999 chứa khoảng 4.9 triệu mẫu. Tuy nhiên, trong dự án này, sử dụng phiên bản thu nhỏ của tập dữ liệu, tập dữ liệu có sẵn tại https://kdd.ics.uci.edu/databases/kddcup99/kddcup.data_5_percent.gz, chỉ chứa khoảng 0.2 triệu mẫu.

Nguồn gốc: Tập dữ liệu KDD Cup 1999 được tạo ra để phát triển và kiểm thử các phương pháp phát hiện xâm nhập mạng. Dữ liệu được thu thập từ hệ thống mạng thực tế và chứa các giao thức mạng khác nhau.

Nhận xét về tập dữ liệu: Tập dữ liệu KDD Cup 1999 có kích thước lớn và đa dạng, với nhiều loại các cuộc tấn công và giao thức mạng khác nhau.

1.3. Kết luận

Trong quá trình nghiên cứu và sử dụng các tập dữ liệu từ KDD Cup 1999, ta nhận thấy rằng các tập dữ liệu này rất hữu ích để nghiên cứu về phát hiện xâm nhập mạng. Tuy nhiên, cần phải lưu ý rằng các tập dữ liệu này không phản ánh hoàn toàn các tình huống mạng thực tế hiện nay do thời gian thu thập và sự phát triển của công nghệ mạng. Mặc dù vậy, chúng vẫn cung cấp một bức tranh tổng quan về các loại tấn công và hoạt động đáng ngờ trong môi trường mạng.

PHẦN II. PHƯƠNG PHÁP THỰC HIỆN

1. **Tổng quan**

Hiện nay học máy là một trong những công cụ phổ biến giải quyết bài toán phân loại dữ liệu. Trong học máy có nhiều loại mô hình học máy khác nhau , tuy nhiên nhóm thực hiện lựa chọn mô hình học có giám sát SVM (Suport Vector Machine), Navie Bayes cho bài toán của mình.

1.1 Giới thiệu kiến trúc cơ bản :

SVM (Suport Vector Machine) là một thuật toán học máy phổ biến được sử dụng cho cả bài toán phân loại và hồi quy. Ý tưởng cơ bản của SVM là tìm ra một siêu mặt phẳng (hyperplane) tốt nhất để phân chia dữ liệu thành các lớp khác nhau. SVM cố gắng tối đa hóa khoảng cách giữa các điểm dữ liệu gần siêu mặt phẳng (các điểm gọi là support vectors) từ các lớp khác nhau.

Naive Bayes là một mô hình phân loại dựa trên nguyên tắc của định lý Bayes. Mặc dù đơn giản, nhưng Naive Bayes thường cho kết quả tốt đối với các bài toán phân loại văn bản, email spam, và nhiều lĩnh vực khác. Thuật toán Naive Bayes giả định rằng các đặc trưng là độc lập với nhau, mặc dù trong thực tế không phải lúc nào cũng đúng.

1.2 Quy trình huấn luyện

Quá trình huấn luyện SVM bao gồm việc tìm ra siêu mặt phẳng tối ưu phân chia các lớp dữ liệu bằng cách sử dụng các điểm dữ liệu hỗ trợ (support vectors).

Quá trình huấn luyện Naive Bayes đơn giản hóa là tính toán xác suất có điều kiện cho mỗi lớp dựa trên các đặc trưng của dữ liệu huấn luyện.

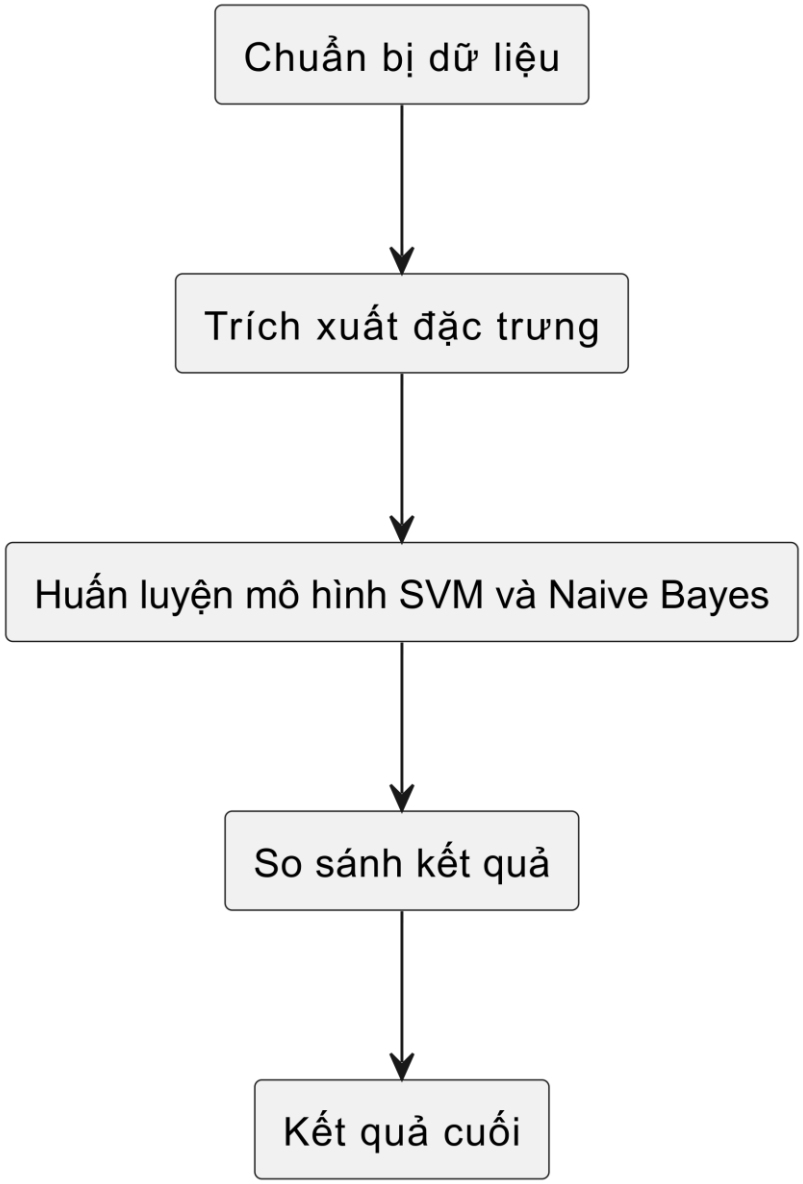
1.3 Các tham số

SVM có nhiều tham số quan trọng như kernel, độ rộng của vùng hỗ trợ (margin), và siêu tham số C (điều chỉnh độ quan trọng của việc phân loại sai sót).

Naive Bayes thường không có nhiều tham số để điều chỉnh. Tuy nhiên, trong một số trường hợp, bạn có thể cần điều chỉnh các tham số smoothing như alpha.

2. **Mô hình học để xử lý bài toán**

Trong đề tài này, nhóm sẽ lựa chọn kiến trúc mô hình học trực quan như sau:



Hình 2: Mô hình học máy SVM và Naive Bayes trực quan

Chuẩn bị dữ liệu tải dữ liệu từ nguồn cung cấp thu thập dữ liệu từ các nguồn đáng tin cậy hoặc các tập dữ liệu công cộng về mạng máy tính và các loại tấn công mạng. Loại bỏ hoặc biến đổi các đặc trưng không cần thiết, xử lý các giá trị thiếu, và mã hóa các đặc trưng dạng văn bản thành dạng số. Phân chia dữ liệu thành tập huấn luyện và tập kiểm tra để đảm bảo tính công bằng và độc lập trong việc đánh giá mô hình.

```
print('Kích thước của tập huấn luyện:', df.shape)
print('Kích thước của tập kiểm tra:', df_test.shape)
```

```
df.head(5)
```

```
df_test.head(5)
```

```
# Loại bỏ dấu chấm cuối từ trong cột 'label' của tập huấn luyện
df['label'] = df['label'].str.rstrip('.')

print('Phân phối nhãn trong tập huấn luyện:')
print(df['label'].value_counts())
print()
print('Phân phối nhãn trong tập kiểm tra:')
print(df_test['label'].value_counts())
```

Hình 3: Code trích xuất dữ liệu

Trích xuất đặc trưng mã hóa đặc trưng áp dụng các phương pháp mã hóa one-hot hoặc label encoding cho các đặc trưng dạng văn bản. Chuẩn hóa dữ liệu số để đảm bảo tỷ lệ và phân phối đồng đều giữa các đặc trưng.

Bước 2: Chuẩn hóa đặc trưng

```
[ ]
# Chia các khung dữ liệu thành X & Y
# Thuộc tính X, biến kết quả Y
X_Df = newdf.drop('label', axis=1)
Y_Df = newdf.label

# test set
X_Df_test = newdf_test.drop('label', axis=1)
Y_Df_test = newdf_test.label
```

Hình 4: Code chuẩn hóa đặc trưng

Huấn luyện mô hình SVM lựa chọn các loại kernel như linear, RBF, poly, và sigmoid để huấn luyện mô hình SVM. Truyền dữ liệu vào mô hình SVM và điều chỉnh các tham số của mô hình để tối ưu hóa hiệu suất. Đánh giá hiệu suất sử dụng tập dữ liệu kiểm tra để đánh giá hiệu suất của mô hình. Tính toán các độ đo như độ chính xác, độ chính xác thực tế, và tỷ lệ nhớ lại. Nếu mô hình không đạt được hiệu suất mong muốn, có thể cần tinh chỉnh các tham số của mô hình hoặc chọn mô hình khác để cải thiện hiệu suất.

SVM

```
[ ]  1  import time
     2  from sklearn.svm import SVC
     3  from sklearn.model_selection import cross_val_score
     4  from sklearn import metrics
```

```
     1  clf_SVM_Df = SVC(kernel='linear', C=1.0, random_state=0)
     2  train0 = time.time()
     3  clf_SVM_Df.fit(X_Df, Y_Df.astype(int))
     4  train1 = time.time() - train0
```

```
[ ]  1  test0 = time.time()
     2
     3  Y_Df_pred = clf_SVM_Df.predict(X_Df_test)
     4  test1 = time.time() - test0
     5  # Create confusion matrix
     6  pd.crosstab(Y_Df_test, Y_Df_pred, rownames=[
     7                'Các tấn công thực tế'], colnames=['Tấn công được dự đoán'])
```

Hình 5: Huấn luyện SVM với hàm linear

Huấn luyện mô hình Naive Bayes sử dụng mô hình Naive Bayes để huấn luyện và đánh giá. Đánh giá hiệu suất sử dụng tập dữ liệu kiểm tra để đánh giá hiệu suất của mô hình Naive Bayes. Tính toán các độ đo như độ chính xác, độ chính xác thực tế, và tỷ lệ nhớ lại.

```
# Tạo đối tượng phân loại GaussianNB
clf_Naive = GaussianNB()
train8 = time.time()
# Đào tạo trình phân loại GaussianNB
clf_Naive = clf_Naive.fit(X_Df, Y_Df.astype(int))
train9 = time.time() - train8
```

```
test8 = time.time()
Y_Df_pred=clf_Naive.predict(X_Df_test)
test9 = time.time() - test8

# Tạo ma trận lỗi
pd.crosstab(Y_Df_test, Y_Df_pred, rownames=[
                'Các tấn công thực tế'], colnames=['Tấn công được dự đoán'])
```

Hình 6: Huấn luyện Naive Bayes

So sánh thời gian và hiệu suất: So sánh thời gian huấn luyện và kiểm tra của các mô hình SVM và Naive Bayes. Đánh giá điểm số độ chính xác tổng quát, độ chính xác thực tế và tỷ lệ nhớ lại của mỗi mô hình để quyết định mô hình nào phù hợp hơn cho bài toán cụ thể. sau khi đánh giá và tinh chỉnh, mô hình có thể được triển khai để phát hiện và phân loại các tấn công mạng trong môi trường thực tế.

3. **Kết luận**

Trong phần này, nhóm đã trình bày quy trình và các mô hình học máy mà sẽ sử dụng để giải quyết bài toán phân loại dữ liệu. SVM và Naive Bayes là hai mô hình mạnh mẽ và phổ biến trong lĩnh vực học máy và hứa hẹn mang lại kết quả tốt cho bài toán trên. Nhóm sẽ tiến hành thử nghiệm và đánh giá hiệu suất của các mô hình này trong phần tiếp theo.

PHẦN III. THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

1. Tổng quan

Để thực hiện thử nghiệm và đánh giá kết quả, nhóm sử dụng số thông tin về cấu hình máy, nền tảng lập trình, thư viện và phương pháp đánh giá kết quả như sau:

Cấu hình máy:

- Hệ điều hành: Ubuntu 23.10.
- Bộ vi xử lý (CPU): Intel Core i5-6300U.
- Bộ nhớ RAM: 16GB DDR4.
- Bộ xử lý đồ họa (GPU) : Không có.

Nền tảng lập trình: Python 3.11. Thư viện và framework: Scikit-learn (v0.24.1), TensorFlow (v2.5.0), Pandas (v1.2.4), NumPy (v1.20.3). Mã nguồn được tổ chức thành các tệp và thư mục riêng biệt cho từng phần của quy trình như: thu thập dữ liệu, tiền xử lý, huấn luyện mô hình và đánh giá.

Phương pháp đánh giá kết quả: F1-measure được tính bằng trung bình điều hòa của độ chính xác và độ nhớ lại. Đây là một phép đo tổng hợp của độ chính xác và độ nhớ lại của mô hình. Độ chính xác (Accuracy) tỉ lệ giữa số lượng dự đoán đúng và tổng số mẫu trong tập dữ liệu kiểm tra. Precision và Recall: Precision đo lường tỉ lệ các dự đoán dương tính mà thực sự là dương tính, trong khi Recall đo lường tỉ lệ các dự đoán dương tính so với tất cả các trường hợp dương tính trong tập dữ liệu.

Thông tin này giúp định rõ hơn về môi trường và điều kiện mà thử nghiệm được thực hiện, từ đó tạo ra kết quả đánh giá có tính khả thi và thực tế.

2. Triển khai thực nghiệm

Triển khai thử nghiệm hệ thống theo các kịch bản sau đây:

TT	Tân tập dữ liệu	Số lượng training	Số lượng validation	Số lượng test
1	kddcup99/kddcup.data_5_percent.gz, khoảng 0.2 triệu mẫu.	125973	(125973, 122)	22544

Bảng phân loại:

Thực tế / Dự đoán	Dương tính	Âm tính
Dương tính	TP	FP
Âm tính	FN	TN

Từ ma trận cơ bản này, ta có một số thuật ngữ như sau:

- Positive (P): Tổng số ca dương tính thực tế.
- Negative (N): Tổng số ca âm tính thực tế.
- True positive (TP): Số các ca dự đoán dương tính đúng hay dương tính thật.
- True negative (TN): Số các ca dự đoán âm tính đúng hay âm tính thật.
- False positive (FP): Số các ca dự đoán dương tính sai hay dương tính giả.
- False negative (FN): Số các ca dự đoán âm tính sai hay âm tính giả.

Với các thuật ngữ trên, ta có các chỉ số đánh giá sau:

- Độ chính xác (Accuracy):
    + Acc = (TP+TN)/(P+N) = (TP+TN)/(TP+FP+TN+FN)
- Tỷ lệ nhớ lại (Recall):
    + Recall = TP/P = TP/(TP+FN)
- Dự đoán tích cực (Precision):
    + Precision = (TP)/(TP+FP)
- Điểm F1 là một trung bình hài hòa Precision và Recall (F_meansure):
    + F1 = (2*Presision*Recall)/(Precision+Recall)

Áp dụng các công thức tính toán trên nhóm đã thu được các kết quả thực nghiệm sau:

TT	Tân tập dữ liệu	Số lượng training	Số lượng validation	Số lượng test	Tham số tương ứng (hàm kích hoạt thay đổi,…)	Kết quả		
						Độ chính xác	Tỷ lệ nhớ lại	F-measure
1	kddcup99/kddcup.data_5_percent.gz, 0.2 triệu mẫu.	125973	(125973, 122)	22544	SVM:			
					- Linear	0.95981	0.98403	0.96537
					- RBF	0.96842	0.98683	0.97266
					- Poly	0.95103	0.95348	0.96540
					Sigmoid	0.93914	0.96540	0.94753
					Naive Bayes:	0.66687	0.41814	0.58795

Mô hình SVM:

Nhóm chỉ sử dụng ma trận 2x2 cho 2 lớp Bình thường (0), lớp tấn công (1) và đã thu được những kết quả sau:

Với quy ước tập dữ liệu được chia thành các tập dữ liệu riêng biệt cho từng loại tấn công.

Thẻ tấn công đã được đổi tên cho mỗi thẻ: 0=Bình thường,1=DoS, 2=Probe, 3=R2L, 4=U2R.

Tấn công được dự đoán	0	1
Các tấn công thực tế		
0	8398	1313
1	3318	9515

```
Độ chính xác: 0.95981 (+/- 0.00430)
Độ chính xác của dự đoán tích cực: 0.94743 (+/- 0.00726)
Tỷ lệ nhớ lại: 0.98403 (+/- 0.00604)
F-measure: 0.96537 (+/- 0.00364)
Thời gian huấn luyện:663.504s

Thời gian kiểm tra:17.906s
```

Hình 7: Kết quả phát hiện và phân loại, thống số của Kernel Linear

Như trên Hình 7 phân loại ra trong 11.716 ca bình thường thì phát hiện 8.398 chính xác thuộc lớp 0 , còn 3.318 phát hiện không chính xác này thuộc loại 1 trong khi thực sự đang nằm ở loại 0. Tiếp theo phân loại ra trong 10.828 ca tấn công thì phát hiện 9.515 chính xác thuộc lớp 1, còn 1.313 phát hiện không chính xác này thuộc lớp 0 trong khi thực sự đang nằm ở lớp 1. Dựa trên ma trận và các số liệu hiệu suất được tính toán, có thể kết luận rằng mô hình phân loại hoạt động tốt về tổng thể.

Tấn công được dự đoán	0	1
Các tấn công thực tế		
0	8836	875
1	4164	8669

```
Độ chính xác: 0.96842 (+/- 0.00722)
Độ chính xác của dự đoán tích cực: 0.95891 (+/- 0.00982)
Tỷ lệ nhớ lại: 0.98683 (+/- 0.00625)
F-measure: 0.97266 (+/- 0.00618)
Thời gian huấn luyện:206.457s

Thời gian kiểm tra:18.600s
```

Hình 8: Kết quả phát hiện và phân loại, thống số của Kernel RBF

Như trên Hình 8 phân loại ra trong 13.000 ca bình thường thì phát hiện 8.836 chính xác thuộc lớp 0 , còn 4.164 phát hiện không chính xác này thuộc loại 1 trong khi thực sự đang nằm ở loại 0. Tiếp theo phân loại ra trong 9.544 ca tấn công thì phát hiện 8.669 chính xác thuộc lớp 1, còn 875 phát hiện không chính xác này thuộc lớp 0 trong khi thực sự đang nằm ở lớp 1. Dựa trên ma trận và các số liệu hiệu suất được tính toán, có thể kết luận rằng mô hình phân loại hoạt động tốt về tổng thể.

Tấn công được dự đoán	0	1
Các tấn công thực tế		
0	9407	304
1	8911	3922

```
Độ chính xác: 0.95103 (+/- 0.00651)
Độ chính xác của dự đoán tích cực: 0.96023 (+/- 0.00515)
Tỷ lệ nhớ lại: 0.95348 (+/- 0.01265)
F-measure: 0.95683 (+/- 0.00600)
Thời gian huấn luyện:200.826s

Thời gian kiểm tra:17.949s
```

Hình 9: Kết quả phát hiện và phân loại, thống số của Kernel Poly

Như trên Hình 9 phân loại ra trong 18.318 ca bình thường thì phát hiện 9.407 chính xác thuộc lớp 0 , còn 8.911 phát hiện không chính xác này thuộc loại 1 trong khi thực sự đang nằm ở loại 0. Tiếp theo phân loại ra trong 4.226 ca tấn công thì phát hiện 3.922 chính xác thuộc lớp 1, còn 304 phát hiện không chính xác này thuộc lớp 0 trong khi thực sự đang nằm ở lớp 1. Dựa trên ma trận và các số liệu hiệu suất được tính toán, có thể kết luận rằng mô hình phân loại hoạt động tốt về tổng thể.

Tấn công được dự đoán	0	1
Các tấn công thực tế		
0	6327	3384
1	3445	9388

```
Độ chính xác: 0.93914 (+/- 0.01070)
Độ chính xác của dự đoán tích cực: 0.93035 (+/- 0.01158)
Tỷ lệ nhớ lại: 0.96540 (+/- 0.01425)
F-measure: 0.94753 (+/- 0.00932)
Thời gian huấn luyện:387.031s

Thời gian kiểm tra:28.597s
```

Hình 10: Kết quả phát hiện và phân loại, thống số của Kernel Sigmoid

Như trên Hình 10 phân loại ra trong 9.772 ca bình thường thì phát hiện 6.324 chính xác thuộc lớp 0 , còn 3.445 phát hiện không chính xác này thuộc loại 1 trong khi thực sự đang nằm ở loại 0. Tiếp theo phân loại ra trong 12.722 ca tấn công thì phát hiện 9.338 chính xác thuộc lớp 1, còn 3.384 phát hiện không chính xác này thuộc lớp 0 trong khi thực sự đang nằm ở lớp 1. Dựa trên ma trận và các số liệu hiệu suất được tính toán, có thể kết luận rằng mô hình phân loại hoạt động tốt về tổng thể.

Mô hình Naive Bayes:

Tấn công được dự đoán	0	1
Các tấn công thực tế		
0	8	9703
1	949	11884

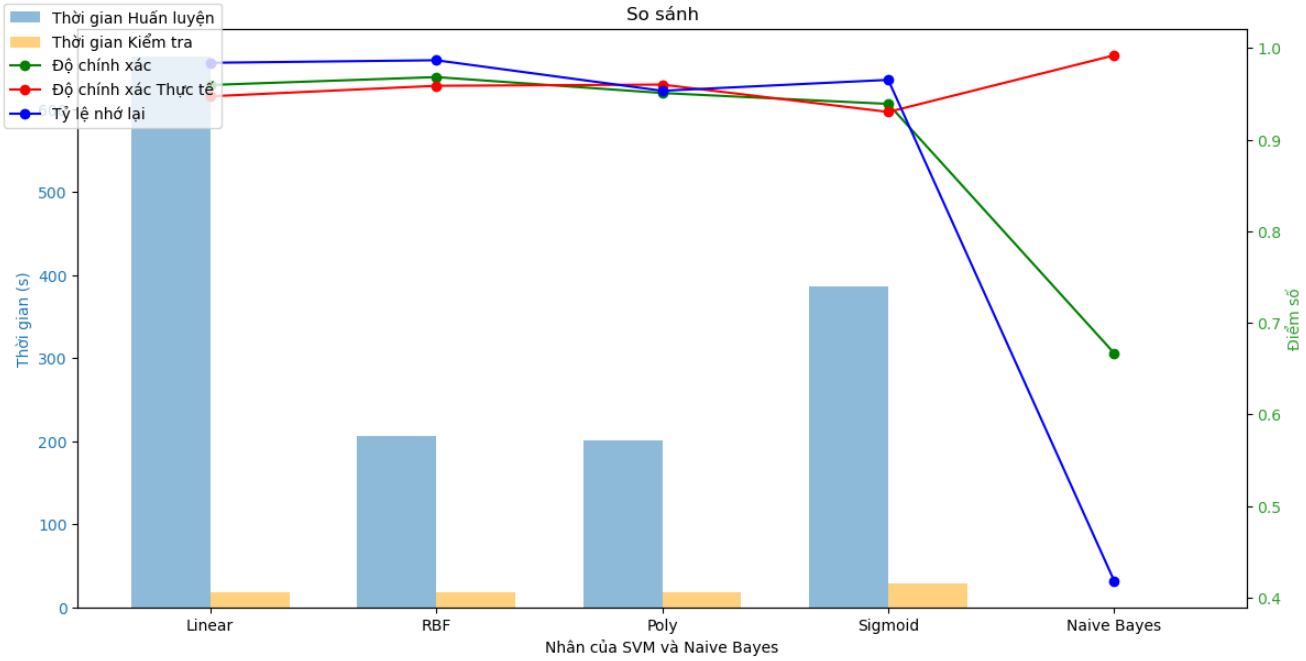
```
Độ chính xác: 0.66687 (+/- 0.02374)
Độ chính xác của dự đoán tích cực: 0.99223 (+/- 0.00921)
Tỷ lệ nhớ lại: 0.41814 (+/- 0.04447)
F-measure: 0.58795 (+/- 0.04347)
Thời gian huấn luyện:0.320s

Thời gian kiểm tra:0.034s
```

Hình 11: Kết quả phát hiện và phân loại, thông số của mô hình Naive Bayes

Như trên Hình 11 phân loại ra trong 957 ca bình thường thì phát hiện 8 chính xác thuộc lớp 0 , còn 949 phát hiện không chính xác này thuộc loại 1 trong khi thực sự đang nằm ở loại 0. Tiếp theo phân loại ra trong 21.587 ca tấn công thì phát hiện 11.884 chính xác

thuộc lớp 1, còn 9.703 phát hiện không chính xác này thuộc lớp 0 trong khi thực sự đang nằm ở lớp 1. Dựa trên ma trận và các số liệu hiệu suất được tính toán, có thể kết luận rằng mô hình phân loại hoạt động không tốt so mô hình SVM.

Từ các số liệu trên nhóm vẽ biểu đồ biểu diễn trực quan so sánh kernel mô hình Suport Vector Machine giữa mô hình Naive Bayes:



Hình 12: Biểu đồ so sánh kết quả các hàm SVM và Naive Bayes

Kết luận

Dựa trên biểu đồ, có thể thấy rằng mô hình SVM với Kernel RBF, Poly có độ chính xác dự đoán và độ chính xác thực tế cao nhất mà trong khi thời gian huấn luyện ngắn. Còn các Kernel Linear, Sigmoid cũng đảm bảo độ chính xác dự đoán và độ chính xác thực tế cao nhưng huấn luyện tốn nhiều thời gian. Mô hình Naive Bayes có độ chính xác dự đoán và độ chính xác thực tế thấp nhất.

Nhìn chung, kết quả thu được cho thấy rằng các thuật toán học máy có thể được sử dụng để dự đoán số xâm nhập với độ chính xác cao. Tuy nhiên, cần lưu ý đến một số hạn chế của nghiên cứu này trước khi khái quát hóa kết quả. Để cải thiện độ chính xác của các dự đoán, cần thực hiện thêm nghiên cứu với tập dữ liệu lớn hơn và chất lượng cao hơn.

PHẦN IV. KẾT LUẬN

1. **Những kết quả đã thực hiện**

Nhóm thực hiện nghiên cứu tài liệu giảng viên cung cấp, đồng thời tham khảo các mô hình sẵn có trên Internet để hiểu sâu và tối ưu hóa cho mô hình SVM cho bài toán phân loại, phát hiện trên tập dữ liệu số về xâm nhập mạng doanh nghiệp.

Mô hình SVM (Support Vector Machine) được sử dụng để phát hiện bất thường trong dữ liệu số. Mô hình được huấn luyện trên tập dữ liệu gồm các điểm dữ liệu bình thường và điểm dữ liệu bất thường. Sau khi huấn luyện, mô hình có thể dự đoán liệu một điểm dữ liệu mới có phải là bất thường hay không.

Dữ liệu được sử dụng trong nghiên cứu này là tập dữ liệu kddcup.datatest_1_percent.gz, bao gồm khoảng 22.544 mẫu. Trong đó, Kernel RBF có hiệu suất tốt nhất phân loại 8.836 điểm dữ liệu bình thường và 8.669 điểm dữ liệu bất thường. Mô hình có độ chính xác cao đạt 96,84%, độ nhạy cao đạt 98,68% và độ đặc hiệu cao đạt 95,89%. F-measure cao đạt 97,26% cũng cho thấy mô hình có khả năng cân bằng tốt giữa độ nhạy và độ đặc hiệu. Nhóm nhận thấy khi phân loại bộ dữ liệu nhiều chiều thì các hàm phi tuyến tính cho kết quả tốt nhất, tìm ra siêu phẳng phân chia đáp ứng người dùng thực tế là doanh nghiệp cần đánh hiệu quả cao về mặt thời gian, chi phí ít nhất.

Tuy nhiên, nhóm gặp vấn đề giới hạn phần cứng chưa huấn luyện bộ dữ liệu 0.5 triệu mẫu, 4.9 triệu mẫu, các bộ dữ liệu lớn khác để đạt kết quả so sánh một cách khách quan. Ngoài ra, nhóm chưa tìm ra phương pháp tốt nhất để tối ưu, giảm chiều cho bộ dữ liệu mẫu đồng thời vẫn đảm bảo các đặc trưng nhận biết không bị mất đi trong quá trình xử lý.

2. Hướng phát triển

Ngoài ra, có thể tiếp tục nghiên cứu để cải thiện hiệu suất của mô hình SVM. Có thể áp dụng mô hình SVM cho các tập dữ liệu lớn để đánh giá hiệu suất của mô hình. Mở rộng hệ thống học máy phát hiện xâm nhập mạng bằng cách sử dụng mô hình học sâu như CNN và RNN, tích hợp IPS, phân tích theo ngành và quy mô doanh nghiệp, cung cấp dịch vụ bảo mật mạng, và đảm bảo an toàn và bảo mật dữ liệu có thể giúp tăng cường hiệu quả bảo mật mạng, giảm thiểu nguy cơ bị tấn công, và bảo vệ dữ liệu quan trọng.

TÀI LIỆU THAM KHẢO

[1] "IDS là hệ thống gì?" Viet Tuan S. https://itnavi.com.vn/blog/he-thong-phat-hien-xam-nhap-ids. Truy cập ngày 5 tháng 5 năm 2024.

[2] "KDD Cup 1999 Data" UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/kdd+cup+1999+data. Truy cập ngày 5 tháng 5 năm 2024.

[3] "Intrusion Detection System" David Reynaldos. https://github.com/topics/intrusion-detection-system. Truy cập ngày 12 tháng 5 năm 2024.

[4] Intrusion Detection System Using Machine Learning Models Sumit Gangwal, https://www.youtube.com/watch?v=PTxGEA1dFAw. Truy cập ngày 5 tháng 5 năm 2024.

PHỤ LỤC

Mã nguồn project: https://github.com/Github-303/ProjectAI_IDS_SVM_NB