

请编辑高亮部分 提交中或英文版本皆可.

Student Name: 黄敏 MinHuang

Email: huangmin@asu.edu

Submission Date: 2024 年 04 月 30 日

Class Name and Term: CSE 534 Spring B

项目 3

I. 项目概述 PROJECT OVERVIEW

以一段话总结本课程项目以及你达成的目标 One paragraph to describe this project and your accomplishment.

通过在 前馈神经网络 (FNN) 在网络流量异常检测中的应用

II. 网络配置 NETWORK SETUP

- 在 Windows 机器上配置运行

III. 软件 SOFTWARE

列出完成本项目时你所用到的软件 Describe major software and network services are used in this project to accomplish your goal.

Pandas,numpy, sklearn, keras, matplotlib, TensorFlow,python3.10, pycharm 开发工具

IV. 项目描述 PROJECT DESCRIPTION

详细叙述你完成本项目的步骤，并给出佐证，如截图，配置文件等，以第一个项目为例，你需要在附录中附上你虚拟机中的 netplan 配置文件 Your work should have evidence and corresponding illustrations, e.g., providing configuration files as attachment in the appendix.

●

- 项目 3 前馈神经网络 (FNN) 在网络流量异常检测中的应用

目标:

1. 了解如何为网络异常检测设置 FNN 训练模型 (使用 NSL-KDD 数据集)
2. 了解前馈神经网络模型在网络攻击检测中的预测

一、任务 1 系统设置

- 1) 任务要求使用 wget 下载" lab-cs-ml-00301.zip",并解压使用,但由于该链接无法访问,所以从 ASU 项目 3 网页上点击下载并解压 <https://raw.githubusercontent.com/SaburH/CSE548/main/lab-cs-ml-00301.zip>
- 2) 将压缩文件放在 Windows 新创建 CSE534_PROJECT3 目录并解压,使用 pycharm 打开,如图 1 所示
- 3) 安装 Python3 和机器学习模块,可以直接在 python 的窗口操作或在终端执行如下命令安装,如图 2 所示


```
pip install keras tensorflow
pip install numpy panda
```

请编辑高亮部分 提交中或英文版本皆可.

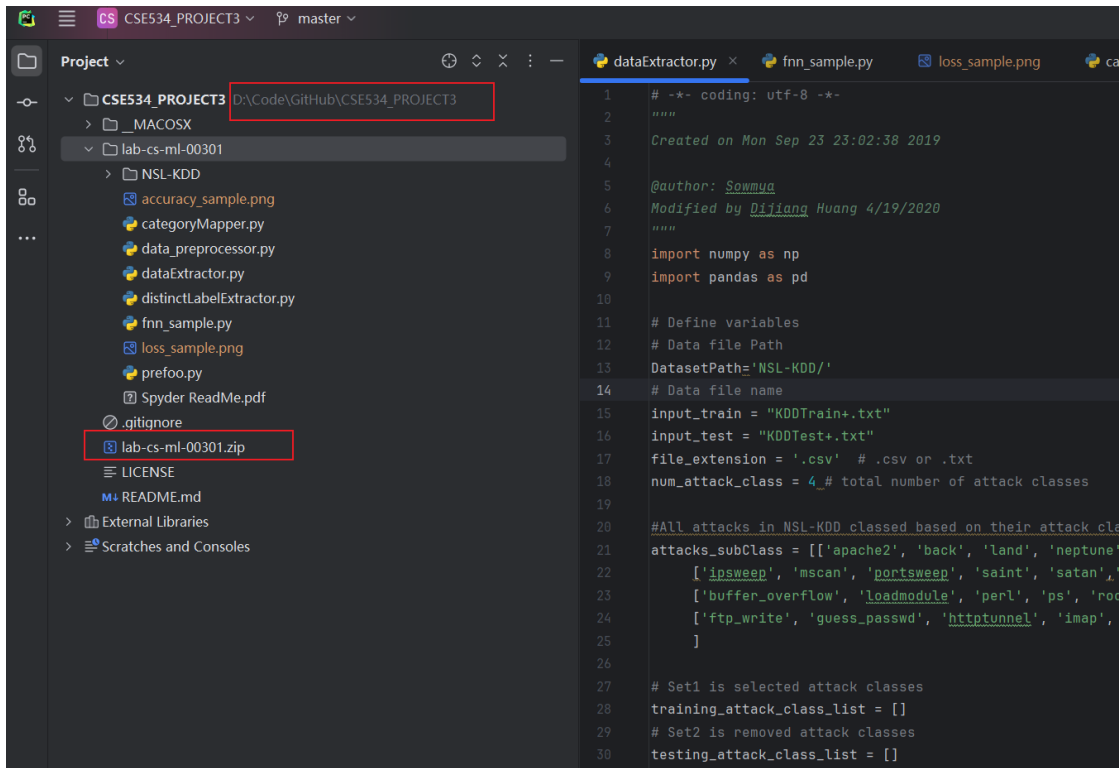


图 1 解压 zip 并使用 pycharm 打开源代码目录

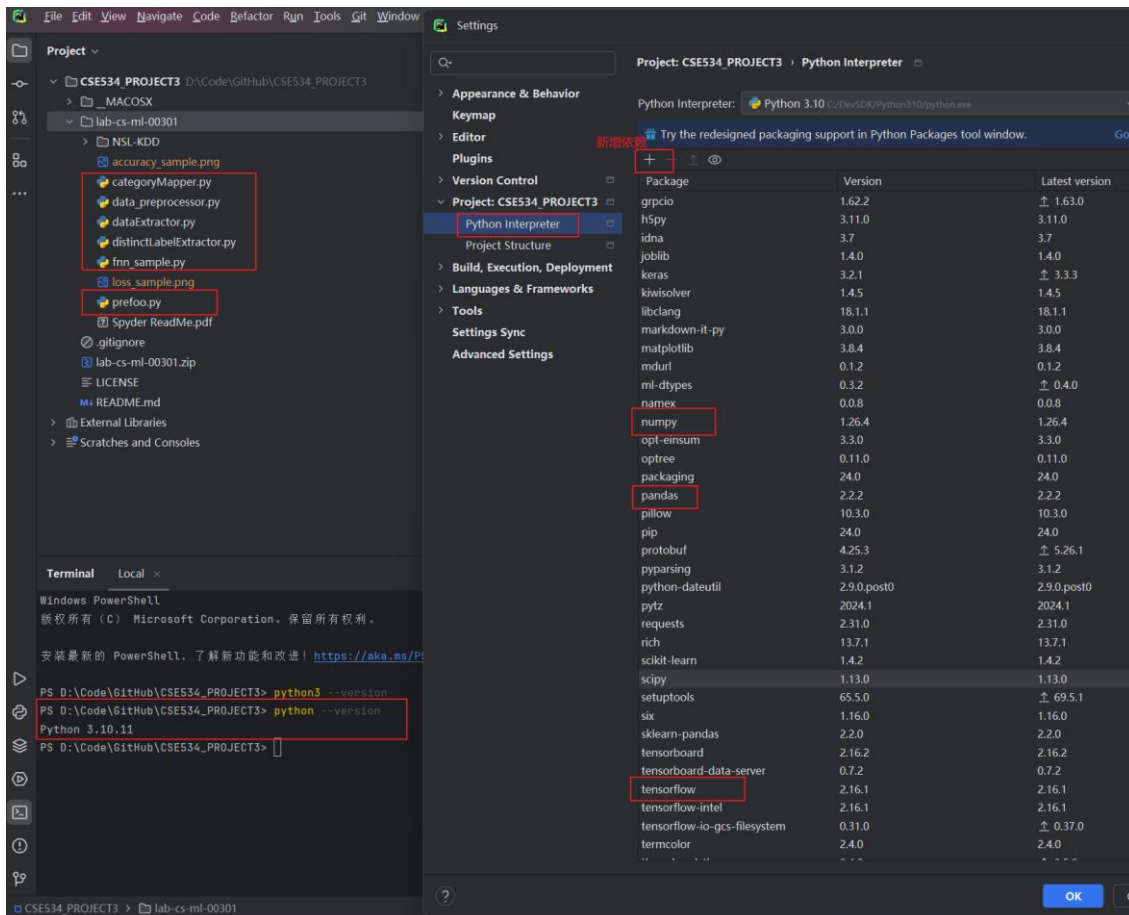
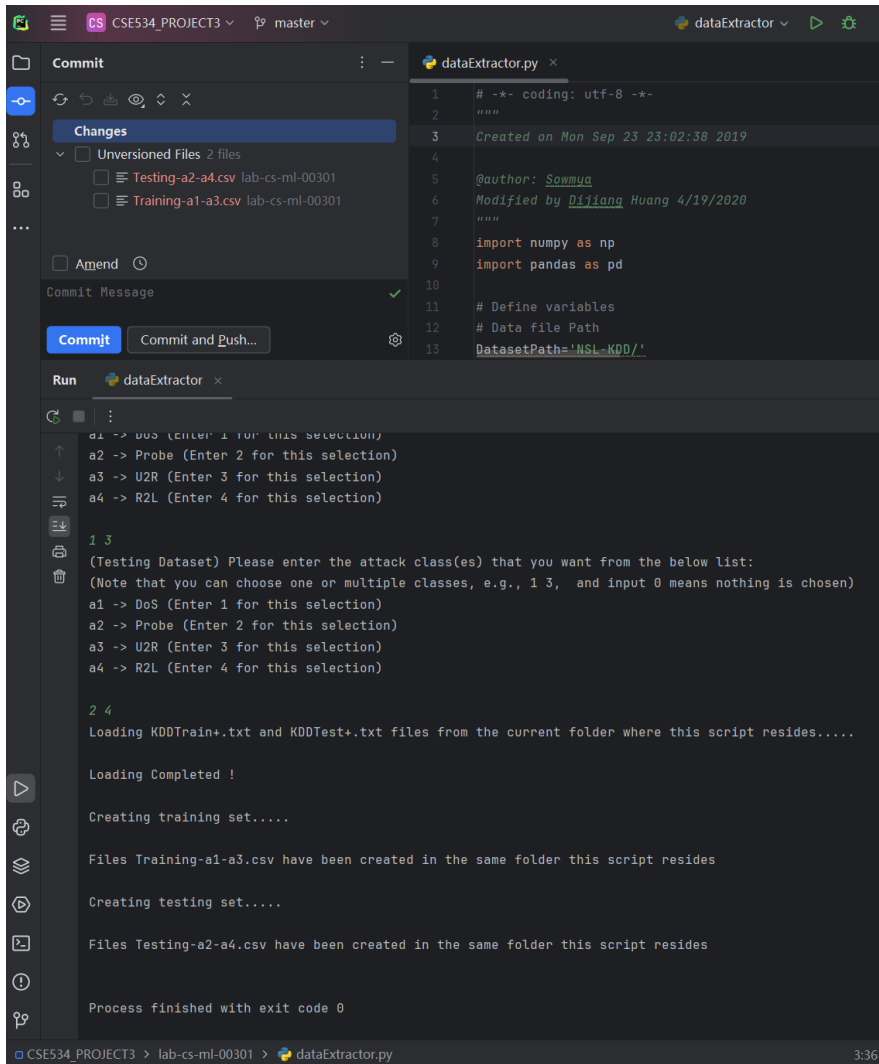


图 2 查看安装的 python 版本和 numpy/pandas/tensorflow 版本

请编辑高亮部分 提交中或英文版本皆可.

二、任务 2 创建用于异常检测的数据模块

- 1) 执行 dataExtractor.py 文件,创建 SA 场景,在 console 输入 1 3 用于创建训练数据集,再次输入 2 4 用于创建测试数据集,如图 3 所示
- 2) 执行 dataExtractor.py 文件,创建 SB 场景,在 console 输入 1 2 用于创建训练数据集,再次输入 1 用于创建测试数据集,如图 4 所示
- 3) 执行 dataExtractor.py 文件,创建 SC 场景,在 console 输入 1 2 用于创建训练数据集,再次输入 1 2 3 用于创建测试数据集,如图 5 所示



The screenshot shows the VS Code interface with the file explorer on the left, the editor in the center, and the Run console at the bottom. The file explorer shows the project structure with files like Testing-a2-a4.csv and Training-a1-a3.csv. The editor displays the dataExtractor.py file with metadata and code. The Run console shows the execution output, including prompts for attack class selection and file creation.

```

1  # -*- coding: utf-8 -*-
2  """
3  Created on Mon Sep 23 23:02:38 2019
4
5  @author: Sowmya
6  Modified by Dijana Huang 4/19/2020
7  """
8  import numpy as np
9  import pandas as pd
10
11  # Define variables
12  # Data file Path
13  DatasetPath='NSL-KDD/'

```

```

a1 -> DDoS (Enter 1 for this selection)
a2 -> Probe (Enter 2 for this selection)
a3 -> U2R (Enter 3 for this selection)
a4 -> R2L (Enter 4 for this selection)

1 3
(Testing Dataset) Please enter the attack class(es) that you want from the below list:
(Note that you can choose one or multiple classes, e.g., 1 3, and input 0 means nothing is chosen)
a1 -> DoS (Enter 1 for this selection)
a2 -> Probe (Enter 2 for this selection)
a3 -> U2R (Enter 3 for this selection)
a4 -> R2L (Enter 4 for this selection)

2 4
Loading KDDTrain+.txt and KDDTest+.txt files from the current folder where this script resides.....

Loading Completed !

Creating training set.....

Files Training-a1-a3.csv have been created in the same folder this script resides

Creating testing set.....

Files Testing-a2-a4.csv have been created in the same folder this script resides

Process finished with exit code 0

```

图 3 SA

请编辑高亮部分 提交中或英文版本皆可.

The screenshot shows a VS Code editor with a commit interface on the left and a terminal window on the right. The commit message is "dataExtractor.py". The terminal window shows the execution of the script, which prompts for attack class selections and outputs the creation of training and testing datasets.

```

1  # -*- coding: utf-8 -*-
2  """
3  Created on Mon Sep 23 23:02:38 2019
4
5  @author: Sommuu
6  Modified by Dijiang Huang 4/19/2020
7  """
8  import numpy as np
9  import pandas as pd
10
11  # Define variables
12  # Data file Path
13  DatasetPath='NSL-KDD/'

```

Commit Message: dataExtractor.py

Run dataExtractor.py

```

a1 -> DoS (Enter 1 for this selection)
a2 -> Probe (Enter 2 for this selection)
a3 -> U2R (Enter 3 for this selection)
a4 -> R2L (Enter 4 for this selection)

1 2
(Testing Dataset) Please enter the attack class(es) that you want from the below list:
(Note that you can choose one or multiple classes, e.g., 1 3, and input 0 means nothing is chosen)
a1 -> DoS (Enter 1 for this selection)
a2 -> Probe (Enter 2 for this selection)
a3 -> U2R (Enter 3 for this selection)
a4 -> R2L (Enter 4 for this selection)

1
Loading KDDTrain+.txt and KDDTest+.txt files from the current folder where this script resides.....

Loading Completed !

Creating training set.....

Files Training-a1-a2.csv have been created in the same folder this script resides

Creating testing set.....

Files Testing-a1.csv have been created in the same folder this script resides

Process finished with exit code 0

```

图 4 SB

请编辑高亮部分 提交中或英文版本皆可。

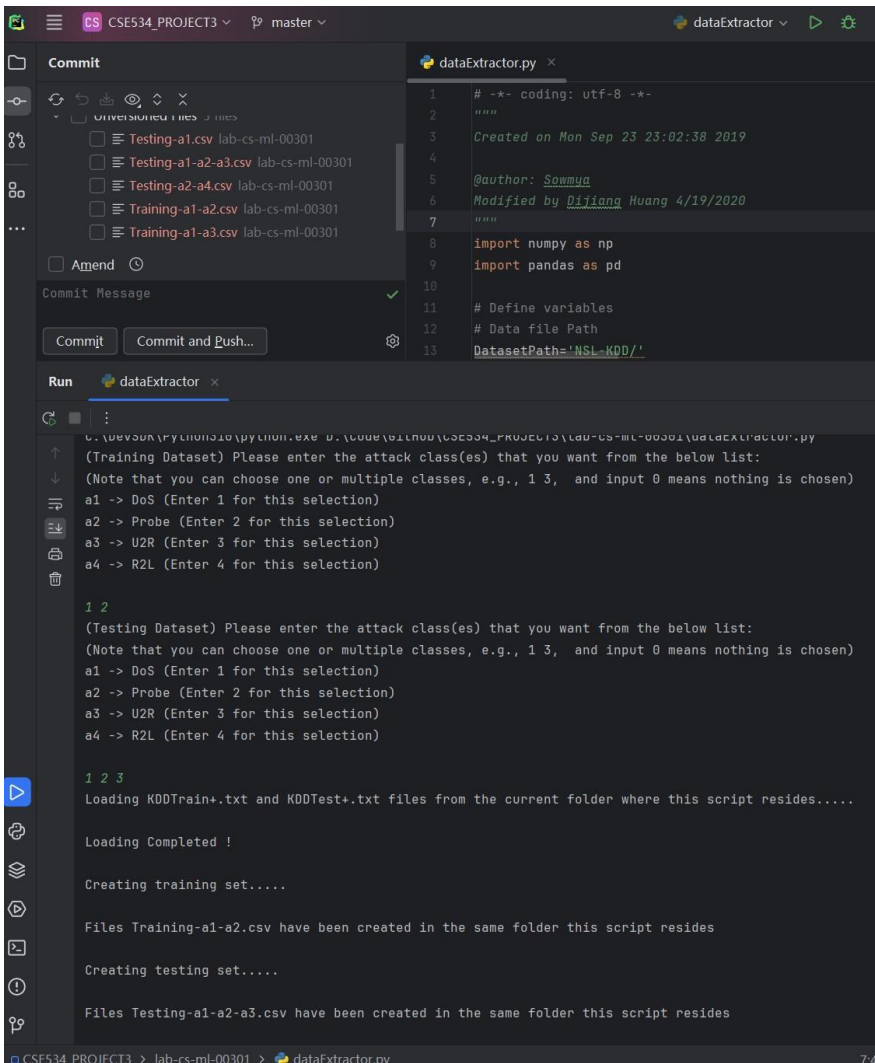


图 5 SC

三、任务 3 异常检测分析

1) 哪种场景产生的测试结果最准确？观察你的模型对所接受训练的攻击类别的变化的预测能力。然后，观察在这些攻击类型子集上进行训练时的预测准确性。观察用特定攻击类型的子集训练模型时，模型的预测效果是否会更好。

回答: 准确率可以通过混淆矩阵来计算

a) 执行 `fnn_sample.py`, 生成 3 个场景的混淆矩阵, 将混淆矩阵的值代入准确率公式计算, 场景 SA 的准确率约等于 0.6896, 场景 SB 的准确率约等于 0.8314, 场景 SC 的准确率约等于 0.8772, 从准确率来看, 场景 SC 的模型具有最高的准确率。

b) 模型在训练数据集包含的攻击类型上表现更好。例如, 场景 SB 中模型在仅对攻击类型'a1'进行训练时, 在测试攻击类型'a1'时表现得很好。说明当模型在特定的攻击类型上进行训练时, 它可能表现出更好的预测效果。

c) 3 个场景的混淆矩阵数据, 如下图 6 和图 7 所示:

场景 SA 的混淆矩阵

[[9634 77]

[3897 1409]]

场景 SB 的混淆矩阵

[[9013 698]

[1368 6092]]

场景 SC 的混淆矩阵

[[9028 683]

[1738 8210]]

请编辑高亮部分 提交中或英文版本皆可。

The screenshot shows a Jupyter Notebook with the following code and output:

```

124 scenario_b_testing_attacks = ['a1']
125 scenario_c_training_attacks = ['a1', 'a2']
126 scenario_c_testing_attacks = ['a1', 'a2', 'a3']
127
128 # 批大小与epoch数
129 batch_size = 10
130 num_epochs = 10
131
132 # 执行三个场景
133 confusion_matrix_a, history_a = execute_scenario(scenario_a)
134 confusion_matrix_b, history_b = execute_scenario(scenario_b)
135 confusion_matrix_c, history_c = execute_scenario(scenario_c)
136
137

```

The output shows the training progress and the confusion matrix for scenario 'a' (SA):

```

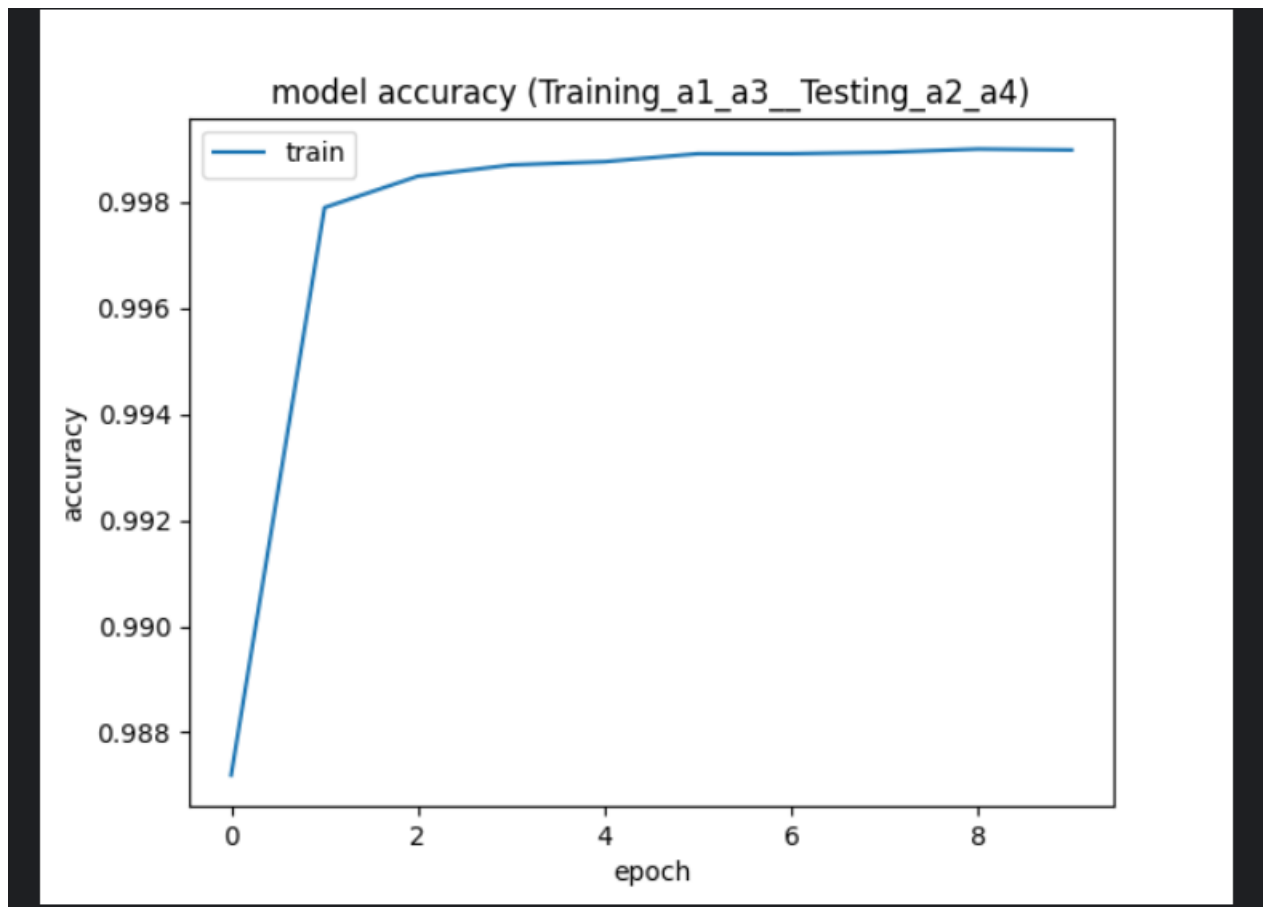
Epoch 5/10 11333/11333 5s 425us/step - accuracy: 0.9977 - loss: 0.0084
Epoch 6/10 11333/11333 5s 437us/step - accuracy: 0.9984 - loss: 0.0069
Epoch 7/10 11333/11333 5s 434us/step - accuracy: 0.9987 - loss: 0.0055
Epoch 8/10 11333/11333 5s 425us/step - accuracy: 0.9987 - loss: 0.0051
Epoch 9/10 11333/11333 5s 406us/step - accuracy: 0.9987 - loss: 0.0052
Epoch 10/10 11333/11333 5s 400us/step - accuracy: 0.9991 - loss: 0.0040
470/470 0s 360us/step
Confusion Matrix:
[ TN, FP ]
[ FN, TP ]=
[[9634  77]
 [3897 1409]]

```

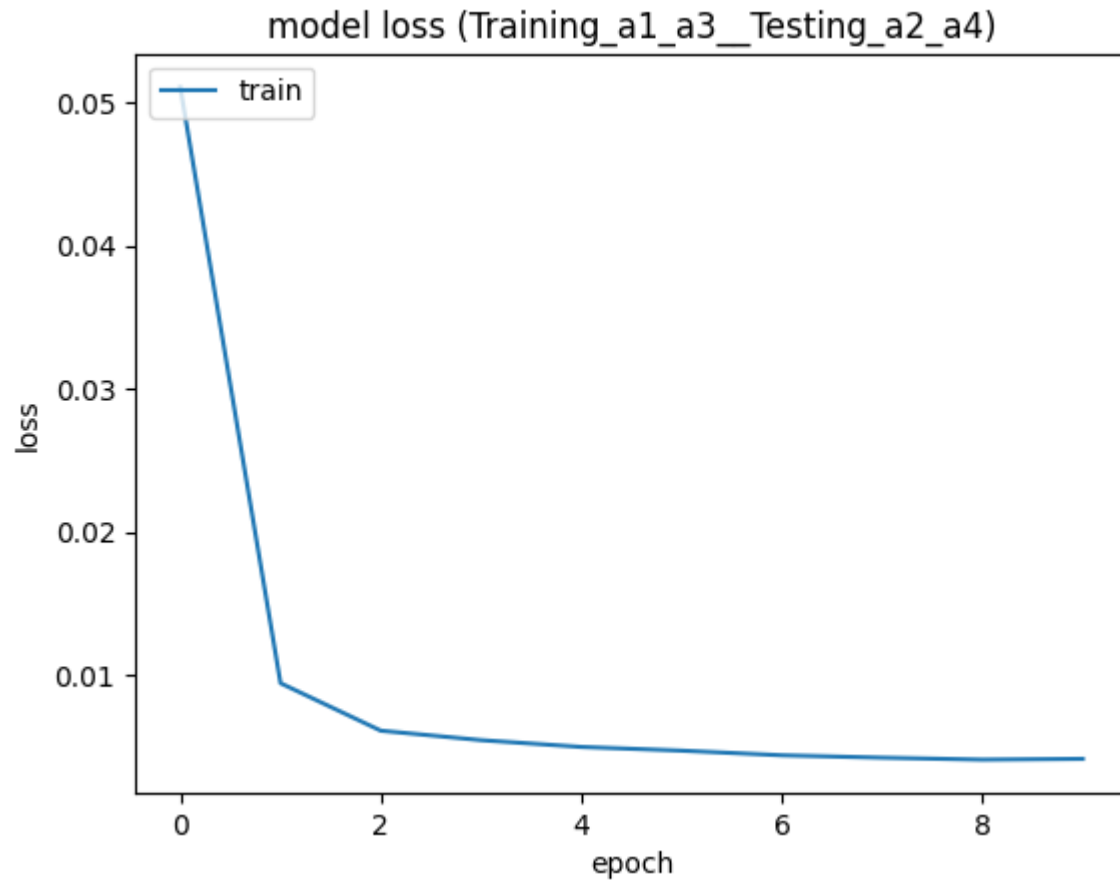
The confusion matrix for SA is:

	Actual SA	Predicted SA
Actual SA	9634	77
Predicted SA	3897	1409

图 6 SA 的混淆矩阵



请编辑高亮部分 提交中或英文版本皆可.



请编辑高亮部分 提交中或英文版本皆可.

commit

dataExtractor.pyfnn_sample.pyloss

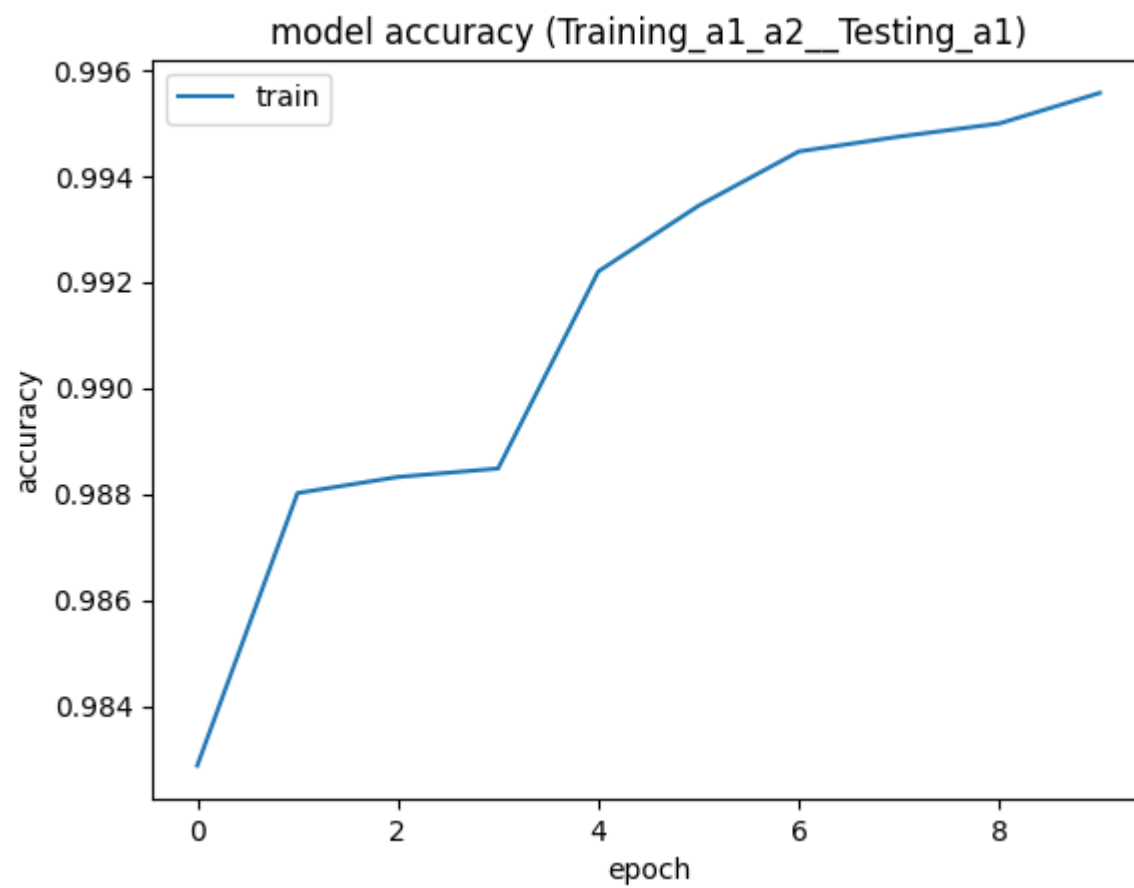
Changes 1 file
Unversioned Files 5 files
Amend

Commit and Push...

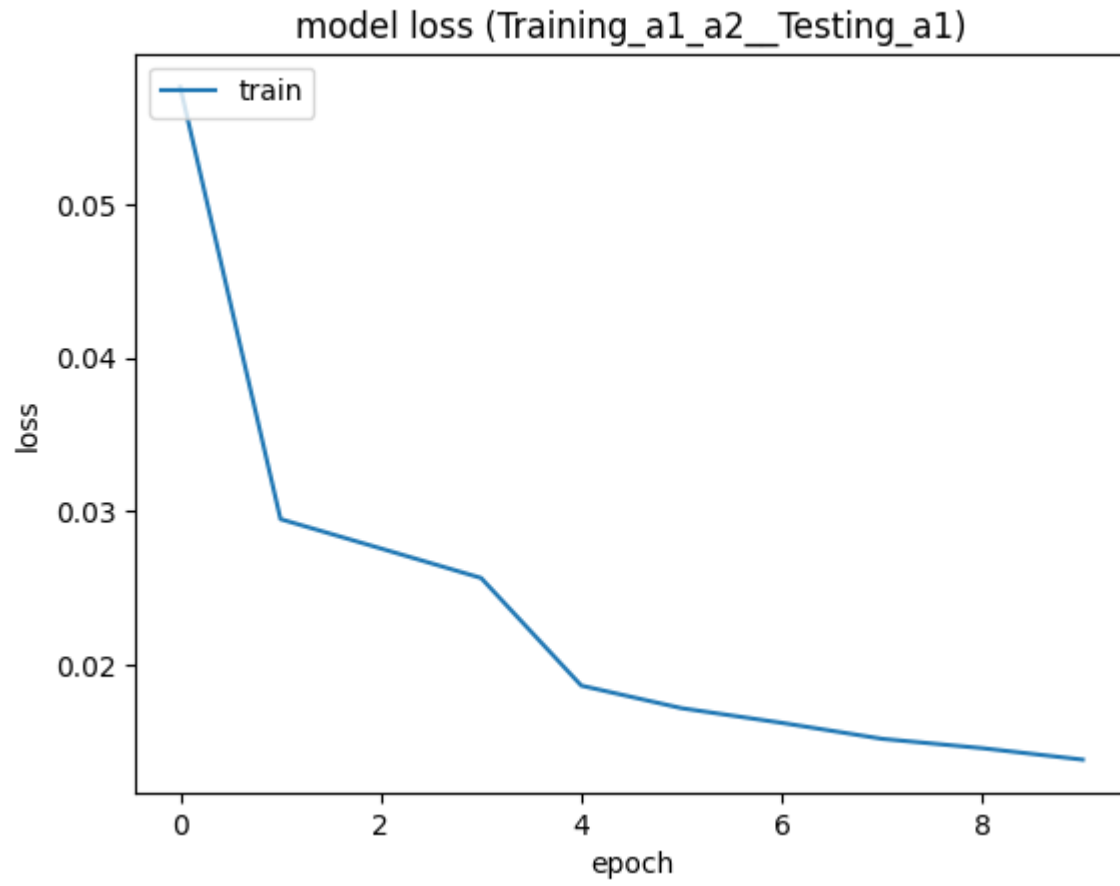
```
83  
84  
85 # 执行特定场景下的数据加载、预处理、模型构建、  
86 3 usages  NoamHuang  
87 def execute_scenario(training_attack_  
    print("Executing Scenario with T  
  
[ FN, TP ]=  
[[9013 698]  
[1368 6092]] SB  
Executing Scenario with Training: ['a1', 'a2'], Testing: ['a1', 'a2', 'a3']  
C:\Dev\SDK\Python310\lib\site-packages\keras\src\layers\core\dense.py:86: UserWarning: Do not pass an 'in  
super().__init__(activity_regularizer=activity_regularizer, **kwargs)  
Epoch 1/10  
12493/12493 ----- 6s 435us/step - accuracy: 0.9668 - loss: 0.1096  
Epoch 2/10  
12493/12493 ----- 5s 421us/step - accuracy: 0.9899 - loss: 0.0228  
Epoch 3/10  
12493/12493 ----- 5s 414us/step - accuracy: 0.9930 - loss: 0.0204  
Epoch 4/10  
12493/12493 ----- 5s 405us/step - accuracy: 0.9937 - loss: 0.0175  
Epoch 5/10  
12493/12493 ----- 5s 422us/step - accuracy: 0.9940 - loss: 0.0169  
Epoch 6/10  
12493/12493 ----- 5s 412us/step - accuracy: 0.9944 - loss: 0.0166  
Epoch 7/10  
12493/12493 ----- 5s 401us/step - accuracy: 0.9944 - loss: 0.0168  
Epoch 8/10  
12493/12493 ----- 5s 413us/step - accuracy: 0.9945 - loss: 0.0152  
Epoch 9/10  
12493/12493 ----- 5s 412us/step - accuracy: 0.9947 - loss: 0.0153  
Epoch 10/10  
12493/12493 ----- 5s 410us/step - accuracy: 0.9949 - loss: 0.0151  
615/615 ----- 0s 373us/step  
Confusion Matrix:  
[ TN, FP ]  
[ FN, TP ]=  
[[9028 683]  
[1738 8210]] SC
```

图 7 SB 和 SC 的混淆矩阵

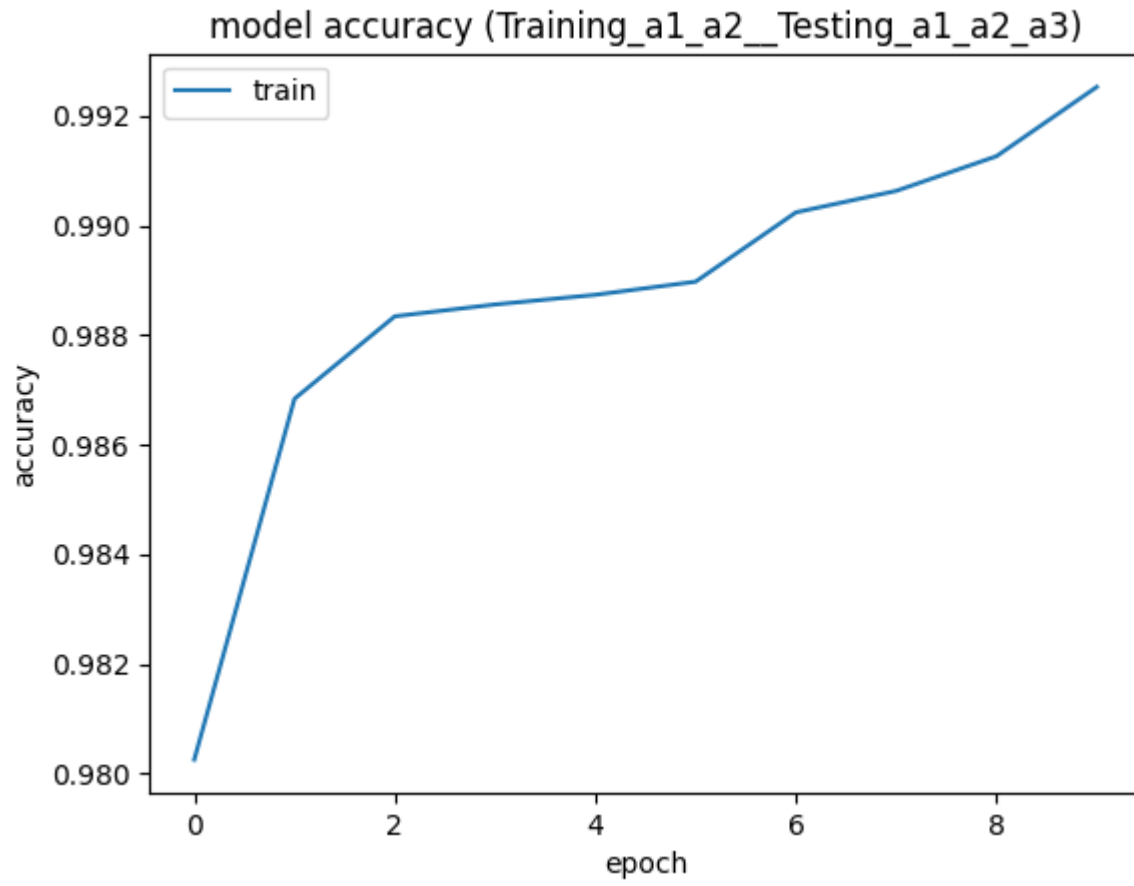
请编辑高亮部分 提交中或英文版本皆可.

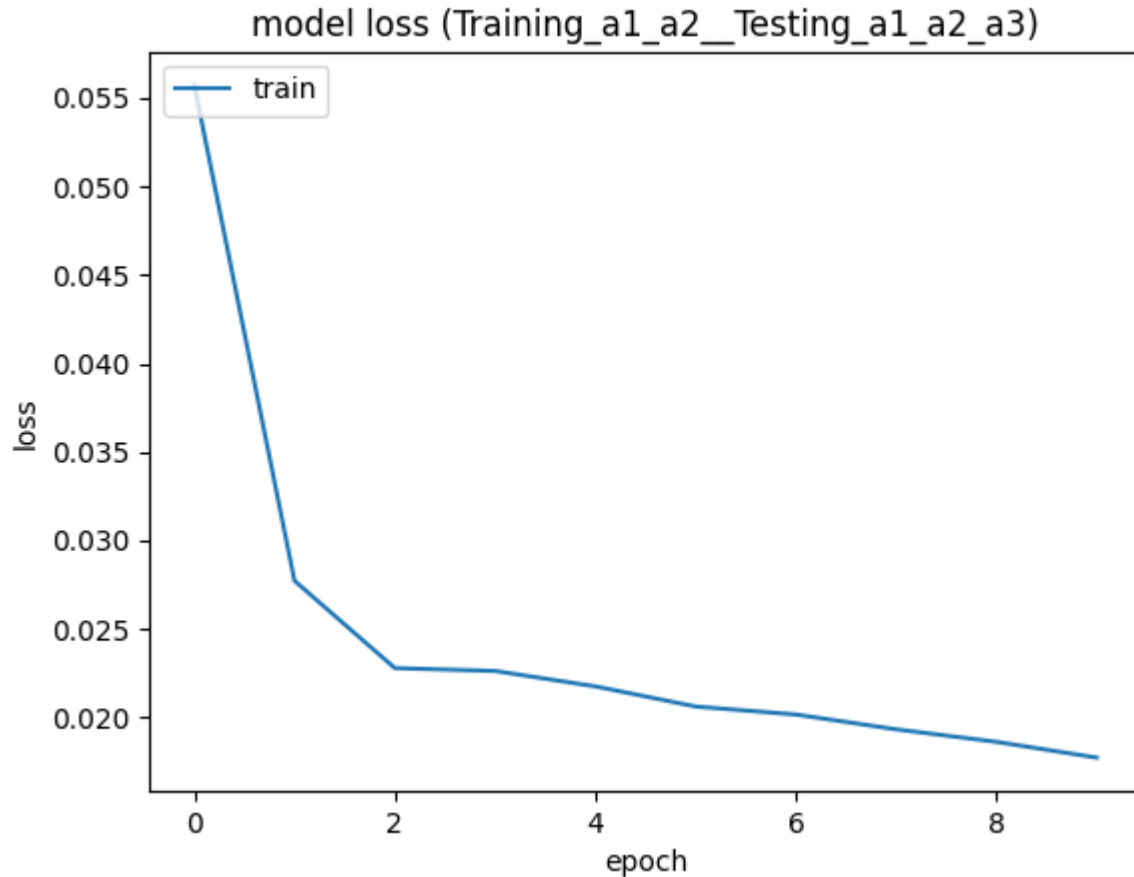


请编辑高亮部分 提交中或英文版本皆可.



请编辑高亮部分 提交中或英文版本皆可.





2) 2. 模型将新的攻击类别（在 SA 和 SC 的测试数据集中）检测为“正常”或“攻击”的平均准确度是多少？（提示：将模型设置为执行二元分类时，模型的预测值可以是：

- 0 → 正常
- 1 → 攻击

该预测与预测的准确性相关。记下该预测的准确性，即对于 SA 中的上述未知攻击 A2 和 A4 或 SC 中的 A3，预测为“正常”或“攻击”）

回答:

- a) 新攻击类别分析：场景 SA 训练数据集包含了 A1 和 A3，测试数据集包含了 A2 和 A4，新攻击类别是 A2 和 A4；场景 SC 训练数据集包含了 A1 和 A2，测试数据集包含了 A1，A2 和 A3，新攻击类别是 A3
- b) 代入准确度公式：计算新攻击类别被检测为“正常”或“攻击”的准确度，需要关注的是真正类（TP）和假负类（FN），代入准确度公式 $\text{准确度} = \text{TP} / (\text{TP} + \text{FN})$
- c) 场景 SA 中对于新的攻击类型 A2 和 A4，模型将其 26.5% 的样本 $\{1409 / (3897 + 1409) = 26.5\%$ 正确预测为“攻击”。
- d) 场景 SC 中对于新的攻击类型 A3，模型将其中 82.5% 的样本 $\{8210 / (1738 + 8210) = 82.5\%$ 正确预测为“攻击”。

3) 3. 未经训练的子集中的攻击与来自经过训练的数据集的攻击之间，有什么区别？（提示：例如，在 SA 中，A1 和 A3 与 A2 和 A4 有什么区别？A1 和 A2 是否属于相同的攻击类别，它们与正常数据之间是否有着相似的区别？通过查看这些攻击的数据，你可以得出观察结果）。

回答:

在场景 SA 中: A1 (Dos 攻击) 和 A3 (U2R 攻击) 与 A2 (Probe 攻击) 和 A4 (R2L 攻击) 的区别:

通过查看这些攻击的数据, 我观察到 A1 和 A3 的攻击模式和 A2、A4 有明显不同。

A1 和 A3 的数据特征可能包括大量网络流量、异常的系统调用等, 而 A2 和 A4 的数据特征可能更多体现在扫描活动、远程访问尝试等方面。它们所针对的攻击目标和手段存在区别。

请编辑高亮部分 提交中或英文版本皆可。

A1 (Dos 攻击) 和 A2 (Probe 攻击) 不属于相同的攻击类别。

A1 是拒绝服务攻击, 目的是耗尽系统资源; 而 A2 是探测攻击, 目的是获取信息, 作为后续攻击的渗透准备。

A1 和 A2 与正常数据之间是否有着类似区别?

通过观察数据, A1 和 A2 与正常数据之间的区别模式是类似的, 它们都表现为异常的网络流量和连接模式。与正常数据相比, 它们可能在网络层面特征上存在明显差异。

因此, 根据数据观察结果, 虽然 A1 和 A2 都是攻击行为, 但它们属于不同的攻击类别, 攻击目标和手段不同。但是与正常数据相比, 它们在网络层面特征上的差异模式是相似的。而 A1、A3 与 A2、A4 则在攻击模式和数据特征上存在更大差距。

4) 4. 预测的准确性是否与攻击的相似性有关? 如果是, 那么相似点是什么? (提示: 如果我们发现模型的预测准确性更好, 则查看数据和攻击类型, 找出可能使预测准确性更好的相似点)。

回答: 是有关的。例如场景 SC 中, 训练集包含 A1 (Dos 攻击) 和 A2 (Probe 攻击) 测试集增加了 A3 (U2R 攻击), 模型表现次于 SB, 但优于 SA, 混淆矩阵显示一些误报和漏报, 这可能是因为 A1 和 A2 虽然属于不同攻击类型, 但都是网络层面的攻击行为, 在很多网络流量和连接模式的特征上是相似的。

V. 项目总结 CONCLUSION

描述从这个项目中吸取的教训, 例如, 任何有趣的发现、提示和技巧。提供关于您的项目的自我评估, 并为此项目提供评论。Describe lesson learned from this project, e.g., any interesting discovers, tips, and tricks. Provide a self-assessment about your project and provide comments to this project.

VI. 附录: 文件 APPENDIX : ATTACHED FILES

提供使用的配置和开发的源文件的列表。在您的配置文件中, 请注明注释。一个好的做法是在您进行更改的地方提供注释, 例如:

Provide a list of used configurations and developed source files. In your configuration file, please with well-marked comments. A good practice is to provide comments where you made changes, something like:

```
// Your Name: comments
```

```
# Your Name: comments
```

```
/*
Your Name: comments
*/
```

注释格式取决于您使用的系统文件和程序 The comment format depends on your used system files and programs.

VII. 参考 REFERENCES

参考是可选的, 可以向阅读你的报告提供链接源以进行验证和学习。Reference is optional, but nice to have to allow others to read your report with additional linked source for validation and learning.

[1] Wireshark, available at <https://www.wireshark.org/>, accessed by 8/31/2018.

[2] Postel, Jon. "RFC 791: Internet protocol." (1981).