|  |
| --- |
| **实操实验室介绍** |

**项目 3**

***前馈神经网络 (FNN) 在网络流量异常检测中的应用***

**类别：**

|  |
| --- |
| CS-ML:机器学习 |

**目标：**

|  |  |
| --- | --- |
|  | 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） |
|  | 了解前馈神经网络模型在网络攻击检测中的预测 |

**实验室的预计时长：**

|  |  |
| --- | --- |
|  | 行家：120 分钟 |
|  | 新手：600 分钟 |

**难度示意图：**

设计

时间

知识

配置

实现

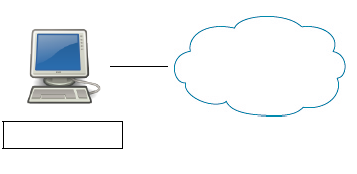
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **难度表格。**   | 衡量指标 | 数值 (0-5) | | --- | --- | | 时间 | 4 | | 设计 | 4 | | 实现 | 3 | | 配置 | 3 | | 知识 | 4 | | 分数（平均） | 3.6 | |

**所需的操作系统：**

|  |
| --- |
| Linux：Ubuntu 18.04 LTS |

**实验室运行环境：**

|  |
| --- |
| ThoTh Lab: https://thothlab.org |



Net 1

互联网

工作站

|  |  |
| --- | --- |
|  | 服务器：Linux (Ubuntu 18.04 LTS) |
|  | 网络设置：连接到互联网 |

**实验室的准备工作：**

|  |
| --- |
| 在 Linux 虚拟机上安装 Python 和 Anaconda 的软件包参考资料 1  NSL-KDD 数据集。参考资料 2  Python 机器学习的基本概念：参考资料 3  基于 Python 的数据预处理：参考资料 4  FNN ML 模型。参考实验室：参考资料 5 |

**实验室概述**

在本实验室中，你将使用 NSL-KDD 数据集和前馈神经网络 (FNN) 进行网络异常检测和攻击分析。你将根据提供的 NSL-KDD 数据集，为若干个不同的数据分析场景创建有针对性的训练和测试数据集。根据每个创建场景的训练和测试结果，你需要解决一些有关基于 ML 的异常检测准确性和效率的问题。

该实验室的评估是基于解决所提供问题的完整性和深度。此外，学生需要演示自己如何完成机器学习任务，并产生良好的机器学习结果，以支持他们解决所提供问题的论点。学生需要提交一系列屏幕截图和相应的图例，说明自己是如何完成所需的机器学习设置、训练和验证的。学生需要提交更新的 Python 编程代码和实验室报告，以便根据测试提供深入的说明。

概括而言，学生将需要做到：

* 用 Python 对机器学习执行数据预处理。
* 实现 Python 程序，以实现所需的机器学习函数/特性。
* 分析和评估机器学习结果，以解决异常检测问题。

**任务 1 系统设置**

在此任务中，你需要检查机器学习环境是否正确设置。按照建议的参考实验室的以下说明，检查机器学习运行环境。

|  |
| --- |
| **建议：** |
| 1. 回顾并练习参考实验室 1 |
| 1. 回顾参考实验室 2 |
| 1. 回顾参考实验室 3 |
| 1. 回顾并练习参考实验室 4（基于 Python 的数据预处理） |
| 1. 回顾并练习参考实验室 5 (FNN)。 |
| 注意：你的服务器上可能已经设置了机器学习服务 (anaconda) 。依照参考实验室 1，验证这些服务是否已正确设置，以执行所需的任务。 |

要执行以下任务，你需要下载数据源和 python 程序。你可以用以下命令来下载实验室资源文件。（用‘wget -version’命令检查 wget 是否已安装。如果 wget 没有安装，则用‘sudo apt install wget’命令来进行安装）

注意：请从下面的 repo 下载代码。不要使用已经存储在虚拟机中的代码。

|  |  |
| --- | --- |
| $ | wget https://gitlab.thothlab.org/thoth-group/ThoThLabResource/raw/master/ lab-cs-ml-00301.zip |
| $ | unzip lab-cs-ml-00301.zip |
| $ | cd lab-cs-ml-00301 |

源代码文件位于文件夹“lab-cs-ml-00301”中。其内容和特点总结如下：

* 有一个文件夹 (NSL-KDD) 中包含可在此实验室中处理和分析的数据。有关数据集的更多信息，请参阅实验室 CS-ML-00101（了解 NSL-KDD 数据集）。
* “Spyder ReadMe.pdf”文件是对如何使用 “Spyder”一种 python 编程 GUI 应用程序的简要说明。
* fnn\_sample.py 文件是提供数据预处理、数据训练和数据分析结果表示的 python 代码。你将用此文件作为开发新 python 代码的起点，以实现其余任务中描述的必需特性。
* *distinctLabelExtractor.py* 文件会加载 NSL-KDD 文件并提取攻击名称和类型。
* *DataExtractor.py* 文件通过从训练和测试数据集中提取攻击类型的子集，创建自定义的训练和测试数据集。
* *categoryMapper.py* 文件为基于字符串的特性创建带标签的类别数据。
* *data\_preprocesor.py* 文件会创建一个 Python 库，该库可以简化基于 Python 的机器学习数据处理解决方案的数据预处理过程。

有关如何使用 *fnn\_sample.py* 的更多信息，请参阅实验室 CS-ML-00201；有关如何使用以下 Python 程序（*distinctLabelExtractor.py、 DataExtractor.py、categoryMapper.py* 和 *data\_preprocesor.py*）的更多信息，请参阅实验室 CS- ML-00200。

**任务 2 创建用于异常检测的数据模块**

在本项目中，我们考虑 NSL-KDD 数据集 *D = (A, N)* ，其中包含攻击子数据集 A 和正常子数据集 N。在本项目中，我们选择 NSL-KDD 数据集中的 *D = KDDT rain + .txt* 用于训练，和 NSL-KDD 数据集中的 *D = KDDTest + .txt* 用于测试。现在，你选择了 4 种攻击类型，如 *A1-A4 A* 和正常数据 *N*，你从用于训练模型的子集 *A' ⊂ A* 和所有正常数据 *N* 开始。四种攻击类型包括：*A*1:拒绝服务（DoS）；*A*2:探测；*A*3:根访问权限 (U2R)；以及 *A*4 :远程到本地 (R2L)。有关这四类攻击及其包含的子类的更多详细信息，请参阅实验室 CS-ML-00101。

在此任务中，你需要用 *DataExtractor.py* 来创建自定义的训练数据集。有关使用此 Python 程序创建自定义训练和测试数据集的更多详细信息，请参阅实验室 CS-ML-00200。在表 CS-ML-00301.1 中，将选择三种攻击场景。你需要创建三个数据集，包括场景 SA、SB 和 SC的训练和测试。

**表 CS-ML-00301.1**

自定义的攻击场景设置

|  |  |  |  |
| --- | --- | --- | --- |
| 数据集 | 场景 A (SA) | 场景 B (SB) | 场景 C (SC) |
| 训练数据集 | *A*1*, A*3*, N* | *A*1*, A*2*, N* | *A*1*, A*2*, N* |
| 测试数据集 | *A*2*, A*4*, N* | *A*1*, N* | *A*1*, A*2*, A*3*, N* |

**任务 3 异常检测分析**

在此任务中，你需要修改提供的 *fnn\_sample.py* 文件，用于获取任务 2 生成的新输入的训练和测试数据集。注意，提供的 *fnn\_sample.py* 文件会选取一个训练数据集，并将其拆分为训练和测试部分。但在此任务中，你需要对其进行修改，以获取通过任务 2 生成的单独的培训和测试数据集。你需要根据三个场景（*S*A、*S*B 和 *S*C）运行三种不同的训练和验证。你应该使 FNN 的训练和测试参数设置相同，即批量大小、期数、神经网络设置，包括第一层节点数、隐藏层节点数、应用的损失函数、阈值等。根据新创建的训练数据集和测试数据集，你应该解决以下问题。请给出你的推理和分析结果。

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。
2. 模型将新的攻击类别（在 *S*A 和 *S*C 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”）

1. 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 *S*A 中，*A*1 和 *A*3 与 *A*2 和 *A*4 有什么区别？*A*1 和 *A*2 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）。
2. 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）。

**交付成果**

学生需要通过一系列屏幕截图来记录实验室的运行过程，并阐释机器学习的训练结果。学生也需要提交一份报告，根据*实验室评估部分*的描述，说明他们可以完成的任务。关于如何提交屏幕截图，请参考《学生指南》第 B.4 节（提交实验室报告）。

**相关信息和资源**

|  |
| --- |
| NSL-KDD 数据集：  https://www.unb.ca/cic/datasets/nsl.html  KDD CUP 存档资料  https://www.kdd.org/kdd-cup  数据预处理：  https://www.shanelynn.ie/select-pandas-dataframe-rows-and-columns-using-iloc-loc-and-ix/  https://medium.eom/@contactsunny/label-encoder-vs-one-hot-encoder-in-machine-learning-3fc273365621  https://scikit-learn.org/stable/modules/generated/sklearn.model\_selection.train-test\_split.html  https://scikit-learn.org/stable/modules/preprocessing.html  搭建 ANN:  https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/ https://machinelearningmastery.com/rectified-linear-activation-function-for-deep -learning-neural-networks/  <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>  https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learning-neural-networks/ |

**实验室评估（100 分）**

在本实验室中，学生需要用基本的 FNN 模型创建一个用于网络入侵检测的异常检测模型。

1. （35 分）仅用实验室的屏幕截图功能记录机器学习的运行结果。对每个给定的训练/测试场景生成的结果，提供充分而简洁的说明。
2. （20 分）提交更新的 *fnn\_simple.py*，并提供足够的注释，用以说明更新的代码。
3. （45 分）提交报告，以便提供深入的分析，从而解决任务 3 中给出的问题。