CICIDS2017数据集(<https://www.unb.ca/cic/datasets/ids-2017.html>)

数据集采集时间为2017年7月3日(星期一)至2017年7月7日(星期五)。其中星期一采集的数据仅有正常的网络流量，其余时间段包含了异常流量。主要的攻击类型包括了暴力破解FTP、暴力破解SSH、DoS、Heartbleed、Web攻击、渗透、僵尸网络和DDos。这8种攻击类型分别在周二至周五的上下午展开。

表1 数据集CICIDS类别分布

|  |  |  |
| --- | --- | --- |
| 类别 | 子类别 | 数量 |
| BENIGN | - | 2273097 |
| FTP-Patator | - | 7938 |
| SSH-Patator | - | 5897 |
| DoS | DoS Hulk | 231073 |
| DoS GoldenEye | 10293 |
| DoS slowloris | 5796 |
| DoS Slowhttptest | 5499 |
| Heartbleed |  | 11 |
| Web Attack | Web Attack Brute Force | 1507 |
| Web Attack XSS | 652 |
| Web Attack Sql Injection | 21 |
| Infiltration | - | 36 |
| Bot | - | 1966 |
| DdoS | - | 128027 |
| PortScan | - | 158930 |

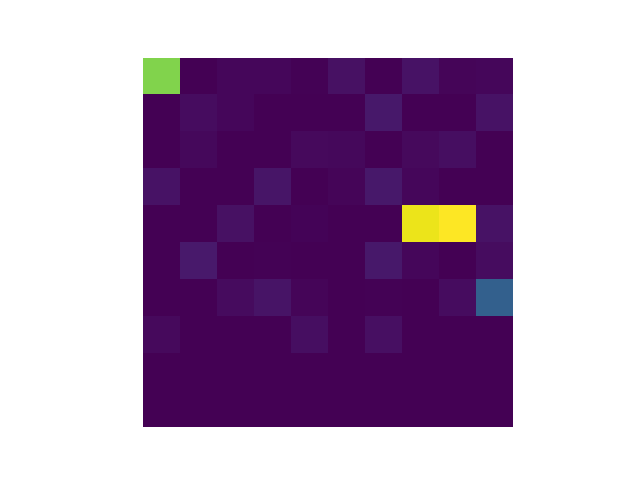
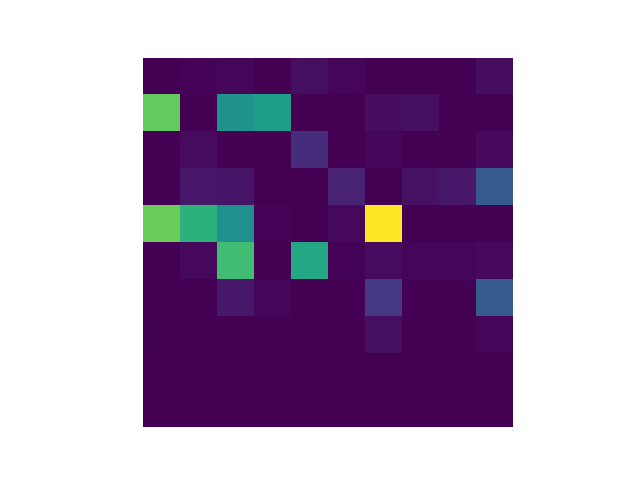
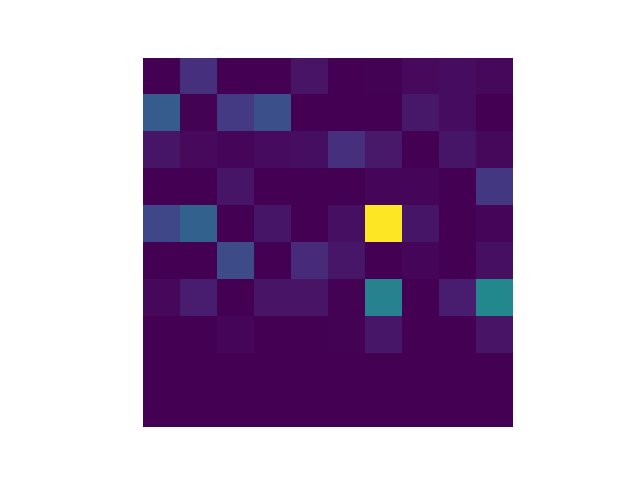
(1)数据预处理

从数据集中可以看到正常流量类数目相对较多，流量呈现高度分布不均，正常流量占到了80%，而对于部分异常流量，其占比不超过1%，对于数据集，我们做了如下调整，剔除了部分数据量极小的异常流量，如Sql Injection类型的异常流量，同时对正常流量抽取了一部分作为实验数据，最后实验数据如表2所示。最后从原来的15个类别变为12个类别。

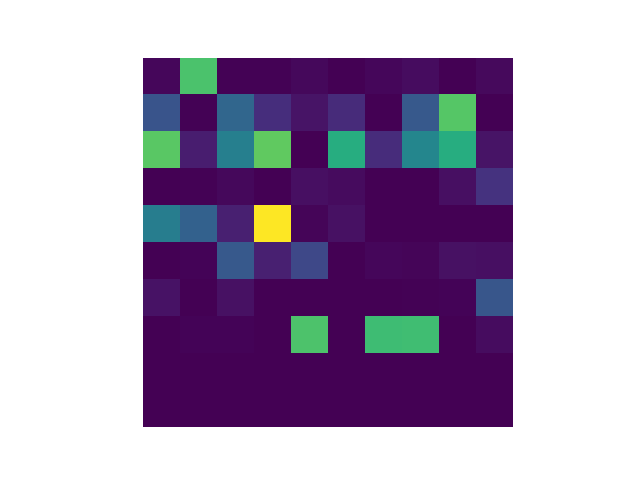
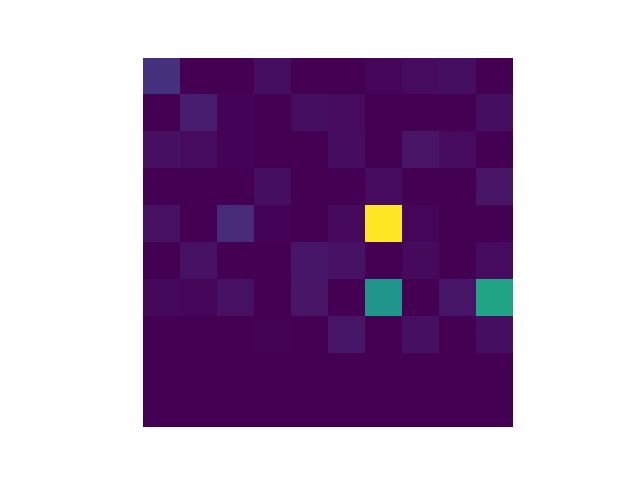
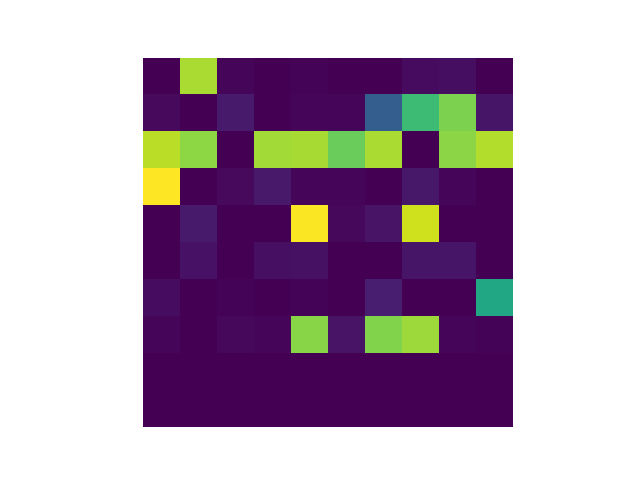
表2 实验数据分布

|  |  |  |
| --- | --- | --- |
| 类别 | 子类别 | 数量 |
| BENIGN | - | 414275 |
| FTP-Patator | - | 7938 |
| SSH-Patator | - | 5897 |
| DoS | DoS Hulk | 231073 |
| DoS GoldenEye | 10293 |
| DoS slowloris | 5796 |
| DoS Slowhttptest | 5499 |
| Web Attack | Web Attack Brute Force | 1507 |
| Web Attack XSS | 652 |
| Bot | - | 1966 |
| DDoS | - | 128027 |
| PortScan | - | 158930 |

每条数据一共有78个属性，刻画了数据包的各种特征，为了利用CNN进行训练，我们在每条数据中补全0，将其扩增为100维，随后转化为大小的灰度图。灰度图样例如图1(a)~(f)所示。

(a) Bot (b) DDos (c) Golden Eye

(d) Hluk (e) Port Scan (f) SSH Patator

图1 灰度图像样例

78列的属性中，均为数值类型，做归一化处理，消除量纲带来的差异：

归一化后，所有的数值均落入区间之间。将数据随机切分，选择70%用于训练使用，剩余30%用作验证集。对于10类标签，使用one-hot编码。

(2)构建CNN网络

构建了一个经典的CNN网络，其具体参数如表3所示。网络结构如图1所示。

表3 CNN 网络具体参数设置

|  |  |  |
| --- | --- | --- |
| layer | output size | kernel |
| conv1 |  | ，32，stride 1 |
| maxpooling |  | ，32，stride |
| conv2 |  | ，64，stride |
| flatten |  | - |
| FC1 |  | Fully Connection |
| FC2 |  | Fully Connection |

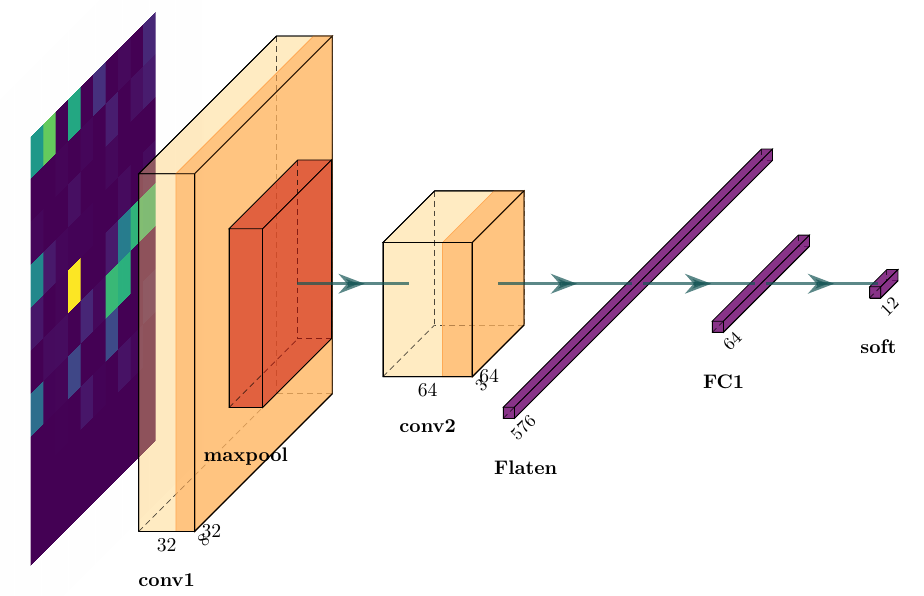


图2 CNN 网络结构

(3) 其余对比模型

对比卷积神经网络分类效果，我们使用了其余5种不同的模型，分别是随机森林、K-近邻算法、支持向量机、多层感知机、朴素贝叶斯、决策树和一般的深度神经网络。

(a).随机森林(Random Forest，RF)

RF对于特征较多的数据时，也可以不用降维，训练速度比较快；同时，在数据集不平衡情况下，其表现效果相对较好。

RF预测结果的混淆矩阵如图2所示。分类效果从混淆矩阵中可以看出，XSS和Brute Force以及Bot三类攻击样本分类相对不是很好，尤其是XSS，因为数据集很少，效果相对不理想。

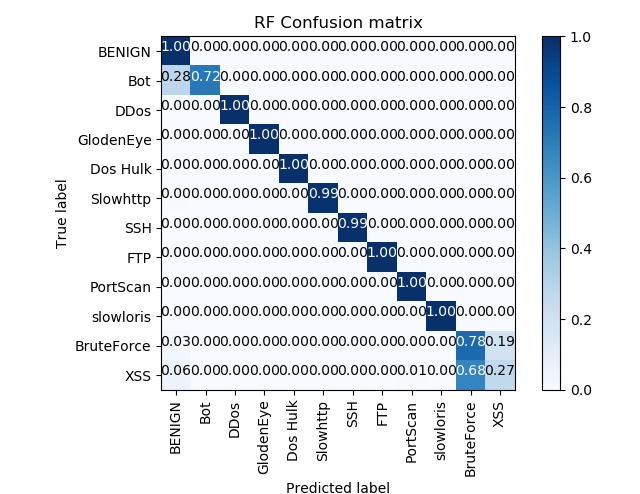


图3 RF 预测结果混淆矩阵

(b) K-近邻算法(K-Nearest Neighbors Algorithm，KNN)

KNN在样本集较大时，其计算量也会随之增大，K值的选择很重要，K值不宜过小，否则容易过拟合；同时K值也不宜过大，否则模型过于简单，忽略了数据中的有效信息。KNN算法的分类效果比RF略好一些，但在XSS攻击类别上同样表现不佳。

对于不同的K的取值，模型表现的精度也不同，实验结果如表所示。其中K取值为3时，模型表现更优。

表4 不同K取值下KNN模型精度

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| K | 平均precision | 平均recall | 平均-score | 加权precision |
| 2 | 0.9167 | 0.8667 | 0.8858 | 0.9960 |
| 3 | 0.9092 | 0.9017 | 0.9033 | 0.9973 |
| 4 | 0.9200 | 0.8725 | 0.8908 | 0.9963 |
| 5 | 0.9067 | 0.8983 | 0.9000 | 0.99703 |

图4为k=3时，KNN模型在测试集上的混淆矩阵。

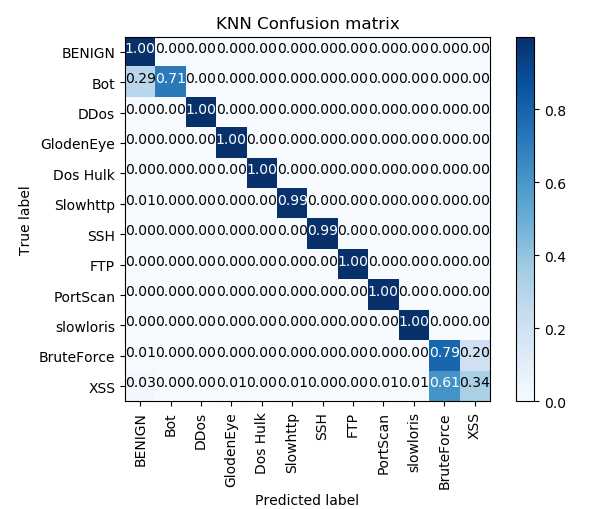


图4 KNN预测结果混淆矩阵

(c) 支持向量机(Support Vector Mechine，SVM)

在训练数据集上，迭代次数分别是200至600时，SVM均不能收敛，并模型的精度相对较低，如表5所示。

表5 不同迭代次数下SVM模型精度

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 迭代次数 | 300 | 400 | 500 | 600 |
| Precision | 0.279 | 0.361 | 0.373 | 0.296 |

虽然使用了高斯核，理论上，在高维空间存在超平面将给定数据分开，但运行时间难以估计。从sklearn文档中得知，当数据量超过10000条时，SVM处理就相对困难了，SVM算法的渐进时间复杂度为，其中为样本数，为特征数。

在本例中，样本数达到了970904个，特征数是78。显然样本数远超10000，因此模型在有限的时间内难以迅速收敛。SVM适用于样本数量有限的情况，其分类效果会比较好，在样本数过多的情况下，相对不适用。

(d) 多层感知机(Multilayer Perceptron)

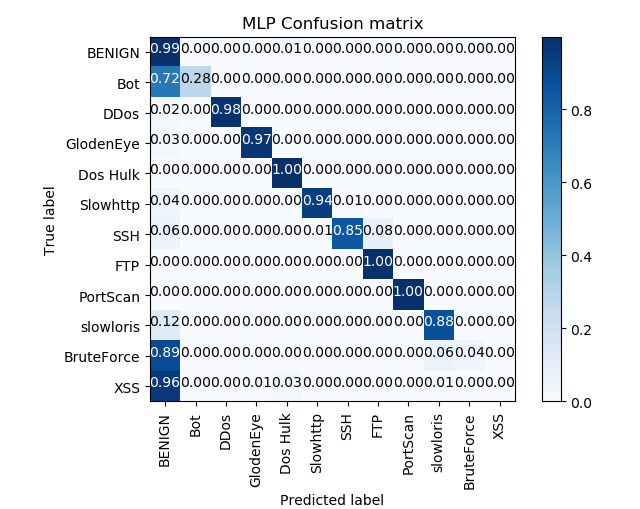


图5 MLP 预测结果混淆矩阵

(e) 朴素贝叶斯(Naïve Bayes，NB)

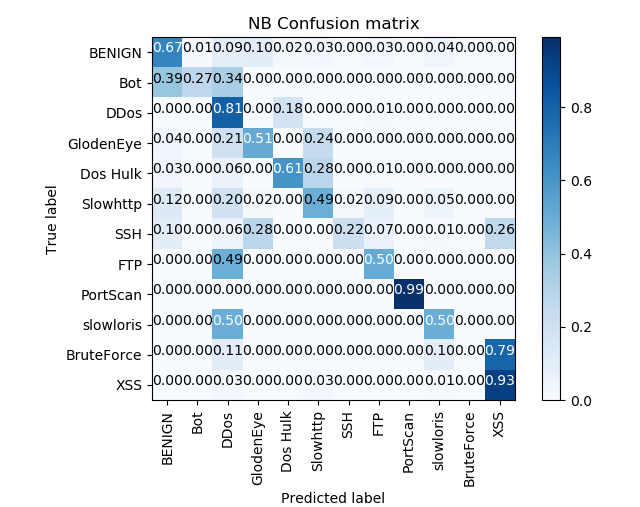


图6 NB预测结果混淆矩阵

(d) 卷积神经网络(CNN，Convolutional Network Neural)

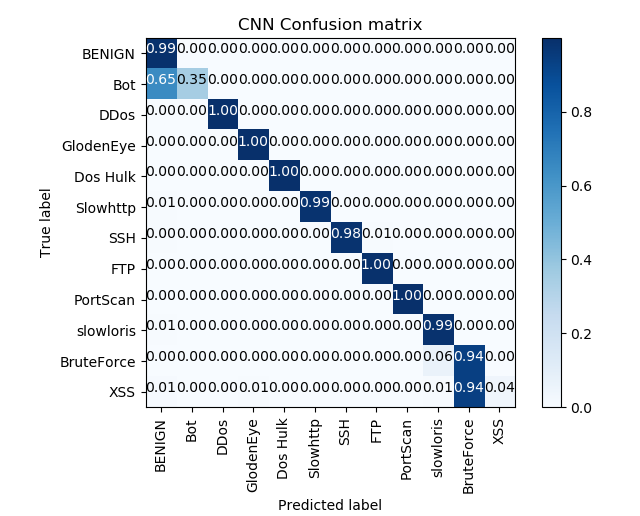


图7 CNN预测结果混淆矩阵

(4) 综合实验对比

|  |  |  |
| --- | --- | --- |
| Network | precision | time(sec) |
| RF | 0.997 | 72 |
| KNN | 0.997 | 1163.18 |
| SVM | 0.373 | 1841.78 |
| MLP | 0.985 | 1438.70 |
| NB | 0.716 | 2.25 |
| CNN | 0.995 | 903.71 |