本项目采用具有无加密流量的诸多现有成果的 KDD99 数据集和包含僵尸网络和加密流量的数据集 CTU13。

KDD99 数据集在单个机器学习模型中训练准确率低，因此项目使用基于朴素贝叶斯模型、决策树模型、KNN 模型的投票器进行集成学习。

下面的结果为单独使用机器学习模型的准确率结果。

```
* 正在执行任务: D:\Py_Lib\python D:\Codefield\Python\exercise\NB-1.py

Best Parameters: {'priors': [0.1, 0.9], 'var_smoothing': 0.0485}
Best CV Score: 0.9250950428209542
Accuracy: 0.7374911284599006
```

KDD99 数据朴素贝叶斯模型训练的准确率

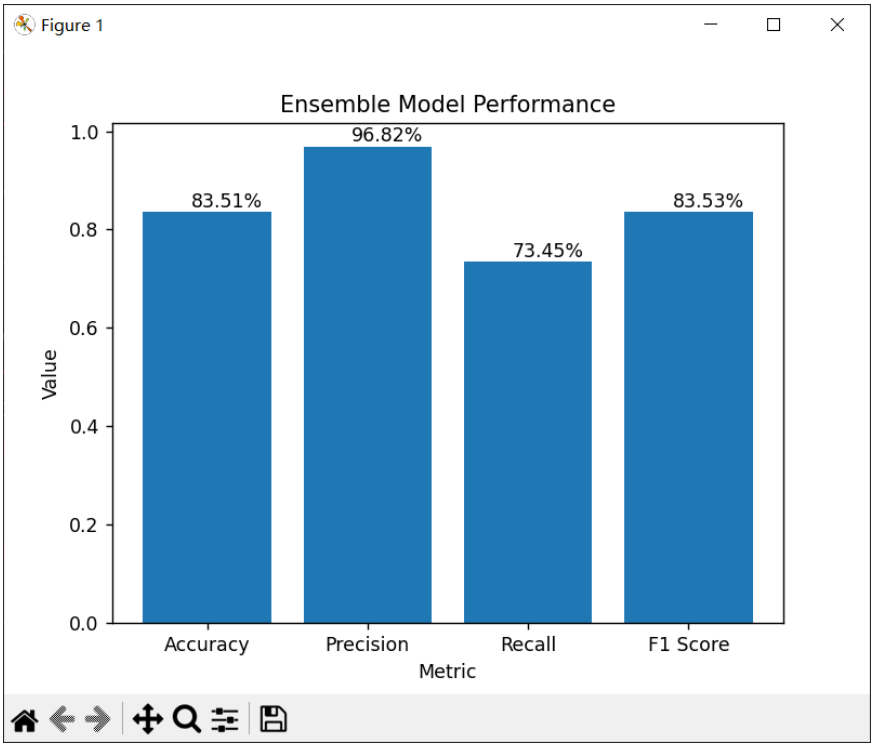
```
Best CV Score: 0.9986901977231402
Accuracy: 0.8010113555713272
```

KDD99 数据 KNN 模型训练的准确率

```
Best CV Score: 0.9986981358060932
Accuracy: 0.7845990063875089
```

KDD99 数据决策树模型训练的准确率

基于每个模型的准确率进行权重分配，构造投票器进行集成学习的成果如下。



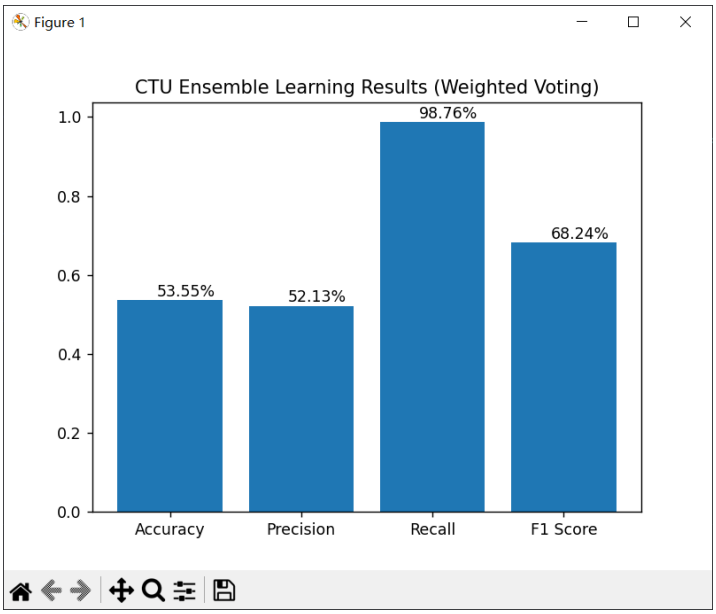
利用 KDD99 数据进行集成学习模型训练的各指标

集成学习的准确率仍不到 90%，采用 CNN 进行训练，使用 TensorFlow 框架搭建模型，卷积层使用 3x3 的卷积核和 32 个输出通道，全连接层将第一层的输出展平成 1024 维向量。同时，为了防止过拟合，使用 dropout 和 L2 正则化技术。其训练成果如下。

```
50 [5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5] [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] 0.589562 0.564
100 [0 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 5 0] [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] 0.9302034 0.921
150 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] 0.9588155 0.948
200 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] 0.97286755 0.974
250 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] 0.97724795 0.98
300 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] 0.9786588 0.975
350 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] 0.9793571 0.98
400 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] 0.9796952 0.978
450 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] 0.9799098 0.985
500 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] 0.9805069 0.98
```

KDD99 数据集 CNN 模型训练的准确率

CTU13 数据集使用基于上述 KDD99 方案的集成学习/CNN 模型效果差，其使用集成学习效果如下。



CTU13 数据集集成学习模型训练的准确率

故采用基于 DNN 的 MOE(Mixture of Experts)模型，每个“专家”网络是一个 DNN，由多个全连接层、ReLU 激活函数和 Dropout 层组成；"门控"网络将输入映射到各个"专家"的权重。

在不处理数据集数据时，其训练效果如下。

```
Epoch [100/100], Step [430/495], Loss: 0.5183
Epoch [100/100], Step [440/495], Loss: 0.5247
Epoch [100/100], Step [450/495], Loss: 0.4825
Epoch [100/100], Step [460/495], Loss: 0.4389
Epoch [100/100], Step [470/495], Loss: 0.5760
Epoch [100/100], Step [480/495], Loss: 0.5562
Epoch [100/100], Step [490/495], Loss: 0.5044
Test Accuracy of the model on the test images: 71.15242732772613 %
```

```
Epoch [100/100], Step [400/495], Loss: 0.4543
Epoch [100/100], Step [410/495], Loss: 0.5428
Epoch [100/100], Step [420/495], Loss: 0.5222
Epoch [100/100], Step [430/495], Loss: 0.5247
Epoch [100/100], Step [440/495], Loss: 0.4544
Epoch [100/100], Step [450/495], Loss: 0.5827
Epoch [100/100], Step [460/495], Loss: 0.4618
Epoch [100/100], Step [470/495], Loss: 0.5739
Epoch [100/100], Step [480/495], Loss: 0.4389
Epoch [100/100], Step [490/495], Loss: 0.4615
Test Accuracy of the model on the test images: 73.8896266784713 %
```

CTU13 数据集未进行特征选择、数据处理前进行模型训练的准确率

其原因是 CTU13 数据集中负例仅占正例 1%不到，且数据集特征冗杂，需要精简。为此，项目采用 SelectKBest 的 chi2 方法（卡方验证）进行打分，保留 8 个最佳特征（Duration,Packets,Total_Bytes,Source_Bytes,ports,Flags,Protocol,Direction）。同时还采用随机欠采样，对正例随机取样，使其和负例数量相近。

优化数据集结构后成果如下。

```
Epoch 484, Loss: 0.24123320113867522
Epoch 485, Loss: 0.24908146534913353
Epoch 486, Loss: 0.24990334955842367
Epoch 487, Loss: 0.25116983978743
Epoch 488, Loss: 0.23958985641864794
Epoch 489, Loss: 0.23079699554613659
Epoch 490, Loss: 0.23798733508946107
Epoch 491, Loss: 0.23763339333901448
Epoch 492, Loss: 0.2304074424252446
Epoch 493, Loss: 0.2366858223187072
Epoch 494, Loss: 0.235616625532774
Epoch 495, Loss: 0.23529702509487313
Epoch 496, Loss: 0.23921525123129997
Epoch 497, Loss: 0.2314795663114637
Epoch 498, Loss: 0.23375156622912202
Epoch 499, Loss: 0.24225288927555083
Epoch 500, Loss: 0.23630767787274506
Saved PyTorch Model State to moe_model.pth
Accuracy: 90.65%
```

优化 CTU13 数据集后进行 MOE 模型训练的准确率

由此可以得出结论，提高恶意流量检测准确率可以采用以下三个策略：

1. 特征选择过滤无用特征
2. 调整数据正负例结构
3. 调整训练模型