

分布式拒绝服务攻击检测的分类算法评估

【摘要】

分布式拒绝服务（DDoS）攻击旨在通过恶意流量耗尽目标网络资源，这对服务的可用性构成威胁。在过去二十年中，随着互联网的发展，已提出许多检测系统，特别是入侵检测系统（IDS）。尽管如此，用户和组织在应对 DDoS 时仍面临持续挑战。IDS 是保护关键网络免受不断演变的侵入活动的第一道防线，但必须始终保持最新状态，以检测任何异常行为，从而维护服务的完整性、机密性和可用性。然而，新检测方法、技术和算法的准确性在很大程度上依赖于为训练目的和通过创建分类器模型进行评估而设计的精心设计的数据集。在这项工作中，我们使用主要的监督分类算法进行实验，以准确区分 DDoS 攻击和合法流量。在所有分类器中，基于树的分类器和基于距离的分类器表现最佳。

【关键词】

机器学习，DDoS，逻辑回归，朴素贝叶斯，SVM，决策树，随机森林，K-最近邻

引言

DDoS 攻击已成为最严重的网络侵入行为之一，对计算机网络基础设施和各种基于网络的服务构成严重威胁[1]。它们之所以突出，是因为可以轻易发起，给组织带来灾难性的损失，而且很难追踪并找出真正的攻击者。DDoS 攻击通过耗尽网络资源来攻击网络的可用性，导致服务拒绝，并且在过去几年中，无论是在数量上还是在体量上都迅速增加。攻击持续时间更短、数据体量更大的趋势变得越来越普遍[6]。大多数现有工作使用 KDD Cup '99 数据集[2]或 DARPA 数据集[3]来检测 DDoS 攻击。然而，随着时间的推移，网络犯罪和攻击以巧妙的方式侵入目标环境。因此，使用包含所有新型攻击签名的最新数据集来训练分类器，将提高分类器的性能。我们使用 CICDDoS2019 数据集进行我们的分析[4]。

我们工作的目标是通过使用 CICDDoS2019 数据集的训练模型来实现多个监督分类器来检测 DDoS 攻击。我们的重点是以更高的准确性减少假报，最终有助于提高生产系统的正常运行时间，以及组织的声誉。

背景与相关工作

基于 Web 服务器日志捕获的特征，如平均数据包大小、入站比特率与出站比特率、源

IP 与目标 IP 及其端口等[5]，可以检测网络流量是否异常。主要有两种类型的拒绝服务攻击。第一种是网络级 DoS 攻击，它耗尽网络资源，因此禁用了实际用户的连接性；另一种攻击是应用级 DoS 攻击，其中服务器资源耗尽，合法用户请求被拒绝。在 DDoS 攻击中，攻击者控制了多个称为僵尸网络的机器，从这些机器上运行称为机器人代码的脚本，并攻击受害者服务器。

有两个主要类别。第一种是反射攻击，另一种是利用攻击。在反射攻击中，攻击者的身份保持不露面，而在利用攻击中，情况并非如此。反射攻击和利用攻击都可以通过应用协议和传输层协议或两者的组合来实现。基于 TCP 的反射攻击包括 MSSQL、SSDP，而基于 UDP 的反射攻击包括 CharGen、NTP、TFTP。

Kurniabudi 分析了巨大网络流量的相关和显著特征。Ring 等人已经确定了 15 个不同的属性来访问单个数据集[8]的适用性。Idhamerd 描述了基于网络熵估计、聚类、信息增益比和树算法[9]的 DDoS 检测半监督 ML 方法。[10]的研究人员提出了 INDB（使用朴素贝叶斯的入侵检测）机制来检测入侵数据包。使用朴素贝叶斯算法背后的原因是它的可预测性特征。Alenezi 和 Reed 在[11]中提出了 IDS 的广泛分类。讨论了 DoS/DDoS 攻击的困难性和特点，并采用三种不同的分类方法对数据进行了分析。Alpna 和 Malhotra 在 KNN 和随机森林[12]的帮助下开发了检测 DDoS 攻击的架构。Singh 等人开发了一种改进的 SVM 算法，用于检测网络攻击[13]。目前存在许多涉及 DDoS 攻击检测的相关工作。然而，这些研究大多评估的是有特定的分类算法的数据集，并试图使用较旧的数据集，如 KDDCup'99 [2]或 DARPA [3]，得到更优性能[14-16]。在本文中，我们使用最近的数据集 CICDDoS2019 [5]对六种不同的分类算法进行了比较分析。

数据集与方法论

数据集包含七个 csv 文件，数据量超过 10GB。我们应用特征提取算法找出最重要的特征，并执行数据预处理技术，如数据清洗、规范化、移除无穷大值。一旦模型准备好，就通过测试集进行访问，通过测量准确度、精确度、召回率、f1 分数、真正例和真负例来评估。如果准确度不可接受，则对每种分类算法进行优化。此外，还分析了训练测试分割比例。

DDoS 攻击通常通过僵尸网络或多个机器人发生。因此，在目标服务器接收数据包时，会有多个 IP 地址或 MAC 地址，但像数据包长度、流量持续时间、前向方向的总数据包数这样的属性使我们能够识别它为真实请求或恶意请求。为了比较数据包，可以应用数据挖掘技术来测量概率或发生次数，以对数据包进行分类。在这里，我们使用以下六种机器学习算法

对异常流量进行分类：逻辑回归、支持向量机、朴素贝叶斯、K 最近邻、决策树和随机森林。

在我们的实验中，我们使用了由 New Brunswick 大学创建的包含 88 个特征的数据集。该数据集可在加拿大网络安全研究所的网站[5]上公开获得。我们收集了不同类型的攻击的数据，如 Portmap、LDAP、MSSQL、UDP、UDPLag 等。如果请求来自合法用户，那么它将被标记为“良性”，否则将被标记为特定的攻击名称。数据集是为分析目的明确创建的，每天都组织起来。CIC 每天都记录了原始数据，包括来自每台服务器机器的网络流量和事件日志。实际的数据集有超过 88 个特征，但 CIC 本身已经做了降维，他们使用了 CIC 流量计-V3[17]，并生成了最重要的 88 个特征进行分析，并提供了 csv 文件。如果有人想通过自己的方式提取特性，可以使用他们共享的 PCAP 文件。

我们已经在这个数据集上做了两种类型的实验。最初我们抽样数据集，选择随机选择 30000 行从每个 csv 文件，加起来 200000 行我们的数据分析样本，这是不平衡数据集。第二个实验我们获得相同数量的良性和攻击数据元组从每个 csv 文件数据集，导致一个完全平衡的训练和测试数据集。

表 1 显示了每个文件包含的记录与正常类的总数，例如标签=“良性”。关于数据集的更多细节可以在[18]上找到。在训练模型之前，将数据集中的 IP 地址转换为数字整数。

CSV File Name	Total Rows	Benign Rows
LDAP	2113234	5124
MSSQL	5775786	2794
NetBIOS	3455899	1321
Syn	4320541	35790
UDP	3782206	3134
UDPLag	725165	4068
Portmap	191694	4734
Total	20364525	56965

表 1 标记数据的分布

我们选择了单变量选择技术。它是一种统计测试，可以用于选择那些与输出标签关系最强的特性。scikit-learn 库提供了 SelectkBest 类，帮助我们实现算法，并给出与类标签最相关的特性的结果。我们已经使用了前 25 个特性来训练我们的模型。为了获得数据集的每个特征的重要性，我们使用了基于树的分类器附带的特征重要性内置类。图 1 展示了前 15 个最重要的特性。

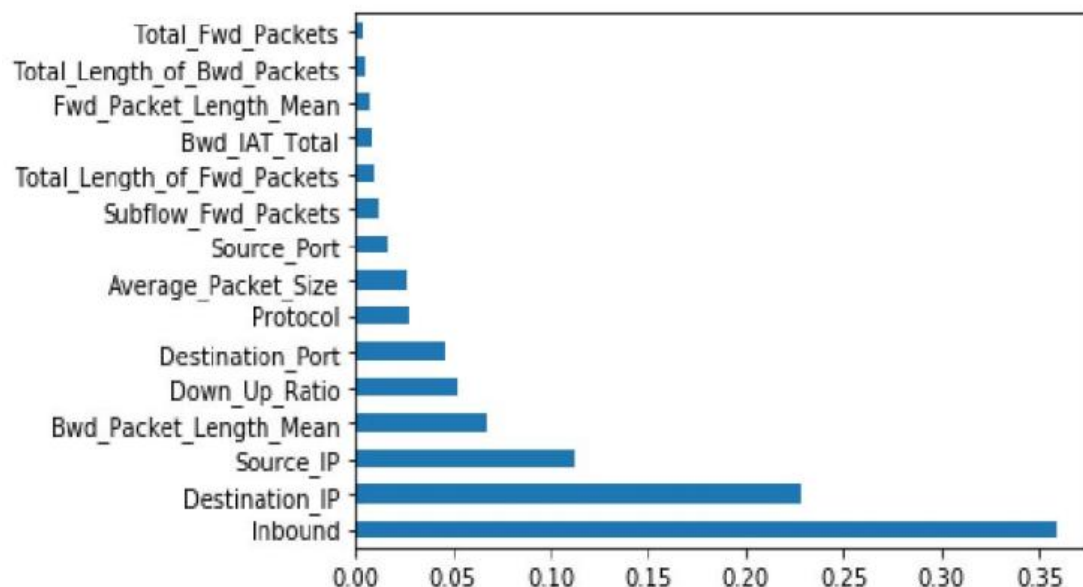


图 1 特征的重要性

实验结果与讨论

1. 评估指标

为了评估分类器的性能，我们使用了基于混淆矩阵的主要性能指标。这个矩阵包含了 ML 模型进行的真实和预测分类的信息。为了公平起见，我们还在结果表中包括了真正例、真负例、假正例和假负例的值。如前一节所述，我们在不平衡数据集和平衡数据集上实现了六种不同的机器学习分类算法。我们使用 Python 的 scikit-learn 库实现。

2. 实验

我们对每个单独的 7 个 csv 数据文件进行了随机抽样，从每个文件中选择 30K、40K 和 50K 个元组，只是为了测量良性流量与攻击流量的比例。实际数据集的良性流量数量较少，而在抽样时，这本身就是有偏见的。平均而言，当使用不平衡数据训练模型时，良性流量与攻击标签相比大约有 0.5%到 0.7%。表 2 显示了类标签的分布。

Sample	Attack (1)	Benign (0)
30K	208710	1263
40K	278302	1698
50K	347780	2220

表 2 当随机选择数据集时，类标签的分布

为了避免对分类模型准确性的偏差问题，我们还创建了平衡数据集，其中我们从每个 7-csv 文件中选择所有的良性流量，并从攻击流量中随机抽取相同数量的元组。我们最终从

所有攻击和良性数据的文件中收集了 105042 行。由于这个数字非常小，我们在现有的数据帧中再次添加相同的数据，以使训练集的大小超过 200K 行，这可以与不平衡的数据集相比较。

3. 结果

每个分类器都使用准确度分数和其他评估指标如精确度、召回率和 f1 分数进行了评估。表 3 显示了不平衡数据集的总体准确度，表 4 显示了平衡数据集的输出结果。数据是基于五轮观察中的最佳值选择的。

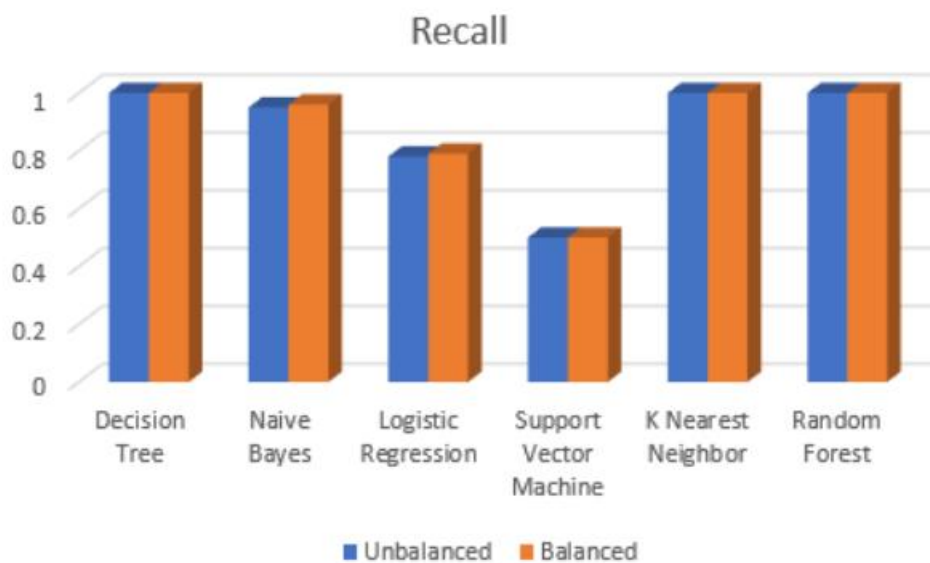
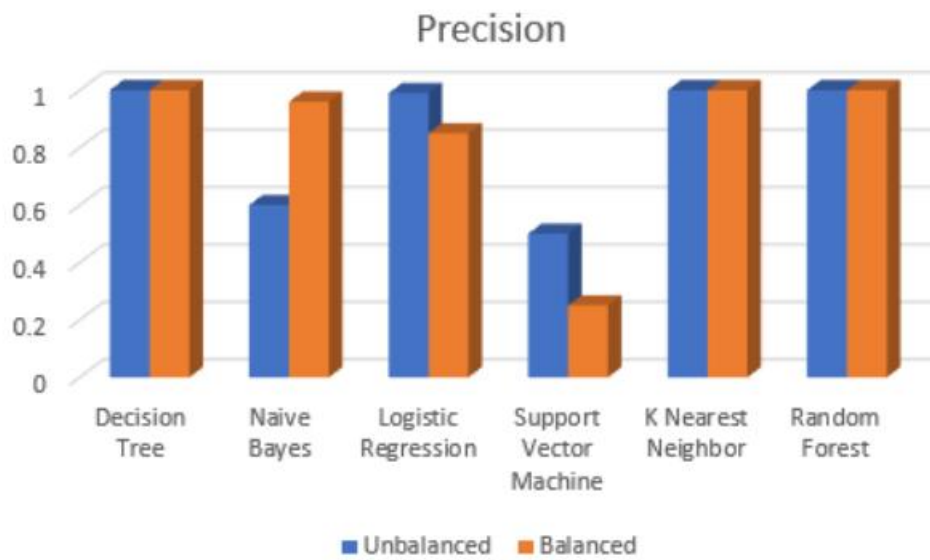
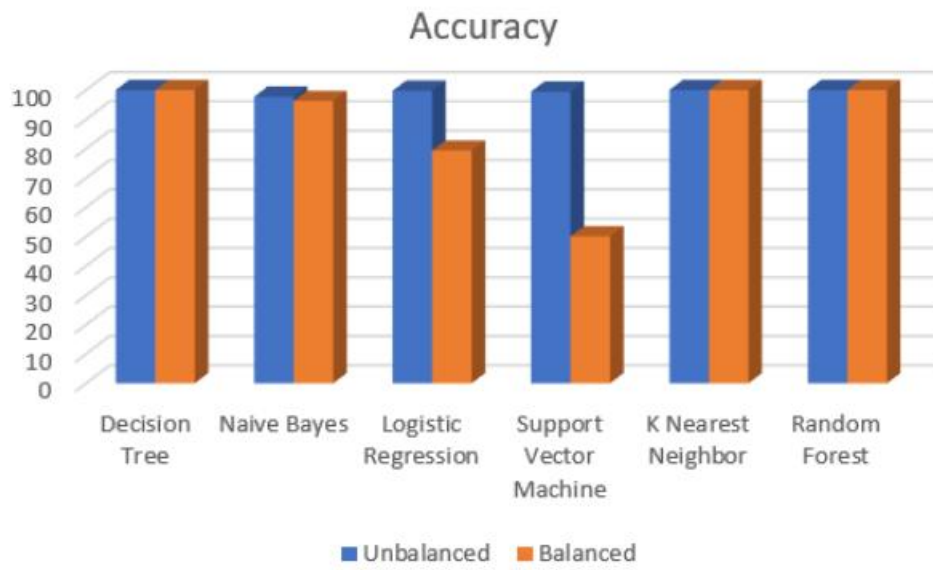
Unbalanced Dataset	TP	TN	FN	FP	Accuracy	macro avg		
						Precession	Recall	F1 Score
Decision Tree	62599	398	3	0	99.99523	1	1	1
Naive Bayes	61199	370	31	1400	97.72857	0.6	0.95	0.66
Logistic Regression	62619	213	164	4	99.73333	0.99	0.78	0.86
Support Vector Machine	62663	0	337	0	99.46507	0.5	0.5	0.5
K Nearest Neighbor	62598	401	0	1	99.99841	1	1	1
Random Forest	62602	397	0	1	99.99841	1	1	1

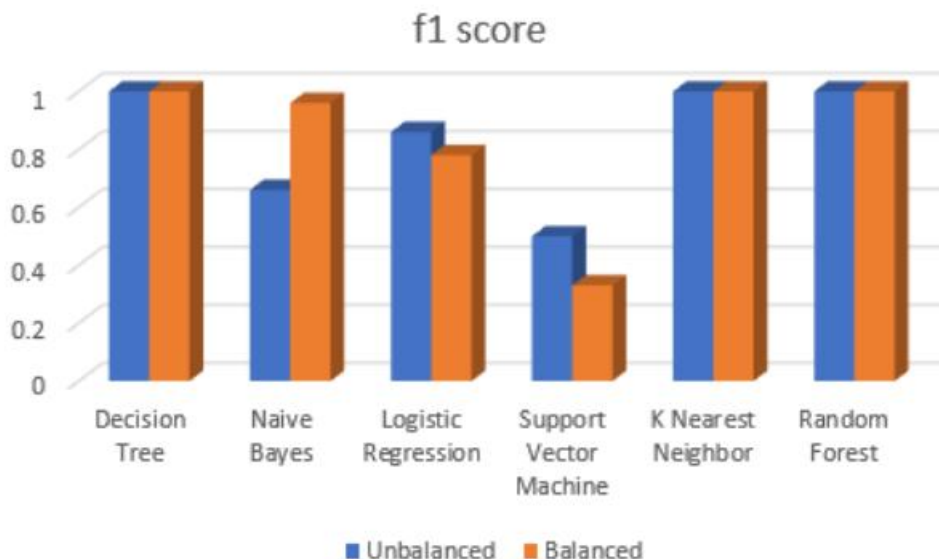
表 3 不平衡数据集的结果

Balanced Dataset	TP	TN	FN	FP	Accuracy	macro avg		
						Precession	Recall	F1 Score
Decision Tree	31577	31449	0	0	100	1	1	1
Naive Bayes	31387	29278	2290	71	96.25392	0.96	0.96	0.96
Logistic Regression	31276	8730	12819	201	79.34185	0.85	0.79	0.78
Support Vector Machine	31577	0	31449	0	50.10154	0.25	0.5	0.33
K Nearest Neighbor	31477	31549	0	0	100	1	1	1
Random Forest	31477	31549	0	0	100	1	1	1

表 4 平衡数据集的结果

由于不平衡数据集偏向攻击类别，所有分类算法的准确度都非常高。但这并不能帮助我们做出选择 DDoS 攻击检测最佳执行算法的决策。这里，除了朴素贝叶斯之外，所有算法在不平衡数据上表现都非常出色。相反，我们注意到平衡数据集的准确度有轻微的变化。如表 4 所示，基于树的算法如决策树、随机森林和基于距离的分类算法 k-最近邻表现最佳，而朴素贝叶斯给出了良好的准确度，但其余的分类技术——SVM 和逻辑回归表现较差。图 2 显示了每种分类算法在不平衡数据集和平衡数据集之间的准确度得分比较。此外，图 3、图 4 和图 5 分别显示了不平衡和平衡数据集之间的精确度、召回率和 F1 分数的比较。





在分析输出后，基于树的分类算法如决策树和随机森林，以及基于距离的分类算法在两种类型的数据集上表现最佳，并且几乎达到了 100% 的准确度。即使在考虑其他指标时，这三种分类器也表现最佳。然而，当每个分类器的参数发生变化时，可以注意到性能有轻微的变化。在这里，我们尝试为每种算法找到最佳性能。

未来工作建议








虽然我们的初步实验结果是有希望的，但这项工作可以在多个方向上扩展：a) 在我们的实验中，由于硬件限制，我们只使用了略多于 200,000 行的数据。将来，我们可以计划选择超过 100 万行的数据集。这将为我们提供更准确的训练模型进行预测。b) 我们可以根据不同类型的 DDoS 攻击进行数据挖掘，因为 Portmap 可能可以通过 K-NN 高效检测，但对于 UDPlag，朴素贝叶斯可能更好。如果这一点得到证实，那么我们可以把所有单独的模型合并到一个单一的模型中，以获得所有类型 DDoS 攻击的近乎 100% 的准确度。c) 我们可以尝试不同的特征选择技术。

结论

在本文中，我们使用了 CICDDoS2019 数据集，这是一个相当新的数据集，包括 DDoS 的最新攻击签名。我们使用主要的监督分类算法进行实验，以从合法流量中准确分类攻击。在所有分类器中，决策树、随机森林和 K-最近邻表现最佳。虽然初步结果很有希望，但我们计划通过扩展数据集并针对不同类型的 DDoS 攻击来扩展这项工作。我们将在未来的工作中专注于这些方向。

附：复现基于平衡数据集的分类模型实验

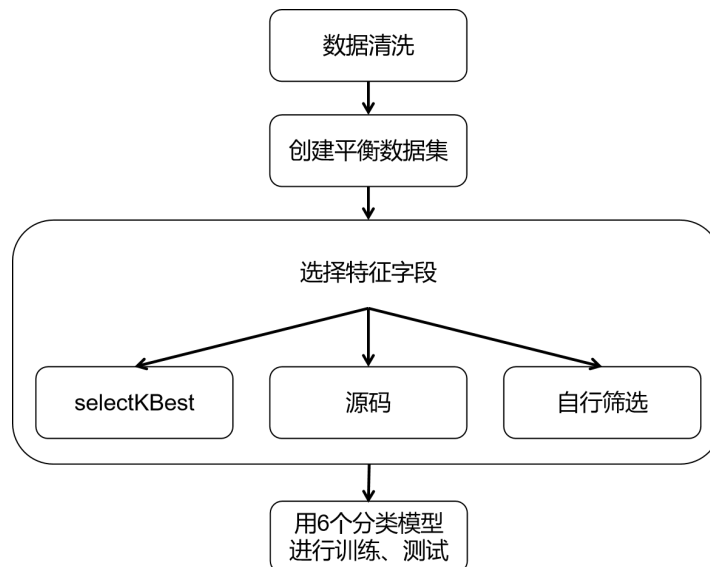
文章中提及，使用的数据集是 7 个 csv 文件，数据量超 10GB。数据集（<https://www.unb.ca/cic/datasets/ddos-2019.html>）有一个名为“CSV-03-11.zip”的压缩包正好与文章描述相符，因而可以判断作者使用这 7 个 csv 文件数据进行训练、测试：

-  LDAP.csv
-  MSSQL.csv
-  NetBIOS.csv
-  Portmap.csv
-  Syn.csv
-  UDP.csv
-  UDPLag.csv

对每个 csv 文件进行数据清洗后，合并到 `ddos_dataset.csv` 文件中（文件过大，没有上传到 github 仓库中）。

按照文章的思路，先提取出所有良性（BENIGN）流量数据，随机选择同等数量的非良性流量数据，构成平衡数据集 `balanced_dataset.csv`。由于本实验不需要和不平衡数据集的结果作比较，没有按照文章思路来对数据进行复制处理。

至此，可以开始使用各个机器学习算法模型进行实验。特征字段的选择很重要，本人用相同的模型参数，对不同的特征字段选择分别做了实验。



1. 使用 `selectKBest` 筛选出最重要的 25 个特征字段

按照文章的思路，利用所有的数据，即 `ddos_dataset.csv`，用 `selectKBest` 训练出最重要

的 25 个特征:

'Source IP',
'Destination IP',
'Timestamp',
'Flow Duration',
'Fwd Packet Length Std',
'Bwd Packet Length Max',
'Bwd Packet Length Min',
'Bwd Packet Length Mean',
'Bwd Packet Length Std',
'Bwd IAT Total',
'Fwd PSH Flags',
'Packet Length Std',
'Packet Length Variance',
'RST Flag Count',
'URG Flag Count',
'CWE Flag Count',
'Down/Up Ratio',
'Avg Bwd Segment Size',
'Init_Win_bytes_forward',
'Init_Win_bytes_backward',
'Active Mean',
'Active Min',
'Idle Mean',
'Idle Min',
'Inbound',
'Label'

6 个分类模型的测试结果如下:

	TP	TN	FP	FN	Accuracy	Precision	Recall	f1-score
Decision Tree	16872	0	2	16910	99.99408	1.00	1.00	1.00
Naive Bayes	15596	1276	406	16506	95.02131	0.95	0.95	0.95
Logistic Regression	14262	2610	242	16670	91.55813	0.92	0.92	0.92
Support Vector Machine	15353	1519	10	16902	95.47419	0.96	0.95	0.95
K Nearest Neighbor	1672	0	0	16912	100.0	1.00	1.00	1.00
Random Forest	16872	0	1	16911	99.99704	1.00	1.00	1.00

能看出 Naive Bayes、Logistic Regression、Support Vector Machine 这三个模型准确度相比

另外三个较低，与文章的描述一致。

总体来说，各个模型的准确度都很高，均在 90%以上，和文章的结果相差很大。对照文章中图 1 给出的 15 个特征字段，本人训练得到的特征字段不完全重合，于是猜测数据清洗或者模型参数与作者使用的不一致。

2. 作者 Github 仓库源码中使用的 20 个特征字段

文章中说使用了 selectKBest 训练得到的 25 个特征字段，但是图 1 只给出了 15 个特征字段。于是尝试找到文章的源码，但源码中没有给出 selectKBest 的训练过程，而是直接列出了 20 个特征字段，而且不是文章中所说的 25 个特征字段。本人有些疑惑，但也使用了这 20 个特征对 6 个分类模型进行训练、测试。

```
'Destination IP',
'Flow Duration',
'Source IP',
'Total Length of Bwd Packets',
'Bwd IAT Mean',
'Fwd IAT Mean',
'Flow IAT Mean',
'Destination Port',
'Bwd Packet Length Mean',
'Source Port',
'Average Packet Size',
'Total Backward Packets',
'Subflow Bwd Packets',
'Fwd Packet Length Mean',
'Packet Length Mean',
'Total Fwd Packets',
'Subflow Fwd Packets',
'Total Length of Fwd Packets',
'Down/Up Ratio',
'Protocol',
'Label'
```

6 个分类模型的测试结果如下：

	TP	TN	FP	FN	Accuracy	Precision	Recall	f1-score
Decision Tree	16872	0	2	16910	99.99408	1.00	1.00	1.00
Naive Bayes	15593	1279	406	16506	95.01243	0.95	0.95	0.95
Logistic Regression	10088	6784	374	16538	78.81246	0.84	0.79	0.78

Support Vector Machine	15280	1592	10	16902	95.25811	0.96	0.95	0.95
K Nearest Neighbor	16872	0	0	16912	100.0	1.00	1.00	1.00
Random Forest	16872	0	2	16910	99.99408	1.00	1.00	1.00

SVM 的准确度有 95%，而文章中只有 50%。原因可能是：本人所找到的源码不是最终的源码，作者使用的 25 个特征字段对 SVM 分类影响很大；平衡数据集的建立有随机性，而模型的训练和测试非常依赖于这些数据。

3. 自行筛选特征字段

在数据清洗阶段，本人已经把 Flow ID、SimillarHTTP 这两个特征字段筛除。

个人认为，还有几个特征字段对 DDoS 检测没有太大意义，需要筛除：Unnamed: 0、Source IP、Source Port、Destination IP、Destination Port、TimeStamp、Inbound。还有 Fwd Header Length.1 内容和 Fwd Header Length 是相同的，也筛除。剩下的特征字段都保留，对 6 个分类模型进行训练、测试。

Protocol: 传输协议。

Flow Duration: 流量持续时间，从第一个数据包到达到最后一个数据包的时间长度。

Total Fwd Packets: 前向（从源到目的）数据包的总数。

Total Backward Packets: 后向（从目的回源）数据包的总数。

Total Length of Fwd Packets: 前向数据包的总字节数。

Total Length of Bwd Packets: 后向数据包的总字节数。

Fwd Packet Length Max/Min/Mean/Std: 前向数据包长度的最大值、最小值、平均值和标准差。

Bwd Packet Length Max/Min/Mean/Std: 后向数据包长度的最大值、最小值、平均值和标准差。

Flow Bytes/s: 流量的每秒字节数，即数据传输速率。

Flow Packets/s: 流量的每秒数据包数。

Flow IAT Mean/Std/Max/Min: 流量的往返时间（Inter-Arrival Time）的平均值、标准差、最大值和最小值。

Fwd IAT Total/Mean/Std/Max/Min: 前向往返时间的总和、平均值、标准差、最大值和最小值。

Bwd IAT Total/Mean/Std/Max/Min: 后向往返时间的总和、平均值、标准差、最大值和最小值。

Fwd PSH Flags/Bwd PSH Flags: 前向和后向数据包中的 PSH（Push）标志的数量，表示数据应该立即被处理。

Fwd URG Flags/Bwd URG Flags: 前向和后向数据包中的 URG（Urgent）标志的数量。

Fwd Header Length/Bwd Header Length: 前向和后向数据包的头部长度。

Fwd Packets/s/Bwd Packets/s: 前向和后向的每秒数据包数。

Min Packet Length/Max Packet Length: 数据包长度的最小值和最大值。

Packet Length Mean/Std/Variance: 数据包长度的平均值、标准差和方差。

FIN Flag Count/SYN Flag Count/RST Flag Count/PSH Flag Count/ACK Flag Count/URG Flag Count/CWE Flag Count/ECE Flag Count: 分别表示 TCP 的结束连接标志、建立连接标志、重置连接标志、推标志、确认标志、连接结束标志和拥塞窗口控制标志的数量。

Down/Up Ratio: 下行（目的到源）与上行（源到目的）流量的比例。

Average Packet Size: 平均数据包大小。

Avg Fwd Segment Size/Avg Bwd Segment Size: 平均前向和后向段大小。

Fwd Avg Bytes/Bulk/Bwd Avg Bytes/Bulk: 前向和后向的数据块平均字节数。

Fwd Avg Packets/Bulk/Bwd Avg Packets/Bulk: 前向和后向的数据块平均数据包数。

Fwd Avg Bulk Rate/Bwd Avg Bulk Rate: 前向和后向的数据块平均速率。

Subflow Fwd Packets/Subflow Fwd Bytes/Subflow Bwd Packets/Subflow Bwd Bytes: 子流的前向数据包数、前向字节数、后向数据包数和后向字节数。

Init_Win_bytes_forward/Init_Win_bytes_backward: 前向和后向的初始窗口字节数。

act_data_pkt_fwd: 前向活跃数据包的数量。

min_seg_size_forward: 前向最小段大小。

Active Mean/Std/Max/Min: 活跃时间段（数据传输时）的平均值、标准差、最大值和最小值。

Idle Mean/Std/Max/Min: 空闲时间段（无数据传输时）的平均值、标准差、最大值和最小值。

	TP	TN	FP	FN	Accuracy	Precision	Recall	f1-score
Decision Tree	16869	3	4	16908	99.97928	1.00	1.00	1.00
Naive Bayes	3510	13362	412	16500	59.22922	0.72	0.59	0.52
Logistic Regression	16673	199	5237	11675	83.90954	0.87	0.84	0.84
Support Vector Machine	16783	89	7384	9528	77.88006	0.84	0.78	0.77
K Nearest Neighbor	16764	108	54	16858	99.52048	1.00	1.00	1.00
Random Forest	16871	1	5	16907	99.98224	1.00	1.00	1.00

Decision Tree、KNN、Random Forest 的结果接近 100%，与文章结果相近。

Naive Bayes、Logistic Regression、SVM 的表现较差，也与文章出入较大，但总的来说，相比前两次的结果，本次结果更能印证文章的结论——“基于树的分类算法如决策树和随机森林，以及基于距离的分类算法在两种类型的数据集上表现最佳，并且几乎达到了 100%的准确度”。

实验总结

本人对文章所描述的平衡数据集相关的操作进行了复现,包括数据清洗、特征字段选择、6 种分类模型,但没有完全复现文章中呈现的结果。本人认为有以下几点原因:

1. 数据清洗过程存在差异;
2. 平衡数据集的建立具有随机性;
3. 特征字段的选择不同;
4. 模型参数设置不同。

不过从本人的 3 个测试的结果来看,文章的结论是正确的——“基于树的分类算法如决策树和随机森林,以及基于距离的分类算法在两种类型的数据集上表现最佳,并且几乎达到了 100%的准确度”。

文章大体的思路很清晰,可是在复现过程中,发现文章有以下这些问题:

1. 细节描述不清,比如,具体使用哪 25 个特征字段,模型使用了什么参数;
2. 没有解释为什么决策树、随机森林、KNN 算法会比其他三种好,而仅仅从测试结果出发来作此结论。