2017-SIGCOMM(Big-DAMA)-Cluster-Based Load Balancing for Beter Network Security

ABSTRACT

在大数据时代，流量正在迅速增加。 因此，通常使用缩放方法。 例如，由多个实例（扩展方法）组成的设备，以及在它们之间分配传入流量的负载平衡器。 虽然最常见的负载平衡方式基于循环法，但某些方法根据设备特定的功能优化实例之间的负载。 例如，扩展代理服务器的负载平衡可以提高缓存命中率。

在本文中，我们提出了一种基于机器学习的安全设备的新型负载平衡方法。 我们提出的负载均衡器使用群集方法，同时保持所有网络安全设备实例的均衡负载。 我们证明，与传统的负载平衡器相比，我们的方法具有可扩展性并可提高实例的机器学习性能。

INTRODUCTION

负载平衡技术在企业和数据中心网络中很普遍，主要用于支持扩展服务。 传统上，负载均衡器（LB）用于Web服务器之间的流分配，其中入口流需要分布在服务器集群中。 负载平衡通常由循环（例如，Round-Robin DNS [13]），IP级别（例如，[3]）或根据当前服务器的负载[2]来使用。 在层4和层5上采用其他负载平衡器（LB）（例如，[6,22]）。 LB还用于其他网络服务，例如网络代理服务器。 这样的LB通常基于代理服务器的缓存内容，其目标是提高缓存命中率而不是实现平衡负载（例如[32]）。

基于机器学习（ML）的网络安全设备（例如，网络入侵检测系统-NIDS）本质上不同于传统的网络服务，例如web和代理服务器。 Web服务器响应查询; 和代理服务器缓存数据，同时提供高缓存命中率。 另一方面，网络安全设备生成统计数据，维持不同阶段（培训/非培训），并在培训阶段根据收集的统计数据生成预测或分类。 因此，优化由多个实例组成的网络安全设备的性能需要不同的负载平衡考虑因素。

基于NIDS的机器学习误用检测（其搜索已知的侵入模式）已被广泛研究和呈现。 例如：Naive Bayes [21]，Hidden Naive Bayes [16]，支持向量机（SVM）[10,18]，K-Nearest Neighbors [1]，Decision Trees [8,15]，Random Forest [33]， 多层感知器（MLP）[20]，反向传播神经网络[5]，自组织映射（SOM）[14]，以及演化模糊神经网络[4]。

目前关于基于误用检测的NIDS的大多数工作都提出了一种集中式解决方案，其中使用单个中央设备来分析和检测所有入口网络流量的模式。 然而，在大数据时代，单个设备训练，处理和聚类/预测所有数据中心流量是不可行的或极其昂贵的。 为了应对增加的入口流量，其他工作提出了分布式ML算法（例如，[11,29]），它们在它们之间共享信息。 另一种方法是使用NIDS设备，该设备由几个“基于ML的集中式”实例组成，其中每个实例独立地分析网络流量的子集。 因此，实例之间不需要共享信息（状态或同步）。 实际上，这种方法可以利用以前的集中误用NIDS解决方案。

在本文中，我们为后一种方法提出了一种基于集群的负载均衡器，旨在最大化NIDS的误用检测性能，同时在其实例之间保持相对平衡的负载。 我们的负载均衡器通过使用ML聚类算法将一组类似的流分配给同一个实例。 根据给定的特征（例如，相同的源子网），通过在它们之间具有一些相关性来定义“相似流”。

MOTIVATION

如今，网络需要支持高容量需求并根据需要进行扩展。 此外，SDN和NFV范例正在将业界从使用特定于供应商的物理网络设备转移到部署在标准COTS服务器上的虚拟设备。 因此，所有网络功能（包括NIDS）都在虚拟化和扩展。

有几种已知的扩展服务的方法：

- 放大。 通过使用更强大的服务器和更多内存和计算资源来支持不断增长的带宽需求。 然而，这种方法可能是不可行的，因为：a）一些机器学习算法不能并行化，并且单个核心的计算容量是有限的。 b）与处理的网络需求相比，训练时间可能比线性增长更快。
- 通过分布式算法在同一NIDS设备的多个实例上进行扩展[11,29]。 这种方法在设计上更具可扩展性; 但是，它会导致额外的网络开销，以便同步设备的实例。
- 通过在多个独立的NIDS实例上对网络流量进行负载平衡来扩展。 负载平衡器应该以最大化整体学习性能（即，对应于检测质量）的方式在实例之间分配流量，同时尽可能保持平衡负载。

在本文中，我们关注后一种方法。 我们认为传统的负载平衡方法（例如，循环[7]和均匀随机流分布），其目的是在实例上具有相等的负载，不适用于基于机器学习的NIDS。 这种传统方法降低了整体机器学习性能（就普遍的ML度量而言：准确性，精确度，召回率，F分数和曲线下面积）。 ML NIDS设备的性能高度依赖于分配给每个实例的流量中不同网络流的相似性。 因此，优化此类NIDS设备的安全性能需要采用新方法对其实例之间的网络流量进行负载平衡。 我们的工作提出了一种新方法，用于在设备的实例（具有相同的功能）上分发类似网络流的组，以实现更好的安全性能，同时在其实例之间保持相对平衡的负载。

找到并概括这种负载平衡方法并非易事。 首先，研究领域相对广泛。 负载均衡器可以使用许多聚类算法，并且NIDS可以使用许多误用检测算法。 适当的评估需要很好地理解现有方法及其各自的领域（例如，统计，集中/分布，离线/在线，基于分类，基于异常，基于知识，基于软计算的方法）。 此外，许多提议的方法没有公开可用的实施用于评估目的。

其次，该领域的公共数据集的可用性和质量是有限的。 KDD CUP '99数据集[27]已经过时并且因为没有真实地代表攻击而受到严厉批评。 NSL-KDD数据集[28]对ML更具挑战性，但仍缺乏现代攻击。 因此，为了开发与数据集无关的负载平衡方法，应使用各种跟踪和数据源。

EVALUATION

在本节中，我们评估了我们的方法; 即，使用基于ML的聚类负载均衡器对误用ML的NIDS所获得的改进。 首先，我们描述我们的评估设置（即跟踪，模型，工具和方法）。 然后，我们通过基于循环的LB向集中式基线模型和分布式模型呈现ML性能比较。 最后，我们展示了我们方法的可扩展性。

- Traces：为了评估，使用众所周知的NSL-KDD [28]数据集。 它是KDD CUP '99 [27]的简化版，更适合机器学习基准测试。 每条记录代表TCP会话中的一系列数据包。 我们使用原始的NSL-Train + / Test +（20%）套装进行培训/测试。 这些功能包括在过去2秒内与之前100个连接到同一目标的所有连接的统计信息，当前TCP会话的统计信息以及专业知识。 记录标记为正常或攻击类别之一：DOS（拒绝服务），PROBE（扫描），R2L（远程到本地，未经授权的远程访问）或U2R（用户到根，未经授权的本地） 访问）。

- 功能预处理：由于评估的误用检测算法需要数字特征，因此可以使用单热编码删除或编码分类特征。 数字特征要么按原样保留，要么根据最小 - 最大值进行缩放。

我们评估两种不同的模型：

- 基线模型Baseline Model（集中式）：所有流量都由单个NIDS实例处理。 ML性能作为与扩展模型进行比较的基线。 在准确度，精确度，召回率，F分数和曲线下面积（AUC）方面的性能指标

是使用对测试数据的5倍交叉验证产生的。

- 扩展模型Scaled-out Model：通过不同的负载平衡方法，流量在同一NIDS设备的多个实例之间进行负载平衡。 首先，我们评估循环和统一随机方法，在所有实例中实现平衡负载，忽略ML实例的功能 - 传统LB.
- 其次，我们将传统的LB方法与基于群集的LB方法进行比较，后者对传入流量进行聚类，并将每个群集分配给NIDS实例。 为此，我们定义了一个聚类模型，它包括聚类算法（表1），聚类特征集（表2）和特定特征预处理设置 - 即聚类模型=聚类算法+特征集+特征 - 预处理。 扩展模型将针对负载平衡，ML性能指标和可伸缩性进行评估，如续集中所述。
- 工具和方法：评估框架基于scikit包1，运行在Python 2.7，Amazon Linux 64位，Intel Xeon E5-2666 v3 @ 2.9GHz，7.5GB RAM上。

Approach	Algorithms
Conventional Load Balancer	Round-Robin, Unified-Random
Cluster-Based Load Balancer	K-means
Scaled-out Misuse Detection NIDS Algorithms	1-NN, Decision-Tree, Random-Forest, Multi-Layer Perceptron, SVM (only for min-max scaled features).

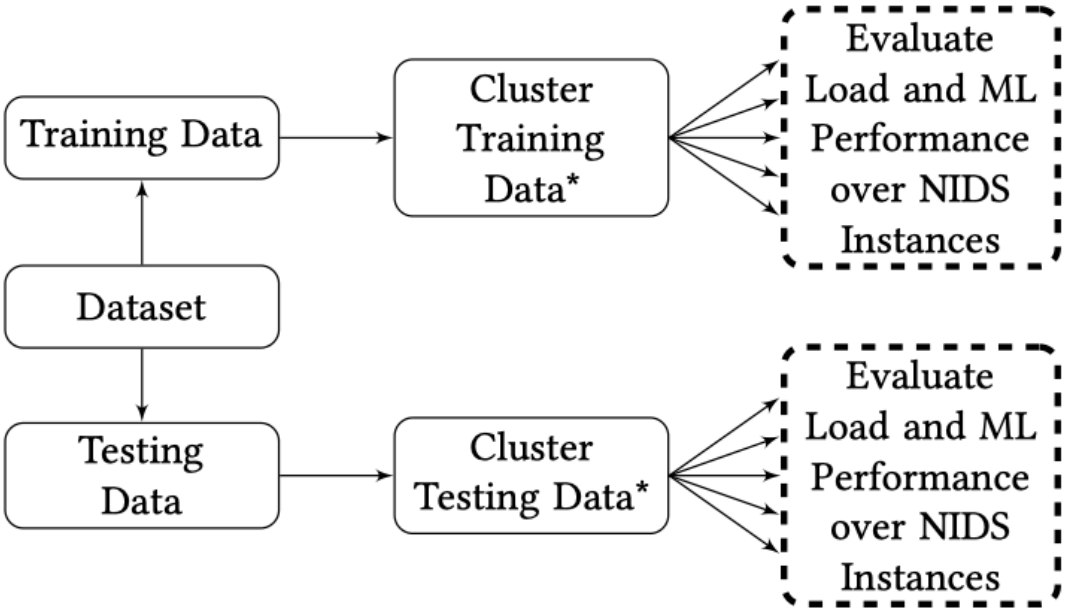
Table 1: The evaluated LB, clustering and misuse detection algorithms for the baseline and scaled-out models.

Clustering Feature Sets	# of Features	Description
1. Same destination	5	Statistics on connections to same destination in last 2 seconds
2. Same service	4	Statistics on connections to same service in last 2 seconds
3. 100 connections same destination	10	Statistics on 100 last connection to same destination
4. Domain expert	13	Features within a connection suggested by domain knowledge
5. TCP features	9	Statistics on individual TCP connection
6. All 2 secs features	9	Union of 2 secs same destination + service
7. All history features	19	Union of 100 connections + all 2 secs features

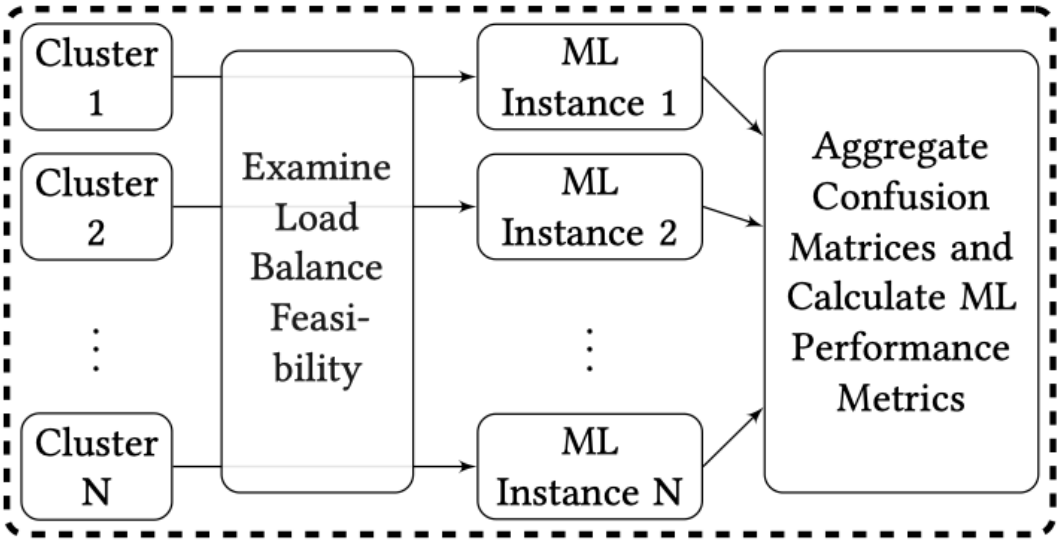
Table 2: Features options for clustering. Each row describes a set of features for traffic clustering.

评估过程如图1（a）所示。 首先，将数据集拆分为训练集和测试集。 然后，针对每个聚类模型训练模型。 针对不同数量的簇k = {3..9}评估负载平衡器聚类模型。

特定k的聚类模型的评估过程如图1（b）所示。 每个集群都分配给NIDS实例，并且所有实例与相同的误用检测算法并行运行。 使用5倍交叉验证对每个实例进行训练和测试。 将所有实例的混淆矩阵和预测概率概括为单个矩阵，其用于ML性能测量的计算：准确度，精度，召回率，F分数和曲线下面积（AUC）。



(a) Workflow of the scaled-out model evaluation (* - using the *same* clustering model)



(b) Workflow of the load and ML performance evaluation over the NIDS instances

Figure 1: The evaluation workflow

- 聚类模型可行性Clustering Models Feasibility：为了定义聚类模型是否可行，我们首先通过聚类大小的最大标准偏差（σ= 6000）和最大和最小聚类之间的最大比率25来定义负载平衡聚类模型。所有簇数（即，超过k = {3..9}）。仅评估负载平衡的聚类模型，并省略不平衡的聚类模型。其次，我们检查NIDS实例使用的ML算法对每个聚类模型的可行性。例如，SVM要求每个集群必须包含多个类的样本。因此，不评估不满足该要求的任何聚类模型（例如，用于特征选项1,2的K均值聚类模型）。 在此标准下，可行的聚类模型是具有所有预处理和编码选项的特征集3和7（表2），以及具有最小 - 最大缩放和单热编码的特征集5。图2显示了每个可行聚类模型的最大和最小聚类之间的最大比率。基于群集的LB的进一步负载平衡改进留待将来工作（有关更多详细信息，请参见§5）。

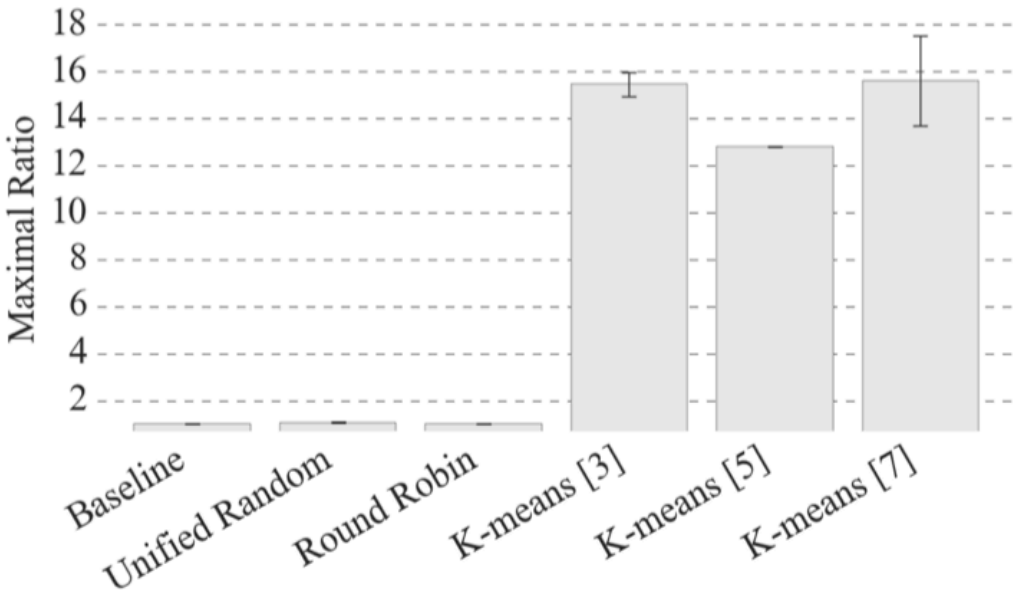


Figure 2: Maximal ratio between the biggest and the smallest cluster sizes, over all cluster sizes (k). Comparison between baseline, conventional LB, and clustering LB (our approach).

- ML性能比较ML Performance Comparison:：我们基于群集的LB显着改善了一些基于ML的误用检测算法（表1），而其他算法的改进则不那么重要。我们主要使用F-score指标，因为它考虑了精确度和召回得分，它与我们评估中的准确度得分几乎相同。 图3显示了决策树算法的平均F分数。可以看出，基线和常规LB方法达到平均F分数约为。 0.8虽然我们的方法取得了显着的改进：所有可行的聚类选项都可以使F得分高于0.9。随机森林算法的ML性能改进非常相似。 另一方面，图4显示了SVM ML算法的平均F分数的比较。可以看出，与基线和传统LB方法相比，SVM几乎没有改善（同样也适用于1-NN算法）。此外，多层感知器性能增益位于中间，传统LB的平均F分数约为0.87，基线为0.91，基于簇的LB为0.92-0.96。

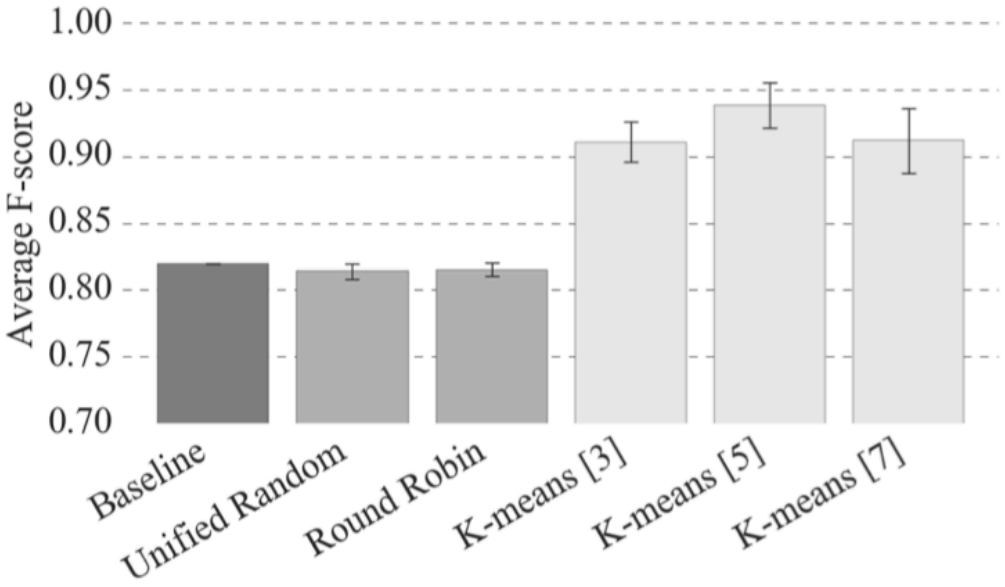


Figure 3: Decision tree average F-score. Comparison between baseline, conventional LB, and clustering LB – our approach.

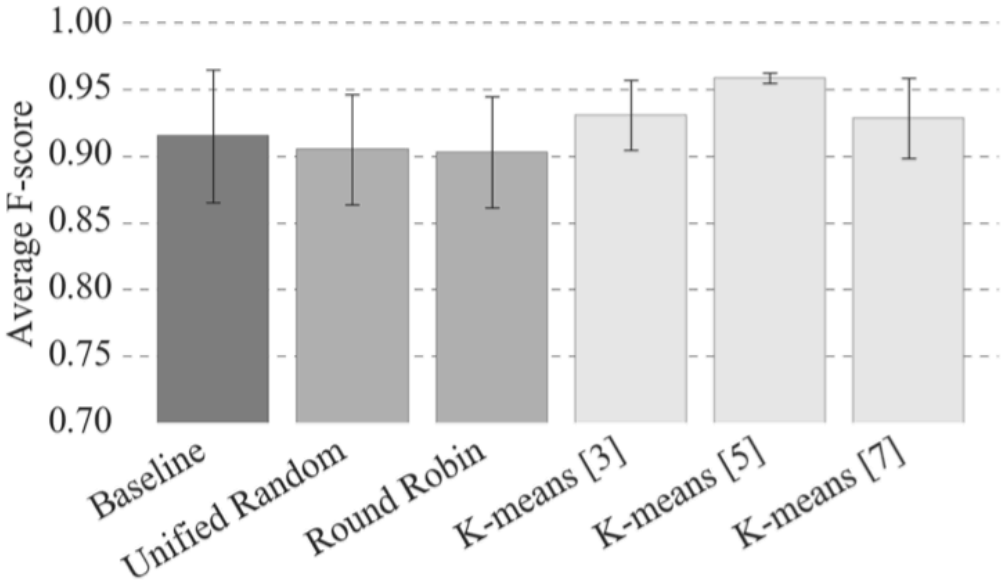


Figure 4: SVM average F-score comparison (same results obtained for 1-NN).

之前的数字显示了平均结果。因此，在下文中，我们放大并呈现我们针对特定设置和指标的一些评估测量，以进一步展示和强调我们的方法改进。图5显示了决策树ML算法的曲线下平均面积，对于U2R类，最罕见的攻击类，因此是最具挑战性的学习。基于簇的LB将AUC提高了大约。与基线和常规LB方法相比，为10%。图6显示了k = 9的决策树ML算法的ROC曲线和AUC。可以清楚地看到，通过基于簇的LB改善了所有ROC曲线，如分别通过图3和5中的F分数和AUC测量所观察到的。讨论：有人可能认为，由于所有基线模型的流量都是集中处理的，因此它的性能应该是最高的（无论是F分数还是AUC）。但是，我们的评估表明了其他方面，尤其是基于决策树的

NIDS（见图3和图5）。决策树算法配置有三个级别。我们假设基于簇的LB实际上采用了较高的决策级别。因此，在已经处理了第一个决策级别之后，每个NIDS实例都采用决策树算法。因此，有效地，基于群集的LB实现了具有更高级别的决策树的性能，即使每个实例配置有三个级别。图7展示了我们假设基于簇的LB有效改进决策树的级别数。它显示了决策树的基线集中式方法的平均F分数，具有三到五个级别（DT3，DT4和DT5，分别为）。当使用基于群集的LB时，获得的具有三个级别（DT3）的决策树的平均F分数等于根据基线模型在集中处理时具有五个级别（DT5）的决策树的实现的F分数。因此，采用基于簇的LB有效地实现了具有五个级别的集中式决策树的F-得分结果，而实际上使用仅具有三个级别的决策树。

图8显示了循环LB和基于簇的LB的流标签分布，其中k = 9。 可以看出，循环上的循环流标签分布非常均匀。 然而，基于K均值的LB导致聚类，其中每个实例包含更高百分比的特定流标签; 因此，允许对此特定流程标签进行更好的培训。 因此，总体上实现了更好的ML性能（例如，如图6中所示）。

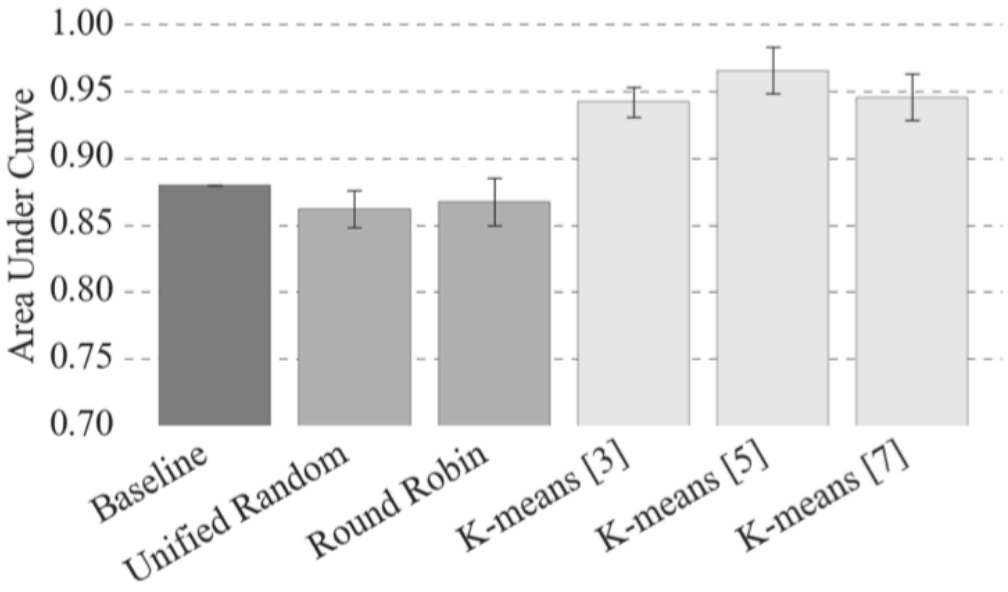


Figure 5: Area under curve (AUC) of the decision-tree ML algorithm for U2R class.

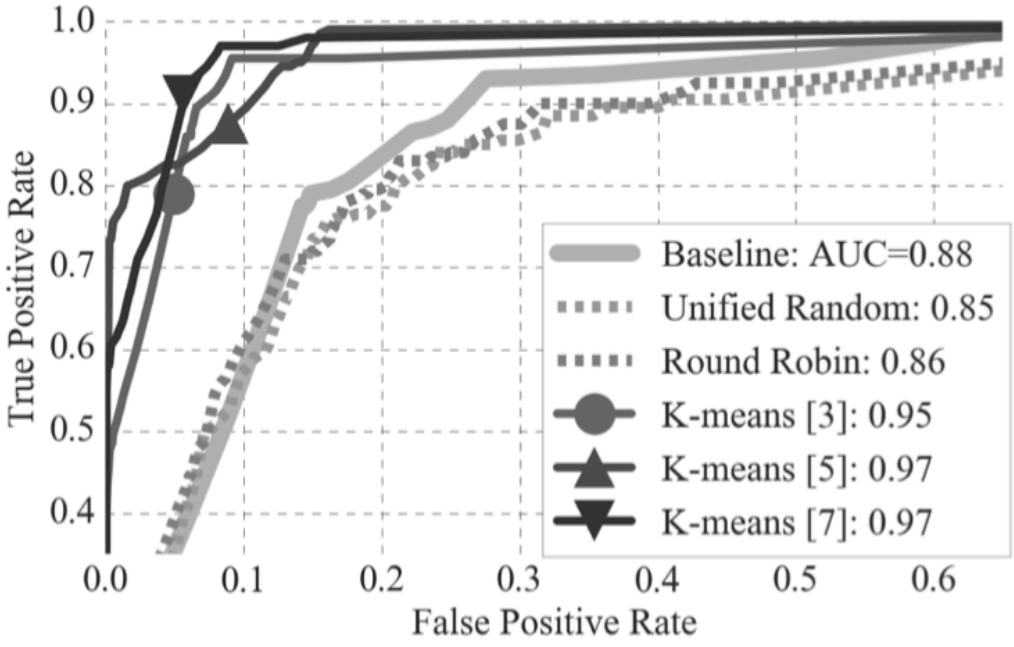
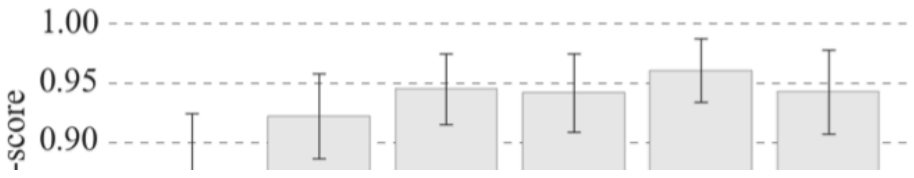


Figure 6: ROC curves and areas of Decision-Tree ML algorithm for U2R (user-to-root) class with 9 instances (k=9). Numbers in the legend indicates the AUC of each clustering model.



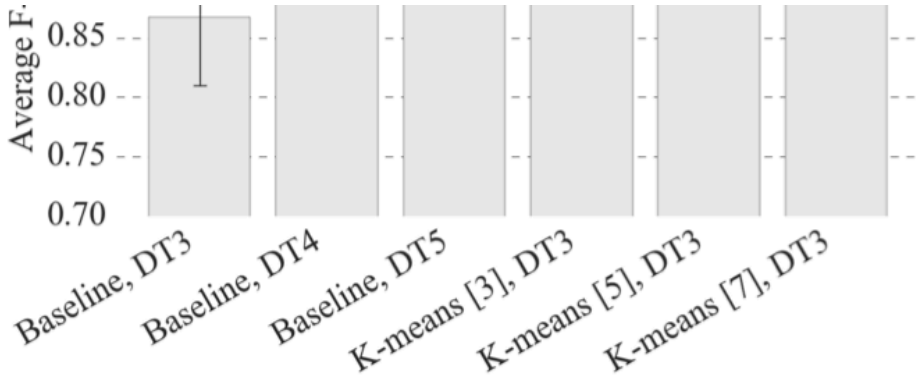
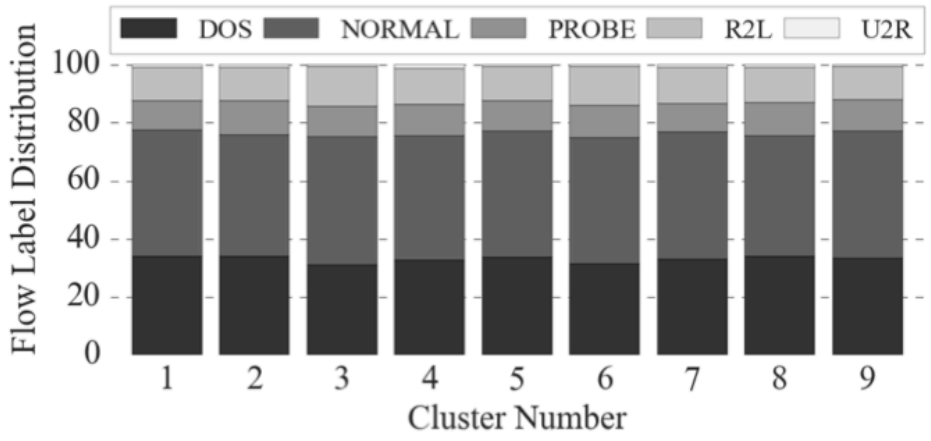
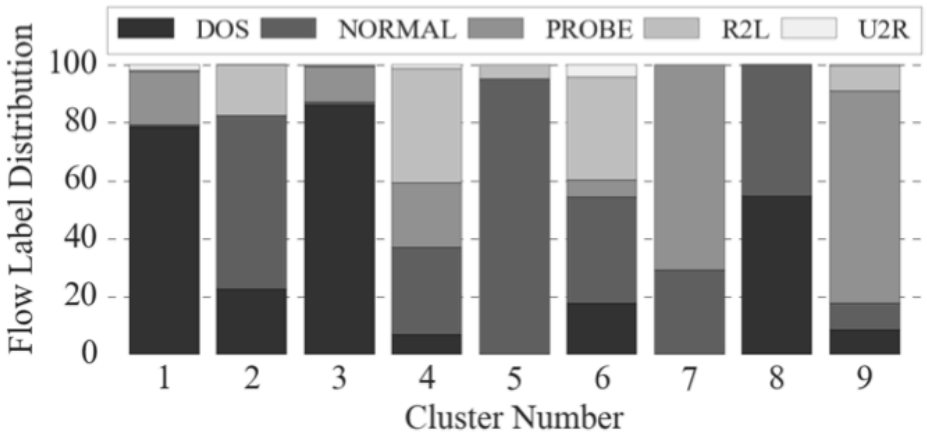


Figure 7: Comparison of the achieved average F-score of decision-tree between between the baseline model – centralized processing with three to five levels (DT3, DT4, DT5), and cluster-based LB with only three levels.



(a) Flow label distribution for round-robin.



(b) Flow label distribution for K-means [3].

Figure 8: Comparison of the flow label distribution over the clusters between conventional round-robin LB and cluster-based LB, for k=9.

可伸缩性Scalability：我们的评估表明，所有可行的聚类选项都可以很好地扩展k（聚类数）。 表3展示了K-means的可扩展性，其中数值特征根据最小 - 最大值缩放。 对于每个k，F得分在所有可行的聚类模型和误用检测算法上取平均值。 可以看出，随着聚类数量的增加，平均F分数略有提高。

K	3	4	5	6	7	8	9
F-score	0.922	0.929	0.932	0.934	0.936	0.938	0.940
Std. Dev	0.044	0.037	0.034	0.032	0.031	0.029	0.029

Table 3: K-means scalability

RELATED WORK

以前的工作介绍了NIDS的聚类和流量相关方法。 但是，如本文所述，没有一种用于LB的横向扩展。 林等人。 在[19]目前的聚类方法中，用于改进监督分类性能和运行时间。 但是，此方法不适用于负载平衡问题，因为实例数是从攻击类的数量导出的，而不是来自流量容量需求。

一些工作[12,17,24,30,31]提出了不同的负载平衡方法和NIDS的改进，它们不是基于ML分类器。 例如，Le等人。 [17]旨在最大化同一节点的流之间的相关性，同时保持跨节点的相等负载。 然而，他们的NIDS模型基于用于检测DDoS攻击的统计算法（CUSUM）。

SUMMARY AND FUTURE WORK

在本文中，我们介绍了为NIDS集成基于ML的负载均衡器的第一步。 我们提出的LB方法使用聚类方法来提高NIDS设备的机器学习性能。 我们证明了与传统的负载平衡器相比，我们的方法具有可扩展性并实现了更好的ML性能。

这项工作可以扩展到几个方向，包括：

- 负载平衡群集改进。 我们应该通过获得更好的负载平衡集群来改进基于集群的LB方法，这可以通过以下几种方式实现：a）优化技术：在本文中，手动选择表2中的集群特征集。 然而，例如可以通过优化方法实现的任何其他特征组合可以进一步改善平衡负载与安全实例的ML性能之间的误用检测权衡。 b）可以在线使用NIDS实例到LB关于负载和流量相似性的反馈，以便在负载平衡和集群精度方面改进集群。 c）其他聚类算法：例如模糊C均值2可用于改善基于ML的NIDS性能。
- 在线评估。 在这项工作中，我们提出了基于ML群集的NIDS负载均衡器的离线评估。 但是，为了处理实际系统中的入口流量，需要在线群集和误用检测算法。 为此，可以使用诸如在线K均值之类的算法。
- 异常检测NIDS。 误用检测算法需要对标记流量进行训练，这可能很难获得。 纯异常检测算法，如一类SVM，iForest和EXPoSE [25]只需要良性流量，可能更适合实际应用。
- 额外的痕迹。 在本文中，我们对NSL-KDD数据集进行了评估。 在我们未来的工作中，我们将使用其他跟踪，例如ISCX 2012数据集[26]，它更现代并包含实际的流量跟踪。