论文阅读分析报告

1.论文信息

Latah, M., Toker, L. An efficient flow-based multi-level hybrid intrusion detection system for software-defined networks. *CCF Trans. Netw.* **3**, 261–271 (2020).

- Received28 July 2019
- Accepted04 September 2020
- Published20 October 2020
- Issue DateDecember 2020

2.论文综述

这篇论文讨论了如何利用软件定义网络（SDN）范式来解决传统网络中的安全挑战，并提出了一种基于流量统计的5级混合分类系统，用于提高入侵检测系统的准确率。SDN作为一种新颖的网络范式，提供了增强的编程能力，可以更有效地解决传统网络中的安全挑战。在该范式中，最关键的元素是控制器，用于管理通信转发元素（如交换机或路由器）的流量。

这个分类系统包括不同级别的算法：k近邻方法（kNN）、极限学习机（ELM）以及分层极限学习机（HELM）。这种多级分类方法可以处理不同类型的数据，从而提高入侵检测的准确率。实验研究表明，该系统实现了良好的准确率（84.29%），并且能够检测新的攻击达到77.18%。

学术界和业界应该继续研究和开发SDN在网络安全领域的应用。此外，了解如何有效地利用流量统计信息对于入侵检测系统的性能提升具有重要意义。在研究入侵检测系统时，选择适当的性能指标并进行全面的评估也是至关重要的。这篇论文提供了一个有前途的方法，通过SDN和机器学习技术来提高网络入侵检测系统的性能，为未来网络安全提供更多的解决方案。

3.文章架构

这段文本介绍了关于软件定义网络（SDN）和入侵检测系统（IDS）的背景信息以及论文的主要贡献和结构。SDN是通过分离控制数据平面和可编程网络的提案发展而来的网络范式，其中控制器是关键的组成部分，它通过OpenFlow协议等南向接口管理和收集基于流量的统计信息。

论文的目标是设计一种高效的SDN中入侵检测系统，能够正确分类已知攻击并检测新攻击，主要基于流量统计数据提供的信息。基于流量的入侵检测方法强调了其计算效率，因为它们仅依赖于数据包头部的检查，而不需要对数据包有效载荷进行分析。与此相反，基于数据包的系统在网络流量加密时无法使用，因此需要将这两种方法结合使用以提供更高级别的保护。

为了提高入侵检测系统的准确性，研究采用了多层次的方法，其中包括K最近邻（kNN）、极限学习机（ELM）和分层极限学习机（H-ELM）分类方法。研究的贡献包括：

1. 设计了一个多层次混合方法，仅基于6个流特征进行入侵检测，这些特征可以轻松地由典型的SDN控制器获得。
2. 使用机器学习算法，如ELM和H-ELM，减少了测试阶段所需的时间。
3. 通过使用标准数据集（NSL-KDD）测试系统，包括一组新攻击，以验证提出系统的有效性。
4. 与使用相同流特征和数据集的先进的有监督机器学习方法进行比较时，准确性从75.75%提高到84.29%，并且能够检测新攻击，达到77.18%。

文章的结构包括相关工作、kNN、ELM和H-ELM的背景介绍、提出的多层次混合分类系统的详细解释、用于训练和测试的数据集的回顾、实验结果的讨论、评估指标的介绍以及最后的总结。

4.相关工作-Related work

文章介绍了多个关于入侵检测系统的研究，每个研究采用不同的方法和技术，以提高对网络入侵的检测和分类能力。以下是每个研究的总结和学习观点：

1. Xing et al. (2004) 的研究采用了一个3级分类模型，使用C4.5算法进行入侵分类。尽管在已知攻击的准确性方面表现出色，但对于未知攻击的检测率较低。这个研究的局限性在于无法有效地检测未知攻击。
2. MLIDS (Al-Nashif et al. 2008) 采用了三个粒度级别的流量分析，以实现高检测率和几乎零的误报率。然而，由于它使用了滑动时间窗口和内容分析，可能会对网络产生负担。
3. Abuadlla et al. (2014) 提出了一个两级入侵检测系统，分别使用两个独立的神经网络进行检测和分类。它在已知攻击和未知攻击的检测方面表现出色，但在第一级中，它需要检查每个数据包的头部，这可能会对网络产生额外负担。
4. Hussain et al. (2016) 提出了一个两级入侵检测系统，使用支持向量机和人工神经网络进行异常检测和滥用检测。它在已知攻击的检测方面表现出色，具有较低的误报率。
5. Amoli and Hämäläinen (2013) 提出了一个实时多级入侵检测系统，用于提高对未知攻击的检测率。它使用多个时间窗口和子空间聚类来检测异常流量。
6. Aziz et al. (2013) 提出了一个三级混合智能方法，使用主成分分析、遗传算法和不同分类器来提高检测准确性。不同分类器在不同类型的攻击检测方面表现出色。
7. Cordella and Sansone (2007) 提出了一个串行多级入侵检测系统，使用学习向量量化分类器和可靠性概念来降低误报率。
8. Gogoi et al. (2013) 提出了一个三级入侵检测系统，采用监督、非监督和基于异常值的技术，表现出高准确性和低误报率。
9. Reddy et al. (2006) 采用基于规则和K均值聚类的两级系统，表现出色。
10. 还有其他研究如Lee et al. (2008)、Rajeswari and Kannan (2008)、Araki et al. (2014) 和Casas et al. (2011) 使用了不同的方法来提高入侵检测系统的性能。

总的来说，这些研究表明，入侵检测系统可以采用多种不同的方法和技术，具体选择取决于应用场景和性能需求。重要的是权衡性能和对网络的负担。某些方法在已知攻击的检测方面表现出色，而另一些则在未知攻击方面表现更好。这些研究为构建用于SDN环境的入侵检测系统提供了有益的经验教训，可以根据实际需求进行选择和定制。这些研究展示了不同方法和技术在入侵检测中的应用，以及它们的性能和局限性。深度学习方法在处理大规模数据时表现出色，但可能需要更多的计算资源。多层级分类模型能够在不同级别上提高准确性，但可能需要更多的特征工程。聚类技术适用于检测异常流量，但需要选择合适的特征和算法。

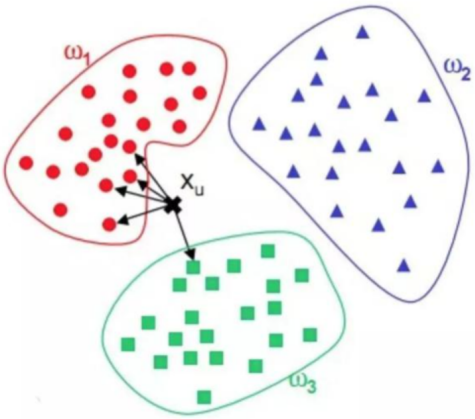
从这些研究中，我们可以学到如何在不同情况下选择合适的方法来构建入侵检测系统。在SDN环境中，流量统计数据可以方便地获取，因此流量特征成为一个重要的数据源。此外，深度学习和多层级分类模型可能是有效的方法来处理这些数据。然而，具体的选择取决于具体的应用场景和性能需求。

5.理论背景-Theoretical background

5.1 KNN邻近算法 K-nearest neighbor algorithm (kNN)

$$d_{(X,Y)} = \sum_{i=0}^{n} \left(X_i - Y_i\right)^2$$

KNN有着非常明显的优点和缺点：优点：精度高、对异常值不敏感、无数据输入假定；缺点：计算复杂度高、空间复杂度高。以下是我对KNN算法的解析：



算法解析：对于一个需要预测的输入向量x，我们只需要在训练数据集中寻找k个与向量x最近的向量的集合，然后把x的类别预测为这k个样本中类别数最多的那一类。如图所示，ω1、ω2、ω3分别代表训练集中的三个类别。其中，与xu最相近的5个点（k=5）如图中箭头所指，很明显与其最相近的5个点中最多的类别为ω1，因此，KNN算法将xu的类别预测为ω1。

基于上述思想给出如下所示的KNN算法：

输入：训练数据集

$$\mathrm{T} = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$

其中：

$$x_i \in X \subseteq R^n$$

为n维的实例特征向量。

$$y_i \in Y = \{c_1, c_2, \dots, c_K\}$$

为实例的类别，其中，i=1,2,...,N，预测实例x。

输出：预测实例x所属类别y。

算法执行步骤：

1. 根据给定的距离量度方法（一般情况下使用欧氏距离）在训练集T中找出与x最相近的k个样本点，并将这k个样本点所表示的集合记为N_k(x)；
2. 根据如下所示的多数投票的原则确定实例x所属类别y：

$$\mathrm{y} = \operatorname{argmax} \sum_{x_i \in N_{k(x)}} I(y_i, c_j), \quad i = 1,2, \dots, N \ ; \ j = 1,2, \dots, K$$

上式中I为指示函数：

$$\mathrm{I}(\mathrm{x}, \mathrm{y}) = \begin{cases} 1, if\ x = y \\ 0, if\ x \neq y \end{cases}$$

通过上述KNN算法原理的讲解，我们发现要使KNN算法能够运行必须首先确定两个因素：（1）算法超参数k；（2）模型向量空间的距离量度。

K值的确定：

- K值的选择在KNN算法中非常重要，影响着模型的性能。

- 如果选择较小的k值，算法对噪声敏感，近邻点的影响较大，可能导致过拟合。
- 如果选择较大的k值，模型相对鲁棒，不容易受噪声影响，但可能导致欠拟合，因为较远的点也会对预测产生影响。
- 通常采用交叉验证的方式选择k值，选取在一定范围内准确率最高的k值作为最终参数。

距离度量方法：

- 距离度量方法用于衡量两个样本点之间的相似程度，距离越短表示相似程度越高。
- 常用的距离度量方法包括闵可夫斯基距离、欧氏距离、曼哈顿距离、切比雪夫距离等。
- 选择距离度量方法要根据具体问题和数据特性来决定，通常欧氏距离是一个常用的默认选择。

综而言之，KNN算法是一种强大的分类方法，但在应用时需要谨慎选择合适的k值和距离度量方法，以确保获得最佳性能。通过交叉验证等方法，可以确定这些参数的最佳取值。

KNN算法的核心：KDTree

KNN分类算法的思想非常简单，就是k个最近邻多数投票的思想，关键就是在给定的距离量度下，如果快速找到预测实例的最近的k个邻居？

初学者一般采用直接暴力寻找的方法，因为k值一般不会取得特别大。确实，特征空间维度不高且训练样本容量小时，暴力寻找方法是可行的，但是当特征空间维度特别高或者样本容量较大时，计算过程就会非常耗时，这种方法就不可行了。

因此，为了快速查找到k个近邻，我们可以考虑使用特殊的数据结构存储训练数据，用来减少搜索次数。其中，KDTree就是最著名的一种。

KDTree的构造

我们使用递归方法来构造KDTree：（1）构造根节点，使根节点对应于k维空间中包含的所有点的超矩形区域；（2）不断地对k维空间进行切分，生成子节点。

构造根节点

首先，在包含所有节点的超矩形区域内选择一个坐标轴和在此坐标轴上的一个切分点，确定一个垂直于该坐标轴的超平面，这个超平面将当前区域划分为两个子区域（即二叉树的左右两个子节点）。

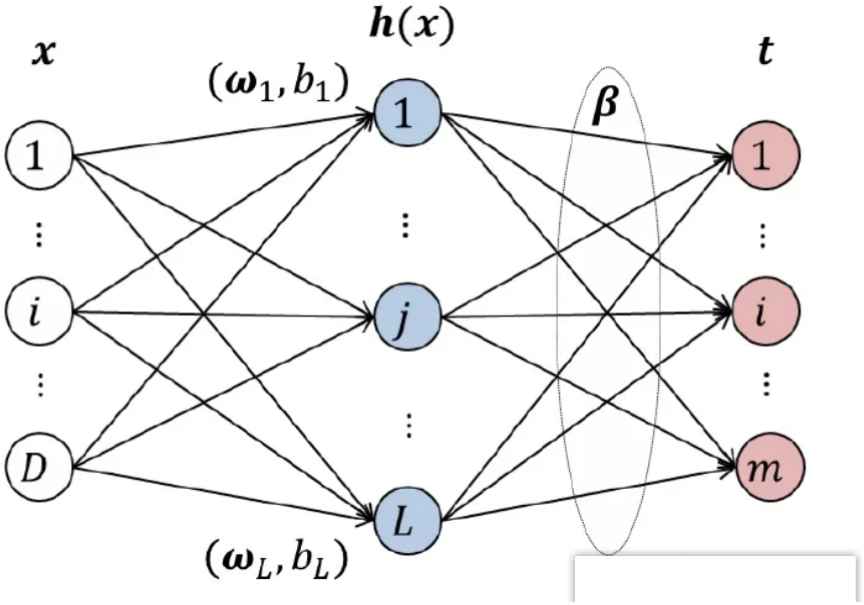
递归构造子节点

通过递归方法对两个子区域进行相同的划分，知道子区域内没有实例时终止（此时只有子节点）。

通常我们循环地选择坐标轴对空间进行划分，当选定一个维度坐标时，我们选择所有训练实例在该坐标轴上的中位数作为切分点。此时，我们构造的KDTree是平衡二叉树，但是平衡二叉树在搜索近邻时并不一定是最高效的。

5.2 极限学习机 Extreme learning machine (ELM)

极限学习机（*ELM*）模型的网络结构与单隐层前馈神经网络（*SLFN*）一样，只不过在训练阶段不再是传统的神经网络中屡试不爽的基于梯度的算法（后向传播），而采用随机的输入层权值和偏差，对于输出层权重则通过广义逆矩阵理论计算得到。所有网络节点上的权值和偏差得到后极限学习机（*ELM*）的训练就完成了，这时测试数据过来时利用刚刚求得的输出层权重便可计算出网络输出完成对数据的预测。



1. 单隐藏层神经网络计算过程如下：
    - 输入值乘以权重值
    - 加上偏置值
    - 进行激活函数计算
    - 对每一层重复步骤1~3
    - 计算输出值
    - 误差反向传播
    - 重复步骤1~6

而 ELM 则对其进行了如下改进：去除步骤4；用一次矩阵逆运算替代步骤6；去除步骤7。

2. 算法学习过程

具体地，ELM 计算过程如下：

$$f_L(x) = \sum_{i=1}^{L} \beta_i g_i(x) = \sum_{i=1}^{L} \beta_i g(w_i * x_j + b_i), j = 1, ..., N$$

式中：L 是隐藏单元的数量，N 是训练样本的数量，beta 是第 i 个隐藏层和输出之间的权重向量；w 是输入和输出之间的权重向量；g 是激活函数；b 是偏置向量；x 是输入向量。

极限学习机的计算过程与标准反向传播神经网络十分类似，但是隐藏层与输出之间的权重矩阵是伪逆矩阵。
将上式可以简写为：

$$T = H\beta$$

$$H = \begin{bmatrix} g(w_1 * x_1 + b_1) & \dots & g(w_L * x_1 + b_L) \\ \vdots & \dots & \vdots \\ g(w_1 * x_N + b_1) & \dots & g(w_L * x_N + b_L) \end{bmatrix}_{N \times L}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m} \quad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m}$$

式中：m 是输出的数量；H 是隐藏层输出矩阵；T 是训练集目标矩阵；

本篇文章中还提到了一下定理：

> 对于矩阵 G 和 A，若 $AGA = A, GAG = G, (AG)^T = AG, (GA)^T = GA$，则 G 是 A 的广义逆矩阵。

了解了上述定理后，现在我们要做的是定义我们的代价函数。如果输入权重和隐层层偏差可以随机选择，那么SLFN是一个线性系统。

由于我们考虑的 ELM 是一个线性系统，那么可以设计优化函数：

$$||H\hat{\beta} - T|| = \min_{\beta} ||H\beta - T||$$

由于 H 是可逆的，所以计算如下：

$$\hat{\beta} = H^{\dagger}T$$

5.3 分层极限学习机 Hierarchical Extreme Learning Machine

分层极限学习机（H-ELM）是一种深度学习模型，旨在实现更好的泛化性能和更快的收敛速度，相对于传统的极限学习机（ELM）方法。H-ELM的训练分为两个阶段：无监督的层次特征表示和有监督的特征分类。

- 在第一阶段，输入实例被转化成ELM特征空间，以提取训练实例中的隐藏信息。然后，通过多层次的无监督学习阶段获取高级别的稀疏特征。每个隐藏层的输出通过激活函数和输出权重进行定义。在每一层的特征提取之后，当前隐藏层的参数保持固定。H-ELM使用随机映射的特征作为特征分类阶段的输入，并应用$\ell 1$惩罚来获得更稀疏和显著的信息。特征自编码器的输入权重通过搜索随机映射的特征空间的路径来获得。H-ELM通过在优化模型中添加稀疏约束来实现普遍逼近能力。

$$\boldsymbol{H_i} = g(\boldsymbol{H}_{i-1} \cdot \beta) \tag{7}$$

- 在监督训练阶段，H-ELM使用传统的ELM方法进行实现。

$$O_\beta = \underset{\beta}{\operatorname{argmin}} \left\{ \|\boldsymbol{H}\beta - X\|^2 + \|\beta\|_{\ell_1} \right\}$$

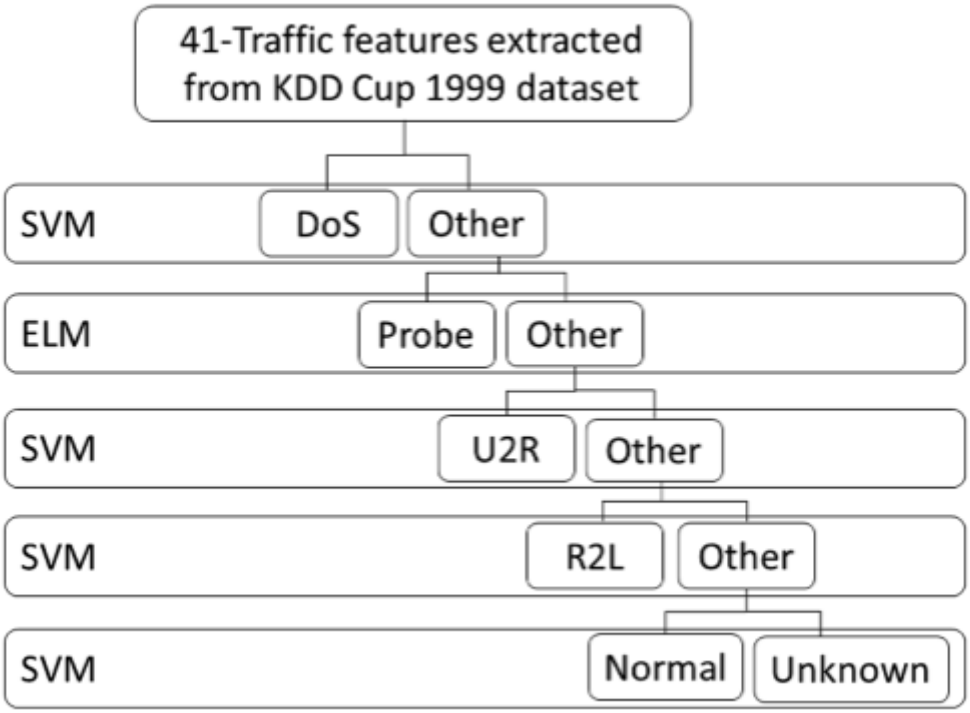
H-ELM是一个有趣的深度学习方法，结合了无监督和有监督学习的元素，以实现更好的泛化性能和更快的训练收敛。通过多层次的特征提取和稀疏化，它可以帮助系统更好地理解输入数据的结构和特征，从而提高分类性能。然而，该方法的效果可能会受到模型结构和超参数的选择的影响，因此需要仔细的调整和实验来确定最佳配置。总的来说，H-ELM代表了深度学习领域的一种有趣探索，可以在某些应用中提供有竞争力的性能。

6.基于流量统计的5级混合分类系统 （ Proposed multi-level hybrid IDS)

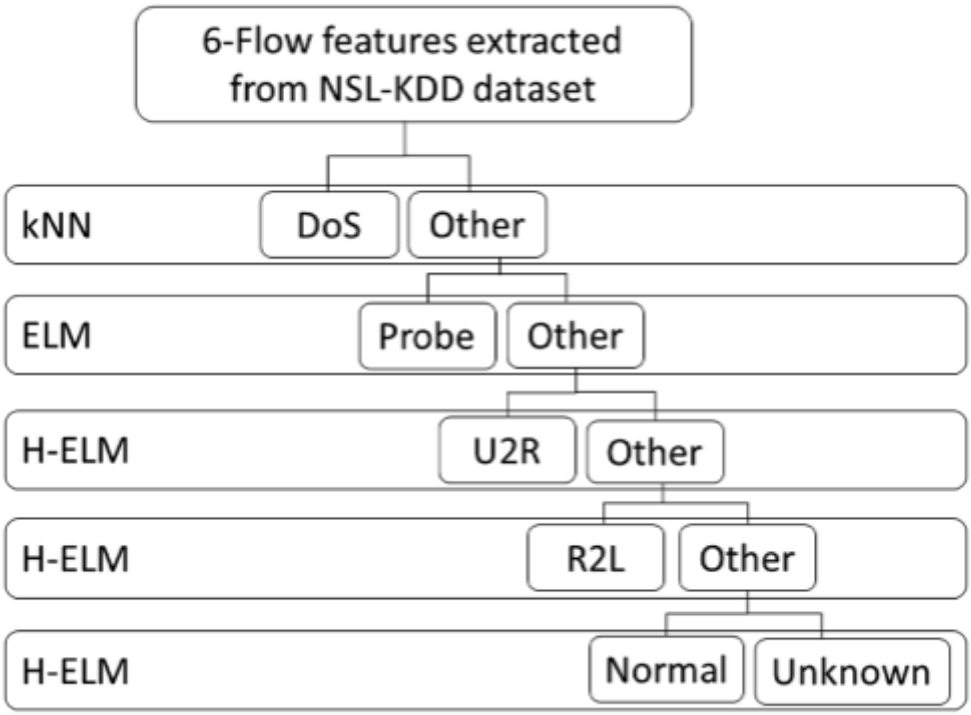
该系统使用流量统计将网络流量分类为不同的类别，并结合不同的分类算法来提高系统的准确性。

在这一部分，作者介绍了他们提出的系统，该系统受到了Al-Yassen等人（2017年）的研究启发，该研究基于SVM和ELM算法的组合构建了一个五级分类模型。作者的系统用于检测各种网络攻击类型，其中DoS（拒绝服务）和Probe攻击在其他攻击类型之前被检测出来，因为它们与正常流量相比具有较低的相似性。与此不同，U2R（用户到根）和R2L（远程到本地）攻击与正常流量模式相对相似，因此它们被视为潜在威胁。具体来说，该系统包括以下五个级别：

1. 第一级别：基于端口的分类，将流量分为TCP、UDP和其他类型。
2. 第二级别：基于流量的分类，将流量分为正常流量和异常流量。
3. 第三级别：基于协议的分类，将流量分为HTTP、FTP、SMTP等不同的协议类型。
4. 第四级别：基于应用程序的分类，将流量分为不同的应用程序类型，如Web浏览器、邮件客户端等。
5. 第五级别：基于行为的分类，将流量分为不同的行为类型，如扫描、攻击等。



与Al-Yassen等人（2017年）的模型类似，作者的模型采用每一层一个分类器的方式构建。第二层使用ELM，因为它在性能上优于其他方法，例如SVM和神经网络。在随后的层中，作者使用H-ELM代替SVM，因为H-ELM比SVM更快，且在泛化性能方面更好。作者的模型使用6个流特征，而不是Cheng等人（2012年）使用的41个流量特征，并且实验基于NSL-KDD数据集，这是KDD Cup 99的增强版本，具体框架图如下所示



7.数据集和选定的特征 （数据集和选定的特征）

在这项研究中，研究人员使用了NSL-KDD数据集，这是KDD Cup 99数据集的增强版本，以解决KDD Cup 99数据集中大量冗余记录的问题。NSL-KDD数据集包括表1中列出的各种特征，作者选择使用其中的一些特征，包括F1、F2、F5、F6、F23和F24，这些特征可以轻松地从SDN控制器中获取。

NSL-KDD数据集包含39种攻击，每种攻击被分类为以下四个类别之一。此外，一组这些攻击也被添加到测试集中。表2展示了NSL-KDD测试集中已知攻击和新攻击记录的分布情况。

- NSL-KDD数据集的特征列表

F. #	Feature name	F. #	Feature name	F. #	Feature name
F1	Duration	F15	Su attempted	F29	Same srv rate
F2	Protocol type	F16	Num root	F30	Diff srv rate
F3	Service	F17	Num file creations	F31	Srv diff host rate
F4	Flag	F18	Num shells	F32	Dst host count
F5	Source bytes	F19	Num access files	F33	Dst host srv count
F6	Destination bytes	F20	Num outbound cmds	F34	Dst host same srv rate
F7	Land	F21	Is host login	F35	Dst host diff srv rate
F8	Wrong fragment	F22	Is guest login	F36	Dst host same src port rate
F9	Urgent	F23	Count	F37	Dst host srv diff host rate
F10	Hot	F24	Srv count	F38	Dst host serror rate
F11	Number failed logins	F25	Serror rate	F39	Dst host srv serror rate
F12	Logged in	F26	Srv serror rate	F40	Dst host rerror rate
F13	Num compromised	F27	Rerror rate	F41	Dst host srv rerror rate
F14	Root shell	F28	Srv rerror rate	F42	Class label

- KDD-测试集中已知和新攻击的分布

	DoS	R2L	U2R	Probe
Known attacks	5741	2199	37	1106
	76.98%	79.85%	18.50%	45.68%
New attacks	1717	555	163	1315
	23.02%	20.15%	81.50%	54.32%

- 每层每个分类器使用的参数

Layer	Classifier	Parameter		
1	kNN	K=65		
2	ELM	N=400		
3	H-ELM	N1	N2	N3
		40	40	200
4	H-ELM	N1	N2	N3
		10	10	300
5	H-ELM	N1	N2	N3
		10	10	200

8.评估指标（Evaluation metrics)

在这项研究中，作者评估了他们的多层次基于流的系统的性能，主要关注准确性（Accuracy）和误报率（False Alarm Rate，FAR）。准确性是通过以下方式计算的：

- 真正例（True positive，TP）是正确分类的攻击实例的数量。
- 真负例（True negative，TN）是正确分类的正常流量实例的数量。
- 假正例（False positive，FP）是错误分类的正常流量实例的数量。
- 假负例（False negative，FN）是错误分类的攻击实例的数量。

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

误报率是通过以下方式计算的：

$$False\ Alarm\ Rate = \frac{FP}{TN + FP}$$

此外，作者还计算了精确度（Precision）、召回率（Recall）和F1分数（F-measure），它们分别通过以下方式计算：

$$Precision = \frac{TP}{TP + FP}$$

$$Recall\ (Detection\ Rate) = \frac{TP}{TP + FN}$$

$$F - measure = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)}$$

- 精确度（Precision）反映了IDS检测到的攻击中实际攻击的百分比。
- 召回率（Recall）表示检测到的攻击占NSL-KDD数据集中所有攻击的百分比。
- F1分数（F-measure）是一种更平衡的度量，综合考虑了精确度和召回率。

自己的看法:

作者使用了一系列常见的性能指标来评估他们的入侵检测系统的性能，这些指标包括准确性、误报率、精确度、召回率和F1分数。这些指标提供了对系统性能的多方面评估，可以帮助确定系统在检测攻击时的效果。然而，需要注意的是，这些指标之间存在权衡关系，改进一个指标可能会导致另一个指标的变化。因此，在使用这些指标时，需要根据具体的应用场景和需求进行权衡和选择，以确保获得最合适的性能评估。此外，数据集的质量和分布也可能对性能评估产生影响，因此需要谨慎处理和分析评估结果。

9.实验结果（Experimental results）

实验被分为两个部分：

1. 评估每个层次的性能。
2. 与其他有监督机器学习方法进行比较。

第一个实验主要评估了模型的性能，展示了每个层次的准确性和误报率，具体实验数据结果如下图所示

Table 4 Results for training and testing stages achieved at each layer of our proposed system

Layer #	Classifier	Detected attack	Training stage		Testing stage	
			Accuracy (%)	False alarm rate (%)	Accuracy (%)	False alarm rate (%)
1	kNN	DoS	97.34	1.38	91.23	3.39
2	ELM	Probe	97.12	0.12	92.61	1.45
3	H-ELM	U2R	99.96	0.0024	99	1.1
4	H-ELM	R2L	99.19	0.0240	86.97	0.94
5	H-ELM	Unknown attacks	90.76	7.28	80.39	5.61
All layers	kNN + ELM + H-ELM	All types	94.11	7.89	84.29	6.3

第二个实验中，作者将他们提出的系统与其他研究中的方法进行了比较，包括Tang等人（2016，2018）、Latah和Toker（2018b）、Dey等人（2018）、Wang等人（2018）以及其他传统的有监督学习方法。结果表明，与传统的有监督机器学习方法相比，作者的系统在准确性、召回率和F1分数方面取得了最高的值。在误报率方面，H-ELM方法实现了最高的精确度，并且具有最低的误报率。作者的系统在检测新攻击方面表现出色，检测率达到了77.18%。此外，作者的方法也优于Wang等人（2018）提出的半监督学习方法以及Dey等人（2018）提出的结合传统监督学习方法和特征选择方法的有监督学习方法，具体数据对比如下所示。

Method	Feature selection	Accuracy (%)	False alarm rate (%)	Precision (%)	Recall (%)	F1-score (%)
Naive Bayes	–	49.88	5.14	80.28	15.83	26.45
Neural Network	–	63.70	7.66	87.88	42.02	56.86
SVM	–	71.40	10.63	87.79	57.80	69.71
Decision Tree	–	74.43	6.43	92.50	59.95	72.75
ELM (N = 1500)	–	74.80	10.17	89.18	63.43	74.13
kNN	–	77.09	4.07	95.33	62.84	75.75
H-ELM (N1 = 30,N2 = 30,N3 = 300)	–	77.59	2.57	96.98	62.59	76.08
H-ELM (N1 = 10,N2 = 10,N3 = 200)	–	80.39	5.61	94.26	69.80	80.21
Simple Deep Neural Network (Tang et al. 2016)	–	75.75	3.21	92.50	59.95	74.13
Semi-supervised Approach (Wang et al. 2018)	–	77.26	N.A	N.A	N.A	N.A
Random Forest (Dey et al. 2018)	Information gain	81.95	N.A	N.A	N.A	N.A
Our approach	–	84.29	6.3	94.18	77.18	84.83
Decision Tree (Latah and Toker 2018b)	PCA	88.74	3.99	83.24	96.5	89.38
GRU-RNN (Tang et al. 2018)	–	89	N.A	89	89.5	89.2

作者还使用了Cbench工具来评估他们系统的吞吐量。为此，他们将系统实现为SDN控制平面中POX控制器的模块。这种方法被认为比将系统实现为控制器的应用更有效，因为他们的6个流特征可以轻松地从控制器中获取，并且增加流的数量会显著增加控制器和应用程序之间的交互。作者以10秒的时间间隔定期收集先前提到的特征。每个Open vSwich连接了具有不同MAC地址的1000个虚拟主机，向POX控制器发送了10,000个Packet-In消息。与基本的转发模块相比，作者的系统在性能方面表现出可接受的表现，这是因为他们仅使用了可以轻松从控制器获取的6个流特征，并且在第2到第5层使用减少测试阶段所需时间的机器学习算法，如ELM和H-ELM。

10.总结Conclusion

在本文中，作者提出了一种用于SDN的高效多层次混合入侵检测方法。该系统基于kNN、ELM和H-ELM方法的组合设计。通过在NSL-KDD数据集上进行的实验研究，显示出我们的方法在与传统的有监督学习方法相比显著提高了整体准确性。此外，该系统能够检测包括在测试集中的新攻击，检测率达到77.18%。未来的工作将集中在改进系统，以实现更低的误报率。