基于 DL 和 TSVM 的入侵检测方法研究

魏明军 彭 宁
(华北理工大学信息工程学院 河北 唐山 063210)

摘 要 为解决网络环境下大量高维数据给入侵检测造成的数据特征提取不当、检测速度慢、检测率低的问题,提出一种基于深度置信网络(Deep Belief Network,DBN)和孪生支持向量机(Twin Support Vector Machine,TSVM)的入侵检测模型(DBN-TSVM-5)。利用五层受限玻尔兹曼机的 DBN 对归一化后的标准数据集进行特征降维,以获得入侵检测数据的最优低维表示,构造多分类 TSVM-5 分类器,对降维后的数据进行识别。经过 KDDCUP99数据集的仿真实验,结果表明,该模型是一种有效的入侵检测模型。

关键词 深度学习 入侵检测 深度置信网络 孪生支持向量机

中图分类号 TP393 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2020.12.052

INTRUSION DETECTION METHOD BASED ON DL AND TSVM

Wei Mingjun Peng Ning
(*College of Information Engineering, North China University of Science and Technology, Tangshan 063210, Hebei, China*)

Abstract An intrusion detection model (DBN-TSVM-5) based on deep belief network (DBN) and twin support vector machine (TSVM) is proposed to solve the problems of improper data feature extraction, slow detection speed and low detection rate caused by a large amount of high-dimensional data in network environment. In this method, DBN of five-layer restricted boltzmann machine was used to reduce the characteristic dimension of the normalized standard data set to obtain the optimal low-dimensional representation of intrusion detection data. Multi-classification TSVM-5 classifier was constructed to identify the data after dimension reduction. The simulation results on KDDCUP99 data set show that the model is an effective intrusion detection model.

Keywords Deep learning Intrusion detection Deep belief network Twin support vector machine

0 引 言

互联网的发展使我们步入科技时代,人们的日常生活也因为网络的普及而简化,伴随而来的网络安全问题则愈演愈烈。比如,各大网络平台用户信息泄露、DoS 攻击、WannaCry 勒索病毒、被黑客窃取计算机全部内存内容的漏洞等。如何有效鉴别各种网络攻击行为是网络安全领域中迫切需要解决的问题。入侵检测是一种积极、主动的网络安全防御技术手段,其通过分析收集计算机关键点的信息,从中发现是否有威胁计算机安全的异常行为[1],若有则及时作出响应告知用户,从而进行紧急处理以保护本地计算机安全。因此入侵检测技术一直是网络安全研究领域中不可规避的重点课题。

研究学者在入侵检测系统(Intrusion Detection Systems,IDS)中尝试引用不同的方法,比如:基于免疫方法[2]、基于神经网络[3]、基于数据挖掘[4]、基于粒子群[5]、基于云计算[6]、支持向量机[7](Support Vector Machine,SVM)等方法,经实验证明这些方法运用到入侵检测中都显示出各自的检测优势。然而,现如今面临网络上数据的海量化、高维化的特点,以及网络攻击方式的多样化、复杂化的趋势,入侵检测技术需要寻求新的突破,在提取数据特征时,能够尽可能保留初始数据的本质特征,并且提高检测率。

深度学习(Deep Learning,DL)被广泛运用于各个研究领域是因为其具有独特的数据特征学习能力。结合计算机视觉可以感知路口车流量,从而动态地控制

收稿日期:2019-07-12。**魏明军**,教授,主研领域:计算机网络,信息安全技术。**彭宁**,硕士生。

十字路口红绿灯时间的长短;融合自然语言处理开发了百度智能机器人小度;在语音识别领域成功地被应用于同声传译技术。将深度学习应用到不同领域且皆取得了很好的成果。

综合 IDS 研究现状,本文将深度学习和 TSVM 相结合,提出一种 DBN-TSVM-5 入侵检测模型。深度学习具有很好的数据特征提取性能,用于数据降维;而 TSVM 比 SVM 具有更高的分类精度,且耗时仅占 SVM 的四分之一[8]。数据采用 KDDCUP99 数据集对 DBN-TSVM-5 模型进行测试与评估,结果表明该模型的性能比传统的入侵检测方法好。

1　深度置信网络

DBN 是 Hinton 等[9]于 2006 年提出的一种深度学习模型,其特征学习能力非常强大,通过逐层提取的方式,可将原始数据转换成为更高层和更抽象的形式。

DBN 由若干层受限玻尔兹曼机(Restricted Boltzmann Machine,RBM)和一层有监督的反向传播(Back-Propagation,BP)网络组成,如图 1 所示。

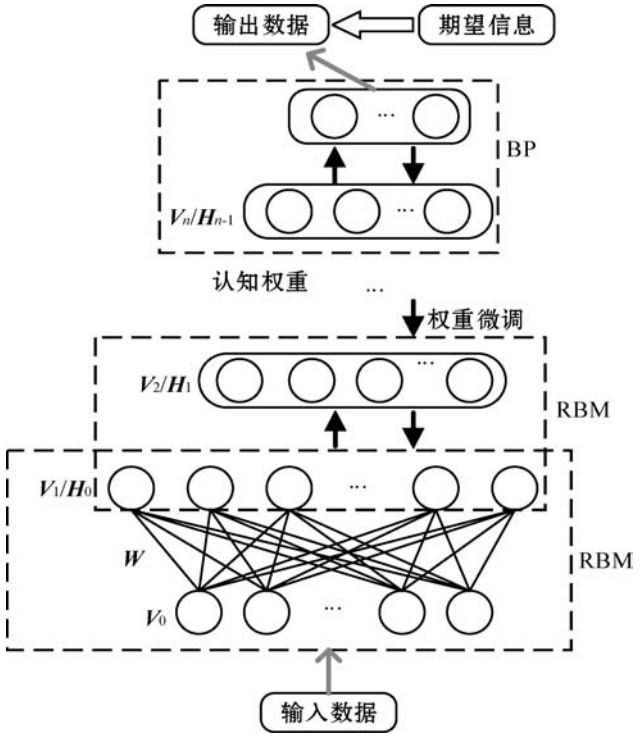

图 1　DBN 模型图

1.1　RBM 模型

1986 年,一个层内无连接、层间全连接的两层网络模型被提出,它就是 RBM,包含了一个可见层 V 和一个隐含层 H,每一层都是由神经元组成,所有神经元都有激活状态 1 和未激活状态 0 两种状态值。$\boldsymbol{W}$ 是两层之间的连接权重矩阵,$\boldsymbol{a}$ 是可见层偏差,$\boldsymbol{b}$ 是隐含层偏差。如图 2 所示。

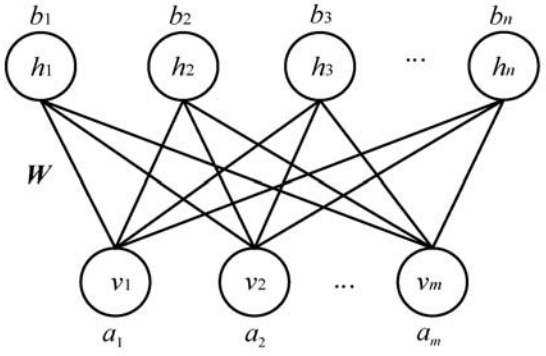

图 2　RBM 模型图

训练 RBM 模型使用的数据特征维数较高时,以 Gibbs 采样方法训练需要足够大的采样步数,可想而知用此方法训练 RBM 会花费相当多的时间,导致最终训练结果不理想。针对该问题,Hinton[10]提出对比散度算法(Contrastive Divergence,CD)。

1.2　CD 对比散度

不同于 Gibbs 采样,CD 算法在一开始将训练样本作为输入加载到可见层,经实验证明,仅需要使用 $k=1$ 步吉布斯次采样,就可以得到很好的近似值[11]。

在已知 $v^{(0)}$ 的情况下,出于 RBM 的连接特性,所有隐含层之间相互独立,利用式(1)计算隐含层第 j 个神经元的状态。

$$P(h_j = 1 \mid v) = \sigma\left(\sum_{i=1}^{m} w_{ij} v_i + b_j\right) \tag{1}$$

确定所有隐含层神经元 $h^{(0)}$ 的状态之后,由于可见层节点之间也是相互独立的,再根据隐含层神经元的状态,利用式(2)重构出可见层第 i 个神经元的状态,到得可见层重构 $v^{(k)}$。

$$P(v_i = 1 \mid h) = \sigma\left(\sum_{j=1}^{n} w_{ij} h_j + a_i\right) \tag{2}$$

算法 1　CD 算法

输入:训练样本 $X^{(S)}$,学习率 $alpha$,最大训练周期 k。

输出:链接权重矩阵 $\boldsymbol{W}$,可见层偏置向量 $\boldsymbol{a}$,隐藏层偏置向量 $\boldsymbol{b}$。

初始化:令可见层神经元的初始状态 $v^{(0)} = X^{(S)}$,$\boldsymbol{W}$、$\boldsymbol{a}$、$\boldsymbol{b}$ 取随机较小的数值。

for $t=0,1,\cdots,k$ do

for $j=1,2,\cdots,n$(对于所有隐含层神经元节点)

依据式(1)计算 $h_j^{(t)} \sim p(h_j|v^{(t)})$

采样:如果 $p(h_j|v^{(t)})>0.5$,则 $h_j^{(t)}=1$,否则 $h_j^{(t)}=0$

for $i=1,2,\cdots,m$(对于所有可见层神经元节点)

依据式(2)计算 $v_i^{(t+1)} \sim p(v_i|h^{(t)})$

采样:如果 $p(v_i|h^{(t)})>0.5$,则 $v_i^{(t+1)}=1$;否则 $v_i^{(t+1)}=0$

利用式(3)更新各个参数:

$$w_{ij}=w_{ij}+alpha(p(h_j|v^{(0)})\cdot v_i^{(0)}-p(h_j|v^{(k)})\cdot v_i^{(k)})$$

$$a_i=a_i+alpha(v_i^{(0)}-v_i^{(k)}) \tag{3}$$

$$b_j=b_j+alpha(p(h_j|v^{(0)})-p(h_j|v^{(k)}))$$

2 对支持向量机

Jayadeva 等[12]在 2007 年提出了 TSVM。TSVM 的基本思想是对正负两类样本点分别构造一个分类超平面。这样有两点好处:一方面使得每一个分类超平面与其中一类样本点尽可能近;另一方面远离另一类样本点。以二维平面内的样本数据为例,对线性 TSVM 分类思想描述如图 3 所示。

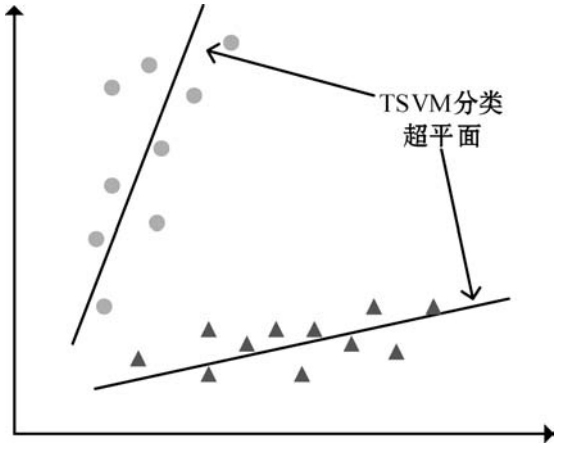

图 3 线性 TSVM 示意图

TSVM 实质上是将传统 SVM 中的一个二次规划问题(Quadratic Programming Problem,QPP)分成两个规模较小的 QPP[13],简化了计算复杂度,使得样本训练时间缩减为传统 SVM 的四分之一,而且还保持了较高的分类精度。

3 DBN-TSVM-5 模型

3.1 模型总体设计

基于 DBN-TSVM-5 的入侵检测方法框架结构图如图 4 所示。

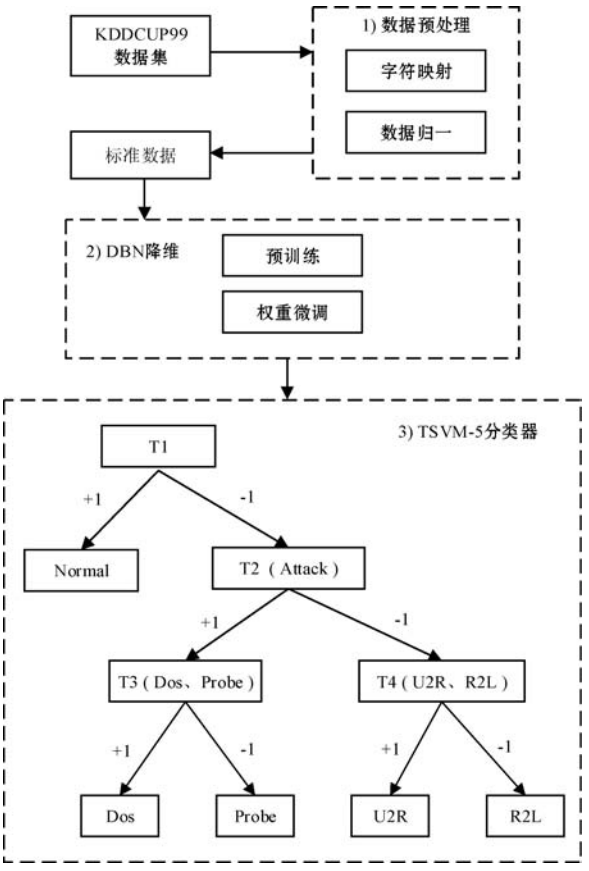

图 4 DBN-TSVM-5 框架结构图

主要有以下 3 个步骤:

1)数据预处理。将 KDDCUP99 数据集通过特征映射的方法,把字符型数据转换为数值;再对数值化后的数据全部进行数据归一化处理,将数值规范到 0 ~ 1 之间,使之成为标准数据集。

2)DBN 降维。经过预训练和权重微调两个过程后,得到 DBN 网络模型最优表达,能够反映标准数据特征的低维数据。

3)多分类 TSVM-5。构造多分类 TSVM-5 分类器,对五类数据进行识别。

3.2 DBN 降维过程

DBN 训练经过预训练和微调两个阶段后方可得到一个可以反映高维、非线性原始数据特征的低维数据,从而实现对标准数据集的最优提取。

1)预训练。将训练集中的数据特征和类标签分离,用无标签的训练集对每一层 RBM 进行自下而上、单独、无监督、基于 CD 算法的训练。输入 $\boldsymbol{V}_0$ 通过 $P(\boldsymbol{H}|\boldsymbol{V}_0)$ 计算出 $\boldsymbol{H}_0$,$\boldsymbol{H}_0$ 根据 $P(\boldsymbol{V}|\boldsymbol{H}_0)$ 计算重构出的 $\boldsymbol{V}_1$ 如果和 $\boldsymbol{V}_0$ 一样,那么隐藏层 $\boldsymbol{H}_0$ 就是 $\boldsymbol{V}_0$ 的另一种表示,如此 $\boldsymbol{H}_0$ 可作为下一层 RBM 的可见层 $\boldsymbol{V}_1$。按照此方法执行每一层 RBM,直至训练完所有的 RBM 层。

2)微调。RBM 自下向上训练完之后,只能保证 RBM 本身这一层内的权值对该层的特征提取是局部

最优;因此还需要反向传播网络微调整个网络参数,达到全局最优。在这一步需要将带有标签的数据附加到顶层,使用这些带标签的数据对网络进行区分性、有监督的、自上而下的训练来对整个网络权值进行调整。BP 网络接受最后一层 RBM 输出的特征向量作为其输入数据[14],将 BP 网络输出层得到的实际输出与期望信息两者之间做减法操作,有差值则进行反向传播。

算法 2　DBN 训练

输入:可视层变量 $V_0 = (v_1, v_2, v_3, \cdots, v_i, \cdots, v_m)$。

输出:参数 $\boldsymbol{W}, \boldsymbol{a}, \boldsymbol{b}$。

(1) 将每一条训练数据 $X^{(S)}$ 赋值给第一个 RBM 的可见层 $\boldsymbol{V}_0$,并用 CD 算法训练第一层 RBM。

(2) 训练完第一层 RBM 后,将其输出结果作为下一层 RBM 的输入,继续用 CD 算法训练下一层。

(3) 迭代步骤(1)和步骤(2)直至训练完所有层的 RBM。

(4) 向前计算完之后,误差为:

$$E(\theta, t) = \frac{1}{2}\sum_{i=1}^{m}(v'_i - v_i)^2 \tag{4}$$

式中:v'_i 为重构后的新特征;v_i 为原始输入的特征。

(5) 利用梯度下降算法进行逆向传播,调整网络中各个权值,使误差达到最小值。权值更新公式为:

$$\Delta w_{ij} = -\varepsilon \frac{\partial E}{\partial w_{ij}} \tag{5}$$

(6) 重复步骤(4) - 步骤(5)过程,直至误差足够小,保存参数。

3.3　多分类 TSVM-5 分类器

本文改进的多分类 TSVM 算法 TSVM-5 设计具体分类步骤如下:

1) 先将 KDDCUP99 中正常样本 Normal 标记为 +1,其余四类攻击样本标记为 -1,通过 T1 分类器筛选出正常样本;

2) 将 Dos 和 Probe 两类样本标记为 +1,U2R 和 R2L 标记为 -1,再将剩余四类样本输入到 T2 分类器中。标记为 +1 的样本传到 T3 分类器,标记为 -1 的传到 T4 分类器中;

3) 通过 T3 分类器,输出标记为 +1 的是 Dos 攻击样本,-1 是 Probe 样本;

4) 通过 T4 分类器的,输出标记为 +1 的是 U2R 攻击样本,-1 是 R2L 样本。

4　实　验

4.1　数据来源与预处理

1998 年,林肯实验室在美国空军局域网进行模拟而采集 9 周的网络数据。随后 Sal Stolfo 教授和 Wenke Lee 教授在此基础之上对网络数据进行分析和预处理形成了 KDDCUP99 数据集,该数据集是 IDS 研究领域中被广泛使用的实验数据之一,其中包含 494 021 个训练样本和 311 029 个测试样本,分布情况如表 1 所示。

表 1　KDDCUP99 数据分布情况

标志	攻击类别	KDDTrain	KDDTest
0	Normal	97 278	60 593
1	Dos	391 458	229 853
2	Probe	4 107	4 166
3	R2L	1 126	16 189
4	U2R	52	228

1) 字符特征数值化。KDDCUP99 数据集每一条数据有 38 个数字型属性和 3 个字符型属性,外加 1 个类标签。字符型属性的数据不利于特征提取和分类算法的识别,需要将其数值化。比如 Protocol_type 这一属性有 3 种类型:Tcp、Udp、Icmp,映射规则设置如下:Tcp = 0, Udp = 1, Icmp = 2,将这些转化为数值类型。

2) 数值归一化。想要数据之间的量纲具有可比性而不对实验造成影响,将训练集和测试集中的全部数据记录,都要进行归一化处理,将数据归一到 0 和 1 之间。经过上一步字符映射处理后,将数据用. csv 格式导出,可以看出 KDDTrain 训练集中的第 20、21 两列全为 0,数据归一后这两列会出现差错,因此归一数据之前,对每列数据最大值最小值进行判定:若差值不为 0,进行归一操作;若差值为 0,则不进行归一操作,给定这列归一后的数据全为 0。

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{6}$$

4.2　参数设置

根据已有研究表明,当 DBN 层数到达 7 层及以上,入侵检测识别的准确率趋于稳定值[15]。为了选取 DBN 模型处理 KDDCUP99 数据具有相对较高准确率的层数,本文选取 2 ~ 7 层 DBN 模型,设置 6 种不同的 DBN 网络结构,如表 2 所示。将 DBN 最后的重构误差作为选取 DBN 层数的依据,结果如图 5 所示。由结果可知采用 5 层 RBM 网络结构的误差最小。

表 2 DBN 层数设置

DBN 层数	节点
2	41 – 80 – 5
3	41 – 100 – 50 – 5
4	41 – 100 – 80 – 50 – 5
5	41 – 100 – 80 – 50 – 10 – 5
6	41 – 100 – 80 – 60 – 30 – 10 – 5
7	41 – 70 – 100 – 80 – 60 – 30 – 10 – 5

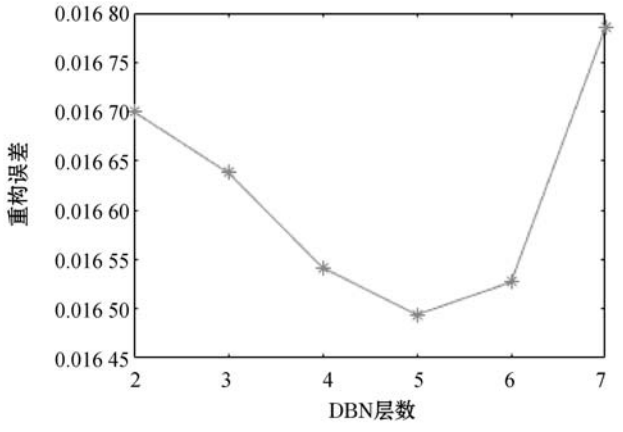

图 5 DBN 不同层数重构误差

KDDCUP99 数据集经过预处理后依然为 41 维特征,因此输入层节点为 41,之后依次选取为 100、80、50、10 和 5,即用 DBN-TSVM-5 网络结构为 41-100-80-50-10-5 对归一后的标准数据进行降维。预训练迭代次数为 30 次,微调权值迭代次数为 100 次。由于 RBF(radial basis function)核函数参数设置少且非线性分类性能好[16],所以本文采用 RBF 作为 TSVM-5 的核函数,设置惩罚因子 C1 = 1,C2 = 1 核函数参数 gamma = 1,最后获得准确率。

4.3 实验结果

采用 Anaconda 的 Python 集成环境,使用 Pycharm 编译器编写程序代码。实验用未改进的二分类对支持向量机模型(TSVM-2)、本文改进的多分类对支持向量机模型(TSVM-5)、基于 DBN 和 TSVM-5 混合模型(DBN-TSVM-5)三种方法对入侵检测进行数据分析。

受到实验硬件环境的限制,分类器的代码数据量太大会导致存储溢出。为了对比分析这三种方法的有效性,随机抽取以下 Data1、Data2、Data3、Data4 四个数据集作为实验数据,见表 3。用 Python 第三方库 sklearn 里的 train_test_split 函数,按照 6∶4的比例将每个数据集分割成训练集和测试集,函数里 straight 参数可以按照数据标签的比例划分,使每个样本类别都分到训练集和测试集。

表 3 实验数据

标志	类别	Data1	Data2	Data3	Data4
0	Normal	2 766	3 169	3 569	3 969
1	Dos	3 576	4 573	5 059	5 559
2	Probe	1 283	1 583	1 966	2 366
3	R2L	323	523	1 226	1 826
4	U2R	52	152	180	280

从准确率(Accuracy,AC)和误报率(False Alarm,FA)来比对三种方法的有效性。表 4 是 TSVM-2、TSVM-5 和 DBN-TSVM-5 三种模型方法在不同数据集上进行实验得到的检测精度和误报率结果。

表 4 实验结果 %

数据集	TSVM-2		TSVM-5		DBN-TSVM-5	
	AC	FA	AC	FA	AC	FA
Data1	87.29	12.72	92.15	8.56	92.20	7.43
Data2	87.71	11.29	93.49	6.95	96.58	4.36
Data3	89.72	10.28	93.96	5.99	97.12	3.87
Data4	90.56	9.43	94.74	5.89	97.93	3.09

可以看出,在不同数据集上 TSVM-5 与 TSVM-2 对比得出,不仅实现了数据样本的多分类,检测率还平均提高了 5.37%。由 TSVM-5 和 DBN-TSVM-5 对比看出,DBN 降维操作可以提取出数据的深度特征,从而更有利于数据的分类与识别,比 TSVM-5 的检测率平均提高了 2.52%。

5 结 语

面对网络环境具有高维、复杂数据的特点,本文提出一种基于 DBN 和改进的 TSVM 入侵检测混合模型。以字符映射和归一化处理后的 KDDCUP99 数据为实验数据进行仿真实验。DBN 具有良好的降维性能,成功地减少了数据特征向量,再把降维后的数据输入到 TSVM-5 多类分类器中检测攻击数据,进行入侵检测识别。实验数据表明,DBN-TSVM-5 模型的检测准确率分别比 TSVM-2 和 TSVM-5 提高了 8.03% 和 2.52%,同时其误报率也有所降低,是一种卓有成效的入侵检测模型。

参 考 文 献

[1] 余淋. 基于深度置信网络的入侵检测研究[J]. 计算机科

学与应用,2018,8(5):687 - 701.
[2] 魏明军,王月月,金建国. 一种改进免疫算法的入侵检测设计[J]. 西安电子科技大学学报(自然科学版),2016,43(2): 126 - 131.
[3] Hodo E, Bellekens X, Hamilton A, et al. Threat analysis of IoT networks using artificial neural network intrusion detection system [C]//2016 International Symposium on Networks,Computers and Communications(ISNCC). IEEE,2016: 1 - 6.
[4] Lee W, Stolfo S J, Mok K W. A data minging framework for building intrusion detection models[C]//Proceedings of the 1999 IEEE Symposium on Security and Privacy. IEEE,1999: 120 - 132.
[5] 张凌杰,张国辉. 基于混合粒子群优化算法的入侵检测研究[J]. 计算机应用与软件,2009,26(4):10 - 12.
[6] 齐玉珠. 基于云计算的入侵检测技术研究[D]. 南京: 南京邮电大学,2014.
[7] Justin V, Marathe N, Dongre N. Hybrid IDS using SVM classifier for detecting DoS attack in MANET application [C]//2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). IEEE, 2017:775 - 778.
[8] 聂盼盼,臧洌,刘雷雷. 基于对支持向量机的多类分类算法在入侵检测中的应用[J]. 计算机应用,2013,33(2): 426 - 429.
[9] Hinton G E, Osindero S, Teh Y W. A fast learning algorithm for deep belief nets[J]. Neural Computation,2006,18(7): 1527 - 1554.
[10] Hinton G E. Training products of experts by minimizing contrastive divergence[J]. Neural Computation,2002,14(8): 1771 - 1800.
[11] 闫涛,周琦. 深度学习算法实践[M]. 北京: 电子工业出版社,2018:351 - 352.
[12] Jayadeva, Khemchandani R, Chandra S. Twin support vector machines for pattern classification[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007,29(5):905 - 910.
[13] 丁世飞,张健,张谢锴,等. 多分类孪生支持向量机研究进展[J]. 软件学报,2018,29(1):89 - 108.
[14] 张克君,鲜敏. 基于 DBN 和 TSVM 的混合入侵检测模型研究[J]. 计算机应用与软件,2018,35(5):319 - 323,339.
[15] 高妮,高岭,贺毅岳. 面向入侵检测系统的 Deep Belief Nets 模型[J]. 系统工程与电子技术,2016,38(9):2201 - 2207.
[16] 高妮,高岭,贺毅岳. 海量数据环境下用于入侵检测的深度学习方法[J]. 计算机应用研究,2018,35(4):1197 - 1200.

(上接第 248 页)

[10] 王鹏. 面向不平衡数据分类问题的核逻辑回归算法的设计与实现[D]. 西安:西安电子科技大学,2015:21 - 38.
[11] 张磊,赵耀,朱振峰. 跨媒体语义共享子空间学习研究进展[J]. 计算机学报,2017,40(6):1394 - 1421.
[12] 张灵均. 多模态数据分类的模糊粗糙方法研究[D]. 天津:天津大学,2017:45 - 80.
[13] 叶婷婷. 多模态特征选择及其在脑疾病分类中的应用研究[D]. 南京:南京航空航天大学,2016.
[14] 杨杨. 面向模态不均衡数据的多模态学习技术研究[D]. 南京:南京大学,2016:35 - 43.
[15] 王世勋. 面向多模态数据的多分类与检索方法研究[D]. 武汉:华中科技大学,2015:35 - 56.
[16] Pereira J C,Coviello E,Doyle G,et al. On the role of correlation and abstraction in cross-modal multimedia retrieval[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2014,36(3):521 - 535.
[17] 刘建伟,刘媛,罗雄麟. 半监督学习方法[J]. 计算机学报,2015,38(8):1592 - 1617.
[18] 杭琦,杨敬辉. 机器学习随机森林算法的应用现状[J]. 电子技术与软件工程,2018(24):125 - 127.
[19] 李芸初. 基于支持向量机的文本分类[J]. 中国新技术新产品,2019(1):23 - 24.
[20] 晋远,孙红三,叶紫,等. 基于大数据 Bayes 分类的家电设备识别算法[J]. 建筑科学,2017,33(4):31 - 38.

(上接第 303 页)

[14] Futrell R, Wilcox E, Morita T, et al. Neural language models as psycholinguistic subjects: representations of syntactic state[EB]. arXiv:1903.03260, 2019.
[15] Lopez-Gazpio I, Maritxalar M, Lapata M, et al. Word n-gram attention models for sentence similarity and inference [J]. Expert Systems with Applications, 2019,132: 1 - 11.
[16] Huang L, Yang Y, Zhao X, et al. Sparse data-basedurban road travel speed prediction using probabilistic principal component analysis[J]. IEEE Access, 2018, 6: 44022 - 44035.
[17] Xu W, Huang L, Fox A, et al. Online system problem detection by mining patterns of console logs[C]//2009 Ninth IEEE International Conference on Data Mining, 2009.
[18] Lou J G, Qiang F, Yang S, et al. Mining invariants from console logs for system problem detection[C]//2010 USENIX Conference on USENIX Annual Technical Conference, 2010.
[19] Zhao X, Rodrigues K, Luo Y, et al. Non-intrusive performance profiling for entire software stacks based on the flow reconstruction principle[C]//Usenix Conference on Operating Systems Design & Implementation, 2016.