ISSN 1673-9418 CODEN JKYTA8
Journal of Frontiers of Computer Science and Technology
1673-9418/2018/12(04)-0575-11
doi: 10.3778/j.issn.1673-9418.1707049

E-mail: fcst@vip.163.com
http://www.ceaj.org
Tel: +86-10-89056056

基于CNN和LSTM深度网络的伪装用户入侵检测*

王 毅[1],冯小年[2],钱铁云[1+],朱 辉[3],周 静[3]
1. 武汉大学 计算机学院,武汉 430072
2. 中国电力财务有限公司,北京 100005
3. 北京汇通金财信息科技有限公司,北京 100094

CNN and LSTM Deep Network Based Intrusion Detection for Malicious Users*

WANG Yi[1], FENG Xiaonian[2], QIAN Tieyun[1+], ZHU Hui[3], ZHOU Jing[3]
1. School of Computer Science, Wuhan University, Wuhan 430072, China
2. China Power Finance Co., Ltd., Beijing 100005, China
3. Beijing Huitong Financial Information Technology Co., Ltd., Beijing 100094, China
+ Corresponding author: E-mail: qty@whu.edu.cn

WANG Yi, FENG Xiaonian, QIAN Tieyun, et al. CNN and LSTM deep network based intrusion detection for malicious users. Journal of Frontiers of Computer Science and Technology, 2018, 12(4): 575-585.

Abstract: The intrusion detection of internal malicious users, as an active security protection technology, has been a hot research topic in recent years. Existing methods are unable to accurately model the users' behavior. This paper proposes a novel CCNN-LSTM method which combines the convolution neural network (CNN) and long short-term memory (LSTM) neural network for camouflage intrusion detection. The basic idea is to use convolution neural network to capture the local correlation in users' activity data, and use long short-term memory neural network to deal with sequential relationship and long-range dependency. The proposed method can automatically learn the representation of data without artificial extraction of complex features, and can also scale to large volume of high dimensional data. The experimental results show that the proposed method has higher detection rate and lower detection cost than a number of baselines.

Key words: intrusion detection of malicious users; depth neural network; convolution neural network; long and short-term memory artificial neural network

* The National Natural Science Foundation of China under Grant No. 61572376 (国家自然科学基金).
Received 2017-07, Accepted 2017-11.
CNKI网络出版: 2017-11-28, http://kns.cnki.net/kcms/detail/11.5602.TP.20171128.0823.002.html

摘 要:用户伪装入侵检测技术作为一种主动式安全防护技术已成为当前的研究热点。现有的用户伪装入侵检测技术存在难以准确建模用户行为模式的缺陷。利用卷积神经网络(convolution neural network,CNN)处理局部关联性数据和特征提取的优势,以及长短期记忆(long short-term memory,LSTM)神经网络捕获数据时序性和长程依赖性的优势,设计了一种结合卷积和长短期记忆的深度神经网络(CCNN-LSTM)用于伪装入侵检测。该方法具有较强的学习能力,能自动学习数据的表征而无需人工提取复杂特征,在面对复杂高维的海量数据时具有较强的潜力。实验结果表明,该方法具有更高的检测率及更低的检测代价,其性能胜过多个基线系统。

关键词:伪装用户入侵检测;深度神经网络;卷积神经网络;长短期记忆人工神经网络

文献标志码:A **中图分类号:**TP311

1 引言

互联网+时代,信息网络已深入国民经济的各个环节,人、物及商业已通过信息网络逐步互联,网络安全问题也日渐凸显出来。从组织信息系统遭受的直接攻击来源看,可将入侵分为外部入侵和内部用户伪装入侵。内部用户伪装入侵是指未授权用户通过伪装成内部合法用户进入系统,访问、修改关键数据或执行其他非法操作的行为[1]。

根据建模用户行为模式所侧重的信息(频率信息、相关信息、转移信息)[2]不同,目前的伪装入侵检测方法分为如下3种类型:(1)基于用户行为频率信息的伪装入侵检测,如文献[3-5],这类方法比较简单、高效,但其忽略了行为序列的时序性,因而其检测率往往较低。(2)基于用户行为相关信息的伪装入侵检测,如文献[6]通过*n*-gram模型考虑了相邻行为间相关信息(局部强相关性),文献[7-8]同时考虑了相邻行为和不相邻行为这两类相关信息。总体来说,这类方法能较好捕获行为序列的局部强相关性以及行为的长程依赖性,但不能准确捕获整个行为序列的时序性,因而其建模用户行为模式的能力有限。(3)基于用户行为转移信息的伪装入侵检测,如文献[1,9-11],这类方法能较好地捕获行为序列的时序信息,但未能兼顾行为的长程依赖性,故建模某些复杂且有长程依赖关系的行为序列时也捉襟见肘。

用户伪装入侵检测关键技术在于如何有效地进行用户行为模式的建模,虽然已有的方法就其用户行为建模能力相比早期的伪装入侵检测方法有较大提升,但面对海量网络数据及复杂高维入侵行为特征等安全挑战时,传统检测技术存在建模能力不足及“维数灾难”等问题。而深度学习运用了分层次抽象思想,能学习非常复杂的函数,能更好地应对大量的高维复杂数据,同时有着更高的建模能力。在深度学习的各种框架中,卷积神经网络(convolution neural network,CNN)需要的参数更少,非常适合处理具有统计平稳性和局部关联性的数据;长短期记忆(long short- term memory,LSTM)神经网络被专门设计用于学习具有长依赖关系的时间序列数据,在学习更高级别特征序列中的长程依赖性和时序性上有着很大的优势。因此,本文就内部用户伪装入侵检测提出了一种结合卷积和长短期记忆的统一网络框架(unified model combined convolution neural network with long short- term memory, CCNN-LSTM)的内部用户伪装入侵检测方法,并进行了大量实验和详细的实验分析。

本文组织结构如下:第2章阐述了CCNN-LSTM网络结构的详细设计和CCNN-LSTM模型学习,并提出了一种新的负样本采集方法;第3章阐述了基于CCNN-LSTM伪装入侵检测的具体流程;第4章通过实验对本文方法进行评估;第5章总结全文并展望未来。

2 CCNN-LSTM模型

2.1 CCNN-LSTM网络结构

2.1.1 问题特征

企业或组织内部常常通过行为监控和收集器进行用户行为的搜集,这些行为按时间段进行截取,便可形成行为流。用户行为流往往具有局部强关联性、长程依赖性以及时序性等。以Shell命令流为例,

对这3种特性进行阐述：(1)用户的Shell命令流中往往有很多局部关联很强的命令块，如“cd ls cat”、“gdb gcc make”等。可以从上述示例发现前一个命令块是用来查看某个文件的，而后者是用来编译某份C++源码的。“查看文件”和“编译C++源码”等用户行为，在系统中具体执行时，往往需要分解为更多的子行为，并由子行为组合而成。从上述观察，可以推断出：用户的抽象行为往往由Shell命令的组合来实现，这正是Shell命令流中很多命令有较强的局部关联性的原因。(2)用户的Shell命令流中某些命令往往和其跨度较远的命令具有很强的关联性，如“login、x…x、logout”和“open、x…x、close”等，这类似于自然语言处理中的长程依赖性。(3)Shell流具有很强的时序性，这往往体现了用户的行为习惯。

卷积神经网络是一种分层的计算模型，随着网络层数的增加其可提取越来越复杂抽象的模式。CNN典型架构为Input→Conv→Pool→FullCon，该架构融合了局部感受野、共享权重以及空间或时间子采样这3种思想。这样的架构使得CNN非常适合处理具有统计平稳性和局部关联性的数据，并使它对平移、缩放、倾斜等具有高度的不变形特性。长短期记忆是一种时间递归神经网络(recurrent neural network，RNN)。它与RNN的区别主要在于它在算法中加入了一个判断信息有用与否的“处理器”，这个处理器作用的结构被称为cell。一个cell中被放置了3扇门，分别叫作输入门、遗忘门和输出门。一个信息进入LSTM的网络中，可以根据规则来判断是否有用。只有符合算法认证的信息才会留下，不符的信息则通过遗忘门被遗忘，在反复运算下解决神经网络中长期存在的问题。因此在卷积和池化层之上选择LSTM可学习更高级别特征序列中的长程依赖性和时序性。

2.1.2　CCNN-LSTM具体结构

基于上述观察和分析，本文结合深度神经网络，设计了一个如图1所示的CCNN-LSTM网络。

首先使用一个卷积层对Shell流进行处理，接着使用基于块的最大池化层[12]进行局部重要特征的挑选，然后使用LSTM层捕获数据的时序性和长程依赖性，最后使用sigmoid激活的全连接层进行正常概率计算。下面将对该网络各层的设计做详细阐述。

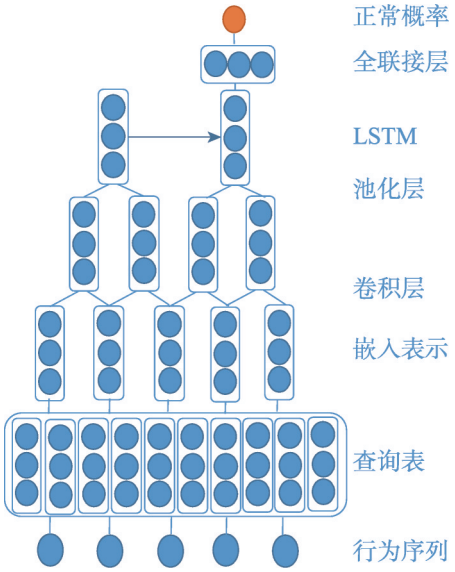


Fig.1　CCNN-LSTM network structure

图1　CCNN-LSTM网络结构示意图

嵌入表示查询层存储了一个各命令的嵌入表 V_C，当该网络面对一个命令序列 $s=[c_1c_2\cdots c_k]$ 时，其中 $v_{c_k}\in\mathbb{R}^{d_c\times 1}$ 表示命令 c_k 的嵌入表示，嵌入表示查询层即对每一个命令 c_k，在 V_C 中查询其嵌入表示。若有命令未曾出现在训练命令集中，即 $c_i\notin C$，直接取查询表 V_C 中第一个位置的嵌入表示 v_0；反之，则取对应命令的嵌入表示 v_{c_i}。经过上述处理后，原始命令token的id序列变成了相应的命令嵌入表示序列 $V_s=[v_{c_1}v_{c_2}\cdots v_{c_k}]$。相比其他的命令表示(如one-hot)，命令嵌入表示具有低维稠密、计算高效的优势。

卷积层对命令嵌入表示序列 V_s 进行卷积操作。具体而言，卷积层设置多个卷积核 $W\in\mathbb{R}^{l_c\times d}$，每个卷积核都对窗口 l_c 中的特征进行概述并且产生一个新的特征。对于窗口 l_c 中的命令嵌入表示 $\boldsymbol{V}_{i:i+l_c-1}$，一个卷积核 W_l ($1\leqslant l\leqslant L$，其中 L 表示该卷积层卷积核的数目)按式(1)产生新的特征 h_i^l：

$$h_i^l=f(W_l\cdot\boldsymbol{V}_{i:i+l_c-1}+b_l) \tag{1}$$

其中，f 是卷积层使用的非线性激活函数(ReLU)；$W_l\in\mathbb{R}^{l_c\times d_c}$ 是该层的第 l 个卷积核；b_l 是该卷积核的偏置项；$\boldsymbol{V}_{i:i+l_c-1}\in\mathbb{R}^{d_c\times l_c}$ 是 $v_i,v_{i+1},\cdots,v_{i+l_c-1}$ 堆叠而成的矩

阵。当一个卷积核对命令嵌入表示序列 V_s 从 $\boldsymbol{V}_{0:l_c-1}$ 到 $\boldsymbol{V}_{k-l_c+1:k}$ 中的每个窗口进行遍历后，便得到了该卷积核所产生的新的特征图，如式(2)所示：

$$H^l=[h_1^l,h_2^l,\cdots,h_{k-l_c+1}^l] \tag{2}$$

将所有卷积核产生的特征图进行堆叠，便可得到新的序列表示 $H_s=[h_1,h_2,\cdots,h_{k-l_c+1}]$，其中 $h_i\in\mathbb{R}^{L\times 1}$ 是拼接 L 个卷积核在第 i 步产生的特征而形成的表示，可被看作用户的抽象行为表示。

池化层对新的序列表示 H_s 进行池化操作。给定一个预定义的块数 N，首先将 H^l 划分为 N 个片段，并将每个片段中的最大值顺序拼接起来，得到一个长度为 N 的向量 $\boldsymbol{p}^{lN}$，如式(3)所示：

$$\boldsymbol{p}^{lN}=chunkMax\{H^l\}=[h_{n_1}^l,h_{n_2}^l,\cdots,h_N^l] \tag{3}$$

将 L 个特征图的 $\boldsymbol{p}^{lN}$ 堆叠起来，便可得到 $P_s=[\boldsymbol{p}_{n_1},\boldsymbol{p}_{n_2},\cdots,\boldsymbol{p}_N]$，其中 $\boldsymbol{p}_{n_i}\in\mathbb{R}^{L\times 1}$ 为 L 个特征图的第 n_i 块最大池化后而形成的向量。

为捕获数据的时序性和长程依赖性，在 P_s 上使用LSTM架构的网络层。该层由多个共享权值的LSTM区块按时间步的顺序连接而成。在每个时间步上，LSTM区块的输出将由3个门遗忘门 f_t、输入门 i_t、输出门 o_t 协同控制，其中每个门都是一个关于历史区块输出 b_{t-1} 和当前时间的输入 p_t 的矢量函数。这些门共同决定怎样更新当前的记忆细胞体状态以及当前的隐藏状态。使用 d_h 来表示LSTM区块中记忆细胞体状态的维度，同时该架构中其他向量也使用相同的维度。在时间步 t 上，LSTM区块对 p_t 按式(4)~(9)所示进行处理：

$$i_t=\sigma(W_i\cdot[h_{t-1},p_t]+b_i) \tag{4}$$

$$f_t=\sigma(W_f\cdot[h_{t-1},p_t]+b_f) \tag{5}$$

$$q_t=\tanh(W_q\cdot[h_{t-1},p_t]+b_q) \tag{6}$$

$$o_t=\sigma(W_o\cdot[h_{t-1},p_t]+b_o) \tag{7}$$

$$c_t=f_t\odot c_{t-1}+i_t\odot q_t \tag{8}$$

$$b_t=o_t\odot\tanh(c_t) \tag{9}$$

其中，σ 是一个逻辑函数；tanh是双曲正切函数；$\odot$ 表示元素乘运算。直观上，可以将遗忘门 f_t 视为控制来自旧区块的信息以多大程度被丢弃的函数，输入门 i_t 控制有多少信息需要被存储在当前记忆细胞体中，输出门 o_t 基于当前记忆细胞体状态决定需要输出什么。

经过LSTM网络的处理后，将最后一步的LSTM区块的输出作为该序列的表示 r_s，即如式(10)所示：

$$r_s=b_T \tag{10}$$

其中，r_s 代表序列的抽象表示；T 表示LSTM网络输入的最大长度，即池化层输出 P_s 的最大长度。若 $|P_s|<T$，则用0向量补齐；b_T 表示处理第 T 步的序列值是LSTM区块的输出。

最后，用一个全连接层直接对该序列表示进行处理，该层使用逻辑函数 σ 作为激活函数，该层得到该序列是用户正常行为序列的概率 $P_\theta(s)$，其按式(11)计算：

$$P_\theta(s)=\sigma(S_\theta(s))=\frac{1}{1+\mathrm{e}^{-S_\theta(s)}} \tag{11}$$

其中，$\theta=\{V_C,W_{hc},W_{hi},W_{hr},W_u\}$ 为模型参数；$W_u\in\mathbb{R}^{m\times 1}$ 是概率输出层的参数。

2.2 CCNN-LSTM模型实现

2.2.1 CCNN-LSTM模型损失函数

根据最大似然估计，本文将损失函数设为模型在训练样本上的负对数似然性，即损失函数 $L(\theta)$ 按式(12)计算：

$$L(\theta)=-\lg P(D|\theta) \tag{12}$$

其中，$\theta=\{V_C,W_{hc},W_{hi},W_{hr},W_u\}$ 为模型参数；D 表示整个训练数据集；$P(D|\theta)$ 表示参数设为 θ 时模型在训练数据集 D 上的似然性。本模型的训练样本分为正、负样本，假设用户的历史行为皆为正常，则用户的历史行为序列片段皆为正样本；对负样本，将在下一节提出改进的采集方法。根据2.1节的描述，可知某个样本为正常的概率为 $P_\theta(s)$，即 $P(y=1|x=s;\theta)=P_\theta(s)$，那么该样本为异常的概率即为 $P(y=0|x=s;\theta)=1-P_\theta(s)$，则有：

$$\lg P(D|\theta)=\sum_{s_i\in D}y_{s_i}\cdot\lg P(y=1|x=s_i;\theta)+(1-y_{s_i})\cdot\lg P(y=0|x=s_i;\theta) \tag{13}$$

其中，y_{s_i} 表示行为序列 s_i 的标签，其取值为1或0，$y_{s_i}=1$ 时代表序列 s_i 为用户的正常序列，$y_{s_i}=0$ 时代表序列 s_i 为用户的异常序列(伪装入侵行为)。

确定了损失函数后，模型的学习问题即转换为最小化损失函数的优化问题。采用Adam优化算法对模型进行优化求解。对概率输出层、卷积层和嵌入层，使用经典的反向传播算法进行误差的反向传播。而对于LSTM层，采用后向传播算法进行误差的反向传播。

2.2.2　基于序列表征的负样本采集方法

伪装入侵检测中，能轻易获得用户的正样本（正常命令序列），却很难获得其负样本（伪装入侵的命令序列）。目前大多数的采集负样本的做法都是直接采集其他用户的所有命令序列或随机采集一定数量的命令序列，然而这种采集方法并未考虑用户自身的特性，即不同特性的用户其未来正常行为和历史正常行为有着不同的“紧密度”。本文认为若一个用户的历史行为比较固化单一，那么其未来行为也极有可能和历史行为相似；然而若一个用户的历史行为极其丰富多变，那么其未来行为可能和历史行为差距较大。

基于上述假设，本文提出了一种基于序列表征的负样本采集算法。简言之，该算法利用用户的历史命令序列表征，计算用户历史数据的散布程度，若散布程度高，则从其他用户命令序列中选择和本用户历史命令序列最不相似的作为其负样本；若散布程度低，则从其他用户命令序列中选择和本用户历史命令序列相似的作为其负样本。本文方法能够较好地在具有一定行为模式的用户之间进行负样本采集。

3　基于CCNN-LSTM的内部用户伪装入侵检测

本文提出的CCNN-LSTM网络运用于内部用户伪装入侵检测的框架如图2所示。

以图2中的用户A为例说明基于CCNN-LSTM伪装入侵检测的具体过程。

（1）使用命令数据搜集及解析模块进行数据收集及预处理，主要收集用户A的Shell命令，按时序排序，截取固定时间窗口内的数据，形成短的命令流数据，并对其进行过滤和主干抽取等预处理。

（2）使用上文提出的负样本采集方法进行负样本的采集，该阶段主要包括：由用户A的正样本（历史行为数据）训练出一个Seq2vec模型，使用该模型计算其他用户正样本的表征，在此基础上使用2.2.2小

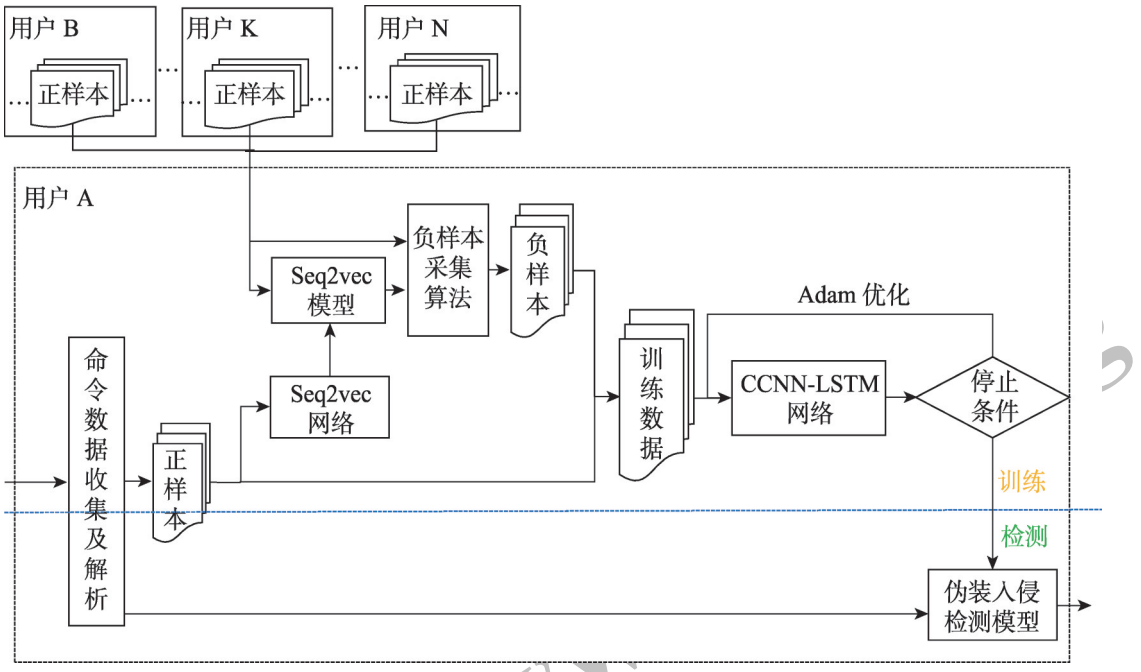


Fig.2　Camouflage intrusion detection framework based on CCNN-LSTM

图2　基于CCNN-LSTM的伪装入侵检测框架

节提出的算法进行负样本采集。

(3)由上述得到的正样本和负样本训练CCNN-LSTM网络得到用户A的伪装入侵检测模型,使用该模型即可检测用户A的行为是否是伪装入侵行为。

对组织信息系统内的关键岗位的用户都进行上述流程的处理,即可得到相应用户的伪装入侵检测模型,从而进行组织信息系统内部用户伪装入侵的检测。

4 实验及分析

4.1 实验数据集及评价指标

为了和其他文献中的方法进行公平对比,本文采用已成为伪装入侵检测领域标准数据集的SEA数据集(http://www.schonlau.net/)[3-4,7]对上述提出的方法进行实验评价。

SEA数据集包含50个用户(User1~User50)的命令文件和一个masquerade_summary.txt文件,其中每个用户文件一行记录一个用户命令(为了保护隐私每个命令的参数未被公开),总共包含15 000个用户命令,前5 000个命令为用户的正常命令,用作训练数据,而后10 000个命令被划分为100个块,这些块中可能散布着入侵者的数据,用作测试数据。masquerade_summary.txt文件保存了一个100×50的矩阵数据,行号代表测试数据中的命令块号,列号代表用户号,矩阵使用0/1来指示某用户测试数据中的某块命令是否为伪装入侵命令块(0代表该块为正常命令块,1代表该块为伪装入侵命令块)。

本文实验遵循数据集固有的划分,即每个用户的前5 000个命令块作为训练数据,后10 000个命令作为测试数据。本文方法需要训练样本和测试样本的长度一致,因此也参照测试数据集中100个命令为一个块(Block)的方式将训练数据集划分为50个块。另外,需要说明的是在进行序列表征学习时,为了能让序列表征模型拥有更多的训练样本,采取了固定窗口在序列上滑动的方式进行样本采取,具体地设置了一个长度为100的窗口,以每步滑动5个命令的步长在每个用户的训练数据上进行滑动,共产生了1 000个用于进行序列表征学习的样本。

实验评价指标方面,除了采用检测率(detection rate,DR)和误报率(false positive rate,FPR)外,也采用了漏检率(miss rate,MR)和检测代价Cost[3],它们的定义如下所示:

$$MR = 1 - DR \tag{14}$$

$$Cost = \alpha \times MR + \beta \times FPR \tag{15}$$

其中,α、β采用$\alpha=1$、$\beta=3$和$\alpha=1$、$\beta=6$两种设定,分别简写为Cost#3和为Cost#6(后文无明确说明情况下,Cost#6也可简写为cost)。

4.2 实验设置

本文实验在Ubuntu 16.04 LTS环境下进行,使用Keras2.0.2神经网络库构建网络,使用Google开源的人工智能系统Tensorflow1.1.0作为后端计算框架。实验中以Normal(0,0.005)分布初始化嵌入层以及卷积层的权重和偏置,以glorot _normal(0)分布初始化LSTM层和全连接层的权重和偏置。使用mini-batch的方式对网络进行小批量梯度更新,当对CCNN-LSTM入侵检测模型进行训练时,*mini-batch-size*设为32,当对序列表征模型进行训练时,*mini-batch-size*设为512。由于用于CCNN-LSTM入侵检测模型的训练样本集较小,将该网络的训练迭代轮数*epoch*设为30,同时设置训练数据集上$loss \leqslant 0.09$时进行早停的规则。对序列表征模型进行训练时,将其迭代轮数*epoch*设为300。使用Adam学习算法进行模型参数学习。

4.3 实验结果与分析

4.3.1 CCNN-LSTM网络超参数分析

(1)Shell命令嵌入表示维度的影响效果

为了探究Shell命令嵌入表示维度对该网络入侵检测性能的影响,本小节设置了8组变换维度的实验。设置的最低维度为10,最高维度为250。先以20为步长,取得4组维度进行实验;再从100以50为步长,取得另外4组维度进行实验。实验过程中,卷积核长度设为3。实验结果如图3所示。该结果显示当嵌入表示的维度从10增加到250时,伪装入侵的漏检率在维度为10时最高为66.23%,而后逐步下降,当维度为70时达到最低漏检率24.97%,最后再缓慢增长到29.87%;检测代价Cost,也大致呈现先下降,再

缓慢增长，最后趋至平稳的规律；而误检率无明显规律可循。从分析极值来看，当维度为10时，3项指标皆为该系列实验值的极值。这表明当该网络维度过低时，网络的学习能力不足，其将样本判断为大类样本的惰性极强。当维度激增至250时，其误检率和检测代价相对 $d=10$ 都有明显下降，表明其相对于 $d=10$ 的网络有了更强的学习能力，然而此时的结果和 $d=50$ 结果类似，并且各项指标皆低于 $d=100$ 时的指标，表明嵌入表示过大时会出现一定性能的下降，维度变大时，理论上网络会有更强的学习能力，但对于小数据来说也存在极大的过拟合风险。因此对于本文的小数据集，反而是在处于中间维度的100维时将达到最好的综合性能。

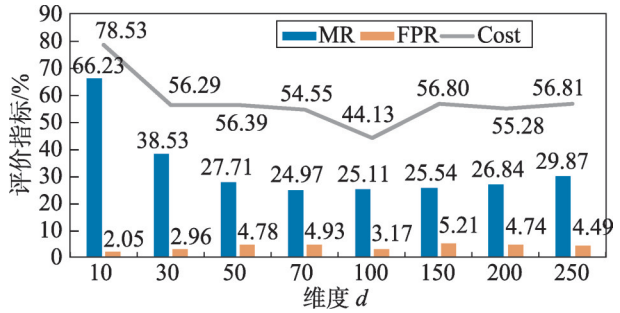


Fig.3　Effect of Shell command embedded dimension on detection result

图3　Shell命令嵌入表示维度对检测效果的影响

（2）卷积核长度的影响效果

本小节设置了6组不同的卷积核长度进行实验，以分析卷积核大小对网络性能的影响。实验过程中，命令嵌入维度设为100。实验结果如图4所示。从图中可以看出，3项指标（漏检率、误检率、检测代价）皆有先下降再上升的规律。除了漏检率是在 $ks=2$ 时达到最低为24.24%，其他两项指标误检率和检测代价都是在 $ks=3$ 时达到最低，分别为44.13%和3.17%。这表明 $ks=2$ 时，网络对攻击样本有最好的检测准确度，但也牺牲了对正常样本的检测准确度。相较之下 $ks=3$ 时虽没有达到对攻击最好的检测率，却达到了最低的误检率，即其对正确样本被误分的比率更小，因此其综合指标cost达到了最低，故 $ks=3$ 时网络的综合性能最好。从图中可以进一步得出这样的结论：①命令序列有强的局部关联性，$ks=1$ 时即上下文窗口为1，每次仅考虑一个命令，该情况退化为忽略局部关联性的情形，此时该网络的3项指标都最高，即网络的检测性能最低；②卷积核长度不宜过大，过大会一定程度损伤网络的检测性能，这是由于较长的卷积核长度会引入数据稀疏性，同时也会使模型参数过多难以训练。

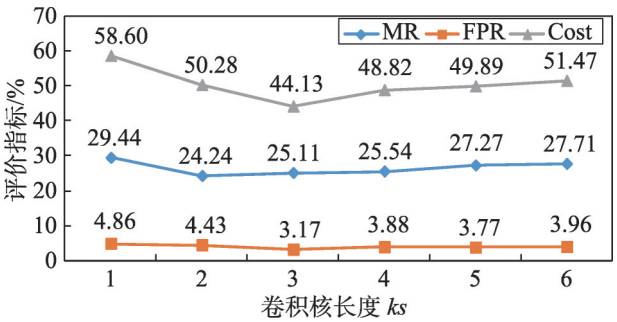


Fig.4　Effect of convolution core length on detection result

图4　卷积核长度对检测效果的影响

4.3.2　负样本采集方法对检测效果的影响

本文提出的基于CCNN-LSTM检测算法本质上是一种监督式算法，而本实验使用的SEA数据集的训练集中只有每个用户的正常样本，因此需要人为采集负样本。据调研，使用SEA数据集进行基于监督式算法的伪装入侵检测方法中，研究者们大多采用从其他用户的正样本中随机采集负样本的策略。第2章提出了一种基于序列表征的负样本采集算法。本小节对负样本采集算法超参数进行分析。

根据2.2.2小节的描述可知，本文提出的负样本采集算法需要设置两个超参数：采集的负样本个数 k、用户正样本散度阈值（mean squared error threshold，MSET）。本小节就这两个超参数进行实验分析，实验过程中采用上一小节调节出的最优网络超参数进行实验。

在进行负样本个数 k 分析时，固定用户散度阈值为5.0，并设置7组 k 值（$k=1,5,10,20,30,40,50$）进行实验。实验结果如图5所示，可以看出随着负样本个数 k 的增大，漏检率和误检率分别呈现线性下降和线性上升的趋势。当 $k=1$ 时，误检率最低仅为0.06%，即模型将99.94%的正样本都判断正确，但漏检率高达100%，即此时模型会将所有的异常样本误判为正

常样本，可见此时模型的区分能力极弱，即当负样本过少时模型几乎无判别能力。随着负样本的增多，模型有更强的负例识别能力，但也降低了模型的正例识别能力。当 $k=5$ 时，该模型的误检率和漏检率分别为25.11%、3.17%，虽然都未达到最佳，但cost最低，即此时模型的综合检测效果最好。这是由于测试集中负样本约占5%，即测试集中正负样本比例约为20∶1。目前的机器学习都建立在训练测试同分布的假设上，按照这个假设训练数据集的正负样本比例也应大致为20∶1，即每个用户训练集中应大致拥有2.5个负样本，但样本规模太小，易受噪声样本的影响，因此5个左右的负样本应是一个比较好的选择，实验结果也证实了这一点。

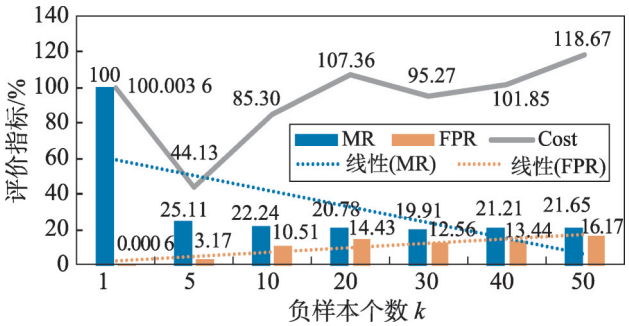


Fig.5 Effect of number of negative samples on detection result

图5 负样本个数对检测效果的影响

进行用户正样本散度阈值分析时，负样本个数固定为5，并设置了7组正样本散度阈值（*MSET*=4.7，4.8，4.9，5.0，5.1，5.2，5.3）进行实验。实验结果如图6所示。从该实验结果可以看出，当散度阈值从最低的4.7按0.1的步长增长到最高的5.3时，入侵的漏检率呈现逐步下降的趋势，而误检率呈现逐步上升的趋势，两者的综合指标cost呈现“两头缓平，中间凹陷”的规律，这和本文的预期基本一致。当用户的散度阈值设置得过小时，算法容易对一些散度较低的用户也采用“挑选最远负样本”模式进行负样本采集，这会造成CCNN-LSTM模型将会学习出更宽泛的正样本区域，即模型对正样本有一定的偏向，能比较好地判断出正样本，但同时也增大了将负样本错判为正样本的风险，因此会呈现漏检率高、误检率低的实验结果；当用户散度阈值设置得过大时，CCNN-LSTM模型对一些散度大的用户也学习出比较“紧致”的正例区域，这样虽然会使模型以比较高的置信度判断出正样本，同时能辨识出更多的负样本，但由于散度大用户的特点，也使模型存在误判更多的正样本为负样本风险，会呈现误检率高、漏检率低的实验结果。从cost指标可以看出，当*MSET*=5.0时，模型的综合检测效果达到最好，此时cost仅为44.13%。

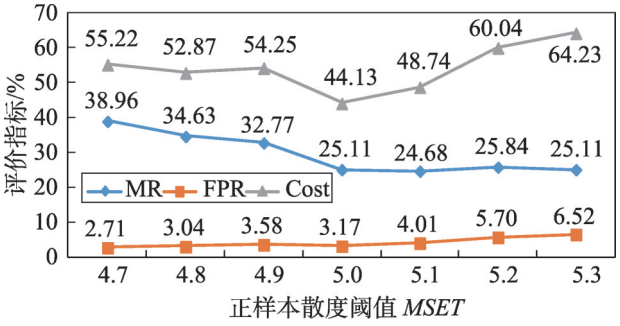


Fig.6 Divergence threshold analysis of user positive samples

图6 用户正样本散度阈值分析

4.3.3 与其他检测方法的比较

（1）与常见深度神经网络的比较

据本文所知，目前尚无深度神经网络方法被运用到伪装入侵检测领域。因此，本小节仅和目前深度学习领域几种经典的深度神经网络进行比较，它们分别是CNN、RNN、LSTM。为了公平比较，上述网络也和本文方法一样设置两个隐藏层，即CNN采用了 Input→Conv→Maxpool→Conv→Maxpool→Dense 的结构，RNN和LSTM都采用了 Input→RNN1(LSTM1)→RNN2(LSTM2)→Dense 的结构，本文方法采用了 Input→Conv→Maxpool→LSTM→Dense 的结构，且本文方法中Conv与LSTM层与上述网络相应层的设计一致。各方法实验结果如表1所示，从入侵检测效果综合指标cost来看，CNN最差，RNN和LSTM次之，本文提出的方法最好。CNN主要获取数据的局部关联性，而RNN和LSTM主要获取数据的序列性，该结果表明用户命令数据的序列性比局部关联性更能体现用户行为模式，从而提高伪装入侵的检测效果。值得注意的是LSTM的cost值比RNN低了4.92%，即LSTM的综合检测效果会略优于RNN，这

是因为LSTM能捕获长程的依赖性引起的。在用户Shell命令中某些命令的出现往往是和自己距离比较远的命令相关联的,如用户先以rlogin命令远程登录了某台线上机器,经过ftp、make、…、launch等操作后,其键入了exit命令登出了线上机,根据用户的历史习惯,其在线下键入launch后可能是键入explore,但若是在线上键入launch则下一个极可能键入exit,即exit命令和离它较远的rlogin命令有了很强的关联性。本文方法不仅捕获了数据的局部关联性、时序性,也捕获了其长程依赖性,因此表现出了最低的检测代价,故本文方法相比其他深度神经网络更适合伪装入侵检测。

Table 1　Comparison with common depth neural networks

表1　与常见深度神经网络的比较　%

Methods	DR	FPR	Cost
CNN	33.33	9.60	124.27
RNN	41.99	**2.83**	74.99
LSTM	51.95	3.67	70.07
Proposed Method	**74.89**	3.17	**44.13**

(2)与其他伪装入侵检测方法的比较

已有众多伪装入侵检测方法被研究者们提出,包括Compression[1]、Sequence Matching[1]、IPAM(incremental probabilistic action modeling)[1]、Uniqueness[1]、Hybrid multistep Markov[1]、Bayes one-step Markov[1]、SVM(support vector machine)[4]、Two-class NB[3]、IWNB(instance- weighted Naive Bayes)[13]、PHMM(profile hidden Markov model)[14]、ECM(eigen co-occurrence matrix)[7],皆在SEA数据集上进行了实验验证,实验结果如表2所示。本小节将简要介绍这些方法,并对实验结果进行分析,给出实验结论。

由表2中可知,入侵检测率上,SVM远远超过大部分方法,达到80.1%,本文方法仅次于该方法达到74.9%。概括来看,基于机器学习的检测方法相比其他非机器学习方法(如Compression、Sequence Matching、Uniqueness等)有更高的检测率;误检率上,Uniqueness最低为1.4%,该方法本质上是一种基于规则的方法,其并无学习过程,故不能学习数据中的知识,虽然其表现出最低的误检率,但检测率极差,实用价值低;入侵检测代价上,当β=6时,ECM表现出极低的检测代价42.7%,本文方法仅次于该方法检测代价为44.1%。而当β=3时,本文方法的入侵代价低至34.6%,为所有方法最佳。和本文方法一样ECM方法也捕获了用户命令的局部相关性和长程依赖性,然而该方法需要根据全局的特征共现矩阵构建用户的众多分层网络,因此难以实现增量训练。本文方法可以通过在线训练的方式实现增量学习和在线学习,且本文方法能应对大数据量的训练数据,有一定的实用性。

Table 2　Comparison with related literature methods

表2　与相关文献方法的比较　%

Methods	DR	FPR	Cost#6	Cost#3
Compression	34.2	5.0	95.8	80.8
Sequence Matching	36.8	3.7	85.4	74.3
IPAM	41.1	2.7	75.1	67.0
Uniqueness	39.4	**1.4**	69.0	64.8
Hybrid multistep Markov	49.3	3.2	69.9	60.3
Bayes one-step Markov	69.3	6.7	70.9	50.8
SVM	**80.1**	9.7	78.1	49.0
Two-class NB	66.2	4.6	61.4	47.6
PHMM	70.0	5.0	60.0	45.0
IWNB	70.2	4.5	56.8	43.3
ECM	72.3	2.5	**42.7**	35.2
Proposed Method	74.9	3.2	44.1	**34.6**

另外,从方法本身的角度来看,统计分析方法(Compression、Sequence Matching、IPAM、Uniqueness、Hybrid multistep Markov、Bayes one-step Markov)表现出了很高的检测代价,表明这类方法具有较差的综合检测效果,仅靠少量统计参数来刻画用户复杂的行为模式,建模能力不足;SVM和Two-class NB都是通过捕获用户命令的频率信息,用机器学习模型来建模用户行为模式。从Cost#3指标来看,两者远超过Uniqueness(利用频率信息),但仅勉强超过Bayes one-step Markov(利用命令转移信息)。一方面表明机器学习方法的用户行为建模能力比传统的统计分析方法要强,另一方面也表明捕获命令序列的何种信息对检测效果也有较大影响。PHMM无论在Cost#6还是Cost#3上,都超过了SVM和Two-NB,其主要捕获了命令序列的转移信息,序列建模能力较强,这也表明命令序列的时序性能较好反应用户

的行为模式。IWNB是一种半监督学习方法,其虽然也是基于频率信息建模用户行为模式,但其在训练过程中运用了无标签的测试数据进行模型的微调,故比SVM、Two-NB、PHMM等方法都要好。ECM方法和本文方法在Cost#6和Cost#3上的值都比较接近,分别达到所有方法的最佳值,表明两者都有较强的综合检测效果。需要注意的是,这两种方法都捕获了数据的局部关联性和长程依赖性,表明这两种信息对用户的行为建模极其有效。另外,本文方法也捕获了数据的时序性,在应对更多的训练数据时会呈现更大的潜力。

5 总结与展望

5.1 本文工作总结

本文研究与分析了现有检测方法,发现目前的检测方法存在难以准确捕获用户复杂行为模式的缺点。然后,以此为切入点,利用CNN捕获数据局部关联性的优势以及LSTM捕获数据序列性和长程依赖的优势,设计了一种CCNN-LSTM网络结构。同时,给出了具体的模型学习过程。最后,编码实现了该方法并在SEA数据集上进行了大量实验,实验结果表明该方法在伪装用户入侵检测上具有优异的综合检测效果,有较强的潜力。针对内部用户伪装入侵检测中负样本获取困难的问题,提出了一种新的用户样本采集方法。即通过用户正常行为的散度,有策略地采集其他用户的样本作为该用户的负样本。实验结果表明,该方法优于现有的负样本采集方法,对后续的伪装入侵检测算法具有较大的补益。

5.2 未来工作展望

未来工作可按如下几方面展开:

(1)面对小、中型数据集,考虑将多种异构模型的结果进行融合,如将CNN、LSTM、SVM、NB等结果进行Stacking集成得到最终检测结果。

(2)内部用户伪装入侵检测可以使用自动编码机或基于LSTM的seq2seq进行检测,概括讲即通过编码器压缩原始输入,然后由解码器重构输入,最后通过比对重构输入和原始输入之间的差距进行正常和异常的判断,从而进行入侵检测。

(3)针对负样本稀少的问题,可以尝试使用生成式对抗网络(generative adversarial networks,GAN)学习出负样本生成器,进行负样本的生成。

References:

[1] Schonlau M, Dumouchel W, Ju Wenhua, et al. Computer intrusion: detecting masquerades[J]. Statistical Science, 2001, 16(1): 58-74.

[2] Li Chao, Tian Xinguang, Xiao Xi, et al. Anomaly detection of user behavior based on Shell commands and co-occurrence matrix[J]. Journal of Computer Research and Development, 2012, 49(9): 1982-1990.

[3] Maxion R A, Townsend T N. Masquerade detection using truncated command lines[C]//Proceedings of the 2002 International Conference on Dependable Systems and Networks, Bethesda, Jun 23-26, 2002. Washington: IEEE Computer Society, 2002: 219-228.

[4] Kim H S, Cha S D. Empirical evaluation of SVM-based masquerade detection using UNIX commands[J]. Computers & Security, 2005, 24(2): 160-168.

[5] Zhao Xiang, Hu Guangyu, Wu Zhigong. Masquerade detection using support vector machines in the smart grid[C]// Proceedings of the 7th International Joint Conference on Computational Sciences and Optimization, Beijing, Jul 4-6, 2014. Washington: IEEE Computer Society, 2014: 30-34.

[6] Hofmeyr S A, Forrest S, Somayaji A. Intrusion detection using sequences of system calls[J]. Journal of Computer Security, 1999, 6(3): 151-180.

[7] Oka M, Oyama Y, Abe H, et al. Anomaly detection using layered networks based on eigen co-occurrence matrix[C]// LNCS 3224: Proceedings of the 7th International Symposium on Recent Advances in Intrusion Detection, Sophia Antipolis, Sep 15-17, 2004. Berlin, Heidelberg: Springer, 2004: 223-237.

[8] Wang Xiuli, Wang Yongji. Masquerader detection based on command closeness model[J]. Acta Electronica Sinica, 2014, 42(6): 1225-1229.

[9] Lane T, Brodley C E. An empirical study of two approaches to sequence learning for anomaly detection[J]. Machine Learning, 2003, 51(1): 73-107.

[10] Tian Xinguang, Sun Chunlai, Duan Miyi. Anomaly detection of user behaviors based on Shell commands and Markov chain models[J]. Journal of Electronics & Information Technology, 2007, 29(11): 2580-2584.

[11] Xiao Xi, Tian Xinguang, Zhai Qibin, et al. Masquerade de-

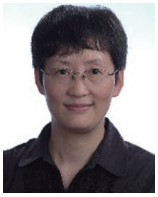
tection based on shell commands and Markov chain models [J]. Journal on Communications, 2011, 32(3): 98-105.
[12] Zhang Jiajun, Zhang Dakun, Hao Jie. Local translation prediction with global sentence representation[C]//Proceedings of the 24th International Joint Conference on Artificial Intelligence, Buenos Aires, Jul 25-31, 2015. Menlo Park: AAAI, 2015: 1398-1404.
[13] Sen S. Using instance-weighted Naive Bayes for adapting concept drift in masquerade detection[J]. International Journal of Information Security, 2014, 13(6): 583-590.
[14] Huang Lin, Stamp M. Masquerade detection using profile hidden Markov models[J]. Computers & Security, 2011, 30 (8): 732-747.

附中文参考文献:

[2] 李超, 田新广, 肖喜, 等. 基于Shell命令和共生矩阵的用户行为异常检测方法[J]. 计算机研究与发展, 2012, 49 (9): 1982-1990.
[8] 王秀利, 王永吉. 基于命令紧密度的用户伪装入侵检测方法[J]. 电子学报, 2014, 42(6): 1225-1229.
[10] 田新广, 孙春来, 段洣毅. 基于shell命令和Markov链模型的用户行为异常检测[J]. 电子与信息学报, 2007, 29(11): 2580-2584.
[11] 肖喜, 田新广, 翟起滨, 等. 基于shell命令和Markov链模型的用户伪装攻击检测[J]. 通信学报, 2011, 32(3): 98-105.

WANG Yi was born in 1993. He received the M.S. degree in computer technology from Wuhan University in 2017. His research interests include Web mining and data management, etc.
王毅(1993—),男,四川南充人,2017年于武汉大学计算机学院获得硕士学位,主要研究领域为Web挖掘,数据管理等。

FENG Xiaonian was born in 1969. He received the M.S. degree in computer science from Huazhong University of Science and Technology in 1998. Now he is a senior engineer at China Power Finance Co., Ltd. His research interests include data security and data management, etc.
冯小年(1969—),男,湖北阳新人,1998年于华中科技大学获得硕士学位,现为中国电力财务有限公司高级工程师,主要研究领域为信息安全,数据管理等。



QIAN Tieyun was born in 1970. She received the Ph.D. degree in computer science from Huazhong University of Science and Technology in 2006. Now she is a professor and Ph.D. supervisor at Wahan University, and the member of CCF, ACM and IEEE. Her research interests include Web mining and data management, etc.
钱铁云(1970—),女,浙江诸暨人,2006年于华中科技大学获得博士学位,现为武汉大学计算机学院教授、博士生导师,CCF会员,ACM会员,IEEE会员,主要研究领域为Web挖掘,数据管理等。

ZHU Hui was born in 1983. He received the M.S. degree in statistics from Guizhou University of Finance and Economics in 2009. Now he works at Beijing Huitong Financial Information Technology Co., Ltd. His research interests include power finance, electronic commerce and project management, etc.
朱辉(1983—),男,河南信阳人,2009年于贵州财经大学获得硕士学位,目前在北京汇通金财信息有限公司工作, 主要研究领域为电力金融,互联网金融,项目管理等。

ZHOU Jing was born in 1984. He received the M.S. degree in business administration from North China Electric Power University. Now he is a vice general manager at Beijing Huitong Financial Information Technology Co., Ltd. His research interests include smart grid, information communication, big data and project management, etc.
周静(1984—),男,安徽马鞍山人,硕士,北京汇通金财信息有限公司副总经理,主要研究领域为智能电网,信息通信,大数据,项目管理等。