5　实验与分析

在 Windows 10 操作系统环境下，基于 TensorFlow-GPU 1.8.0 深度学习框架、Python3.6 开发语言，设计实现了算法模型代码。对本文数据预处理中的特征约简，以及完整模型的效果进行了对比实验，来验证方法的有效性。

（1）冗余特征约简效果对比

为了验证冗余特征约简效果，需要在数据预处理部分采用和不采用约简步骤，得到两组不同的规整特征向量，输入到本文模型中进行训练、测试，将得到约简前后的测试精度进行对比，从而说明冗余特征约简的有效性。冗余特征约简的效果对比如表 1 所示：

表 1 冗余特征约简对比

	未使用冗余特征约简	使用冗余特征约简
训练样本数	500 万	500 万
测试样本数	29.2 万	29.2 万
测试精度	96.79%	96.66%
平均测试时间	61.79s	45.01s

从表中可以看出，在相同训练样本和测试样本下，冗余特征约简后，精度较未约简降低了 0.13%，原因是约简特征时，去除特征的权值不为 0，因此其对最终的精度是有一定影响的。但是去除特征后时间缩短了将近 17s。因此，冗余特征约简是有效的，其能够在不大幅度影响精度的条件下，节省比较可观的时间。

在使用冗余特征约简方法时，考虑了模型训练中的损失值变化如图 2 所示：

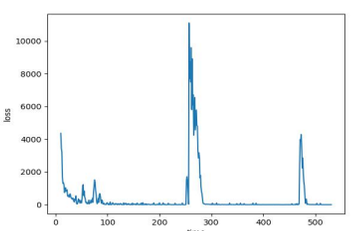

图 2 模型训练损失变化图

可以看出，模型训练损失值在三个时间段内都很高，然后通过后训练损失值逐渐降低，最后损失值在 0 附近波动，说明训练对部分数据已经有了很好的特征提取，因此能够很准确地识别训练数据。

同时，经过对数据进行分析，找到了为什么会出现两次损失的原因：不同标签的数据分布不均匀，可能某批次的数据中包含与前几次训练标签不同的数据，因此在该轮训练中损失值会非常高，然后通过多轮训练将损失值再次降低，最终将损失值降低到 0 附近，实现对数据集的完整训练。

（2）模型效果对比

未来验证本文基于 DNN 的异常行为检测模型的效果，都采用本文的数据预处理方法，对比测试了基于 DBN 和 SVM 算法的两种情况，性能对比如表 2 所示：

表 2 三种建模方法对比

	本文模型	基于 DBN 的模型	基于 SVM 的模型
训练样本数	500 万	500 万	500 万
测试样本数	29.2 万	29.2 万	29.2 万
测试精度	96.66%	93.10%	83.23%
平均测试时间	45.01s	46.23s	101.53s

从上表可以看出，本文模型在测试精度和平均测试时间来看为三种方法中最佳的，达到了 96.66%，超出了基于 DBN 方法的 93.10%，以及基于 SVM 方法的 83.23%。同时，本文模型的平均耗时也是三者中最少的。

6　结束语

本论文提出了一种基于深度学习的网络异常行为检测方法，其核心是利用历史网络数据，建模并训练网络异常行为检测模型，进而实现快速、高精度的异常行为检测。通过对比实验与分析，验证了本文方法中冗余特征约简，以及整体模型的有效性。下一步工作，将继续关注如何提高模型的时间和精度性能，以及考虑模型的持续演进问题。

参考文献：

[1]唐赞玉，刘宏. 多阶段大规模网络攻击下的网络安全态势评估方法研究[J]. 计算机科学，2018，045（001）：245-248.

[2]陈雷. 网络安全态势评估与预测关键技术研究[D]. 2015.

[3]Sung W T. Multi-sensors data fusion system for wireless sensors networks of factory monitoring via BPN technology[J]. Expert Systems with Applications，2010，37（3）：2124-2131.

[4]裴卫杰，庞天杰. 一种基于动态填充的不完备数据聚类算法[J]. 太原师范学院学报：自然科学版，2018，17（1）：50-55.

[5]Li X，Li X，Zhao Z. Combining deep learning with rough set analysis：A model of cyberspace situational awareness[C]// International Conference on Electronics Information & Emergency Communication. IEEE，2016.

[6]Yao S，Wang J，Xu F，et al. Uncertainty measurement and attribute reduction in incomplete neighborhood rough set[J]. Journal of Computer Applications，2017.

[7]Liu Y，Wang W，Liu Y. Improved incomplete discernibility relation and its extended rough set model[J]. CHINA SCIENCEPAPER，2018，v.13（05）：71-75.

[8]高泽芳，胡娜. 基于深度置信网络的网络安全态势感知与预测[J]. 移动通信，2018，42（11）：41-47.

[9]Shone N，Ngoc T N，Phai V D，et al. A Deep Learning Approach to Network Intrusion Detection[J]. IEEE Transactions on Emerging Topics in Computational Intelligence，2018，2（1）：41-50.

[10]Roy S S，Mallik A，Gulati R，et al. A Deep Learning Based Artificial Neural Network Approach for Intrusion Detection[C] //International Conference on Mathematics and Computing. Springer，Singapore，2017.

[11]Changjian Z，Zhenyu S I，Jinge X，et al. Study on cyberspace situation awareness modeling method based on Deep Learning[J]. Journal of Northeast Agricultural University，2013，44（5）：144-149.

[12]Alom Z，Bontupalli V R，Taha T M. Intrusion Detection Using Deep Belief Network and Extreme Learning Machine[J]. International Journal of Monitoring and Surveillance Technologies Research，2015，3（2）：35-56.

浅析基于深度学习的恶意软件检测

◆黎臻　罗栗

（中国电子科技集团公司第三十研究所　四川　610041）

摘要：互联网软件既为人们生活提供了便利，但也带来了一定的潜在危险，故意设计的恶意软件会给网络安全带来威胁。本文对恶意

软件检测技术进行了简要分析，结合机器学习提出了一种基于深度学习的恶意软件检测方法，结合使用卷积神经网络和长短期记忆网络，以此提高检测率。

关键词：恶意软件；深度学习；特征检测

如今，各种各样的互联网软件为人们生活提供了便利，但也带来了一定的潜在危险。恶意软件是故意设计以危害计算机、智能设备，窃取机密信息，渗透网络或破坏关键的基础设施等为目的的一种网络威胁。恶意软件的类型多种多样，如病毒，蠕虫，特洛伊木马，逻辑炸弹，间谍软件，广告软件，垃圾邮件，弹出窗体等。据卡巴斯基实验室最近的一份报告称[1]，由于恶意软件攻击，大约在两年之内全球的金融机构共盗窃了十亿美元。为了遏止恶意软件造成的巨大损失和损害，恶意软件检测已成为计算机安全的热点研究方向。

1　恶意软件检测的发展现状

近年来，关于恶意软件检测的学术研究数量迅速增加。在早期，基于签名的检测方法被广泛使用。Malhotra 等人[2]提出一种新颖的算法将恶意软件分类为干净、正常恶意软件和多态、变质恶意软件。该方法使用基于签名的模式匹配技术计算任意两个文件之间的相似性得分，然后使用 DBScan 算法对所选特征进行分类。基于签名的检测方法可以快速有效地应对已知的恶意软件，但对于零时差恶意软件的效果则不佳。

随着时间的流逝，研究人员开始使用基于行为、启发式和模型检查的检测技术，以及新技术，例如基于深度学习、云、移动设备和物联网的检测。

Cimitile A 等人[3]提出了一种基于模型检查的方法来推断移动恶意软件的系统树。通过在 droid-Sapiens 工具中实施该方法，证明了移动恶意软件家族来自祖先，并且根据其展示的有效载荷来影响自己的后代。

Azmoodeh A 等人[4]提出了一种应用于物联网基于机器学习的方法，通过监视 Android 设备的功耗来检测勒索软件攻击。具体来说，该方法可监控不同进程的能耗模式，以对来自非恶意应用程序的勒索软件进行分类。实验证明了该方法在准确率，查全率，准确率和 F 度量方面优于 K 最近邻，神经网络，支持向量机和随机森林。

即使每种检测方法都有其自己的优势，但是没有一种检测方法可以检测到所有恶意软件。当恶意软件的复杂性（未知恶意软件，新一代恶意软件，混淆恶意软件）增加时，所有检测方法的检测率都会降低。可以看出，基于签名方法、启发式的方法以及大多数情况下的基于移动设备和物联网的方法要比基于行为、模型检查、云和深度学习的方法表现更差。这是因为后者的这些方法可以更有效地检测未知和混淆的恶意软件。

结合多种恶意软件检测方法可以提供更好的检测机制，例如，将基于行为的行为与基于模型检查的方法相结合，并同时使用深度学习和云技术一定会提供更好的检测机制。此外，使用区块链和大数据等新技术可能会提供更多机会来构建更有效的检测器。

2　深度学习的特点

深度学习是数据挖掘和机器学习的一个新领域，已在各种各样的应用领域取得了成功，例如计算机视觉、音频和自然语言处理。卷积神经网络（Convolutional Neural Network, CNN）是深度学习中最具代表性的算法之一。CNN 的最显著特征是它通过权重共享，局部场和空间中的子样本的思想来减少了大量的计算。它在一组 CNN 神经元之间具有相同的权重，通过卷积运算在局部场上挖掘特征信息。CNN 直接将原始数据的向量表示作为输入，并以端到端的结构前向传播输出分类或回归结果。通过反向传播算法更新 CNN 的神经元权重。CNN 的典型应用是通过多个卷积层和池化层来识别手写数字。每个卷积层输出一组特征图，而每个特征图表示通过一个特定的卷积滤波器提取的高级特征。并且池化层主要使用局部相关原理完成下采样，因此后续的卷积层可以从更全局的角度提取特征，这些大大减少了用于训练深度网络的权重参数和计算的数量。

多层深度学习架构在特征学习方面具有卓越的能力。更重要的是，深度学习架构通过分层预训练克服了学习上的困难，即从最低层到最高层对多层特征检测器进行预训练以构建最终的分类模型。这启发我们设计基于深度学习的恶意软件检测架构。

3　基于深度学习的恶意软件检测

近年来，不少深度学习的研究工作已经投入到恶意软件检测上。Zhu D 等人[5]提出了 DeepFlow，这是一种基于深度学习的新颖方法，可直接从 Android 应用程序中的数据流中识别恶意软件。在数千种良性软件和恶意软件上测试 DeepFlow，结果表明 DeepFlow 可以达到 95.05％的高检测 F1 分数，优于传统的基于机器学习的方法，这表明了深度学习技术在恶意软件检测中的优势。

我们提出一种基于深度学习的恶意软件检测方法，该方法使用 CNN 和长短期记忆网络（Long Short-Term Memory，LSTM）。对于恶意软件检测，实际上执行二进制分类任务。我们的方法接收原始文件数据作为输入，并输出表明该恶意软件可能性的识别概率。具体而言，检测过程可以分为两个阶段。第一阶段是预处理恶意软件样本数据，它采用可执行文件的二进制形式，从中生成灰度图像，并使用反编译工具提取操作码序列和元数据特征。因此，此阶段将生成适当的数据格式，作为后续 CNN 和 LSTM 网络的输入。第二阶段应用检测网络的核心过程，该过程分别采用 CNN 和 LSTM 网络，从灰度图像和操作码序列中学习。为了优化检测性能，我们使用堆栈集成来集成两个网络的输出和元数据特征。实际上从原始数据中学习了三种不同的特征集。首先，检测网络通过 CNN 从灰度图像中学习恶意文件结构特征，然后通过 LSTM 从操作码序列中学习恶意代码模式特征。这两个特征集整合并完善局部模式信息。同时，添加了一些简单的元数据特征来描述全局信息。最后使用卷积块对整合后的三种特征进行卷积操作，学习并压缩特征，最终输出的预测结果。

4　结语

深度学习能够处理非结构化的数据，具有强大的表征学习能力，通过深层的网络学习数据的特征，同时能大大降低特征量。并且许多现有的应用实例也表明了深度学习在分类任务上具有高准确率。这让基于深度学习的恶意软件检测存在独有的优势，启发我们设计基于深度学习的恶意软件检测架构。

参考文献：

[1]“KasperskySecurity Bulletin 2016. Overall statistics for 2016,”https://securelist.com/kaspersky-security-bulletin-2016-execuive-summary/76858/.

[2]Aashima Malhotra, Karan Bajaj. A hybrid pattern based text mining approach for malware detection using DBScan[J]. CSI Transactions on ICT, 2016.

[3]Cimitile A, Martinelli F , Mercaldo F, et al. Model Checking for Mobile Android Malware Evolution[C]// International Fme Workshop on Formal Methods in Software Engineering. ACM, 2017.

[4]Azmoodeh A, Dehghantanha A, Conti M, et al. Detecting crypto-ransomware in IoT networks based on energy consumption footprint[J]. Journal of Ambient Intelligence and Humanized Computing, 2017.

[5]Zhu D, Jin H, Yang Y, et al. DeepFlow: Deep learning-based malware detection by mining Android application for abnormal usage of sensitive data[C]// 2017 IEEE Symposium on Computers and Communications (ISCC). IEEE, 2017.

Scrum 在民航 IT 研发应用实践

◆李圣霞　刘盼　夏侯康　罗军　程宇

（广东机场白云信息科技有限公司　广东　510470）

摘要：随着民航领域的逐步发展，对信息化的要求越来越高，同时民航领域业务需求也具备一定特殊性，对研发质量和效率都带来挑战，而敏捷 Scrum 的应用，通过多次迭代，小批量发布的方式，降低了每一次上线的发布风险，提升了产品质量，从而提高了客户满意度。

关键词：敏捷；Scrum；民航

1　研究背景

随着社会的进步和管理水平的日益提高，软件开发与各个行业和领域都密不可分，民航单位作为运输行业的一大重点，日常运营管理越来越精细化，对信息化的程度要求越来越高。第一，用户迫切需要信息化的管理软件，来帮助自己梳理和记录日常业务流程，提高自身管理水平；第二，用户对软件不熟悉，对系统的需求是模糊的、不明确的，只有在软件研发的过程中，才能逐渐明确下一步的需求计划；第三，因为民航业务的特殊性，对安全和质量的要求很高，软件系统的质量问题，对日常业务影响较大，且一般影响范围较广。

在这种情况下，传统的瀑布模型已经无法适应不断变动的需求，因为需求变更在瀑布开发模型中，尤其是开发后期带来的成本是高昂的，不确定的需求也会导致开发过程中工期、成本、质量的多重风险，并最终导致项目的失败。

2　敏捷及 Scrum 简介

敏捷是一种通过创造变化和响应变化在不确定和混乱的环境中取得成功的能力。敏捷软件开发是基于敏捷宣言定义的价值观和原则的一系列方法和实践的总称。

敏捷开发遵循十二条基本原则，如下：

第一，我们最重要的目标，是通过及早和持续不断地交付有价值的软件使客户满意。

第二，欣然面对需求变化，即使在开发后期也一样。为了客户的竞争优势，敏捷过程掌控变化。

第三，经常地交付可工作的软件，相隔几个星期或一两个月，倾向于采取较短的周期。

第四，业务人员和开发人员必须相互合作，项目中的每一天都不例外。

第五，激发个体的斗志，以他们为核心搭建项目。提供所需的环境和支援，辅以信任，从而达成目标。

第六，不论团队内外，传递信息效果最好效率也最高的方式是面对面的交谈。

第七，可工作的软件是进度的首要度量标准。

第八，敏捷过程倡导可持续开发。责任人、开发人员和用户要能够共同维持其步调稳定延续。

第九，坚持不懈地追求技术卓越和良好设计，敏捷能力由此增强。

第十，以简洁为本，它是极力减少不必要工作量的艺术。

第十一，最好的架构、需求和设计出自自组织团队。

第十二，团队定期地反思如何能提高成效，并依此调整自身的行为表现。

Scrum 是敏捷开发的一种最基本框架，也是最流行的一种框架，尤其是对于初次进行敏捷开发的企业尤其适用。Scrum 原始含义是指英式橄榄球次要犯规时在犯规地点对阵争球。传统的“接力式”的开发模式已经不能满足快速灵活的市场需求，而整体或“橄榄球式”的方法——团队作为一个整体前进，在团队的内部传球并保持前进，这也许可以更好地满足当前激烈的市场竞争。

Scrum 框架包括 3 个角色（Product Owner、Scrum Master、开发团队）、3 个工件（Product Backlog、Sprint　Backlog、产品增量）、5 个事件（Sprint、Sprint 计划会议、“每日站会”、Sprint 评审会议、Sprint 回顾会议）、5 个价值（承诺、专注、开放、尊重、勇气）。

3　Scrum 应用实践中的问题和解决方案

在民航系统管理软件的开发过程中，采用传统瀑布模型，因为客户需求的不确定性、各单位部门之间沟通协调的复杂性、对接其他系统数据的困难性等各种因素，往往前期需求调研阶段特别长，因为工期紧张容易造成延期问题，并带来质量的隐患，容易导致客户满意度的降低。