基于神经网络和 NLP 的软件需求安全分析研究

孙宝华[1,3] 胡 楠[3] 李东洋[2,3]
(吉林大学 长春 130012)[1] (东北大学 沈阳 110819)[2]
(国网辽宁省电力有限公司 沈阳 110004)[3]

摘 要 为了对软件需求的不完备性和歧义性程度进行识别,搭建软件需求和标准规范之间的桥梁,提出一种基于自然语言处理(Natural Language Processing,NLP)和神经网络的分析评价模型。首先,从国际标准化组织(ISO)、开源 Web 应用程序安全计划(OWASP)和 PCI 目录等标准出发,识别出多个安全性规范特征,找到文本蕴涵关系;然后,利用蕴涵结果以及文本注释来训练神经网络模型,以预测文档中的某个语句是否存在于安全标准中。所提模型对每个蕴涵配置的预测性能进行了评价,结果表明:蕴涵配置 9 的平均 F-得分最高,为最佳完备性预测器。且在最优和最差配置下,所提模型的性能均优于常用的空模型。

关键词 软件需求,自然语言处理,神经网络模型,安全性,空模型,蕴涵关系
中图法分类号 TP391 **文献标识码** A

Analysis Research of Software Requirement Safety Based on Neural Network and NLP

SUN Bao-hua[1,3] HU Nan[3] LI Dong-yang[2,3]
(Jilin University,Changchun 130012,China)[1] (Northeastern University,Shenyang 110819,China)[2]
(State Grid Liaoning Electric Power Co.,Ltd.,Shenyang 110004,China)[3]

Abstract To identify the incompleteness and ambiguity of software requirements and build a bridge between software requirements and standard specifications,this paper proposed a model of analysis and evaluation based on the Natural Language Processing (NLP) and neural network. Firstly, from ISO, the open-source Web application security plan (OWASP) and the PCI directory,multiple security specification features are identified,and text implication relationships are found. Then, the implication results and text annotations are used to train the neural network model to predict whether a certain statement in the document is available. The proposed model evaluates the performance of each implication configuration. The results show that the average F- score of the implicative configuration 9 is the highest,which is the best completeness predictor. Moreover,the performance of the proposed model is better than that of the null model under optimal and worst allocation.

Keywords Software requirements,Natural language processing,Neural networks model,Security,Null model,Implication relationships

1 引言

在软件需求分析[1]阶段,考虑软件的安全性特征具有重要意义,优点如下:减少缺陷、较早地发现错误、减少改动和标准化的相关成本,以及向利益相关方提供关于安全漏洞和安全要求的技术知识[2]。由于软件需求的理想特性包括准确性、可验证性和无歧义性[3],因此有必要开发出对安全需求进行分析和评价的技术。

很多应用使用了需求工程阶段的自动化,该过程包括开发项目的需求定义、规范、架构、设计以及软件需求综合。文献[4]提出了一种基于 UML 安全特性验证的方法,该方法在 UML 需求模型类图和顺序图的基础上,为核心类的安全特性自定义构造型、标记和约束。文献[5]开发了一个分析需求文档的词汇和语法工具,即需求规范的质量分析器。该工具对可能在软件后期开发中造成歧义的潜在语言缺陷进行检测,但其功能局限于可读性分析。文献[6-7]对可用性、可扩展性和安全性等需求进行了半自动化分类。文献[8]选择理想的安全规格,提供了安全性需求的分类,该方法旨在通过向用户建议不同的安全性需求,并基于用户选择生成软件需求,从而在安全和可用性间维持平衡。其虽然提升了用户便利,但该设计限制了可用特征,使得用户无法自由地选择列表外的特征。

以上模型有的使用了 NLP,有的则没有;有的仅对软件需求文档的安全特征进行分类,但不做进一步处理;有的对需求文档的完备性和歧义性进行识别,但未通过与标准的比较来分析需求文档的完备性和歧义性。由于软件需求具有自由连续的固有特性,基于自然语言处理[9](NLP)和神经网络模型,本文提出了一个半自动化方法,以评价软件需求文档在某些安全特性方面的完备性和歧义性。本文的主要工作有:1)为软件需求的语义分析提供了一个通用架构;2)根据给定的安全需求文档,完成了对语义分类的解释。

孙宝华(1973—),男,硕士生,高级工程师,主要研究方向为电力企业信息化、数据库应用、软件安全;**胡 楠**(1982—),男,博士生,高级工程师,主要研究方向为人工智能、数据分析;**李东洋**(1990—),男,硕士生,主要研究方向为电力企业信息化、数据库应用。

2　提出的分析方法

本文在分析软件需求的完备性和歧义性时，利用了自然语言学习和神经网络等技术，具体流程如图 1 所示。

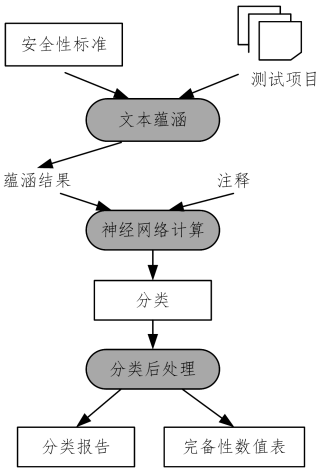

图 1　提出的端到端流程

首先，本文从国际标准化组织(ISO)、开源 Web 应用程序安全计划(OWASP)和 PCI 目录等标准出发，识别出与身份验证、授权、访问控制、数据完整性和加密等特征相关的安全规范。然后，从软件项目相关的 15 个软件需求文档集合[10]中提取实验数据。由于采集的标准文档和从利益相关方获得的需求文档都是自由连续文本，因此本文使用自然语言处理(NLP)技术对文本进行解析。接着，使用基于机器学习的蕴涵算法对每个测试文档与标准进行比较，从而识别出每个标准语句与每个测试文档语句之间的关系(即蕴涵或非蕴涵)。其中，“文本蕴涵”定义为两个语句间的单向关系[11]。给定两个文本片段(T)和假设(H)，如果从 T 可以推断出 H 的意义(即 T 蕴涵 H)，则两者存在蕴涵性。这里利用开源的 EOP 并对蕴涵性进行修改，以确定两个语句之间是否存在语义相似性。

所提 EOP 平台是一个用于文本推理的通用架构，如图 2 所示，其包括两个单独模块，即语言分析管道(LAP)和蕴涵核心(EC)，该平台还包括词汇和句法资源。

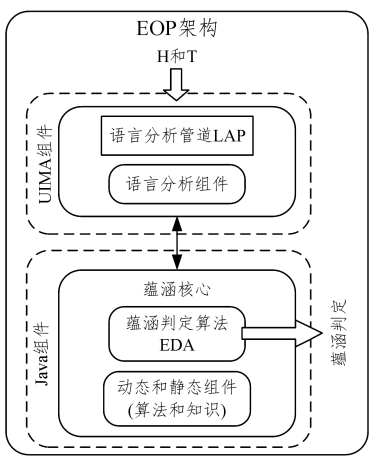

图 2　EOP 框架

EOP 的输入是成对的文本和假设，其输出为蕴涵判定和置信度得分。其中，主要组件 LAP 和蕴涵判定算法(EDA)共同组成了 NLP 模块。3 种主要的蕴涵算法分别为树编辑距离算法[12]、基于粒子群优化(PSO)的编辑距离算法[13]以及最大熵分类蕴涵判定算法[14]。

在 EOP 模块完成处理后，将结果数据用于神经网络模块，在该模块进行构建、训练和评估，并将其用于特定操作符的预测，使用的 3 个操作符为：完备、歧义和缺失。这些操作符反映了在文本处理和神经网络阶段推导出的标准语句与测试文档语句之间的语义关系。

3　方法的实现

3.1　语义蕴涵的实现

为评价标准文档(T)中的语句与测试文档(H)中的语句间的蕴涵关系，本文在 EOP API 上开发了一个 Java 应用，即安全性需求分析(SRA)。SRA 在 EOP 处理之前对输入进行预处理：以串行或并行方式运行 EOP；将结果处理为格式化的报告文件。具体步骤如下。

3.1.1　预处理与 EOP

预处理的主要任务是将标准文档和测试文档转换为单独事务，其中每个事务包括两个语句(一个语句来自标准文档，另一个来自测试文档)和蕴涵配置(建立在 EOP 内置配置上的 SRA 中 9 个预定义包之一)，利用蕴涵配置对这两个语句的蕴涵关系进行评价。对标准文档和测试文档中的每条语句进行比较评价。合并后的标准文档(例如 ISO 和 OWASP)包括 239 条语句，合并后的测试文档(即从 13 个不同的客户需求文档中提取出的安全性文档)包括 18 条语句。当比较这些文档的蕴涵关系时，其中存在 19 359(239 * 81)条特有的语句对，当使用预定义蕴涵配置对每个语句对进行评价时，共得到 173 231(239 * 81 * 9)个事务。

准备好一个事务后，可以将其以串行和并行的方式与其他事务一起处理，即 EOP 处理。在实践中推荐采用并行处理方式。

3.1.2　后续处理

本文使用的 9 个蕴涵配置如表 1 所列。

表 1　蕴涵配置

	LAP	EDA	组件
配置 1	开放 NLP 标注器	最大 ENT 分类	
配置 2	Malt 解析器	最大 ENT 分类	词袋得分；Verb Ocean 词典资源 Verb Ocean 词汇资源
配置 3	Malt 解析器	最大 ENT 分类	词袋得分；Wordnet 词汇资源
配置 4	树标注器	最大 ENT 分类	词袋得分；Wordnet 词汇资源；Verb Ocean 词汇资源；
配置 5	Malt 解析器	最大 ENT 分类	词袋得分；Wordnet 词汇资源；Verb Ocean 词汇资源；与 6,7 不同的模型文件
配置 6	Malt 解析器	最大 ENT 分类	词袋得分；Wordnet 词汇资源；Verb Ocean 词汇资源；与 5,7 不同的模型文件
配置 7	Malt 解析器	最大 ENT 分类	词袋得分；Wordnet 词汇资源；Verb Ocean 词汇资源；与 5,6 不同的模型文件
配置 8	开放 NLP 标注器	编辑距离	固定权重标记编辑距离；基于阈值的模型
配置 9	开放 NLP 标注器	编辑距离 PSO	固定权重标记编辑距离

收集每个事务的蕴涵判定和置信结果，以及关于该数据的其他数据，例如涉及的语句、所使用的蕴涵配置、处理类型以及比较的持续时间。将所有收集到的数据格式化为 CSV 报告。每个报告包含了 9 个蕴涵配置、1 个标准语句以及与之对应的所有其他测试文档语句(总计 81 个)的事务，即最后每个报告中总计得到 729(1 * 81 * 9)个事务。大部分蕴含配置根据不同的特征集(例如 EDA、LAP 或组件)来区分，然而所有的 9 个配置都包含了各自的训练模型，该模型创建于学习阶段并被应用到测试阶段。

3.1.3 注释

本文添加了一个额外的空字段，以对 81 个蕴涵事务(语句对)的每个事务进行手工注释，其数值为 3 个操作符之一："c"表示完备，"a"表示歧义，"n"表示无。总计注释了 19 个报告，代表着 1539(81 * 19)个蕴涵事务对。在神经网络模型的训练阶段使用这些注释来创建一个操作符分类器，以预测语义，其蕴涵结果是"完备""歧义"或"无"。

3.2 神经网络的实现

即使一个蕴涵判定在语句中具有相同语义方面的判定并不准确，但数据中可能存在一些可被用于识别，并正确预测语句间的语义关系模式。为预测一个标准语句与一个测试文档语句在语义上是否匹配，本文通过 Python 内的 Keras API，使用神经网络(NN)建立了一个操作符分类器，如图 3 所示。从蕴涵报告的特定数据上对分类器进行训练，以预测标准语句和测试语句间的蕴涵事务结果是否表明该匹配是"完备""歧义"还是"无"。为实现这一目的：1)需要对蕴涵报告中的数据进行选择性的提取、格式化和合并；2)需要针对每个层精心设计 NN 模型；3)必须在提取数据上对模型进行训练。

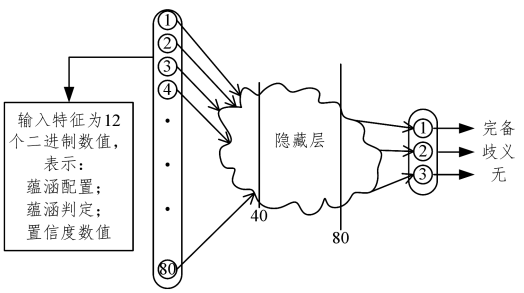

图 3 本文所用的神经网络模型

3.2.1 模型创建

在 NN 模块中，逐层形成一个顺序模型，每层描述神经元的隐藏单元数量、随机权值初始化和激活函数。第一层包含输入特征(向量维度)的数量的额外描述符，由包含蕴涵配置(9 个数值)、判定(2 个数值)和置信度(1 个数值)的数据文件所表示，即每个蕴涵事务总计为 12 个输入特征。一层的输出为下一层的输入，直至到达最后一层，并在该层进行分类。

如图 3 所示，输入层包括 80 个神经元，两个隐藏层各包含 40 和 80 个神经元，最后一层即输出层，包含 3 个神经元。权值初始化选择了基于高斯初始化程序的 he_normal 函数。输入层和隐藏层均使用修正线性单元(relu)激活函数，输出层使用了 softmax 函数，可较好地适用于分类问题。

通过初始化、激活和优化器的组合对模型参数选择进行迭代测试，其中优化器用于模型编译过程，初始化、激活和优化器的可用选项如表 2 所列。通过 448 个测试对 1539 个语句对进行评价，并将预测分类(表示"完备""歧义"或"无")与实际分类(注释数据)进行比较。结果表明：通过高斯和均值分配数据中初始状态的最优值，最优层初始化器与正常和均匀权值相关，而最后一层的激活函数取决于求解问题的类型，此处 Softmax 是最佳选择，因为其能够确保输出层得到合理的格式化，支持在最优操作符匹配(即完备)和较差匹配之间的明确区分以进行模型预测分类。

表 2 层激活函数、初始化器和优化器

2D 层的初始化器	激活函数	优化器
uniform	softmax	SGD
lecun_uniform	softplus	RMSprop
normal	softsign	Adagrad
zero	relu	Adadelta
glorot_normal	tanh	Adam
glorot_uniform	sigmoid	Adamax
he_normal	hard_sigmoid	Nadam
he_uniform	linear	

3.2.2 模型训练

本文在格式化的数据上对形成的模型进行训练，以建立一个能够基于两个语句对应的蕴涵事务(蕴涵配置、判定和置信度)。首先必须使用损失函数、优化器和要观察的指标对模型进行编译，以配置训练过程。

由于存在 3 种可供预测的分类，因此这是一个多分类问题，其损失函数是"分类交叉熵"。模型选择 Adamax 优化器，训练过程中使用的参数如下：数据、目标、批大小、代数、验证集分割以及可选的回调函数。将创建的数据和目标文件读入内存中，将其作为模型拟合的前两个参数。接下来的两个参数描述每次要处理数据的样本数量(批大小)，以及对所有数据进行代数迭代的次数。本文发现批大小为 300，代数为 50000 时能够得到最优结果，这说明在模型训练过程中，数据处理每次处理 300 个条目(蕴涵配置、判定和置信度)，直到处理至数据末端，这个过程重复 50000 次。验证集分割设为0.2(即：留出 80%的训练数据，另外 20%作为验证数据)。回调函数允许在训练过程中的特定阶段进行自定义处理，例如，在每个阶段获得统计信息和其他状态信息。另外，本文在给定阶段保存最优的模型权重，以保证验证的准确性。

3.2.3 后续处理

在完成模型训练后，可以基于蕴涵结果(蕴涵配置、判定和置信度)的输入数据，使用该模型预测描述两个语句间的语义关系操作符(即"完备""歧义"或"无")。另外，为了确定给定标准语句在测试文档内是否得到满足，需要将一组规则应用到给定测试语句相关的预测操作符上。由此，将得到一个分类结果，表明该标准语句在整个测试文档内是否得到满足。若要满足该标准语句分入"完备"类(即测试文档中找到了该标准语句)，那么至少要在标准语句和一个测试文档语句之间存在一个"完备"预测。若要将该标准语句分入"歧义"类，则在标准语句和每个测试文档语句之间必须不存在任何一个"完备"预测，而且必须至少存在一个"歧义"预测。若要将标

准语句分入“缺失”(或“无)类，则标准语句和每个测试文档之间必须仅存在“无”预测。在针对每个标准语句重复上述过程后，将得到每个标准语句和作为一个整体测试文档之间的预测关系(即操作符“完备”“歧义”或“缺失”)。

4　模型评价与分析

4.1　模型评价

为度量 NN 训练模型的分类预测性能，本文应用了空模型概念，并将其与训练模型进行比较评估。空模型是一种未经训练即进行预测的简单方法。由于其预测结果与输入数据不相关，空模型可被作为较差情况下的性能基准[15]。由于存在 3 种可能的预测分类(0 表示“缺失”、1 表示“歧义”、2 表示“完备”)，本文使用了 3 个不同的空模型，每个空模型代表一种不同的潜在分类。例如，第一个空模型 null(0)表示“缺失”分类，则其预测为：所有标准语句在相应的测试文档中均为“缺失”；第二个空模型 null(1)的预测为：所有标准语句在相应的测试文档中均是“歧义”；而第三个空模型 null(2)则预测所有的标准语句为“完备”，也即测试文档内满足标准。

对于每个项目的测试文档，空模型的平均 F-得分结果如表 3 所列。由表 3 可知，null(0)的最高平均 F-得分为 0.34，其次为 null(1)的 0.17，最低为 null(2)的 0.08。从空模型 F-得分的分布结果可以发现，数据是不平衡的，即：标准语句在测试文档中的 0 或“缺失”较多，其次为“歧义”，而“完备”则最少。由于大部分标准语句在相应的测试文档中被标注为缺失，因此 null(0)在这个不平衡的数据上得到了最佳的预测性能。

表 3　空模型的 F-得分

项目	null(0)	null(1)	null(2)
1	0.42	0.15	0.02
2	0.25	0.11	0.15
3	0.15	0.20	0.15
4	0.25	0.25	0.04
5	0.36	0.07	0.11
6	0.25	0.04	0.25
7	0.42	0.11	0.04
8	0.25	0.11	0.15
10	0.70	0.02	0.02
12	0.00	0.85	0.02
13	0.42	0.11	0.04
14	0.56	0.02	0.07
15	0.42	0.11	0.04
平均	0.34	0.17	0.08
最小	0.00	0.02	0.02
最大	0.70	0.85	0.25

接下来，以性能最优的空模型 null(0)为基准，对训练后的 NN 模型进行评价。表 4 列出了空模型与本文训练模型的 F-得分比较。根据蕴涵配置将性能最差和最优的模型与空模型 null(0)进行比较，并给出了增量。值得一提的是，所有增量均为正数，这表明在每个案例中，训练后的模型的预测结果均优于空模型。其中，最差的 NN 模型平均比空模型的 F-得分高 0.14，最佳的 NN 模型则比空模型的 F-得分高 0.45。蕴涵配置 9 的模型也被纳入了评价，其平均比空模型的 F-得分高出 0.36，进一步证明了本文模型的优秀性能。

表 4　空模型与 NN 训练模型的 F-得分比较

项目	最差模型	null(0)增量	最佳模型	null(0)增量	蕴涵配置 9	null(0)增量
1	0.51	0.09	0.91	0.49	0.91	0.49
2	0.37	0.12	0.73	0.48	0.73	0.48
3	0.21	0.06	0.63	0.48	0.63	0.48
4	0.35	0.10	0.72	0.47	0.49	0.24
5	0.39	0.03	0.79	0.43	0.65	0.29
6	0.37	0.12	0.69	0.44	0.57	0.32
7	0.51	0.09	0.79	0.37	0.64	0.22
8	0.33	0.08	0.68	0.43	0.52	0.27
10	0.74	0.04	0.94	0.24	0.79	0.09
12	0.79	0.79	0.94	0.94	0.79	0.79
13	0.59	0.17	0.89	0.47	0.84	0.42
14	0.64	0.08	0.84	0.28	0.84	0.28
15	0.46	0.04	0.79	0.37	0.70	0.28
平均	0.48	0.14	0.80	0.45	0.70	0.36
最小	0.21	0.03	0.63	0.24	0.49	0.09
最大	0.79	0.79	0.94	0.94	0.91	0.79

4.2　分类报告分析

本文围绕给定软件需求文档在安全方面的完备性分析和表示，得出两类结果。第 1 类结果为分类报告，对于给定项目，每个蕴涵配置选项的 NN 模型使用已定义的操作符(“完备”“歧义”“缺失”)对标准语句是否存在于测试文档进行预测；第 2 类结果为完备性数值表，其通过模型预测来判定给定项目的测试文档为“完备”“歧义”“缺失”的百分比。

分类的两个列表即预测列表和实际列表，表示标准语句在单个测试文档中的完备性，可被用于计算以下指标：准确度、精度、召回率和 F-得分。这些指标代表了在静态标准语句集合上，输入测试文档(13 个项目)和蕴涵配置(配置 1 至配置 9 以及合并配置)的 130 个所有可能组合中的一种组合。

为了确定整体性能最佳的蕴涵配置选项，本文计算了每个蕴涵配置选项在每个项目上的平均 F-得分，蕴涵配置情况如图 4 所示，图中的横坐标表示合并蕴涵配置。结果表明蕴涵配置 9 的平均 F-得分最高(得分为 0.71)，因此配置 9 成为最佳候选。蕴涵配置 9 包括开放 NLP 标注器、编辑距离 PSO 以及固定权重标记编辑距离(组件)。虽然合并蕴涵配置 1～9 的所有数据，但并未得到最高 F-得分。这说明合并的蕴含配置选项会导致神经网络出现混淆，使得预测性能下降。蕴涵配置 9 成为模型预测的最佳候选，是因为在蕴涵文本处理阶段，标准文档和测试文档之间几乎每个语句的比较都导向了非蕴涵性分类。虽然 EOP 中的蕴涵判定常常会偏向“缺失”蕴涵关系，蕴涵判定和置信度的结合使得 NN 能够在模型预测过程中发现并利用更加清晰的描述模式。

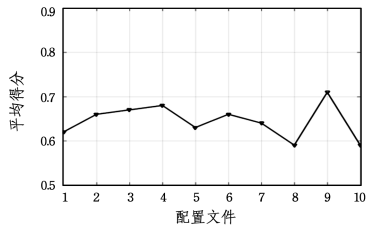

图 4　蕴涵配置与 F 得分关系

从预测分类结果中，可以推导出在测试文档和蕴含配置的每个组合中得到的完备性。对于蕴涵配置 9，其结果样例的完备性数值如表 5 所列。

表 5 蕴涵配置 9 的完备性数值表

（单位：%）

测试文档	完备性	歧义性	缺失性
项目 1	12.1	31.6	47.4
项目 2	31.6	31.6	36.8
项目 3	31.6	47.4	21.1
项目 4	5.3	21.1	73.7
项目 5	21.1	5.3	73.7
项目 6	21.1	21.1	57.9
项目 7	10.5	36.8	52.6
项目 8	26.3	21.1	52.6
项目 10	10.5	10.5	78.9
项目 12	31.6	63.2	5.3
项目 13	15.8	21.1	63.2
项目 14	31.6	5.3	63.2
项目 15	26.3	31.6	42.1

结束语 在部署一个软件开发项目之前，先定义完备且无歧义的安全性需求，可有效减少缺陷，较早发现错误，有助于软件供应商和利益相关方识别安全特性。因此，有必要开发出对安全性需求进行分析和评价的形式化方法。本文通过实证识别出安全性需求的不完备和歧义的程度。提出的方法使用语义蕴涵和神经网络建模，根据安全标准对给定的安全性需求文档进行评价，以确认完备性程度。评价结果验证了本文模型的有效性。

未来，本文将所提模型应用到更大的数据环境，并加入额外的操作符，例如“矛盾”。将标准语句限制为仅适用于给定软件项目的语句，分析额外的非功能性软件需求。

参 考 文 献

[1] 陈志慧. 基于 Event-B 的软件需求形式化建模技术的研究[D]. 成都：电子科技大学，2013.

[2] MALHOTRA R，CHUG A，HAYRAPETIAN A，et al. Analyzing and evaluating security features in software requirements[C]// International Conference on Innovation and Challenges in Cyber Security. 2016：26-30.

[3] 熊伟，王娟丽，蔡铭. 基于 QFD 技术的软件可信性评估研究[J]. 计算机应用研究，2010，27(8)：2991-2994.

[4] 王飞，郭渊博，李波，等. 安全苛求软件需求规格中的安全特性验证方法[J]. 计算机应用，2013，33(7)：2041-2045.

[5] KNAUSS E，OTT D. (Semi-) automatic Categorization of Natural Language Requirements[C]// International Working Conference on Requirements Engineering：Foundation for Software Quality. Springer International Publishing，2014：39-54.

[6] 白川，张璇，王旭，等. 可信软件非功能需求可满足性经济学方法分析[J]. 计算机工程与应用，2017，53(22)：249-257.

[7] 张璇，李彤，王旭，等. 可信软件非功能需求形式化表示与可满足分析[J]. 软件学报，2015，26(10)：2545-2566.

[8] TAKAHASHI T，KANNISTO J，HARJU J，et al. Expressing Security Requirements：Usability of Taxonomy-Based Requirement Identification Scheme[C]// IEEE World Congress on Services. IEEE Computer Society，2014：121-128.

[9] 徐戈，王厚峰. 自然语言处理中主题模型的发展[J]. 计算机学报，2011，34(8)：1423-1436.

[10] RANTOS K，MARKANTONAKIS K. Analysis of Potential Vulnerabilities in Payment Terminals[M]// Secure Smart Embedded Devices，Platforms and Applications. Springer New York，2014：311-333.

[11] 倪盛俭. 汉语文本蕴涵识别研究[D]. 武汉：武汉大学，2013.

[12] 李睿，曾俊瑀，周四望. 基于局部标签树匹配的改进网页聚类算法[J]. 计算机应用，2010，30(3)：818-820.

[13] 周冬梅. 基于演化算法的智能学习与优化方法的研究[D]. 无锡：江南大学，2015.

[14] 伦向敏，侯一民. 运用迭代最大熵算法选取最佳图像分割阈值[J]. 计算机工程与设计，2015，40(5)：1265-1268.

[15] GOLIA S，SIMONETTO A. Treating ordinal data：a comparison between rating scale and structural equation models[J]. Quality & Quantity，2015，49(3)：903-915.

（上接第 335 页）

[4] IEEE 802. 15. 4-2011. Low-Rate Wireless Personal Area Networks (LRWPANs)，Standard for Information Technology Standard. Rapid[EB/OL]. https：//ieeexplore. ieee. org/document/6581828.

[5] GUGLIELMO D D，SEGHETTIA，ANASTASI G，et al. A performance analysis of the network formation process in IEEE 802. 15. 4e TSCH wireless sensor/actuator networks[C]// Computers and Communication. IEEE，2014：3385-3391.

[6] VOGLI E，RIBEZZO G，GRIECO L A，et al. Fast join and synchronization schema in the IEEE 802. 15. 4e MAC[C]// Wireless Communications and NETWORKING Conference Workshops. IEEE，2015：85-90.

[7] DUY T P，DINH T，KIM Y. A Rapid Joining Scheme based on Fuzzy Logic for Highly Dynamic IEEE[J]. International Journal of Distributed Sensor Networks，2016，12(8)：1482-1493.

[8] OJO M，GIORDANO S，PORTALURI G，et al. An energy efficient centralized scheduling scheme in TSCH networks[C]// IEEE International Conference on Communications Workshops. IEEE，2017：570-575.

[9] KIM J Y，CHUNG S H，HA Y V. A fast joining scheme based on channel quality for IEEE802. 15. 4e TSCH in severe interference environment[C]// Ninth International Conference on Ubiquitous and Future Networks. IEEE，2017：427-432.

[10] PENG D，ROUSSOS G. Adaptive time slotted channel hopping for wireless sensor networks[C]// Computer Science and Electronic Engineering Conference. IEEE，2012：29-34.

[11] PALATTELA M R，ACCETTURA N，VILAJOSANA X，et al. Standardizedprotocol stack for the Internet of (important) thing[J]. IEEE CommunSurv Tutor，2013，15：1389-1406.

[12] DUY T P，KIM Y H. An efficient joining scheme in IEEE 802. 15. 4e[C]// International Conference on Information and Communication Technology Convergence. IEEE，2015：226-229.

[13] Berkeley'sopenwsn[EB/OL]. http：//openwsn. berkeley. edu