

中 国 矿 业 大 学

信息安全管理

与工程报告

报告题目： **基于动态威胁值与故障树的信息安全**

**风险评估模型研究**

组 长： **刘佳静**

负责专题： **构建故障树、计算动态威胁值和风险矩阵**

组 员： **骆信智**

负责专题： **英文论文研究、机器学习预测风险值**

2019年 4 月 徐州

《信息安全管理与工程》课程评分表

开课学期：2018-2019(二) 姓名： 刘佳静 学号： 08163340 专业班级： 信安16-3班

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 毕业要求 | 课程教学目标 | 考查方式与考查点 | 占比 | 得分 |
| 1 | 5.1 | 目标1：能够检索信息安全的最新进展资料。了解信息安全管理与工程的总体知识结构；了解现代信息安全管理与工程存在的主要安全问题；.熟悉现代信息安全管理与工程的基本原理和基本方法； | 期末考试(考查-报告)；  能够通过文献索引、网络数据库检索等途径全面、准确地检索并引用最近3年的参考资料。考查点包括参考文献的相关性、完整性、及时性、参考文献引用是否符合规范。撰写报告，能够清晰论述所搭建安全评估模型 | 40% |  |
| 2 | 10.2 | 目标2：开展简单的信息安全管理体系的设计，初步掌握利用风险评估基本模型来分析评估现代计算机网络安全的基本能力，初步掌握对信息系统实例进行安全工程分析和实施步骤规划的能力。 | 期末考试（考查—系统；  结合具体应用背景，完成的评估系统，功能是否实现，完成是否合理。考查点包括专题的选题是否新颖、总结分析是否得当、指出的优缺点是否准确、能否提出问题并给出自己的解决思路，功能模块是否完整。 | 60% |  |
| 总分 | | | | 100% |  |

评阅人：

2019年 4 月 22 日

《信息安全管理与工程》课程评分表

开课学期：2018-2019(二) 姓名： 骆信智 学号： 08163337 专业班级：信安16-3班

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 毕业要求 | 课程教学目标 | 考查方式与考查点 | 占比 | 得分 |
| 1 | 5.1 | 目标1：能够检索信息安全的最新进展资料。了解信息安全管理与工程的总体知识结构；了解现代信息安全管理与工程存在的主要安全问题；.熟悉现代信息安全管理与工程的基本原理和基本方法； | 期末考试(考查-报告)；  能够通过文献索引、网络数据库检索等途径全面、准确地检索并引用最近3年的参考资料。考查点包括参考文献的相关性、完整性、及时性、参考文献引用是否符合规范。撰写报告，能够清晰论述所搭建安全评估模型 | 40% |  |
| 2 | 10.2 | 目标2：开展简单的信息安全管理体系的设计，初步掌握利用风险评估基本模型来分析评估现代计算机网络安全的基本能力，初步掌握对信息系统实例进行安全工程分析和实施步骤规划的能力。 | 期末考试（考查—系统；  结合具体应用背景，完成的评估系统，功能是否实现，完成是否合理。考查点包括专题的选题是否新颖、总结分析是否得当、指出的优缺点是否准确、能否提出问题并给出自己的解决思路，功能模块是否完整。 | 60% |  |
| 总分 | | | | 100% |  |

评阅人：

2019年 4 月 22 日

**摘 要**

随着人们对信息安全风险评估的深入研究，研究者研发了众多的信息安全风险评估方法中，但常规方法都有一个缺陷，即过分依赖人的经验判断，风险模型受人为因素影响过大，不利于风险评估的准确性与独立性。

基于这个现状，我们提出了基于动态威胁值和故障树的信息安全风险评估模型，该模型引入了动态威胁值的概念，有效体现了风险因素随时间变化的灵活性和与时间联系的紧密性；采用故障树来研究风险因素的风险值，亦可体现风险因素的关联性，有效避免人为因素对风险评估的干扰。

**论文从大量安全事件集合中整理提取出有效的风险因素，得到系统一段时间内的风险变化情况，利用动态威胁值与时间的紧密联系，增加数据的准确性，利用故障树建立评估逻辑模型，构建风险矩阵，并进行未来的风险预测。**

我们的创新之处在于：

1、利用风险矩阵预测不能直接求解得系统风险值，为风险的预测和评估提供了极大便利；

2、将故障树与动态威胁值相结合，体现了系统中风险因素之间的相互作用，便于管理人员对系统风险的把握，未雨绸缪。

3、计算相关公式时，尽量引入时间参数，有效减少同一风险因素特殊周期内的波动对整个信息安全风险评估的影响，使结果更加全面、客观。

该论文有图19幅，表18个，参考文献8篇。

**关键词**：动态威胁值 故障树 矩阵预测 风险评估

**Abstract**

With the in-depth study of information security risk assessment, researchers have developed a number of information security risk assessment methods, but the conventional methods have a flaw, that is, relying too much on human experience, the risk model is too much influenced by human factors. Not conducive to the accuracy and independence of risk assessment.

Based on this situation, we propose an information security risk assessment model based on dynamic threat values ​​and fault trees. This model introduces the concept of dynamic threat values, which effectively reflects the flexibility of time factors and the close relationship with time. Using the fault tree to study the risk value of risk factors can also reflect the correlation of risk factors and effectively avoid the interference of human factors on risk assessment.

**This paper extracts effective risk factors from a large number of security incidents, obtains the risk changes of the system over a period of time, makes use of the close relationship between dynamic threat value and time, increases the accuracy of data, establishes the evaluation logic model using fault tree, builds the risk matrix, and carries out future risk prediction.**

Our innovations are:

1, using the risk matrix prediction can not directly solve the system risk value, which provides great convenience for risk prediction and evaluation;

2. Combine the fault tree with the dynamic threat value, which reflects the interaction between risk factors in the system, which is convenient for managers to grasp the system risk and plan ahead.

3. When calculating the relevant formula, try to introduce time parameters to effectively reduce the impact of fluctuations in the special period of the same risk factor on the entire information security risk assessment, so that the results are more comprehensive and objective.

The paper has 19 pictures, 18 tables, and 8 references.

**Keywords**: Dynamic threat value Fault tree Matrix prediction Risk assessment

**目录**

[**1 绪论 1**](#_Toc7114882)

[1.1 概述 1](#_Toc7114883)

[1.2研究目标 2](#_Toc7114884)

[1.3 研究原理与方法 2](#_Toc7114885)

[1.4 意义 2](#_Toc7114886)

[**2 整体框架和流程图 3**](#_Toc7114887)

[2.1 整体框架 3](#_Toc7114888)

[2.2 处理数据流程图 4](#_Toc7114889)

[**3 概念算法介绍 4**](#_Toc7114890)

[3.1 故障树 4](#_Toc7114891)

[3.2 动态威胁值 6](#_Toc7114892)

[**4 英语文献介绍 6**](#_Toc7114893)

[4.1 背景 6](#_Toc7114894)

[4.2 故障树流程 7](#_Toc7114895)

[4.3 相关研究结果 7](#_Toc7114896)

[4.4 工作人员无意损坏公司安全案例 8](#_Toc7114897)

[4.5 为什么需要故障树分析 8](#_Toc7114898)

[4.6 最小割集 9](#_Toc7114899)

[4.7例一 10](#_Toc7114900)

[4.8 例二 11](#_Toc7114901)

[4.9 总结 12](#_Toc7114902)

[**5案例介绍 13**](#_Toc7114903)

[5.1 案例背景介绍 13](#_Toc7114904)

[5.2 案例数据的选取和整理 14](#_Toc7114905)

[**6 风险值计算 17**](#_Toc7114906)

[6.1确定故障树的结构 18](#_Toc7114907)

[6.2 计算威胁值 19](#_Toc7114908)

[6.3 匹配脆弱性 24](#_Toc7114909)

[6.4 确定权重 25](#_Toc7114910)

[6.5 风险值计算 25](#_Toc7114911)

[**7 预测实验 26**](#_Toc7114912)

[**8 结论和展望 33**](#_Toc7114913)

[**参考文献 35**](#_Toc7114914)

1 绪论

# 1.1 概述

由于信息技术的飞速发展，关系国计民生的关键信息基础设施的规模越来越大，同时也极大地增加了复杂程度，企业机构越来越重视信息安全风险评估工作，提倡风险评估制度化。

信息安全的基本属性主要表现在以下5个方面：

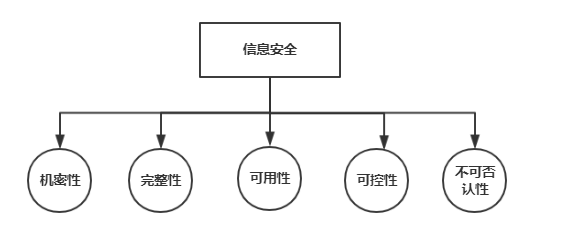


图1-1 信息安全基本属性

(1)保密性（Confidentiality）即保证信息为授权者享用而不泄漏给未经授权者。  
 (2)完整性（Integrity）即保证信息从真实的发信者传送到真实的收信者手中，传送过程中没有被非法用户添加、删除、替换等。  
 (3)可用性（Availability）即保证信息和信息系统随时为授权者提供服务，保证合法用户对信息和资源的使用不会被不合理的拒绝。  
 (4)可控性（Controllability）即出于国家和机构的利益和社会管理的需要，保证管理者能够对信息实施必要的控制管理，以对抗社会犯罪和外敌侵犯。  
 (5)不可否认性（Non-Repudiation）即人们要为自己的信息行为负责，提供保证社会依法管理需要的公证、仲裁信息证据。

风险评估（Risk Assessment） 是指，在风险事件发生之前或之后（但还没有结束），该事件给人们的生活、生命、财产等各个方面造成的影响和损失的可能性进行量化评估的工作。即，风险评估就是量化测评某一事件或事物带来的影响或损失的可能程度。

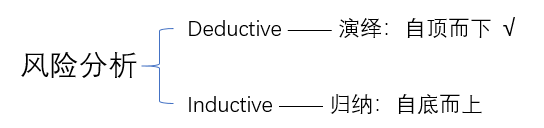


图1-2 风险分析

从信息安全的角度来讲，风险评估是对信息资产（即某事件或事物所具有的信息集）所面临的威胁、存在的弱点、造成的影响，以及三者综合作用所带来风险的可能性的评估。作为风险管理的基础，风险评估是组织确定信息安全需求的一个重要途径，属于组织信息安全管理体系策划的过程。

# 1.2研究目标

我们的研究目标是建立一种基于动态威胁值和故障树的信息安全风险评估模型，根据相关网络安全事件，计算相应的风险值，并对未来风险值进行预测，根据误差不断修正模型，提高模型的准确性。

# 1.3 研究原理与方法

信息安全风险评估模型是针对信息系统自身的资产、脆弱性、威胁以及安全三要素直接的相互关系建立的数学逻辑模型。

下图为信息安全风险评估原理图

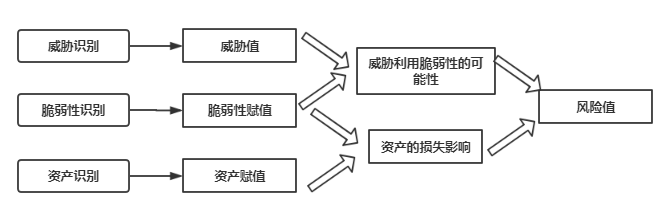


图1-3 信息安全评估原理图

# 1.4 意义

信息安全风险评估是加强信息安全保障体系建设和管理的关键环节。通过开展信息安全风险评估工作，可以发现信息安全存在的主要问题和矛盾，找到解决诸多关键问题的办法。

信息作为一种资源，它的普遍性、共享性、增值性、可处理性和多效用性，使其对于人类具有特别重要的意义。信息安全的实质就是要保护信息系统或信息网络中的信息资源免受各种类型的威胁、干扰和破坏，即保证信息的安全性。根据国际标准化组织的定义，信息安全性的含义主要是指信息的完整性、可用性、保密性和可靠性。信息安全是任何国家、政府、部门、行业都必须十分重视的问题，是一个不容忽视的国家安全战略。

2 整体框架和流程图

# 2.1 整体框架

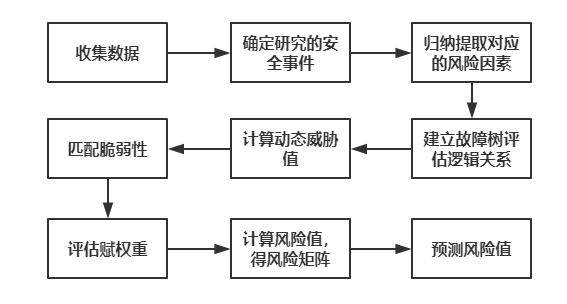


图2-1 整体框架图

# 2.2 处理数据流程图

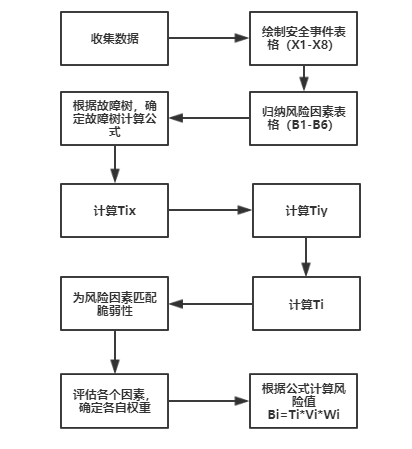
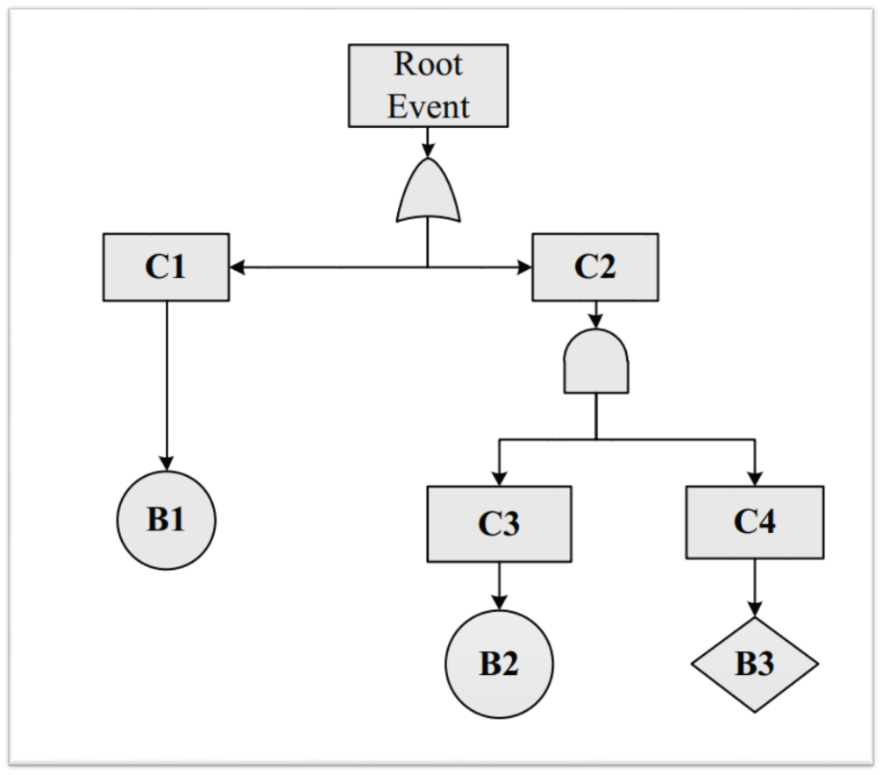


图2-2 数据处理流程图

3 概念算法介绍

# 3.1 故障树

Fault Tree Analysis (FTA)：故障树从一个顶部或根事件开始，该事件是故障的最终结果，表示需要预测可靠性和可用性数据来解决的问题。这种技术定性或定量地确定了系统故障的概率，通过识别顶级事件的根本原因，并提出缓解这些原因的对策。使用FTA处理意外内部安全事件的优点是，它以结构化、图形化的方式显示根事件的所有潜在事件组合。这种结构将分解为基本事件或根本原因，并使解释或沟通对策变得容易。



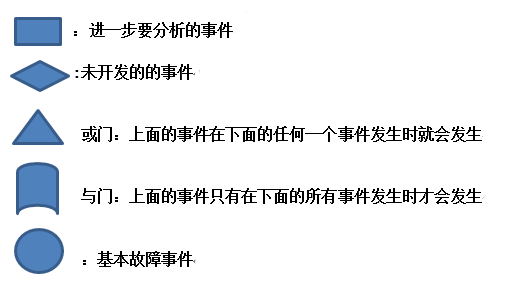


图3-1 故障树示例

故障树计算方法

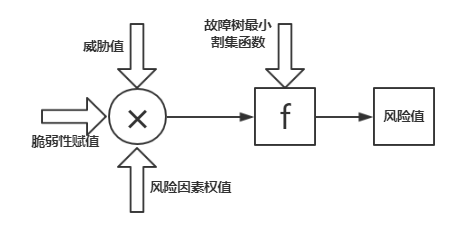


图3-2 故障树计算方法

计算公式：

 （3-1）

其中Ti为动态威胁值，Vi为实际采样获得的脆弱性赋值，Wi为每个风险因素的权重赋值。

# 3.2 动态威胁值

互联网网络安全环境是时刻变化的。由于某些特定原因，同一时期的某一项安全事件是随时可能发生突变的。这就造成外部网络环境的不稳定性。同时，也表明得出一个能够实时地表示网络安全情况的统计数值的重要性。基于此，动态威胁值应运而生。

其他成熟的分析方法的量化值都是由安全专家对每个风险因素进行的经验赋值，其威胁值不但受人为因素影响较大，而且通常是一个恒定值。但同一个威胁常数不能够准确反映系统外部的真实威胁值。

动态威胁评估值是风险值计算中的重要的参考因素。通过一个案例分析，对外部网络环境涉及的安全事件进行概率统计，得出威胁故障树中每一个风险因素的动态威胁值。

动态威胁值计算公式

 （3-2）

Ti为动态威胁值数值

Tix为本次安全威胁出现的次数与前面周期次数和的比值

Tiy为该安全威胁在同一时期与其他安全威胁的比值

4 英语文献介绍

# 4.1 背景

一个组织的信息(系统)可能并不总是由于意外的内部威胁而导致妥协，但仍然具有破坏性。根据人为错误的严重程度，它有可能导致财务不确定性、沟通中断、组织声誉受损或公司不稳定。如果一个严重的错误意外地发生，它可能会使系统容易受到恶意内部人员的攻击，从而预料到任何漏洞。除了本身是一种威胁之外，如果将意外错误作为脆弱性加以利用，还会提高风险水平。为了保护信息系统的机密性、完整性和安全性，组织需要了解员工的行为，从而增加发生意外内部安全事件的风险。评估有助于了解恶意内部人员是否利用意外错误提交和/或隐藏恶意活动。这提供了对导致这些错误的人类行为的洞察。

# 4.2 故障树流程

分析风险的方法属于演绎(即自顶向下)或归纳(即自底向上)方法。使用自顶向下方法，推理工作从一般到更具体。然而，在自底向上的方法中，推理从具体到一般或更广泛的观察。FTA是演绎的，或者基于自顶向下的方法。它是一种图形表示，用于评估从根事件或失败导致根本原因的路径。故障树标识所有可能发生根事件的场景。FTA是一种被广泛应用和接受的分析系统安全性的技术，它基于这样一个概念:系统或子系统的故障可以由较低层系统或子系统的故障引起。与FTA的概念类似，员工无法检测到意外错误可能会导致意外的内部安全事件或发生根事件。因此，基于自顶向下方法的FTA将更有效地分析意外错误。故障树为我们提供了一个倒置的树状结构，其中根事件或故障位于顶部，而根原因或基本事件位于底部。通过计算最小割集(MCS)来分析生成的树。MCS列出威胁代理程序或每个集合中的基本事件，这些事件的存在将增加主事件发生的可能性。MCS解码失败事件，并为解释其封装的威胁代理的对策提供了一种便利。

故障树从一个顶部或根事件开始，该事件是故障的最终结果，表示需要预测可靠性和可用性数据来解决的问题。这种技术定性或定量地确定了系统故障的概率，通过识别顶级事件的根本原因，并提出缓解这些原因的对策。使用FTA处理意外内部安全事件的优点是，它以结构化、图形化的方式显示根事件的所有潜在事件组合。这种结构将分解为基本事件或根本原因，并使解释或沟通对策变得容易。

# 4.3 相关研究结果

许多研究论文都提出了针对恶意内部人的解决方案，一些论文也承认无意内部人在定义“内部威胁”[4]International Data Corporation (IDC) 的研究：员工引发的意外安全事件比恶意的内部攻击更频繁，对信息安全的影响可能更大[5]Cisco 的调查：员工行为是导致数据泄漏的主要原因、四分之一的公司甚至没有适当的访问和使用公司信息的安全策略、揭示了员工故意忽视或绕过安全政策、将个人和公司数据置于危险境地的原因[6]。

FTA技术被成功地应用于包括软件工程在内的许多领域。一项研究提出了利用FTA方法进行人为因素分析，寻找导致软件事故的人为因素[7]。

# 4.4 工作人员无意损坏公司安全案例

CERT的《内部威胁的CERT指南》[3]一书讨论了内部事件，并将其归类为破坏、窃取知识产权和欺诈。我们对这本书的研究确定了利用用户错误创建和/或隐藏欺诈的案例。

在一个处理残疾索赔的部门，一位主管用她自己的账户修改了索赔，并在近两年的时间里每月直接向她的未婚夫支付残疾赔偿金。当她更改位置时，疏忽更新她的访问权限使她能够修改数据并批准更改。这两个职位都使用相同的应用程序，但是在输入、批准和授权支付医疗和伤残索赔方面的角色不同。当她被提升时，她被授权使用新的访问级别，但是管理员没有撤销她以前的访问级别。因此，她最终可以完全访问应用程序，而不需要其他任何人授权系统中的交易(支付)。

我们从三个不同的因素来分析这个案例：管理人的过失；了解员工的可靠性与安全性之间的关系；一连串的错误。在本例中，管理员更新访问权限的疏忽或过失被证明是一个漏洞，恶意内部人员利用该漏洞进行欺诈。它还强调了在员工的可靠性和安全性之间划清界限的重要性。这个漏洞是否激发了员工去制造欺诈?或者这是一个恶意的内部人士预期任何漏洞将被利用来制造欺诈的案件? 结果也可以看作是级联误差，例如;信息安全政策本身的缺失或执行不力，导致管理者的疏忽。管理员的疏忽导致恶意的内部人员拥有过多的访问控制，最终导致欺诈。

教训是：监控内部人员恶意行为的控件将无法检测到意外的内部人员。减轻系统中因用户错误而产生的漏洞;为了理解人为错误所导致的因果关系，有必要进行根本原因分析。

# 4.5 为什么需要故障树分析

围绕用户错误有一个恒定的不确定性和可变性。意外错误的风险很难完全降低，因为它包含安全的一个基本方面，即,人们。组织需要审查他们的安全风险评估模型或方法，以纳入来自意外内部人员的风险。标准与技术研究所(NIST)指出，风险评估的最终结果是确定风险，即，危害发生的程度和可能性。因此，风险评估应该是减轻意外内部威胁的一个组成部分。

CERT基于内部威胁研究的案例，建立了内部威胁的价值模型。我们对价值模型的研究表明，它主要假定“内部人士”是恶意的。它不考虑“内部人士”的意外或疏忽。此外，模型的行为方面、技术攻击方面和防御方面都是基于这一基本假设推导出来的，因此仅限于内部人员的恶意视图。FTA的绩效模型侧重于行政和技术控制，以降低内部威胁的风险。该模型假定技术控制的组织防御机制能够检测恶意活动，这可能也适用于意外错误。

但是，一些旨在积极影响恶意内部人员的人类行为的行政控制假设，在无意的内部人员身上是无效的。

下表一显示了一些行政控制的无效之处意外。

表4-1 行政控制的无效表

|  |  |  |
| --- | --- | --- |
| 政策 | 描述 | 对无意的内部人员无效 |
| 员工干预 | 积极的干预措施，如员工协助或咨询，试图直接降低不满，减少内部人员的不当行为或技术行为。 | 如果没有不适当的行为或不满，那么就没有员工干预。 |
| 制裁 | 旨在激励内部人员减少其不适当的行为或技术行为以避免额外制裁的惩罚性措施。 | 如果对无意的内部人员实施制裁，可能会降低员工的士气并导致生产力下降。 |
| 训练 | 目前仅限于员工对计算机和网络系统的适当使用以及滥用后果的教育。 | 如果没有从无意错误或社会工程方面接受培训，仅在某种程度上有效，例如员工可能误以为他/她的行为是正确的，但实际上，他们可能是无意的错误。 |

应对意外内部威胁的措施应该是主动的，而不是被动的。FTA可以是识别、捕获、评估和管理意外威胁影响的一种方法。针对恶意内部人员的技术和行政控制可能被证明是被动的，而不是针对意外内部人员的主动控制。来自过去事件的信息可以用来识别当前环境中存在的漏洞。这些类比可以用来创建一个故障树来预测已知或未知的人类威胁，从而导致意外的内部事件。由树开发的结构化因果关系探索了原因和事件可能相互作用的路径。这种探索通过树中的逻辑关系预测风险代理的行为。

# 4.6 最小割集

MCS是FTA的重要组成部分，具有重要的结构意义。顾名思义，它只包含发生事件所需的最小故障。故障树可以用MCS进行定性评价。割集的临界性与集中基本事件的数量成反比。例如，如果割集很长，那么它就不那么脆弱，反之亦然。同样，包含一个事件的集合表示单点故障，并且具有很高的脆弱性。

组织只能被保护到风险可接受的程度，这就创建了一个安全性和生产力之间的平衡。国家研究所的标准与技术(NIST) SP 800-30[8]将风险定义为“特定威胁源行使特定潜在脆弱性的可能性的函数，以及该不利事件对组织的影响的结果”。换句话说，当威胁与脆弱性相交叉时，风险就是。如果我们将基本事件视为威胁代理，将MCS视为脆弱性，那么MCS在环境中的存在就表明了风险的存在。

# 4.7例一

“2009年9月1日，我们的一名员工不小心将私人信息附加到一个错误的电子邮件地址，包括姓名、电话号码、相关保险单号码、出生日期、社会关系和安全号码。”

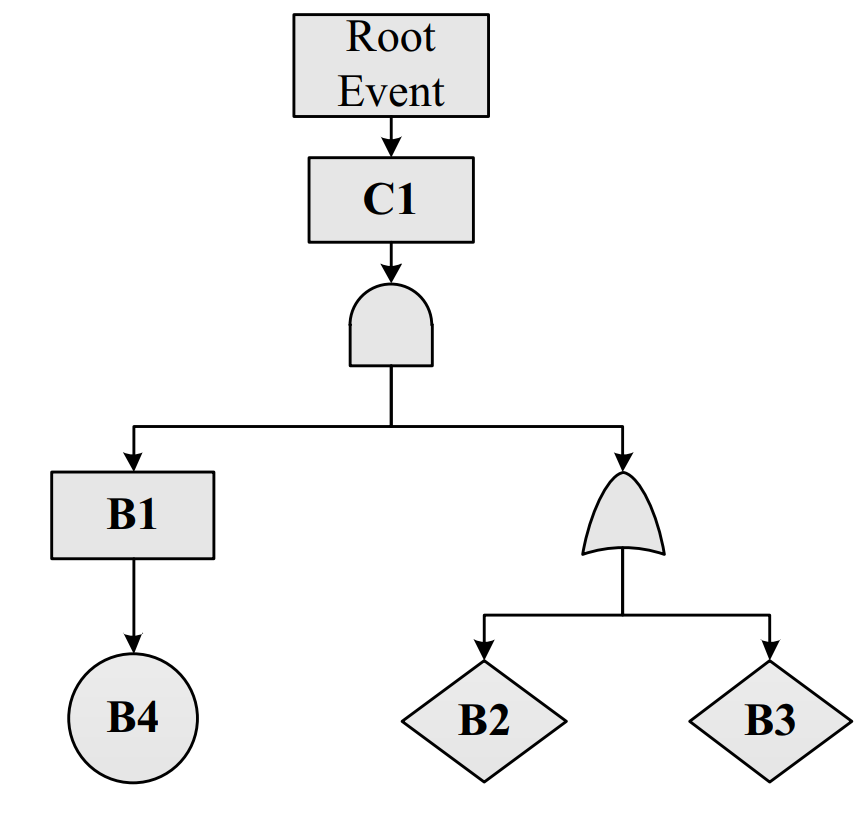


图4-1 例子一故障树分析图

根事件:出站电子邮件中的意外数据泄漏

C1:电子邮件收件人输入错误

B1:未能验证电子邮件收件人

B2:缺乏监控出站邮件的政策和程序

B3:缺乏监控出站邮件的技术控制

B4:缺乏尽职调查

 （4-1）

 （4-2）

 （4-3）

 （4-4）

解决措施：

适当的政策来控制电子邮件中发送的敏感信息

提高对政策的意识和理解

# 4.8 例二

“该公司在满足来自美国的信息自由要求时，意外地交出了在世老兵的数据。这项要求是要从一个已故退伍军人数据库中取得数据，但是数据库中却增加了2 257名活着的退伍军人的数据，而且可能会增加到4 000多名。这些数据包括姓名、社会保险号、出生日期和军事任务。”

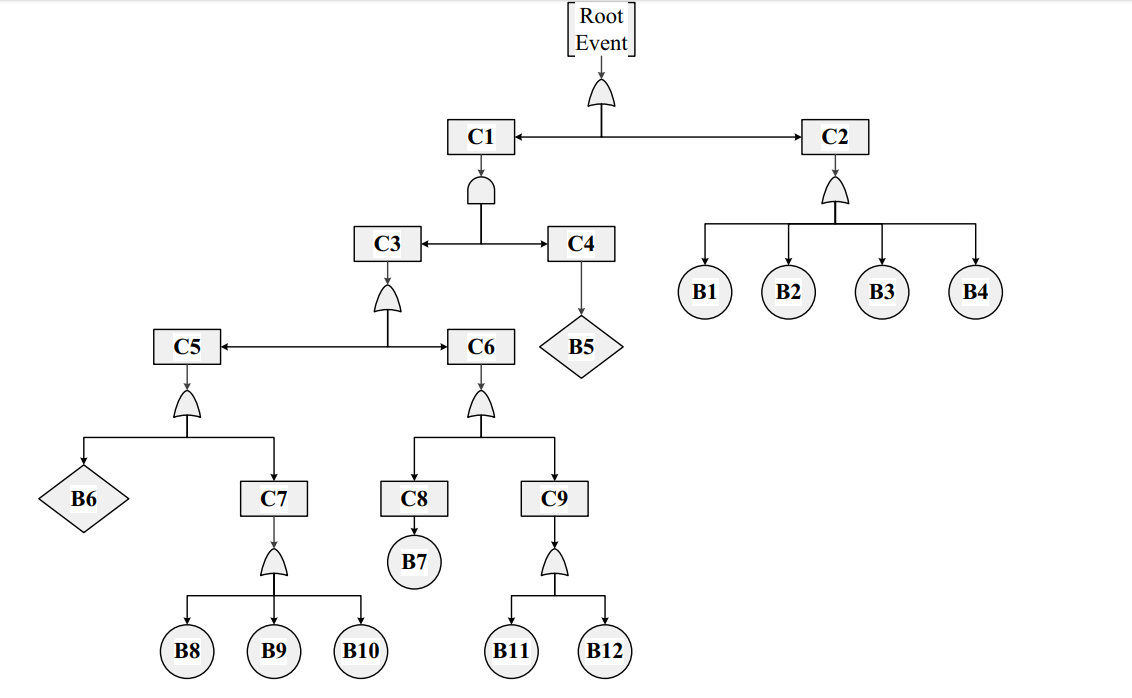


图4-2 例子二故障树分析图

根事件:意外数据泄漏

C1:员工过失

C2:错误的/不正确的行动

C3:过多的特权或访问控制权利

C4:缺乏尽职调查

C5:不遵守访问控制程序

C6:未能维护最低权限

C7:不遵守信息安全政策

C8:审计结果执行不力

C9:缺乏职责分工

B1:注意力不集中

B2:误解

B3:错误的优先级

B4:错误传达

B5:缺乏/不完善的质量控制程序/监督

B6:缺乏信息安全政策

(信息)管理不善

B8:缺乏对信息安全政策的认识或理解

无视信息安全政策

B10:围绕信息安全政策开展工作

B11:安全管理员业务知识不足

B12:许可蠕变

 （4-5）

 （4-6）

 （4-7）

 （4-8）

 （4-9）

 （4-10）

 （4-11）

 （4-12）

 （4-13）

 （4-14）

# 4.9 总结

由于涉及人为错误，因此很难发现和减轻意外的内部安全事件。用于检测员工恶意行为的框架和模型很可能无法检测到无意的内部人员，因为没有恶意意图。最好的解决办法是理解错误背后的人类行为。通过执行FTA，我们可以分析所有可能导致人为错误导致安全事件的场景。此外，对导致基本事件的根源事件的评估反映了不同的行为模式，因为每个人都是不同的。现有的用于检测恶意内部活动的解决方案无法替代，但需要为无意的内部人员提供单独的或互补的缓解策略。FTA对事故内部安全事件进行了大量的人为因素探究，有助于理解和解读有效的应对措施。这项研究将告诉组织和研究人员，意外的内部威胁也需要与恶意威胁同等的关注，因为人为错误的后果可能是严重的。

5案例介绍

# 5.1 案例背景介绍

国家互联网应急中心（英文：National Internet Emergency Center，缩写CNCERT或CNCERT/CC）全称是国家计算机网络应急技术处理协调中心，其成立于2002年9月，是中央网络安全和信息化委员会办公室领导下的国家级网络安全应急机构。

其致力于建设国家级的网络安全监测中心、预警中心、应急中心，以支撑政府主管部门履行网络安全相关的社会管理和公共服务职能，支持基础信息网络的安全防护和安全运行，支援重要信息系统的网络安全监测、预警和处置。

CNCERT作为国家互联网安全应急体系的核心技术协调机构，在协调国内安全应急组织（CERT）共同处理互联网安全事件方面发挥着重要作用。

2003年，CNCERT在我国大陆31个省、自治区、直辖市成立分中心，完成了跨网络、跨系统、跨地域的公共互联网网络安全应急技术支撑体系建设，形成了全国性的互联网网络安全信息共享、技术协调能力。

事件发现：CNCERT依托公共互联网网络安全监测平台开展对基础信息网络、金融证券等重要信息系统的自主监测。同时还通过与国内外合作伙伴进行数据和信息共享，以及通过热线电话、传真、电子邮件、网站等接收国内外用户的网络安全事件报告等多种渠道发现网络攻击威胁和网络安全事件。

预警通报：CNCERT依托对丰富数据资源的综合分析和多渠道的信息获取实现网络安全威胁的分析预警、网络安全事件的情况通报、宏观网络安全状况的态势分析等，为用户单位提供互联网网络安全态势信息通报、网络安全技术和资源信息共享等服务。

应急处置：对于自主发现和接收到的危害较大的事件报告，CNCERT及时响应并积极协调处置，重点处置的事件包括：影响互联网运行安全的事件、波及较大范围互联网用户的事件、涉及重要政府部门和重要信息系统的事件、用户投诉造成较大影响的事件，以及境外国家级应急组织投诉的各类网络安全事件等。

测试评估：作为网络安全检测、评估的专业机构，按照“支撑监管，服务社会”的原则，以科学的方法、规范的程序、公正的态度、独立的判断，按照相关标准为政府部门、企事业单位提供安全评测服务。CNCERT还组织通信网络安全相关标准制定，参与电信网和互联网安全防护系列标准的编制等除此之外，还承担全国互联网金融风险的技术监测和分析工作。

# 5.2 案例数据的选取和整理

我们选取了2017、2018年一共24份月安全报告，提取总结了相关安全事件的数据，充分考虑时间因素对安全事件的影响，总结出8类安全威胁事件：新增病毒、被篡改网站、被植入后门网站、网站页面仿冒、新增漏洞、远程攻击、垃圾邮件、木马劫持。具体计算见下一章。

将部分信息安全保护评测项与中心统计安全事件一一对应，归纳提取8个持续统计的安全事件对应的6个风险因素。

6个风险因素分别是“非法外联”“后门程序”“访问控制”“恶意代码”“身份鉴别”“非法入侵”。

8个安全事件分别是“新增病毒”“被篡改网站”“被植入后门网站”“网页仿冒”“新增漏洞”“远程攻击”“木马劫持”“垃圾邮件”。得事件和风险因素对照表：

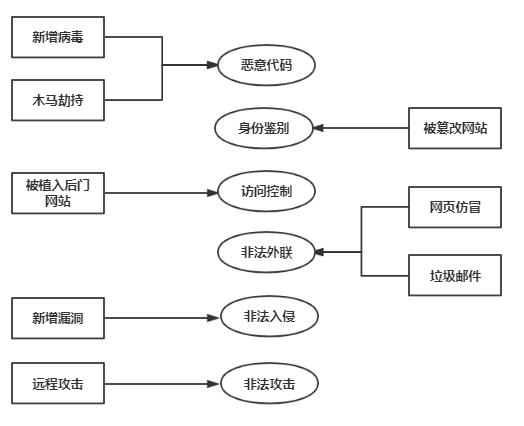


图5-1 事件和风险因素对照图

下表为2017年的安全事件原始数据：

表5-1 2017年的安全事件原始数据表

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 项目\月 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 新增病毒（万） | 188 | 118 | 128 | 144 | 159 | 532 | 182 | 208 | 148 | 113 | 100 | 75 |
| 被篡改网站 | 4981 | 4493 | 5252 | 6312 | 6245 | 3669 | 6468 | 6109 | 5494 | 5163 | 2368 | 4130 |
| 被植入后门网站 | 3265 | 3228 | 5422 | 3940 | 4916 | 4226 | 4146 | 4247 | 2825 | 2180 | 2504 | 3029 |
| 网站页面仿冒 | 2442 | 2120 | 3198 | 1810 | 1696 | 1515 | 1304 | 1572 | 1261 | 1219 | 1365 | 1349 |
| 新增漏洞 | 552 | 998 | 1066 | 1279 | 1399 | 1359 | 1380 | 1380 | 1538 | 1364 | 2112 | 1554 |
| 远程攻击 | 512 | 893 | 964 | 1156 | 1226 | 1267 | 1195 | 1277 | 1433 | 1104 | 1784 | 1367 |
| 垃圾邮件 | 7379 | 6434 | 12334 | 11178 | 7147 | 9324 | 9102 | 9854 | 8991 | 8998 | 8629 | 9407 |
| 被木马劫持IP（万） | 133 | 66 | 78 | 89 | 105 | 483 | 141 | 169 | 105 | 84 | 64 | 44 |

下表为2018年的安全事件原始数据：

表5-2 2018年的安全事件原始数据表

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 项目\月 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 新增病毒（万） | 83 | 52 | 75 | 85 | 102 | 86 | 86 | 81 | 68 | 64 | 84 | 78 |
| 被篡改网站 | 4101 | 3678 | 2559 | 1866 | 1799 | 1669 | 1658 | 1620 | 1164 | 612 | 1357 | 1376 |
| 被植入后门网站 | 2606 | 1718 | 2858 | 3280 | 3014 | 2734 | 3019 | 2934 | 2290 | 2507 | 2513 | 2317 |
| 网站页面仿冒 | 1554 | 583 | 3791 | 3440 | 5729 | 3963 | 4967 | 5819 | 7993 | 5548 | 6469 | 5324 |
| 新增漏洞 | 1392 | 824 | 1902 | 1130 | 1480 | 1020 | 1020 | 1249 | 1366 | 817 | 810 | 1206 |
| 远程攻击 | 1250 | 663 | 1631 | 992 | 1362 | 929 | 926 | 1166 | 1232 | 735 | 700 | 1067 |
| 垃圾邮件 | 9560 | 6853 | 9751 | 8829 | 9200 | 8903 | 8935 | 8975 | 9717 | 8252 | 8487 | 8278 |
| 被木马劫持IP（万） | 52 | 29 | 42 | 64 | 70 | 56 | 58 | 51 | 40 | 41 | 60 | 54 |

下图为2017年各类安全事件统计图：

图5-2 2017年各类安全事件统计图

下图为2018年各类安全事件统计图：

图5-3 2018年各类安全事件统计图

合并为6个风险因素后后，下图为2017年各类风险因素统计表：

表5-3 2017年各类风险因素统计表

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 项目\月 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 恶意代码(万） | 321 | 184 | 206 | 233 | 264 | 1015 | 323 | 377 | 253 | 197 | 164 | 119 |
| 身份鉴别 | 4981 | 4493 | 5252 | 6312 | 6245 | 3669 | 6468 | 6109 | 5494 | 5163 | 2368 | 4130 |
| 访问控制 | 3265 | 3228 | 5422 | 3940 | 4916 | 4226 | 4146 | 4247 | 2825 | 2180 | 2504 | 3029 |
| 非法外联 | 9821 | 8554 | 15532 | 12988 | 8843 | 10839 | 10406 | 11426 | 10252 | 10217 | 9994 | 10756 |
| 非法入侵 | 552 | 998 | 1066 | 1279 | 1399 | 1359 | 1380 | 1380 | 1538 | 1364 | 2112 | 1554 |
| 非法攻击 | 512 | 893 | 964 | 1156 | 1226 | 1267 | 1195 | 1277 | 1433 | 1104 | 1784 | 1367 |

下图为2018年各类风险因素统计表：

表5-4 2018年各类风险因素统计表

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 项目\月 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 恶意代码(万） | 135 | 81 | 117 | 149 | 172 | 142 | 144 | 132 | 108 | 105 | 144 | 132 |
| 身份鉴别 | 4101 | 3678 | 2559 | 1866 | 1799 | 1669 | 1658 | 1620 | 1164 | 612 | 1357 | 1376 |
| 访问控制 | 2606 | 1718 | 2858 | 3280 | 3014 | 2734 | 3019 | 2934 | 2290 | 2507 | 2513 | 2317 |
| 非法外联 | 11114 | 7436 | 13542 | 12269 | 14929 | 12866 | 13902 | 14794 | 17710 | 13800 | 14956 | 13602 |
| 非法入侵 | 1392 | 824 | 1902 | 1130 | 1480 | 1020 | 1020 | 1249 | 1366 | 817 | 810 | 1206 |
| 非法攻击 | 1250 | 663 | 1631 | 992 | 1362 | 929 | 926 | 1166 | 1232 | 735 | 700 | 1067 |

下图为2017年各类风险因素统计图：

图5-4 2017年各类风险因素统计图

下图为2018年各类风险因素统计图：

图5-4 2018年各类风险因素统计图

6 风险值计算

# 6.1确定故障树的结构

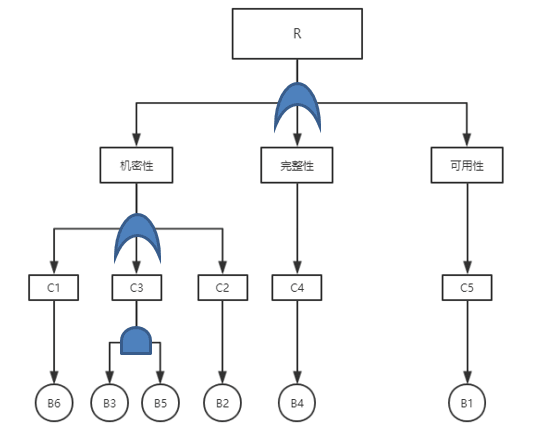


图6-1故障树分析图

各符号代表如下：

C1:产品选择模块，

C2：鉴别措施模块，

C3：非授权访问模块，

C4：物理隔离模块，

C5：病毒防护模块，

R：信息系统风险值，

B1：恶意代码，

B2：身份鉴别,

B3：访问控制，

B4：非法外联，

B5：非法入侵，

B6：非法攻击，

X1：新增病毒，

X2：被篡改网站，

X3：被植入后门网站，

X4：网站页面仿冒，

X5：新增漏洞，

X6：远程攻击，

X7：垃圾邮件，

X8：木马劫持事件。

通过上行算法求最小割集为：

 （6-1）

 （6-2）

 （6-3）

 （6-4）

 （6-5）

 （6-6）

# 6.2 计算威胁值

6.2.1 公式

 （6-7）

Ti为动态威胁值数值

Tix为本次安全威胁出现的次数与前面周期次数和的比值

Tiy为该安全威胁在同一时期与其他安全威胁的比值

公式的两个变量都引进了时间函数使Ti值更加客观准确。

6.2.2 Tix的计算

采用RM算法，将本周期的统计安全事件除以最近半年的同类安全事件总数，得到比例值（比如2017年1月的值除以2017上半年的值总和，2017年9月除以2017下半年的值总和）。

 （本案例中n=6） （6-8）

下表为2017年Tix计算表：

表6-1 2017年Tix计算表

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 项目\月 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 恶意代码(万） | 0.144 | 0.083 | 0.093 | 0.105 | 0.119 | 0.457 | 0.225 | 0.263 | 0.177 | 0.137 | 0.114 | 0.083 |
| 身份鉴别 | 0.161 | 0.145 | 0.170 | 0.204 | 0.202 | 0.119 | 0.218 | 0.205 | 0.185 | 0.174 | 0.080 | 0.139 |
| 访问控制 | 0.131 | 0.129 | 0.217 | 0.158 | 0.197 | 0.169 | 0.219 | 0.224 | 0.149 | 0.115 | 0.132 | 0.160 |
| 非法外联 | 0.148 | 0.128 | 0.233 | 0.195 | 0.133 | 0.163 | 0.165 | 0.181 | 0.163 | 0.162 | 0.159 | 0.171 |
| 非法入侵 | 0.083 | 0.150 | 0.160 | 0.192 | 0.210 | 0.204 | 0.148 | 0.148 | 0.165 | 0.146 | 0.226 | 0.167 |
| 非法攻击 | 0.085 | 0.148 | 0.160 | 0.192 | 0.204 | 0.211 | 0.146 | 0.156 | 0.176 | 0.135 | 0.219 | 0.168 |

下表为2018年Tix计算表：

表6-2 2018年Tix计算表

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 项目\月 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 恶意代码(万） | 0.170 | 0.102 | 0.147 | 0.187 | 0.216 | 0.178 | 0.188 | 0.173 | 0.141 | 0.137 | 0.188 | 0.173 |
| 身份鉴别 | 0.262 | 0.235 | 0.163 | 0.119 | 0.115 | 0.106 | 0.213 | 0.208 | 0.149 | 0.079 | 0.174 | 0.177 |
| 访问控制 | 0.161 | 0.106 | 0.176 | 0.202 | 0.186 | 0.169 | 0.194 | 0.188 | 0.147 | 0.161 | 0.161 | 0.149 |
| 非法外联 | 0.154 | 0.103 | 0.188 | 0.170 | 0.207 | 0.178 | 0.157 | 0.167 | 0.200 | 0.155 | 0.168 | 0.153 |
| 非法入侵 | 0.180 | 0.106 | 0.245 | 0.146 | 0.191 | 0.132 | 0.158 | 0.193 | 0.211 | 0.126 | 0.125 | 0.186 |
| 非法攻击 | 0.183 | 0.097 | 0.239 | 0.145 | 0.200 | 0.136 | 0.159 | 0.200 | 0.211 | 0.126 | 0.120 | 0.183 |

下图为2017年Tix计算图：

图6-1 2017年Tix计算图

下图为2018年Tix计算图：

图6-2 2018年Tix计算图

6.2.3 Tiy计算

Tiy为该安全威胁在同一时期与其他安全威胁的比值，m为安全个数因素，本案例中m=6。

 （6-9）

下表为2017年Tiy计算表：

表6-3 2017年Tiy计算表

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 项目\月 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 恶意代码(万） | 0.017 | 0.010 | 0.007 | 0.009 | 0.012 | 0.045 | 0.014 | 0.015 | 0.012 | 0.010 | 0.009 | 0.006 |
| 身份鉴别 | 0.256 | 0.245 | 0.185 | 0.244 | 0.273 | 0.164 | 0.270 | 0.246 | 0.252 | 0.255 | 0.125 | 0.197 |
| 访问控制 | 0.168 | 0.176 | 0.191 | 0.152 | 0.215 | 0.189 | 0.173 | 0.171 | 0.130 | 0.108 | 0.132 | 0.145 |
| 非法外联 | 0.505 | 0.466 | 0.546 | 0.501 | 0.386 | 0.484 | 0.435 | 0.460 | 0.470 | 0.505 | 0.528 | 0.513 |
| 非法入侵 | 0.028 | 0.054 | 0.037 | 0.049 | 0.061 | 0.061 | 0.058 | 0.056 | 0.071 | 0.067 | 0.112 | 0.074 |
| 非法攻击 | 0.026 | 0.049 | 0.034 | 0.045 | 0.054 | 0.057 | 0.050 | 0.051 | 0.066 | 0.055 | 0.094 | 0.065 |

下表为2018年Tiy计算表：

表6-4 2018年Tiy计算表

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 项目\月 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 恶意代码(万） | 0.007 | 0.006 | 0.005 | 0.008 | 0.008 | 0.007 | 0.007 | 0.006 | 0.005 | 0.006 | 0.007 | 0.007 |
| 身份鉴别 | 0.199 | 0.255 | 0.113 | 0.095 | 0.079 | 0.086 | 0.080 | 0.074 | 0.049 | 0.033 | 0.066 | 0.070 |
| 访问控制 | 0.127 | 0.119 | 0.126 | 0.167 | 0.132 | 0.141 | 0.146 | 0.134 | 0.096 | 0.135 | 0.123 | 0.118 |
| 非法外联 | 0.540 | 0.516 | 0.599 | 0.623 | 0.656 | 0.665 | 0.673 | 0.676 | 0.742 | 0.743 | 0.730 | 0.690 |
| 非法入侵 | 0.068 | 0.057 | 0.084 | 0.057 | 0.065 | 0.053 | 0.049 | 0.057 | 0.057 | 0.044 | 0.040 | 0.061 |
| 非法攻击 | 0.061 | 0.046 | 0.072 | 0.050 | 0.060 | 0.048 | 0.045 | 0.053 | 0.052 | 0.040 | 0.034 | 0.054 |

下图为2017年Tiy计算图：

图6-3 2017年Tiy计算图

下图为2018年Tiy计算图：

图6-4 2018年Tiy计算图

6.2.4 动态威胁值

动态威胁值计算公式

 （6-10）

Ti为动态威胁值数值

Tix为本次安全威胁出现的次数与前面周期次数和的比值

Tiy为该安全威胁在同一时期与其他安全威胁的比值

下表为2017年Ti计算表：

表6-5 2017年Ti计算表

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 项目\月 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 恶意代码(万） | 0.145 | 0.083 | 0.093 | 0.105 | 0.119 | 0.459 | 0.226 | 0.264 | 0.177 | 0.138 | 0.115 | 0.083 |
| 身份鉴别 | 0.302 | 0.285 | 0.251 | 0.318 | 0.339 | 0.202 | 0.347 | 0.321 | 0.313 | 0.309 | 0.148 | 0.241 |
| 访问控制 | 0.213 | 0.218 | 0.289 | 0.219 | 0.291 | 0.253 | 0.279 | 0.282 | 0.198 | 0.158 | 0.187 | 0.216 |
| 非法外联 | 0.526 | 0.484 | 0.594 | 0.538 | 0.408 | 0.511 | 0.465 | 0.495 | 0.498 | 0.531 | 0.551 | 0.541 |
| 非法入侵 | 0.088 | 0.160 | 0.165 | 0.198 | 0.219 | 0.213 | 0.159 | 0.158 | 0.179 | 0.161 | 0.252 | 0.182 |
| 非法攻击 | 0.089 | 0.156 | 0.164 | 0.197 | 0.211 | 0.218 | 0.155 | 0.165 | 0.188 | 0.146 | 0.238 | 0.180 |

下表为2018年Ti计算表：

表6-6 2018年Ti计算表

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 项目\月 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 恶意代码(万） | 0.170 | 0.102 | 0.147 | 0.187 | 0.216 | 0.179 | 0.188 | 0.173 | 0.141 | 0.137 | 0.188 | 0.173 |
| 身份鉴别 | 0.329 | 0.347 | 0.199 | 0.152 | 0.139 | 0.137 | 0.228 | 0.221 | 0.157 | 0.085 | 0.186 | 0.190 |
| 访问控制 | 0.205 | 0.160 | 0.217 | 0.262 | 0.228 | 0.220 | 0.243 | 0.231 | 0.176 | 0.210 | 0.203 | 0.190 |
| 非法外联 | 0.561 | 0.527 | 0.628 | 0.646 | 0.688 | 0.688 | 0.691 | 0.696 | 0.768 | 0.759 | 0.749 | 0.707 |
| 非法入侵 | 0.192 | 0.121 | 0.259 | 0.157 | 0.202 | 0.142 | 0.165 | 0.201 | 0.219 | 0.134 | 0.131 | 0.196 |
| 非法攻击 | 0.193 | 0.107 | 0.250 | 0.154 | 0.208 | 0.144 | 0.165 | 0.207 | 0.218 | 0.132 | 0.125 | 0.191 |

下图为2017年Ti计算图：

图6-5 2017年Ti计算图

下图为2018年Ti计算图：

图6-6 2018年Ti计算图

# 6.3 匹配脆弱性

为了最大限度的降低人为因素对模型评判的干扰，我们将脆弱性赋值分为6个档次：

表6-7 脆弱性赋值表

|  |  |
| --- | --- |
| 脆弱性赋值 | 相应的解释 |
| 0 | 在评估中处于安全状态 |
| 0.2 | 个别地方处于不安全状态 |
| 0.4 | 部分地方处于不安全状态 |
| 0.6 | 很多地方处于不安全状态 |
| 0.8 | 绝大多数地方处于不安全状态 |
| 1 | 没有采取安全防范措施，极度不安全 |

根据安全事件发生的次数、频率，我们对各个风险因素对应的脆弱性进行赋值：

下表为2017年各风险因素对应的脆弱性

表6-8 2017年各风险因素对应的脆弱性表

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 项目\月 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 恶意代码(万） | 0.4 | 0.3 | 0.3 | 0.3 | 0.3 | 0.8 | 0.4 | 0.4 | 0.3 | 0.3 | 0.3 | 0.2 |
| 身份鉴别 | 0.7 | 0.7 | 0.8 | 0.9 | 0.9 | 0.6 | 0.9 | 0.9 | 0.8 | 0.7 | 0.4 | 0.6 |
| 访问控制 | 0.3 | 0.3 | 0.5 | 0.4 | 0.5 | 0.4 | 0.4 | 0.4 | 0.3 | 0.2 | 0.3 | 0.3 |
| 非法外联 | 0.3 | 0.2 | 0.7 | 0.5 | 0.2 | 0.4 | 0.3 | 0.4 | 0.3 | 0.3 | 0.3 | 0.4 |
| 非法入侵 | 0.1 | 0.3 | 0.3 | 0.4 | 0.5 | 0.4 | 0.5 | 0.5 | 0.5 | 0.4 | 0.7 | 0.5 |
| 非法攻击 | 0.1 | 0.3 | 0.3 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.5 | 0.3 | 0.6 | 0.4 |

下表为2018年各风险因素对应的脆弱性

表6-9 2018年各风险因素对应的脆弱性表

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 项目\月 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 恶意代码(万） | 0.2 | 0.2 | 0.2 | 0.2 | 0.3 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| 身份鉴别 | 0.6 | 0.6 | 0.4 | 0.4 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.2 | 0.3 | 0.3 |
| 访问控制 | 0.3 | 0.2 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.2 | 0.3 | 0.3 | 0.2 |
| 非法外联 | 0.4 | 0.1 | 0.6 | 0.5 | 0.7 | 0.5 | 0.6 | 0.6 | 0.9 | 0.6 | 0.7 | 0.6 |
| 非法入侵 | 0.5 | 0.2 | 0.7 | 0.4 | 0.5 | 0.3 | 0.3 | 0.4 | 0.4 | 0.2 | 0.2 | 0.4 |
| 非法攻击 | 0.4 | 0.2 | 0.6 | 0.3 | 0.4 | 0.3 | 0.3 | 0.4 | 0.4 | 0.2 | 0.2 | 0.3 |

# 6.4 确定权重

根据现实情况，在相关国家要求中对非法外联的要求格外严格，并且明确要求信息系统不能存在非法外联情况，一旦出现，立刻停止系统运行，因此，经小组讨论，得各风险因素权重赋值如下：

恶意代码占比重为1，身份鉴别占比重为1，访问控制占比重为1，

非法外联占比重为1.2，非法入侵占比重为1，非法攻击占比重为1

# 6.5 风险值计算

根据公式，计算风险值如下：

表6-10 风险值表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | B1  恶意代码 | B2  身份鉴别 | B3  访问控制 | B4  非法外联 | B5  非法入侵 | B6  非法攻击 | 风险值 |
| U1 | 0.063 | 0.088 | 0.028 | 0.093 | 0.002 | 0.002 | 0.245 |
| U2 | 0.027 | 0.076 | 0.028 | 0.059 | 0.010 | 0.008 | 0.171 |
| U3 | 0.032 | 0.076 | 0.063 | 0.248 | 0.011 | 0.009 | 0.366 |
| U4 | 0.038 | 0.113 | 0.035 | 0.167 | 0.016 | 0.014 | 0.333 |
| U5 | 0.046 | 0.119 | 0.057 | 0.055 | 0.020 | 0.016 | 0.238 |
| U6 | 0.447 | 0.046 | 0.043 | 0.112 | 0.019 | 0.018 | 0.624 |
| U7 | 0.098 | 0.126 | 0.046 | 0.093 | 0.014 | 0.012 | 0.329 |
| U8 | 0.125 | 0.111 | 0.048 | 0.121 | 0.014 | 0.014 | 0.371 |
| U9 | 0.067 | 0.099 | 0.022 | 0.097 | 0.018 | 0.018 | 0.280 |
| U10 | 0.046 | 0.093 | 0.014 | 0.102 | 0.014 | 0.010 | 0.251 |
| U11 | 0.035 | 0.025 | 0.019 | 0.101 | 0.038 | 0.029 | 0.191 |
| U12 | 0.023 | 0.060 | 0.026 | 0.117 | 0.019 | 0.016 | 0.216 |
| U13 | 0.048 | 0.082 | 0.021 | 0.129 | 0.018 | 0.015 | 0.276 |
| U14 | 0.025 | 0.079 | 0.011 | 0.039 | 0.006 | 0.004 | 0.147 |
| U15 | 0.040 | 0.035 | 0.025 | 0.209 | 0.034 | 0.028 | 0.312 |
| U16 | 0.056 | 0.022 | 0.034 | 0.181 | 0.011 | 0.009 | 0.267 |
| U17 | 0.068 | 0.019 | 0.028 | 0.270 | 0.020 | 0.019 | 0.376 |
| U18 | 0.052 | 0.018 | 0.024 | 0.210 | 0.009 | 0.008 | 0.288 |
| U19 | 0.055 | 0.030 | 0.029 | 0.241 | 0.010 | 0.009 | 0.335 |
| U20 | 0.049 | 0.029 | 0.027 | 0.269 | 0.016 | 0.015 | 0.362 |
| U21 | 0.037 | 0.017 | 0.016 | 0.392 | 0.020 | 0.017 | 0.464 |
| U22 | 0.036 | 0.007 | 0.021 | 0.261 | 0.006 | 0.005 | 0.310 |
| U23 | 0.055 | 0.022 | 0.020 | 0.295 | 0.006 | 0.004 | 0.376 |
| U24 | 0.049 | 0.022 | 0.018 | 0.238 | 0.015 | 0.012 | 0.322 |

即风险矩阵如下：

R=

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0.063 | 0.088 | 0.028 | 0.093 | 0.002 | 0.002 | 0.245 |
| 0.027 | 0.076 | 0.028 | 0.059 | 0.010 | 0.008 | 0.171 |
| 0.032 | 0.076 | 0.063 | 0.248 | 0.011 | 0.009 | 0.366 |
| 0.038 | 0.113 | 0.035 | 0.167 | 0.016 | 0.014 | 0.333 |
| 0.046 | 0.119 | 0.057 | 0.055 | 0.020 | 0.016 | 0.238 |
| 0.447 | 0.046 | 0.043 | 0.112 | 0.019 | 0.018 | 0.624 |
| 0.098 | 0.126 | 0.046 | 0.093 | 0.014 | 0.012 | 0.329 |
| 0.125 | 0.111 | 0.048 | 0.121 | 0.014 | 0.014 | 0.371 |
| 0.067 | 0.099 | 0.022 | 0.097 | 0.018 | 0.018 | 0.280 |
| 0.046 | 0.093 | 0.014 | 0.102 | 0.014 | 0.010 | 0.251 |
| 0.035 | 0.025 | 0.019 | 0.101 | 0.038 | 0.029 | 0.191 |
| 0.023 | 0.060 | 0.026 | 0.117 | 0.019 | 0.016 | 0.216 |
| 0.048 | 0.082 | 0.021 | 0.129 | 0.018 | 0.015 | 0.276 |
| 0.025 | 0.079 | 0.011 | 0.039 | 0.006 | 0.004 | 0.147 |
| 0.040 | 0.035 | 0.025 | 0.209 | 0.034 | 0.028 | 0.312 |
| 0.056 | 0.022 | 0.034 | 0.181 | 0.011 | 0.009 | 0.267 |
| 0.068 | 0.019 | 0.028 | 0.270 | 0.020 | 0.019 | 0.376 |
| 0.052 | 0.018 | 0.024 | 0.210 | 0.009 | 0.008 | 0.288 |
| 0.055 | 0.030 | 0.029 | 0.241 | 0.010 | 0.009 | 0.335 |
| 0.049 | 0.029 | 0.027 | 0.269 | 0.016 | 0.015 | 0.362 |
| 0.037 | 0.017 | 0.016 | 0.392 | 0.020 | 0.017 | 0.464 |
| 0.036 | 0.007 | 0.021 | 0.261 | 0.006 | 0.005 | 0.310 |
| 0.055 | 0.022 | 0.020 | 0.295 | 0.006 | 0.004 | 0.376 |
| 0.049 | 0.022 | 0.018 | 0.238 | 0.015 | 0.012 | 0.322 |

7 预测实验

上述算法模型计算单一风险值需要固定的附近六个月的数据，但是在现实情况下往往需要及时甚至超前做出决策减小风险保证安全，所以我们创新性地基于机器学习的各个算法根据已有的分险因子做出了不同模型并综合进行比较判断，最终完成大致预测，达到和计算不相上下的成果。以下为代码和部分注释。

首先导入所需Python模块

* 数据处理：pandas，numpy
* 画图（数据可视化）：seaborn，matplotlib
* 机器学习：sklearn

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

%matplotlib inline

from sklearn.linear\_model import LogisticRegression

from sklearn.svm import SVC,LinearSVC

from sklearn.ensemble import RandomForestClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn.naive\_bayes import GaussianNB

from sklearn.linear\_model import Perceptron

from sklearn.linear\_model import SGDClassifier

from sklearn.tree import DecisionTreeClassifier

读入数据集，测试集和真实数据

train\_df = pd.read\_csv("data.csv")

raw\_df = pd.read\_csv("test.csv")

true\_df = pd.read\_csv("true.csv")

将矩阵前20行数据作为训练数据集

train\_df

表7-1 训练数据集表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | t | B1 | B2 | B3 | B4 | B5 | B6 | R |
| 0 | U1 | 0.052 | 0.220 | 0.069 | 0.185 | 0.011 | 0.009 | 0.467 |
| 1 | U2 | 0.023 | 0.190 | 0.070 | 0.118 | 0.048 | 0.040 | 0.375 |
| 2 | U3 | 0.026 | 0.191 | 0.157 | 0.496 | 0.054 | 0.047 | 0.768 |
| 3 | U4 | 0.032 | 0.282 | 0.086 | 0.334 | 0.082 | 0.071 | 0.726 |
| 4 | U5 | 0.038 | 0.298 | 0.143 | 0.110 | 0.101 | 0.082 | 0.543 |
| 5 | U6 | 0.373 | 0.115 | 0.107 | 0.224 | 0.095 | 0.089 | 0.811 |
| 6 | U7 | 0.081 | 0.314 | 0.116 | 0.187 | 0.072 | 0.058 | 0.650 |
| 7 | U8 | 0.104 | 0.277 | 0.120 | 0.241 | 0.071 | 0.068 | 0.699 |
| 8 | U9 | 0.056 | 0.247 | 0.056 | 0.193 | 0.092 | 0.089 | 0.590 |
| 9 | U10 | 0.038 | 0.231 | 0.034 | 0.204 | 0.072 | 0.050 | 0.527 |
| 10 | U11 | 0.029 | 0.061 | 0.047 | 0.202 | 0.188 | 0.146 | 0.448 |
| 11 | U12 | 0.019 | 0.151 | 0.065 | 0.233 | 0.095 | 0.080 | 0.490 |
| 12 | U13 | 0.040 | 0.205 | 0.053 | 0.259 | 0.088 | 0.077 | 0.586 |
| 13 | U14 | 0.021 | 0.198 | 0.027 | 0.079 | 0.028 | 0.018 | 0.316 |
| 14 | U15 | 0.033 | 0.087 | 0.062 | 0.419 | 0.171 | 0.138 | 0.687 |
| 15 | U16 | 0.046 | 0.054 | 0.086 | 0.361 | 0.055 | 0.046 | 0.512 |
| 16 | U17 | 0.057 | 0.048 | 0.069 | 0.540 | 0.099 | 0.093 | 0.744 |
| 17 | U18 | 0.043 | 0.045 | 0.060 | 0.420 | 0.044 | 0.039 | 0.550 |
| 18 | U19 | 0.046 | 0.075 | 0.073 | 0.482 | 0.051 | 0.045 | 0.651 |
| 19 | U20 | 0.041 | 0.072 | 0.068 | 0.538 | 0.080 | 0.076 | 0.732 |

t列作为标记不参与训练，将t列去除

train\_df = train\_df.drop(['t'], axis=1)

test\_df = raw\_df.drop(['t'], axis=1)

train\_df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 20 entries, 0 to 19

Data columns (total 7 columns):

B1 20 non-null float64

B2 20 non-null float64

B3 20 non-null float64

B4 20 non-null float64

B5 20 non-null float64

B6 20 non-null float64

R 20 non-null float64

dtypes: float64(7)

memory usage: 1.2 KB

发现数据格式为float64，处理转化为int

for col in train\_df.columns:

train\_df[col]=train\_df[col].map(lambda x:-1 if str(x)=='nan' else int(x\*1000))

for col in test\_df.columns:

test\_df[col]=test\_df[col].map(lambda x:-1 if str(x)=='nan' else int(x\*1000))

true\_df['R']=true\_df["R"].map(lambda x:-1 if str(x)=='nan' else int(x\*1000))

进一步处理，查看最后数据集大小

X\_train = train\_df.drop('R',axis=1)

Y\_train = train\_df['R']

X\_test = test\_df.copy()

X\_train.shape, Y\_train.shape, X\_test.shape

((20, 6), (20L,), (4, 6))

使用不同方式进行模型训练和预测

*# Logistic Regression()*

from sklearn.utils import shuffle

X\_shuffle, Y\_shuffle = shuffle(X\_train, Y\_train)

logreg = LogisticRegression()

logreg.fit(X\_shuffle, Y\_shuffle)

Y\_pred1 = logreg.predict(X\_test)

acc\_log = round(logreg.score(X\_train, Y\_train) \* 100, 2)

acc\_log

100.0

*# Support Vector Machines*

svc = SVC()

svc.fit(X\_train,Y\_train)

Y\_pred2 = svc.predict(X\_test)

acc\_svc = round(svc.score(X\_train, Y\_train) \* 100, 2)

acc\_svc

100.0

*# k-Nearest Neighbors*

knn = KNeighborsClassifier(n\_neighbors = 3)

knn.fit(X\_train, Y\_train)

Y\_pred3 = knn.predict(X\_test)

acc\_knn = round(knn.score(X\_train, Y\_train) \* 100, 2)

acc\_knn

35.0

*# Gaussian Naive Bayes*

gaussian = GaussianNB()

gaussian.fit(X\_train, Y\_train)

Y\_pred4 = gaussian.predict(X\_test)

acc\_gaussian = round(gaussian.score(X\_train, Y\_train) \* 100, 2)

acc\_gaussian

100.0

*# Perceptron*

perceptron = Perceptron()

perceptron.fit(X\_train, Y\_train)

Y\_pred5 = perceptron.predict(X\_test)

acc\_perceptron = round(perceptron.score(X\_train, Y\_train) \* 100, 2)

acc\_perceptron

40.0

*# Linear SVC*

linear\_svc = LinearSVC()

linear\_svc.fit(X\_train, Y\_train)

Y\_pred6 = linear\_svc.predict(X\_test)

acc\_linear\_svc = round(linear\_svc.score(X\_train, Y\_train) \* 100, 2)

acc\_linear\_svc

45.0

*# Stochastic Gradient Descent 随机梯度下降，这个每次运行结果会不一样*

sgd = SGDClassifier()

sgd.fit(X\_train, Y\_train)

Y\_pred7 = sgd.predict(X\_test)

acc\_sgd = round(sgd.score(X\_train,Y\_train) \* 100, 2)

acc\_sgd

25.0

*# Decision*

decision\_tree = DecisionTreeClassifier()

decision\_tree.fit(X\_train, Y\_train)

Y\_pred8 = decision\_tree.predict(X\_test)

acc\_decision\_tree = round(decision\_tree.score(X\_train, Y\_train) \* 100,2)

acc\_decision\_tree

100.0

*# Random Forest*

random\_forest = RandomForestClassifier(n\_estimators=100)

random\_forest.fit(X\_train,Y\_train)

Y\_pred9 = random\_forest.predict(X\_test)

random\_forest.score(X\_train,Y\_train)

acc\_random\_forest = round(random\_forest.score(X\_train, Y\_train) \* 100, 2)

acc\_random\_forest

100.0

将9种算法的评分进行比较

models = pd.DataFrame({

'Model': ['Support Vector Machines', 'KNN', 'Logistic Regression', 'Random Forest', 'Naive Bayes', 'Perceptron', 'Stochastic Gradient Decent', 'Linear SVC', 'Decision Tree'],

'Score': [acc\_svc, acc\_knn, acc\_log, acc\_random\_forest, acc\_gaussian, acc\_perceptron, acc\_sgd, acc\_linear\_svc, acc\_decision\_tree] })

models.sort\_values(by='Score', ascending=False)

表7-2 算法的评分表

|  |  |  |
| --- | --- | --- |
|  | Model | Score |
| 0 | Support Vector Machines | 100.0 |
| 2 | Logistic Regression | 100.0 |
| 3 | Random Forest | 100.0 |
| 4 | Naive Bayes | 100.0 |
| 8 | Decision Tree | 100.0 |
| 7 | Linear SVC | 45.0 |
| 5 | Perceptron | 40.0 |
| 1 | KNN | 35.0 |
| 6 | Stochastic Gradient Decent | 25.0 |

通过与真实值的比较，可以看到预测的大致趋势与实际吻合，等级划分可以使预测结果更加准确。

submission = pd.DataFrame({

"t": raw\_df["t"],

"R": true\_df["R"]\*0.001,

"r1": Y\_pred1\*0.001,

"r2": Y\_pred2\*0.001,

"r3": Y\_pred3\*0.001,

"r4": Y\_pred4\*0.001,

"r5": Y\_pred5\*0.001,

"r6": Y\_pred6\*0.001,

"r7": Y\_pred7\*0.001,

"r8": Y\_pred8\*0.001,

"r9": Y\_pred9\*0.001,

})

submission.to\_csv('./submission.csv', index=False)

submission.head()

表7-3 真实值与预测值比较表

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | t | R | r1 | r2 | r3 | r4 | r5 | r6 | r7 | r8 | r9 |
| 0 | U21 | 0.947 | 0.732 | 0.65 | 0.65 | 0.65 | 0.732 | 0.527 | 0.732 | 0.527 | 0.744 |
| 1 | U22 | 0.597 | 0.550 | 0.65 | 0.65 | 0.65 | 0.744 | 0.550 | 0.732 | 0.550 | 0.550 |
| 2 | U23 | 0.714 | 0.550 | 0.65 | 0.65 | 0.65 | 0.650 | 0.550 | 0.732 | 0.512 | 0.550 |
| 3 | U24 | 0.638 | 0.732 | 0.65 | 0.55 | 0.65 | 0.650 | 0.527 | 0.732 | 0.447 | 0.527 |

可以看到优秀的训练模型可以大致预测出趋势，可以保证在直接计算无法实现时仅凭当月风险因素预测出总分险值R，及时采取对策降低损失。

8 结论和展望

本文通过建立基于动态威胁值和故障树的风险评估模型，根据相关公式计算得到了可靠的风险矩阵，有效限制了风险评估中的不可控因素，并利用不同机器学习算法预测了相应的风险值。预测结果与实际结果的误差在允许范围内，进一步证明了实验模型的正确性。

模型的准确性与数据的数量有很大关系，未来我们会丰富数据来源，扩大数据库信息规模，并且结合专业知识，利用爬虫爬取更多相关数据，扩大数据集，修正模型，训练出更可靠的结果，来验证模型设想。

利用动态威胁值与故障树对信息系统的风险进行预测，在实际生活中也会发挥很大的作用。风险值的精确预测，有利于系统管理员对特殊周期风险因素提前采取安全防范措施，减少系统损坏和遭受攻击的概率，提高系统的可靠性和稳定性，避免造成巨大损失。

参考文献

[1]Pallavi Patil,Pavol Zavarsky,Fault Tree Analysis of Accidental Insider Security Events[J].2012 International Conference on Cyber Security,2012(21):113-118

[2]刘忻.基于动态威胁值与故障树的信息安全风险评估模型研究[D].兰州：兰州大学图书馆，2014

[3]D. Capelli, A. Moore, R. Trzeciak, “The CERT Guide to Insider Threats”, 2012.

[4] M. Alawneh, I. Abbadi, “Defining and Analyzing Insiders and their Threats in Organizations”, 2011.

[5] B. Burke, C. Christiansen, “Insider Risk Management: A Framework Approach to Internal Security”, 2009. 9. Available: http://www.rsa.com/ solutions/business/insider\_risk/wp/10388\_219105.pdf

[6] Cisco, “Data leakage worldwide: The effectiveness of security policies”, 2008. Available: http://www.cisco.com/en/US/solutions/collateral/ ns170 /ns896/ns895/white\_paper\_c11-503131.pdf

[7] Y. Zheng, R. Xu, “A Human Factors Fault Tree Analysis Method for Software Engineering”,2008年

[8] G. Stonebumer, A. Goguen, A. Feringa, “Risk Management Guide for Information Technology Systems” Recommendations of the National Institute of Standards and Technology, 2002. Available: http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf