学术写作与学术规范

学术论文写作

苏小红

哈尔滨工业大学 计算机学院

学术论文正文

学术论文正文的基本组成

- Outline
  - Title
  - (1) Abstract
  - (2) Introduction（含Your contribution)
  - (3) Previous Work（Related Work)
  - (4) Our Solution（Motivation and theoretical Support)
  - (5) Experiments （Experimental Support)
  - (6) Discussion（Performance Analysis，Threats To Validity)
  - (7) Conclusion
  - (8) Acknowledgement（Optional)
  - (9) References
  - (10) Appendix（Optional)

并列式
递进式
总分式
分总式
…

相关工作

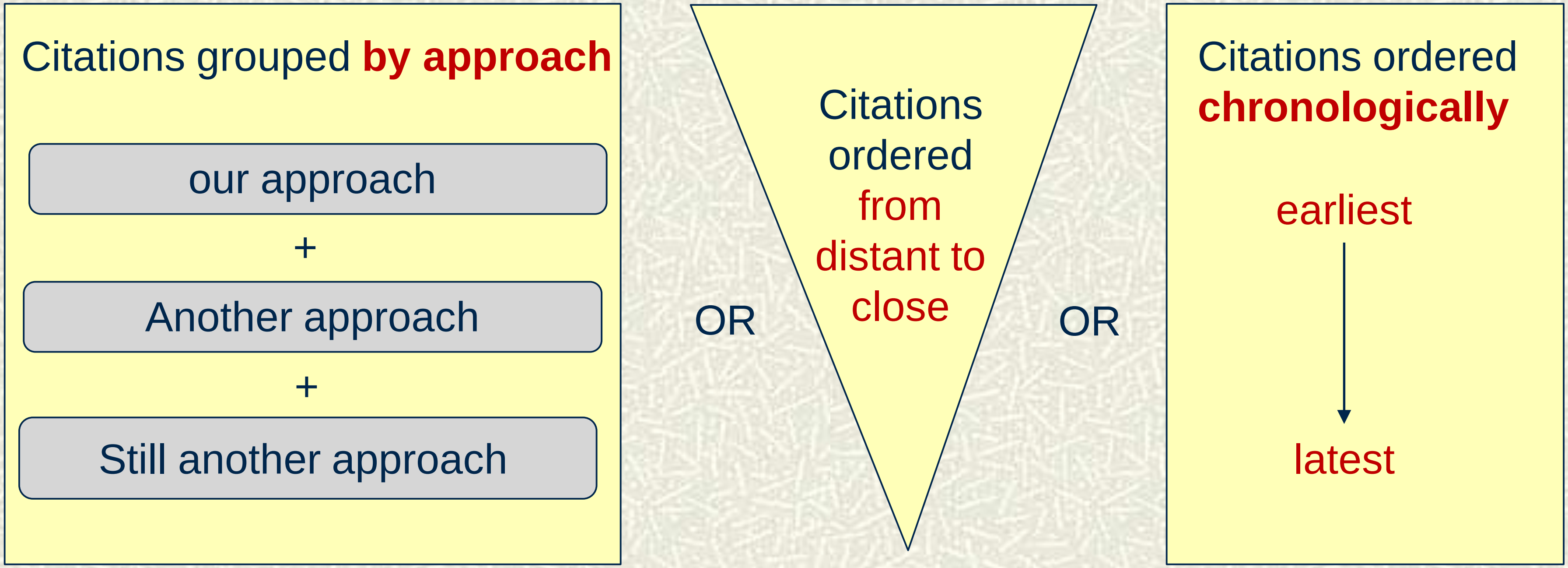
(Related Work)

相关工作写什么?

- Related/Previous Works
  - 你做的与前人有什么不同?
    - a) 将历史上前人的工作分类
    - b) 对每项重要的历史工作进行简短的回顾
    - c) 与自己的工作进行比较，强调和前人工作的不同
- 对文献要梳理出一条主线
  - 文献分类综述，各类要平衡
  - 引最新的，补上投稿前的文献
  - 直接相关，权威性，顶级期刊

相关工作的写作要点

Order of Citations



Citations grouped **by approach**

our approach

+

Another approach

+

Still another approach

OR

Citations ordered from distant to close

OR

Citations ordered **chronologically**

earliest

latest

- 问题为主线
- 研究者为主线

- 按主题/方法梳理
  - 按相关性由弱到强
  - 按时间由远及近
  - 或两种组合

相关工作示例

以研究者为主线：(研究者)做了…

Clemens *etal.* [14], **Wang** *etal.* [15, 16] proposed that chemisorption of…Via extensive quantum chemistry calculations, **Wang** *etal.* [18] found that ·OH is the key in low temperature oxidation and H abstractions by O_2 and peroxides are major ways to produce…

高水平期刊 (Science 2014, 345: 1599-1602)：以问题为主线

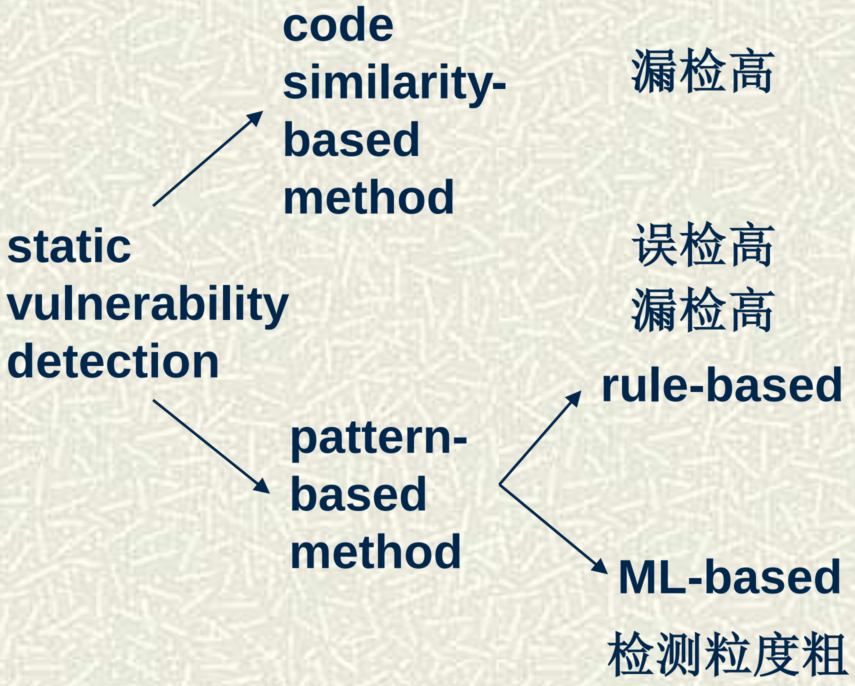
Oxygen activation at the metal-support interface is widely regarded as the key step in roomtemperature CO oxidation [13–17], but substantial debate remains regarding the nature of the active site [9, 12, 17–23]. Experimental studies indicate that materials lacking OH groups are inactive [24, 25], yet, the dominant mechanistic models vary in the suggested role of support OH groups and generally highlight the possible role of oxygen vacancies [16, 17, 22–24, 26]

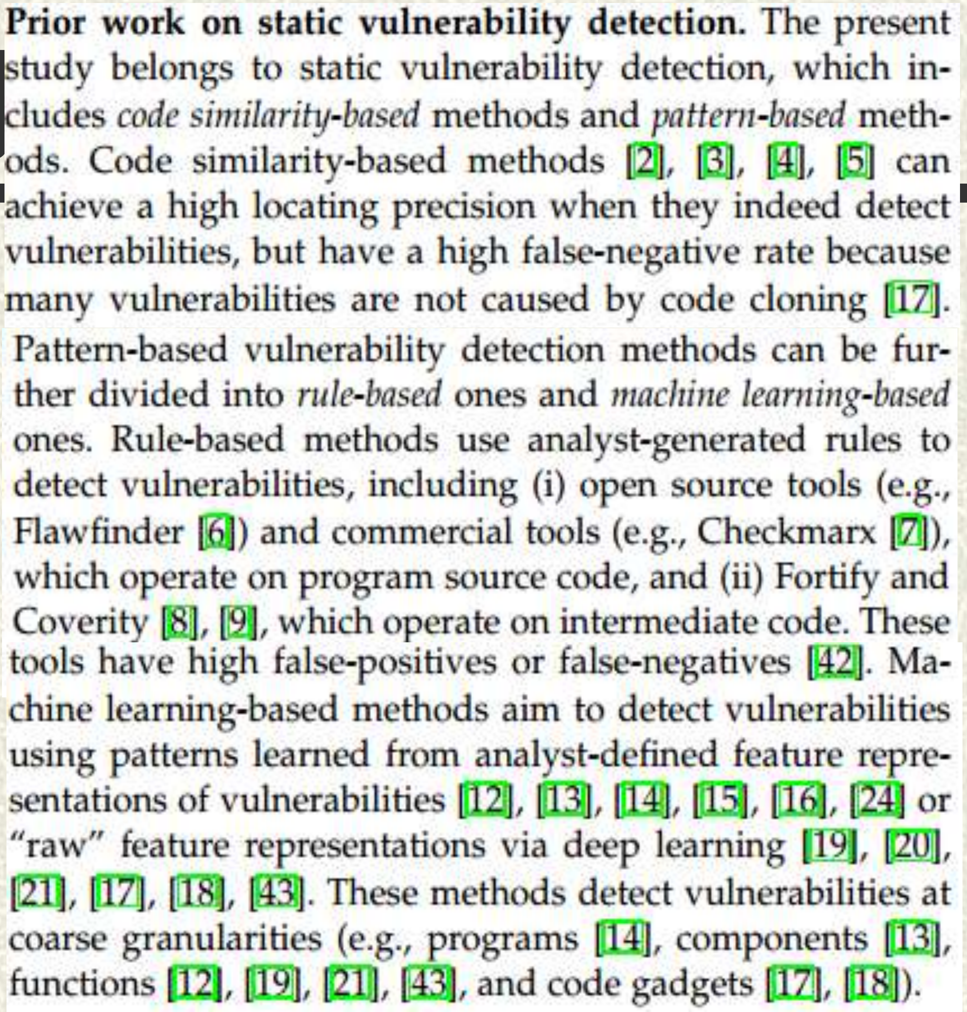
} 问题是什么

} 围绕着问题做了什么

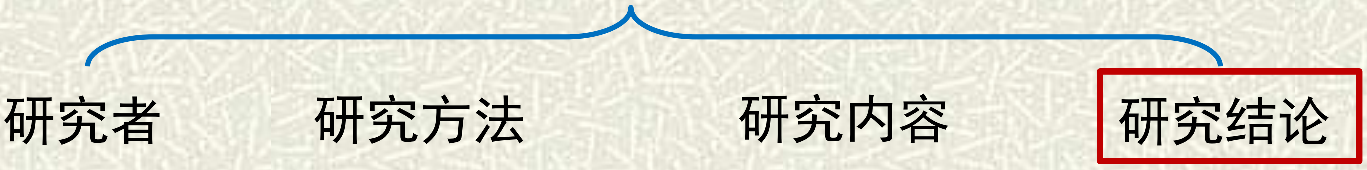
相关工作示例

按方法梳理，问题主线




Prior work on static vulnerability detection. The present study belongs to static vulnerability detection, which includes *code similarity-based* methods and *pattern-based* methods. Code similarity-based methods [2], [3], [4], [5] can achieve a high locating precision when they indeed detect vulnerabilities, but have a high false-negative rate because many vulnerabilities are not caused by code cloning [17]. Pattern-based vulnerability detection methods can be further divided into *rule-based* ones and *machine learning-based* ones. Rule-based methods use analyst-generated rules to detect vulnerabilities, including (i) open source tools (e.g., Flawfinder [6]) and commercial tools (e.g., Checkmarx [7]), which operate on program source code, and (ii) Fortify and Coverity [8], [9], which operate on intermediate code. These tools have high false-positives or false-negatives [42]. Machine learning-based methods aim to detect vulnerabilities using patterns learned from analyst-defined feature representations of vulnerabilities [12], [13], [14], [15], [16], [24] or "raw" feature representations via deep learning [19], [20], [21], [17], [18], [43]. These methods detect vulnerabilities at coarse granularities (e.g., programs [14], components [13], functions [12], [19], [21], [43], and code gadgets [17], [18]).

论述一个研究/发现　　重点要论述什么？



有问题的文献综述语句

Recently, with the help of computational chemistry modelling, detailed reaction

研究方法

pathways at the initial stage including O_2 interacting with active sites and the

研究内容

production of important intermediates have been investigated [17-21]. Via extensive

研究结论是什么？？

quantum chemistry calculations, Wang *etal*. [18] found that ·OH is the key···

研究方法 研究者 研究结论

存在问题：研究综述部分层次比较混乱，有的句子缺乏研究结论。

可以修改为：

Recently, computational chemistry modelling have been a powerful tool for···. Computational studies indicated that the ·OH is the key of ···

本文工作

（Our Solution ）

写作要点和范式

- **Our Solution/Work**
  - **Purpose**
    - **Introduce your work**
    - **Theoretical support to your work**
  - **Style**
    - **Motivation**
    - **Definition, notation**
    - **Theoretical analysis：Lemma，Theorem，Proof**
      - **Put tedious details in Appendix**
    - **Algorithm，Pseudo-code**
    - **Diagram，Explanations**
    - **If you were the reader, what questions you want to ask?**

写作的基本要求

- 提出**论点**, 通过**论据** (事实和/或 数据) 对论点加以**论证**
  - 论点明确, 论据充分, 论证合理
  - 事实准确, 数据准确, 计算准确, 语言准确
  - 内容丰富, 文字简练, 避免重复、 繁琐
  - 条理清楚, 逻辑性强, 表达形式与内容相适应
  - 不泄密, 对需保密的资料应作技术处理

写作的基本原则

1. 以作者的基本论点为轴线

- 对新发现的问题
  - 详尽分析和阐述，并加以严密论证
- 对一般性的问题
  - 只需作简明扼要的叙述
- 对与基本观点不相干的问题
  - 完全不费笔墨，哪怕只有一句一字

写作的基本原则

⌗ 2. 注重论据的科学性和准确性

- 用事实或数据说明论点, 形成材料与观点的统一
- 数据、图表准确
- 遣词造句准确，避免词不达意
- 计算结果实事求是，避免粗心大意

写作的基本原则

3. 支撑材料的准备

- 按来源可以分为三类：
  - 直接材料：亲自调查或科学实验得到的材料
  - 间接材料：引用文献中的或由他人提供的材料
  - 发展材料：整理、分析、研究而形成的材料

写作的基本原则

4. 选择材料的基本原则

- **必要**：不可或缺
- **充分**：量要充足
- **真实**：不可编造
- **准确**：符合实际，不断章取义
- **典型**：有代表性和说服力
- **新颖**：新鲜而不陈旧

写作的基本原则

5. 论点要鲜明

- A paper claims:
  - "To the best of our knowledge, this is the **most sophisticated** neural network solution ever mentioned in the literature."
- Reviewer:
  - **What problem does it solve?** What is the benchmark? **I can't measure " sophisticated "** :)

写作的基本原则

6. 动机和理由要真实可信

- Keep your reasons real
- Consider the opening sentence of this (fictional) introduction:
  - “Machine learning has gathered a lot of interest recently. Deep Learning is now a popular tool. We therefore use it to...”
- Reviewer:
  - This was your one chance to convince me of the problem you're working on. And now you told me you're working on it because it is popular...

写作的基本原则

7. 不要提出未被验证的论点

- **Never make a claim that is not directly validated by a theorem, an experiment, or a reference.**
- **E.g.**
  - **If you're claiming that a model has vanishing gradients, calculate the norm of the gradients!.**
- **Make claims that are useful to tell the story.**
- **More claims is not always better.**
- **It's important to write the paper first to know what claims you have to make, and what experiments/ theory are needed to validate your claims.**

写作的基本原则

8. 论点和方法的唯一保证就是：证明它

- Only have theorems that improve your story,
  - E.g.
    - You made a claim that your algorithm approximates some loss function ->
    - prove a theorem quantifying this approximation.
- Unless proofs are important for the story.
  - Do **not** say "Here are some guarantees from our algorithm".
- **Introduce and justify its existence first.**

写作的基本原则

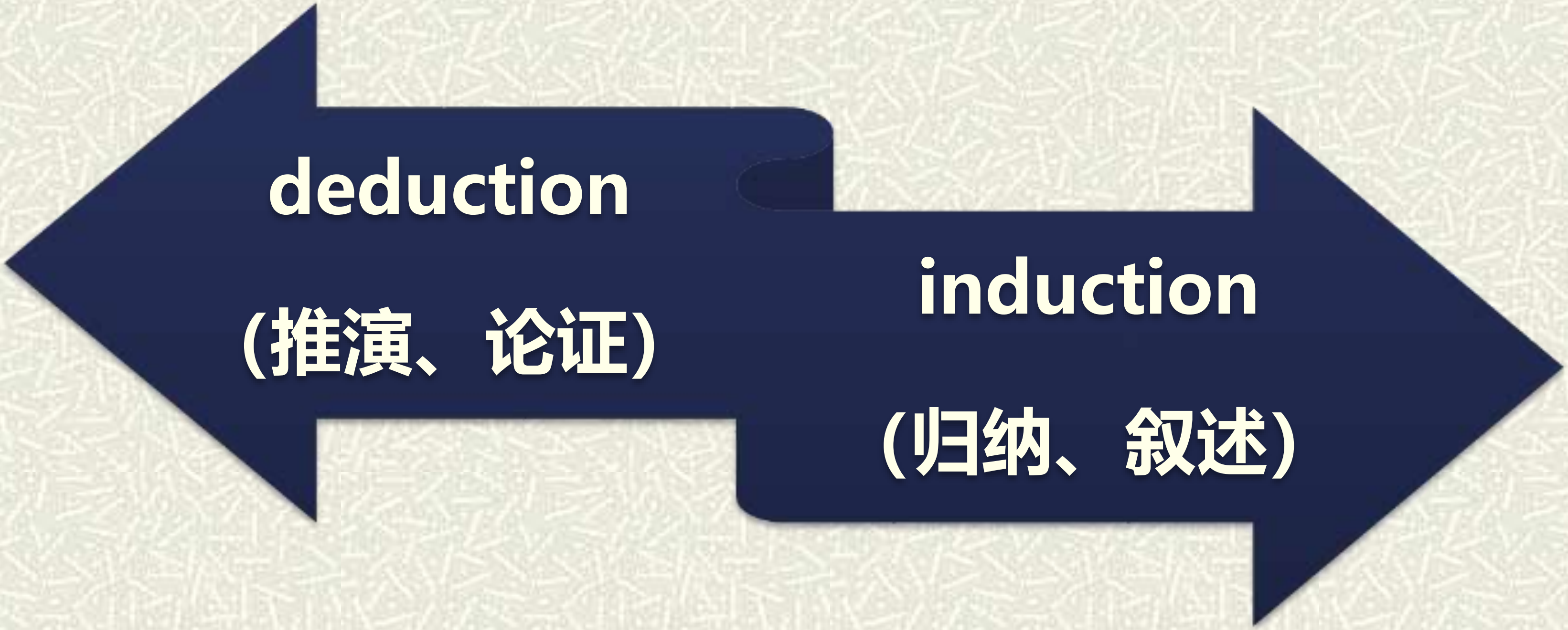
9. “读者思维”， reader-friendly

- write to be understood
- For every line in your paper, ask questions about your reader's mental model:
  1. What does my reader understand up to this point?
  2. What is my reader thinking at this point?
  3. How will my next narrative change that?

写作的基本原则

10.灵活运用行文推进模式

- 确保语义连贯性，逻辑顺畅性



deduction

（推演、论证）

induction

（归纳、叙述）

多以 induction 的论述形式为主
穿插 deduction 的推演与理论性分析

写作的基本原则

deduction（推演、论证）

- argument（争议）引出，注重理论性、论证性分析
- 像是“证公式”，适合要进行论证的场合
- 写作较“烧脑”，对读者的专业要求高
- 多见于数据部分

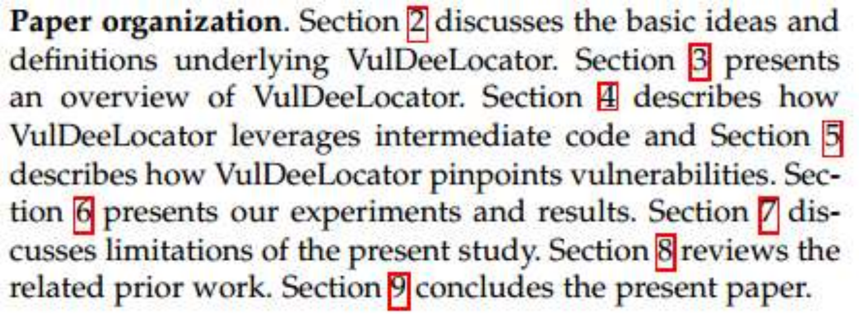
induction（归纳、叙述）

- 由 problem（问题）引出，注重问题引领
- 像是“讲故事”，叙事性结构为主
- 写作无需太“烧脑”，，对读者专业要求低
- 多见于 Introduction 部分

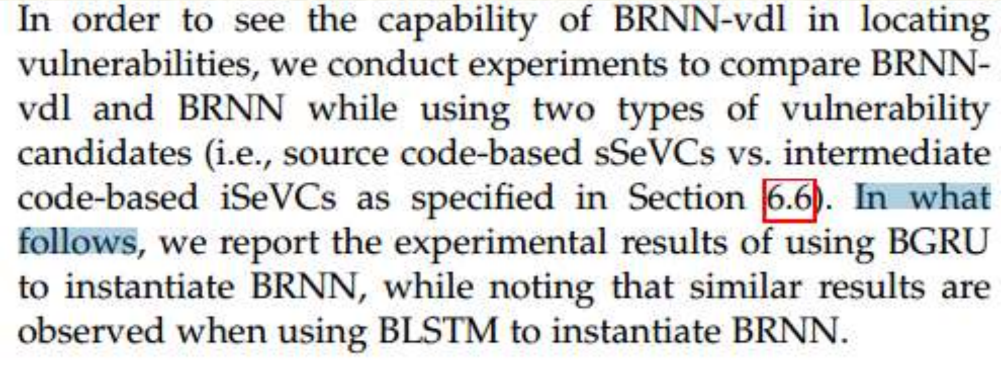
写作的基本原则

11. 前后段落间，环环相扣

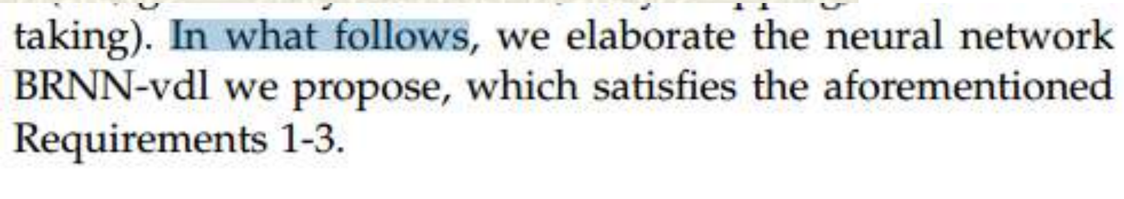
- **Before switching sections**, **always have the last paragraph of the previous one introduce it.**
- **More importantly, explain why the next section is needed.**



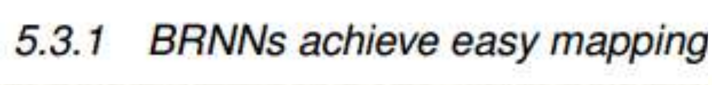
Paper organization. Section 2 discusses the basic ideas and definitions underlying VulDeeLocator. Section 3 presents an overview of VulDeeLocator. Section 4 describes how VulDeeLocator leverages intermediate code and Section 5 describes how VulDeeLocator pinpoints vulnerabilities. Section 6 presents our experiments and results. Section 7 discusses limitations of the present study. Section 8 reviews the related prior work. Section 9 concludes the present paper.



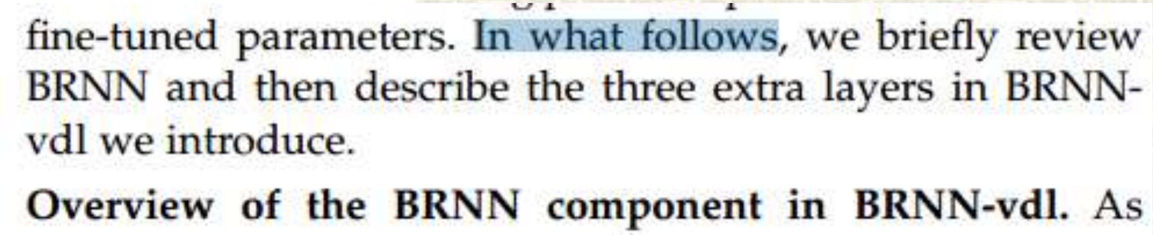
In order to see the capability of BRNN-vdl in locating vulnerabilities, we conduct experiments to compare BRNN-vdl and BRNN while using two types of vulnerability candidates (i.e., source code-based sSeVCs vs. intermediate code-based iSeVCs as specified in Section 6.6). In what follows, we report the experimental results of using BGRU to instantiate BRNN, while noting that similar results are observed when using BLSTM to instantiate BRNN.



taking). In what follows, we elaborate the neural network BRNN-vdl we propose, which satisfies the aforementioned Requirements 1-3.



5.3.1 BRNNs achieve easy mapping



fine-tuned parameters. In what follows, we briefly review BRNN and then describe the three extra layers in BRNN-vdl we introduce.

Overview of the BRNN component in BRNN-vdl. As

写作的基本原则

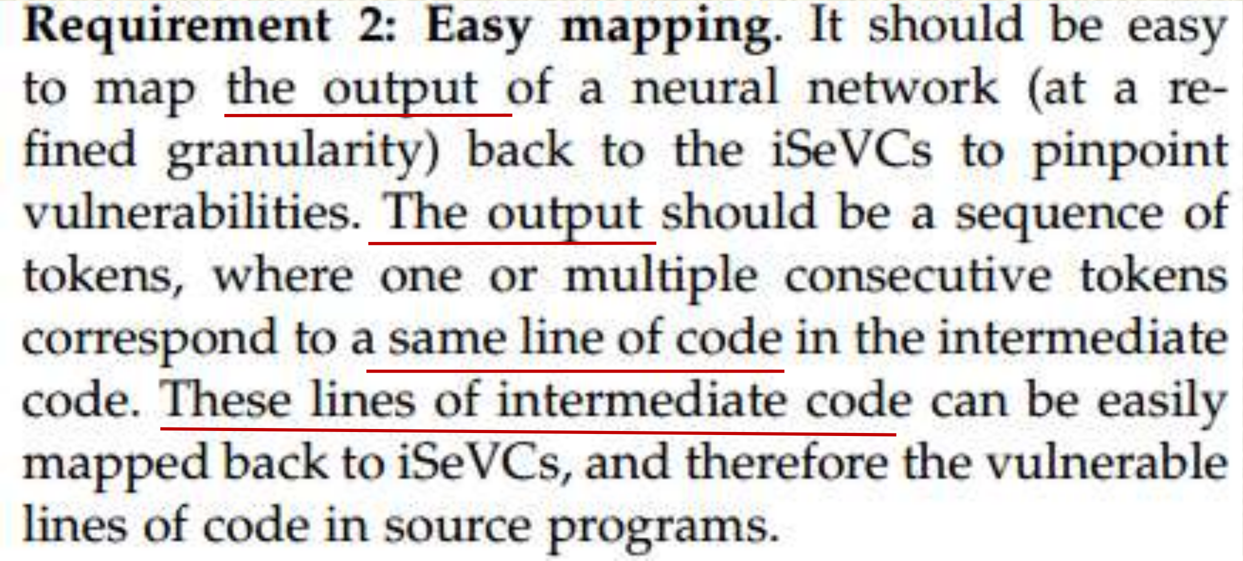
12. 前后语句间，逻辑关系清晰

- 当描述内容发生转变时，要学会使用“信号词/线索词”来帮助读者理清脉络
  - in this study
  - Result shows
  - the result of the research
- 避免前后两句话跳跃幅度过大

写作的基本原则

13. 新旧信息的衔接要流畅连贯

- 段落结构：
  - 新（信息），旧 +新，旧+新，旧+新，…
  - 新（信息），旧+新，（新+旧），旧+新，…

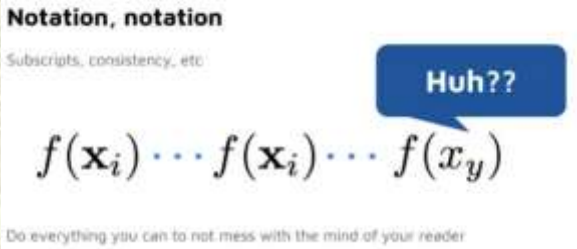


Requirement 2: Easy mapping. It should be easy to map the output of a neural network (at a refined granularity) back to the iSeVCs to pinpoint vulnerabilities. The output should be a sequence of tokens, where one or multiple consecutive tokens correspond to a same line of code in the intermediate code. These lines of intermediate code can be easily mapped back to iSeVCs, and therefore the vulnerable lines of code in source programs.

写作的基本原则

14. 保持一个流畅的故事情节

- 尽早用概念解释符号
- Don't confuse or frustrate your readers, by...
  - Switching context "mid way" / "mid flight"
  - Using undefined notation
  - Changing notation

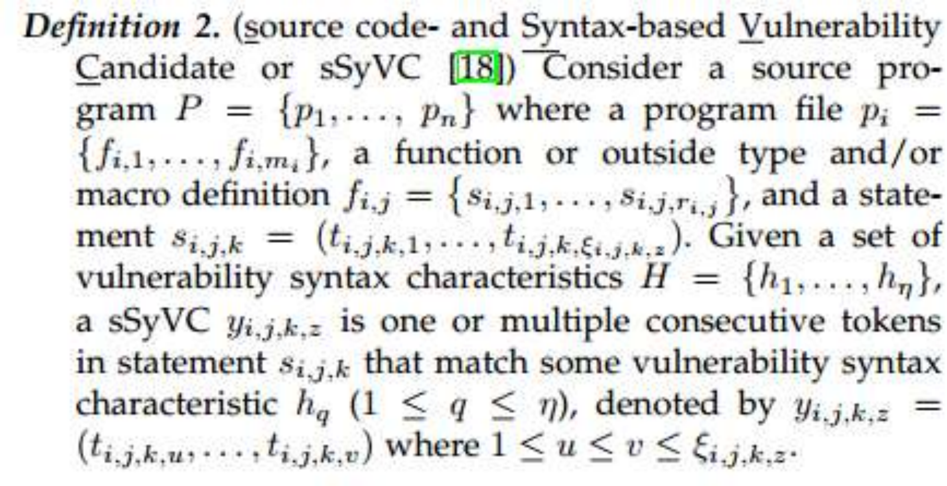


Notation, notation

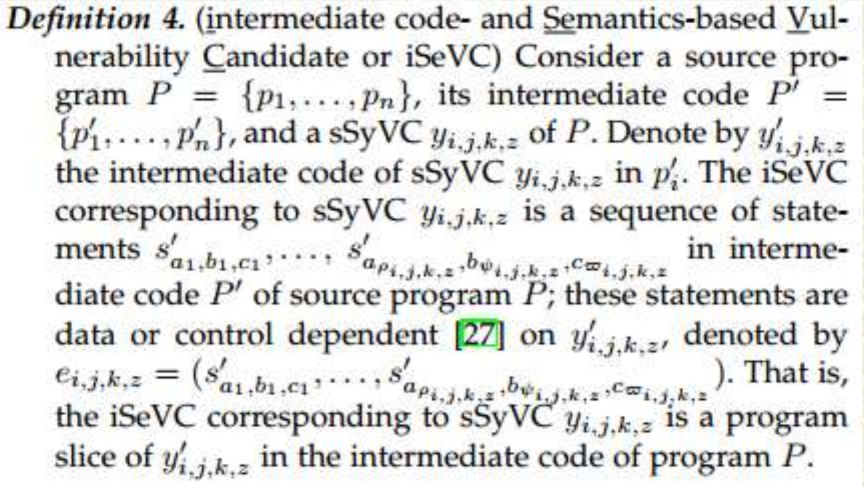
Subscripts, consistency, etc

$$f(\mathbf{x}_i) \cdots f(\mathbf{x}_i) \cdots f(x_y)$$

Do everything you can to not mess with the mind of your reader



***Definition* 2.** (source code- and Syntax-based Vulnerability Candidate or sSyVC [18]) Consider a source program $P = \{p_1, \ldots, p_n\}$ where a program file $p_i = \{f_{i,1}, \ldots, f_{i,m_i}\}$, a function or outside type and/or macro definition $f_{i,j} = \{s_{i,j,1}, \ldots, s_{i,j,r_{i,j}}\}$, and a statement $s_{i,j,k} = (t_{i,j,k,1}, \ldots, t_{i,j,k,\xi_{i,j,k,z}})$. Given a set of vulnerability syntax characteristics $H = \{h_1, \ldots, h_\eta\}$, a sSyVC $y_{i,j,k,z}$ is one or multiple consecutive tokens in statement $s_{i,j,k}$ that match some vulnerability syntax characteristic h_q $(1 \leq q \leq \eta)$, denoted by $y_{i,j,k,z} = (t_{i,j,k,u}, \ldots, t_{i,j,k,v})$ where $1 \leq u \leq v \leq \xi_{i,j,k,z}$.

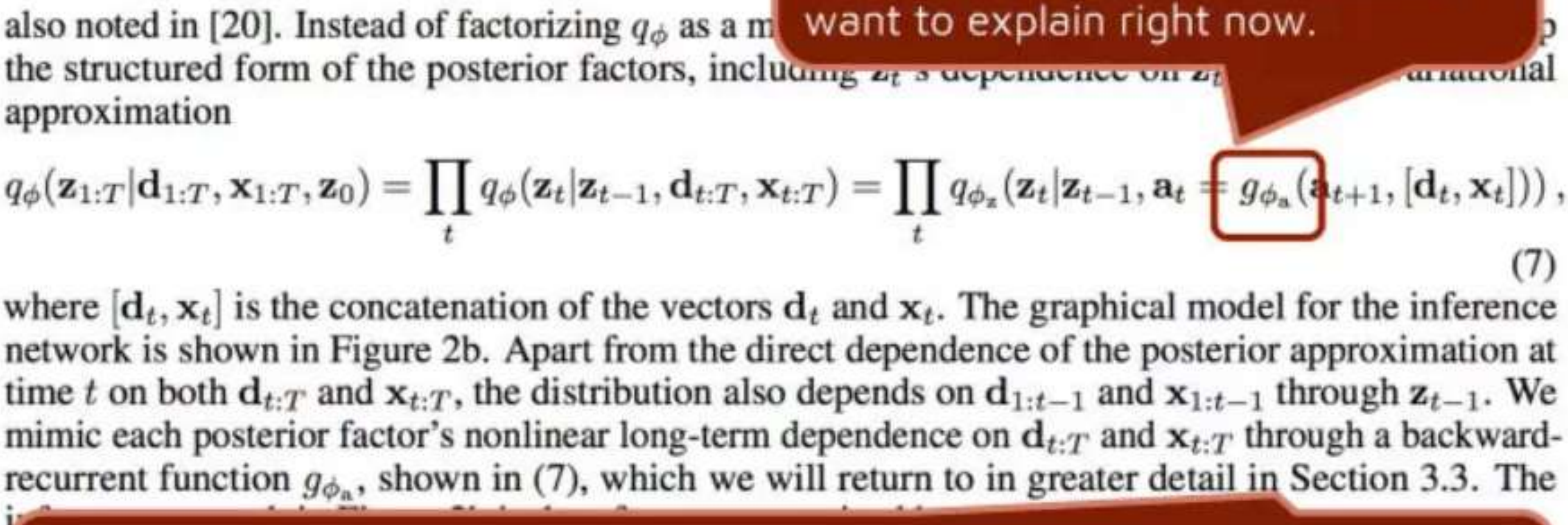


***Definition* 4.** (intermediate code- and Semantics-based Vulnerability Candidate or iSeVC) Consider a source program $P = \{p_1, \ldots, p_n\}$, its intermediate code $P' = \{p'_1, \ldots, p'_n\}$, and a sSyVC $y_{i,j,k,z}$ of P. Denote by $y'_{i,j,k,z}$ the intermediate code of sSyVC $y_{i,j,k,z}$ in p'_i. The iSeVC corresponding to sSyVC $y_{i,j,k,z}$ is a sequence of statements $s'_{a_1,b_1,c_1}, \ldots, s'_{a_{\rho_{i,j,k,z}},b_{\psi_{i,j,k,z}},c_{\varpi_{i,j,k,z}}}$ in intermediate code P' of source program P; these statements are data or control dependent [27] on $y'_{i,j,k,z}$, denoted by $e_{i,j,k,z} = (s'_{a_1,b_1,c_1}, \ldots, s'_{a_{\rho_{i,j,k,z}},b_{\psi_{i,j,k,z}},c_{\varpi_{i,j,k,z}}})$. That is, the iSeVC corresponding to sSyVC $y_{i,j,k,z}$ is a program slice of $y'_{i,j,k,z}$ in the intermediate code of program P.

* These ideas were picked up from Chris Maddison & Danny Tarlow, when writing "A* sampling" (NIPS best paper 2014)

"Top of mind" example

We had to introduce a function and notation here. It's important, but not directly applicable to what we want to explain right now.



also noted in [20]. Instead of factorizing q_ϕ as a m p the structured form of the posterior factors, includ ariational approximation

$$q_\phi(\mathbf{z}_{1:T}|\mathbf{d}_{1:T},\mathbf{x}_{1:T},\mathbf{z}_0) = \prod_t q_\phi(\mathbf{z}_t|\mathbf{z}_{t-1},\mathbf{d}_{t:T},\mathbf{x}_{t:T}) = \prod_t q_{\phi_\mathrm{z}}(\mathbf{z}_t|\mathbf{z}_{t-1},\mathbf{a}_t = \boxed{g_{\phi_\mathrm{a}}}(\mathbf{a}_{t+1},[\mathbf{d}_t,\mathbf{x}_t]))\,, \tag{7}$$

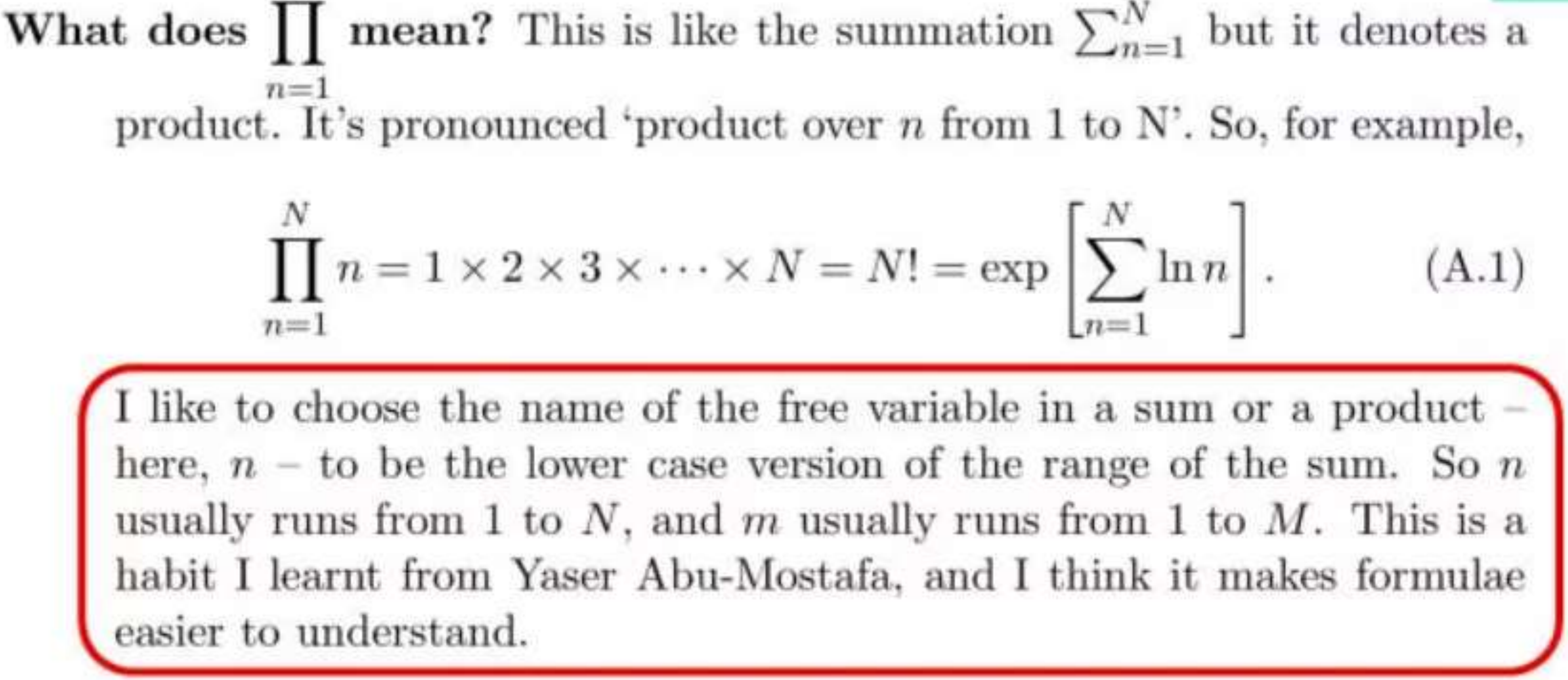
where $[\mathbf{d}_t,\mathbf{x}_t]$ is the concatenation of the vectors $\mathbf{d}_t$ and $\mathbf{x}_t$. The graphical model for the inference network is shown in Figure 2b. Apart from the direct dependence of the posterior approximation at time t on both $\mathbf{d}_{t:T}$ and $\mathbf{x}_{t:T}$, the distribution also depends on $\mathbf{d}_{1:t-1}$ and $\mathbf{x}_{1:t-1}$ through $\mathbf{z}_{t-1}$. We mimic each posterior factor's nonlinear long-term dependence on $\mathbf{d}_{t:T}$ and $\mathbf{x}_{t:T}$ through a backward-recurrent function g_{ϕ_a}, shown in (7), which we will return to in greater detail in Section 3.3. The

We want to keep a **flowing story line**. A reader would be worried that the notation and function is not explained. So, we describe what it is, where it is introduced, and tell the reader *not to worry*: we'll explain it soon :)

写作的基本原则

注意符号的自明性

Notation, notation



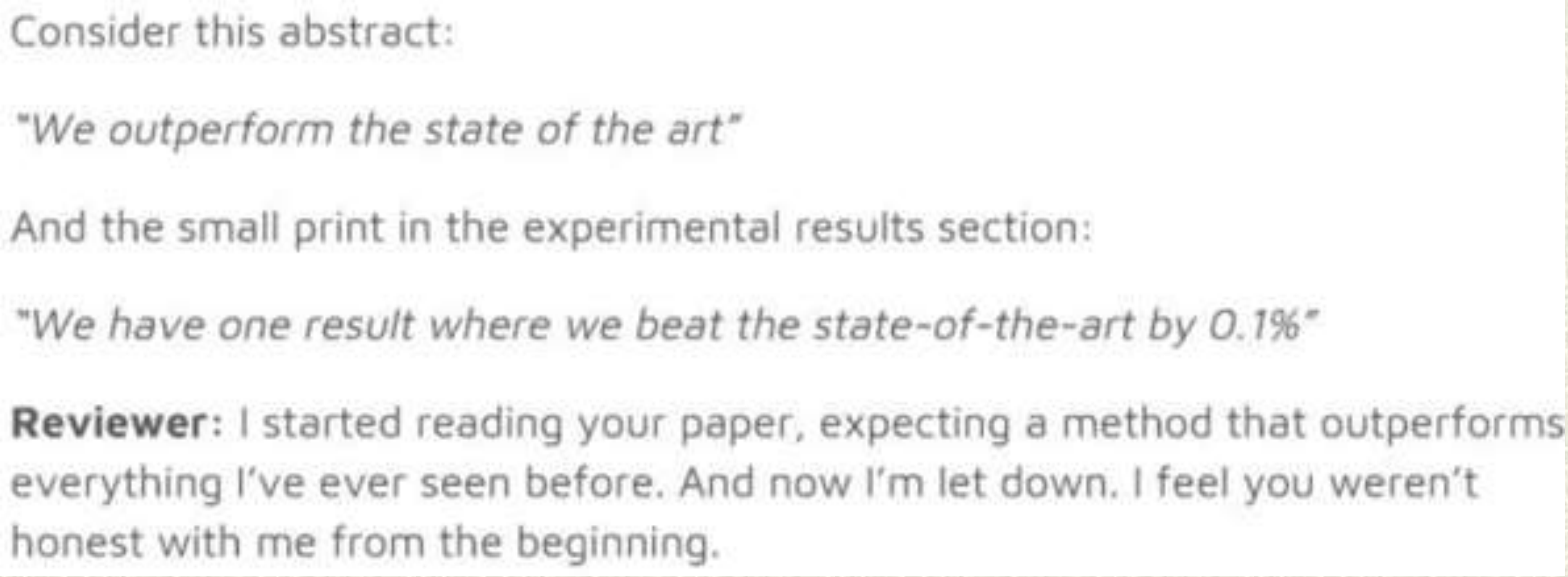
What does $\prod_{n=1}^{N}$ mean? This is like the summation $\sum_{n=1}^{N}$ but it denotes a product. It's pronounced 'product over n from 1 to N'. So, for example,

$$\prod_{n=1}^{N} n = 1 \times 2 \times 3 \times \cdots \times N = N! = \exp\left[\sum_{n=1}^{N} \ln n\right]. \tag{A.1}$$

I like to choose the name of the free variable in a sum or a product – here, n – to be the lower case version of the range of the sum. So n usually runs from 1 to N, and m usually runs from 1 to M. This is a habit I learnt from Yaser Abu-Mostafa, and I think it makes formulae easier to understand.

写作的基本原则

- 15.实事求是，不要夸大和过分吹嘘
- Be academically honest. Don't oversell.



Consider this abstract:

"We outperform the state of the art"

And the small print in the experimental results section:

"We have one result where we beat the state-of-the-art by 0.1%"

Reviewer: I started reading your paper, expecting a method that outperforms everything I've ever seen before. And now I'm let down. I feel you weren't honest with me from the beginning.

写作的基本原则

- 16.信息的排布，最关键的信息要占据C位
  - 段首主题句，是一个段落的C位
  - 主句、主语、句首词，是一句话的 C 位

划分语句的成分，分析清楚语句中需要包含的内容

分析清楚语句的重点

- 哪一部分内容应放在前面?
- 哪一部分内容应放在主句?

重点内容没有前置的例子

语句1：

By combining density functional calculations and DSC experiments, this work provides a

非重点应放置在后面

novel insight into the promotional roles of adsorbed water in low rank coal oxidation.

Density functional calculations reveals that

语句2：

It has been found that a critical moisture content exists at which the rate of oxidation

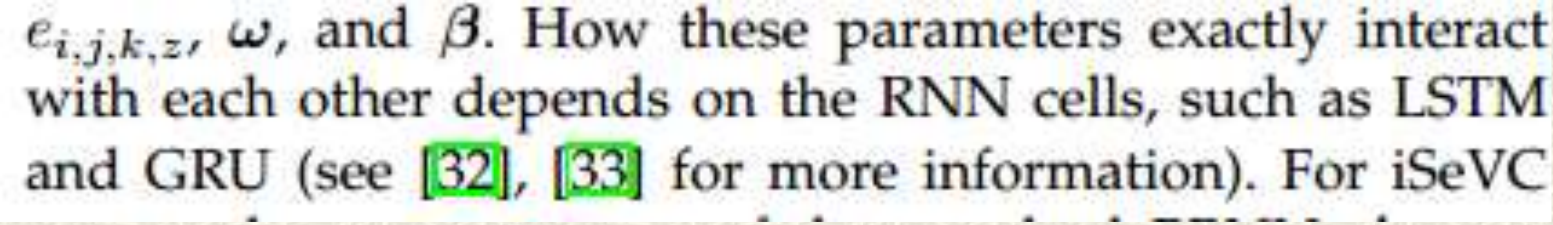
重点信息放在了从句中

reaches a maximum value

重点强调的信息应放置在主句中

写作的基本原则

- 17.不重要的信息，给出引用，详见参考文献



$e_{i,j,k,z}$, ω, and β. How these parameters exactly interact with each other depends on the RNN cells, such as LSTM and GRU (see [32], [33] for more information). For iSeVC

写作的基本原则

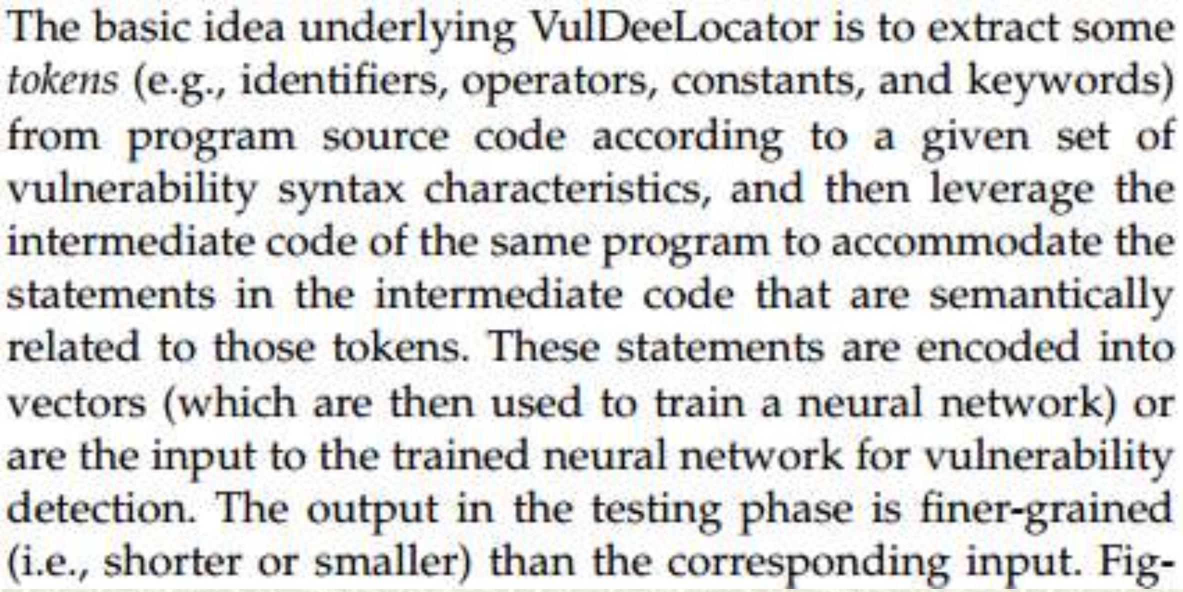
- **18.增大信息密度**
  - 在篇幅有限的情况下，论文能向读者传达多少信息?
  - 实词和其它虚词的比值越大，信息密度大
  - 减少无用的词

写作的基本原则

- **19.期刊和会议限制论文篇幅，隐含着对信息量的要求**
  - 《Chinese physics Letters》的投稿规范
    - 要尽可能多地给出有关研究的信息，尽可能少地运用investigate(调查)， study(研究)， discuss (讨论)等词
      - “The cross section is (6.25±0.02)
      - “The cross section is measured”

写作的基本原则

- 20.用最准确简洁的语言传达你的信息
- Write to discover/understand (for yourself)
  - Be precise in what you are trying to do.
  - Use simple language. If you can't describe your idea in 2-3 simple sentences, maybe you don't understand it that well yourself. Work at it until you can.



The basic idea underlying VulDeeLocator is to extract some *tokens* (e.g., identifiers, operators, constants, and keywords) from program source code according to a given set of vulnerability syntax characteristics, and then leverage the intermediate code of the same program to accommodate the statements in the intermediate code that are semantically related to those tokens. These statements are encoded into vectors (which are then used to train a neural network) or are the input to the trained neural network for vulnerability detection. The output in the testing phase is finer-grained (i.e., shorter or smaller) than the corresponding input. Fig-

简洁的重要性

为什么要简洁?

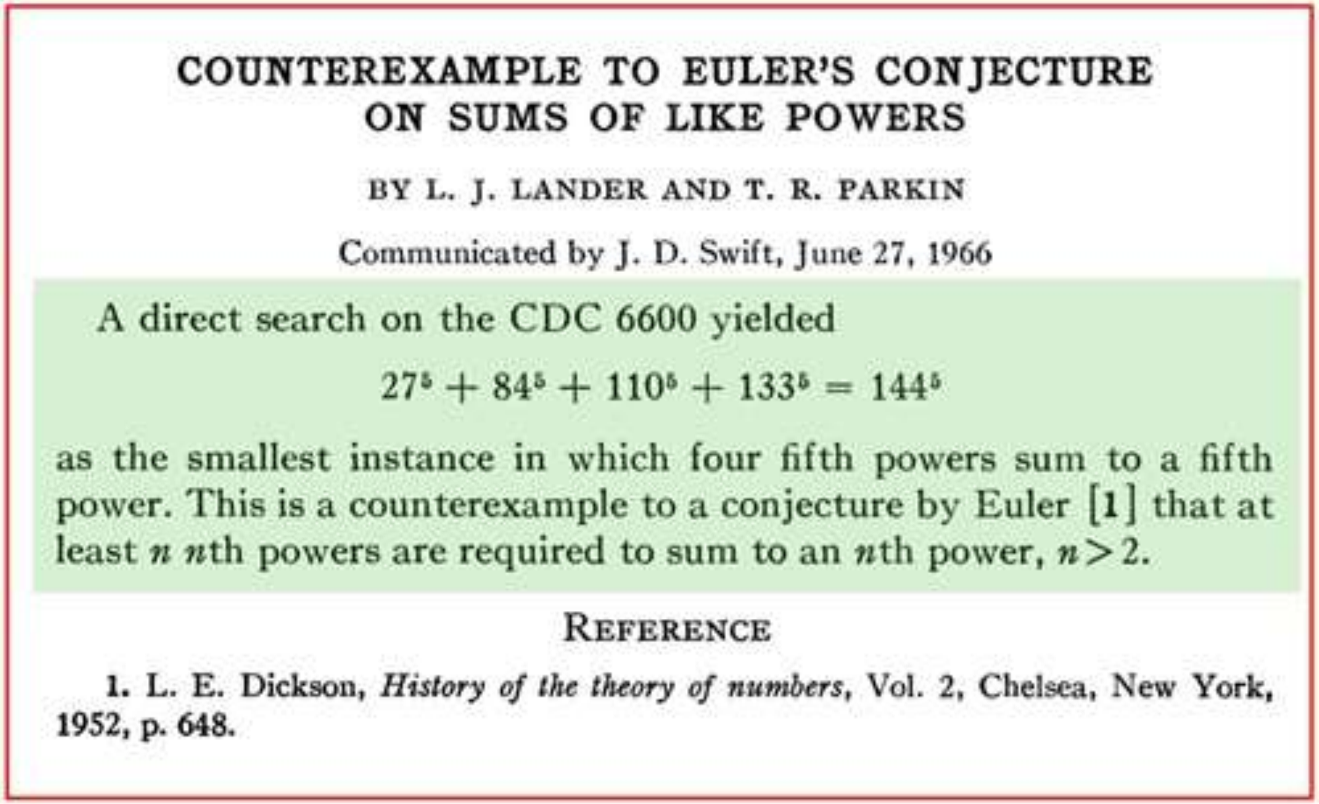
- 审稿人审稿时的心理：快速找到拒稿的证据
- Write to get accepted （for the reviewer)
  - Reviewers are the unpaid, overworked, gate-keepers of science. Don't waste their time.
  - Not all reviewers will be familiar with your work.
  - It is up to you to bring your message across in the clearest way possible!
  - Reviewers usually have less than 1h per paper, sometimes only 30min.
  - They are basically trying to answer the question "How can I justify rejecting this?"

简洁论文的典范

- Watson 与 Crick发现DNA双螺旋结构的论文
  - 发表在《Nature》上
  - 只有约 500 字和一幅 DNA 的双螺旋图
  - 使作者获得了诺贝尔生物医学奖

- Penzias 和 Wilsoh 发现宇宙大爆炸的3K背景辐射的技术观测论文
  - 只有一页篇幅
  - 使作者获得了诺贝尔物理学奖

简洁论文的典范



COUNTEREXAMPLE TO EULER'S CONJECTURE ON SUMS OF LIKE POWERS

BY L. J. LANDER AND T. R. PARKIN

Communicated by J. D. Swift, June 27, 1966

A direct search on the CDC 6600 yielded

$$27^5 + 84^5 + 110^5 + 133^5 = 144^5$$

as the smallest instance in which four fifth powers sum to a fifth power. This is a counterexample to a conjecture by Euler [1] that at least n nth powers are required to sum to an nth power, $n > 2$.

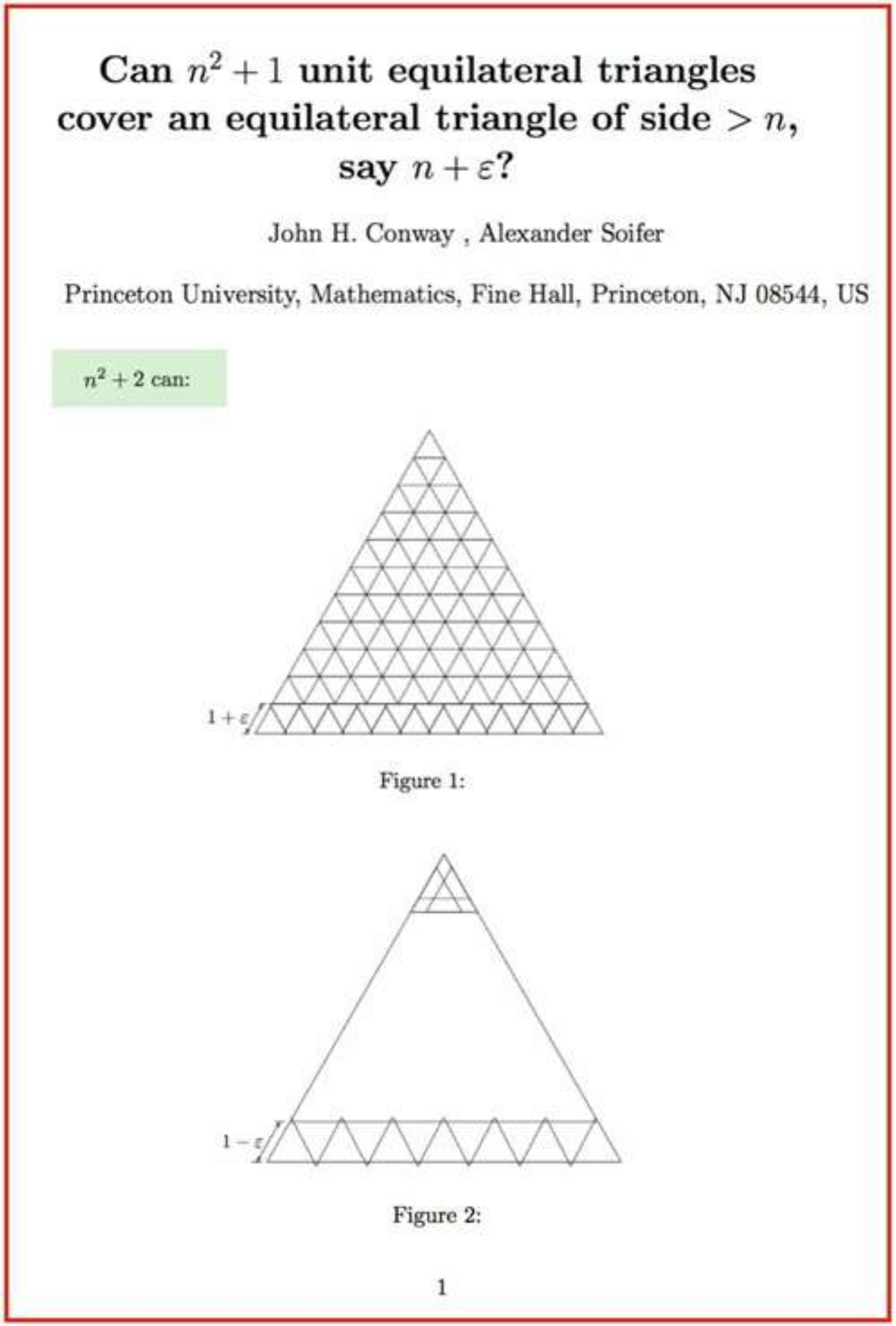
REFERENCE

1. L. E. Dickson, *History of the theory of numbers*, Vol. 2, Chelsea, New York, 1952, p. 648.

- 1966年，Lander 和 Parkin 写了一篇关于欧拉猜想（跟费马大定理相关）的论文，只用了两句话
- 1995年怀尔斯证明了费马大定理的论文足足有108页长

最正文

- 有史以来最短的数学论文
- n^2+ 1个单位正三角形可以覆盖边长 > n，例如 n+ε，的正三角形吗?
- 正文内容**只用两个字和两幅图**，便完成了他们所需要的证明



Can n^2+1 unit equilateral triangles cover an equilateral triangle of side $> n$, say $n+\varepsilon$?

John H. Conway , Alexander Soifer

Princeton University, Mathematics, Fine Hall, Princeton, NJ 08544, US

n^2+2 can:

$1+\varepsilon$

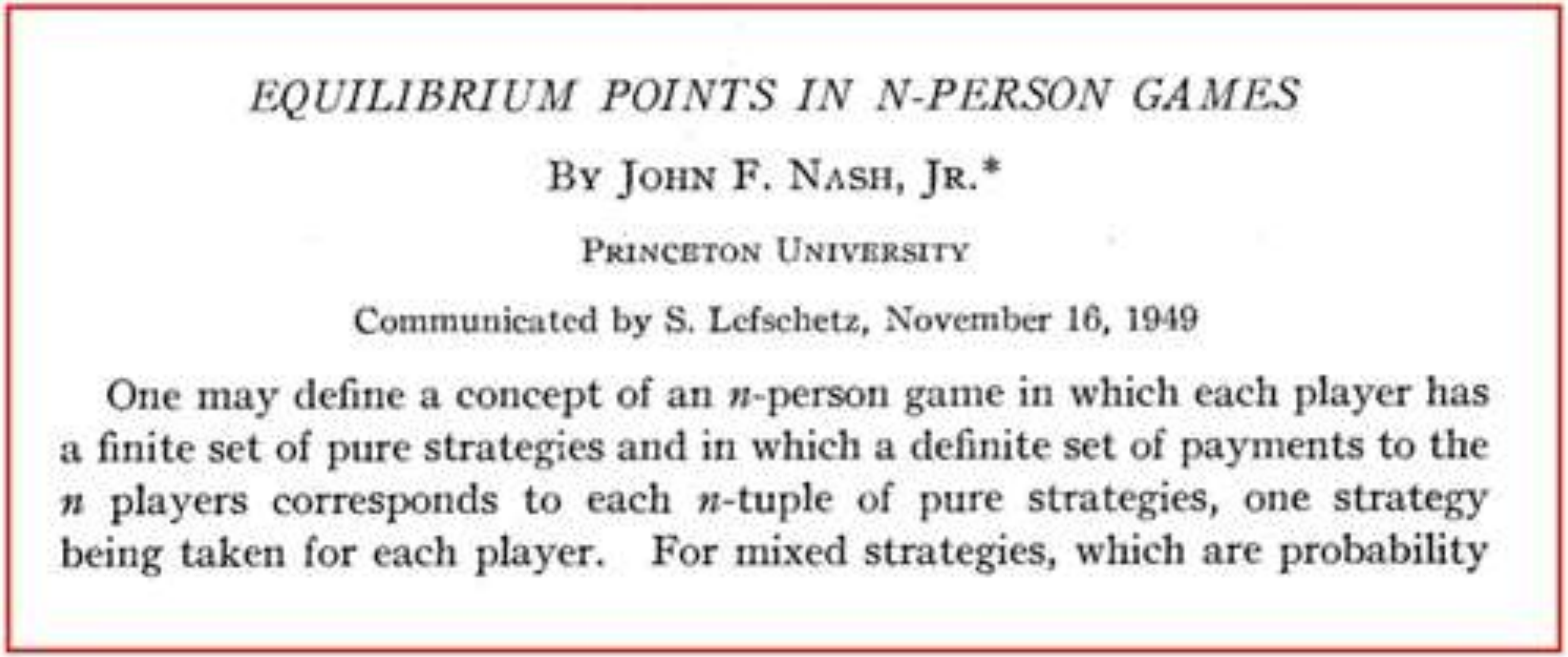
Figure 1:

$1-\varepsilon$

Figure 2:

1

史上最伟大的论文



EQUILIBRIUM POINTS IN N-PERSON GAMES

By John F. Nash, Jr.*

Princeton University

Communicated by S. Lefschetz, November 16, 1949

One may define a concept of an *n*-person game in which each player has a finite set of pure strategies and in which a definite set of payments to the *n* players corresponds to each *n*-tuple of pure strategies, one strategy being taken for each player. For mixed strategies, which are probability

- 21岁的普林斯顿在读研究生**约翰•纳什**，在1950年1月发表的**《N人博弈中的均衡点》**
- **333个字，概述了博弈论的基础原理：纳什均衡**
- 使其获得了诺贝尔经济学奖
- 其博士论文也是出了名的短，只有26页

实验和讨论

（ Experiments and Discussion ）

实验和讨论，写什么?

- Experiments and Discussion
  - 你发现了什么？如何解释你所获的结果?
  - 验证提出的方法和思路
    - a)实验设计
    - b)结果比较
    - c)分析讨论
    - d)实验结论

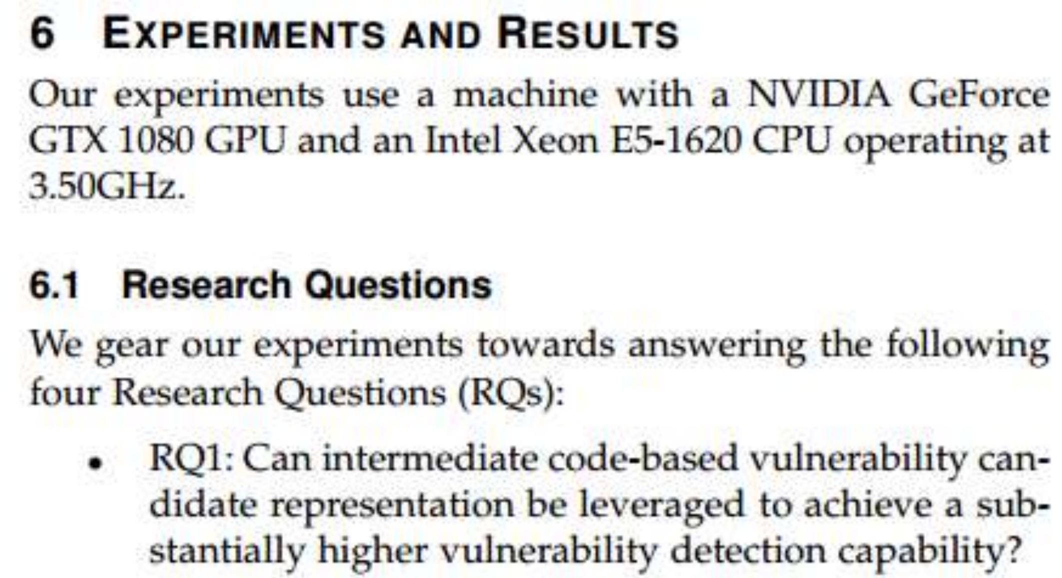
实验

实验部分写什么?

- Experiments and Results
  - Purpose
    - Experimental support to your work
  - Style
    - Experimental design
    - Evaluation Metrics
    - Be sure that other researchers can repeat your experiments according to your descriptions
    - Ensure that the results answer the main study question
    - Compare the results with other studies
    - Performance Analysis
    - What is revealed by the experiments?

实验部分的写作原则

⌗ 1. 读者思维



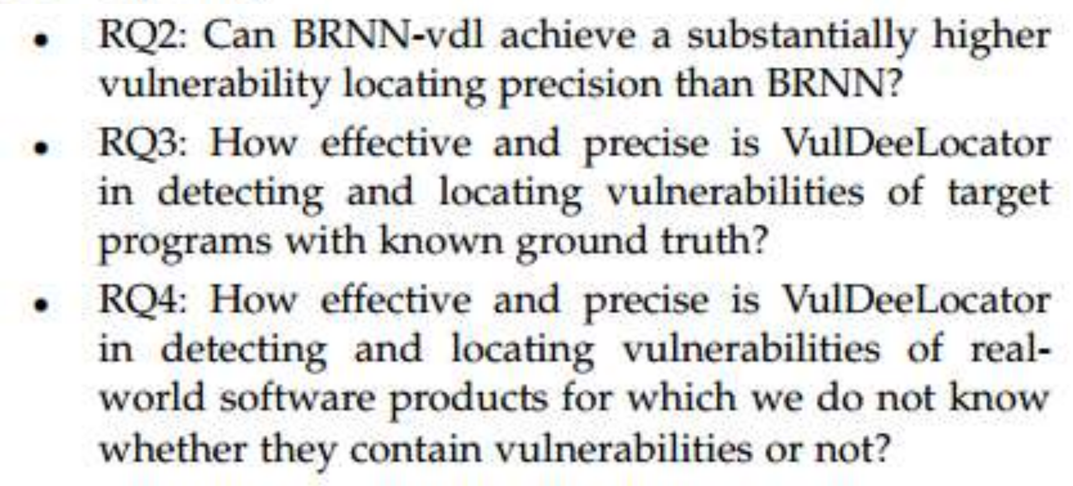
6 Experiments and Results

Our experiments use a machine with a NVIDIA GeForce GTX 1080 GPU and an Intel Xeon E5-1620 CPU operating at 3.50GHz.

6.1 Research Questions

We gear our experiments towards answering the following four Research Questions (RQs):

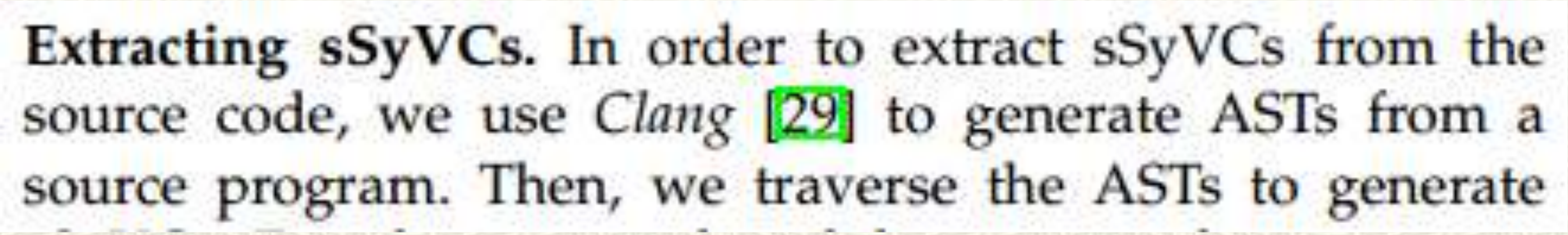
- RQ1: Can intermediate code-based vulnerability candidate representation be leveraged to achieve a substantially higher vulnerability detection capability?



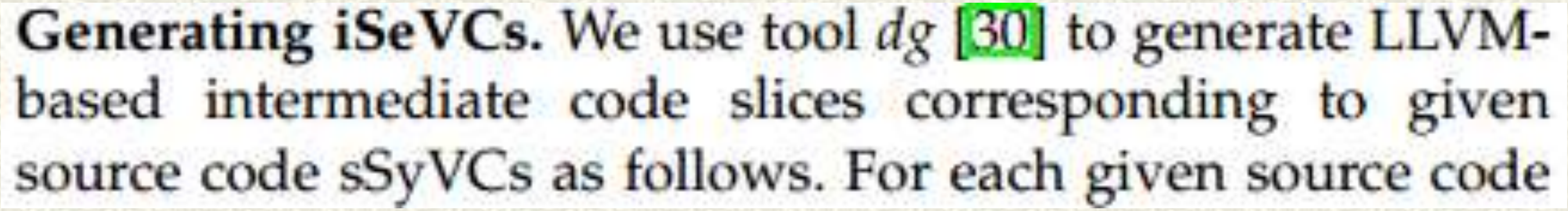
- RQ2: Can BRNN-vdl achieve a substantially higher vulnerability locating precision than BRNN?
- RQ3: How effective and precise is VulDeeLocator in detecting and locating vulnerabilities of target programs with known ground truth?
- RQ4: How effective and precise is VulDeeLocator in detecting and locating vulnerabilities of real-world software products for which we do not know whether they contain vulnerabilities or not?

实验部分的写作原则

- 2.实验方法的描述要具体，真实
  - 可以见参考文献



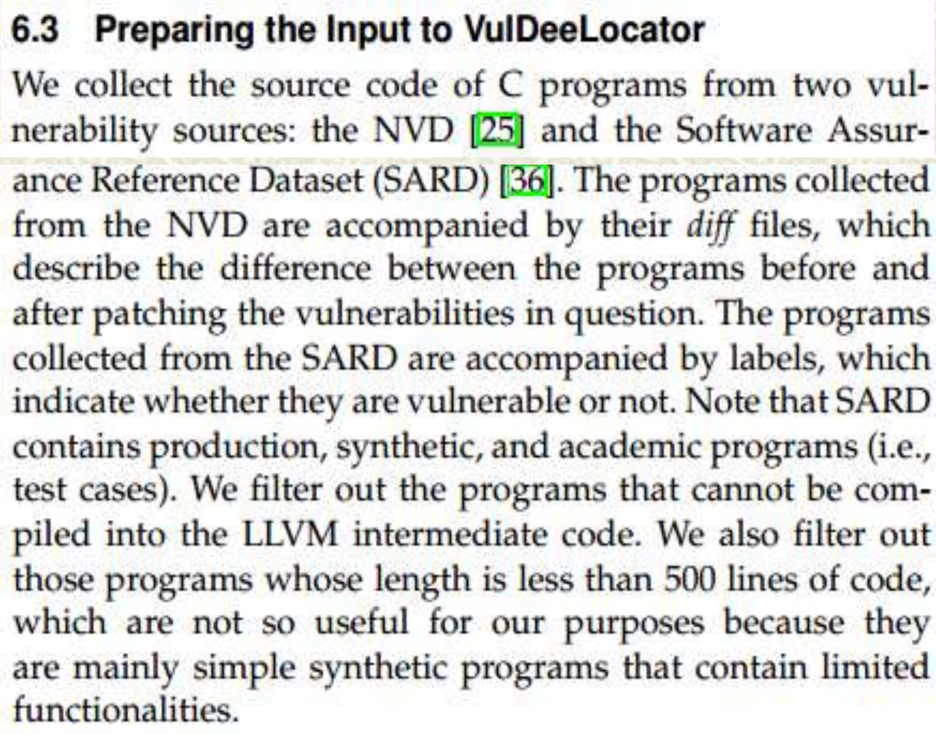
Extracting sSyVCs. In order to extract sSyVCs from the source code, we use *Clang* [29] to generate ASTs from a source program. Then, we traverse the ASTs to generate



Generating iSeVCs. We use tool *dg* [30] to generate LLVM-based intermediate code slices corresponding to given source code sSyVCs as follows. For each given source code

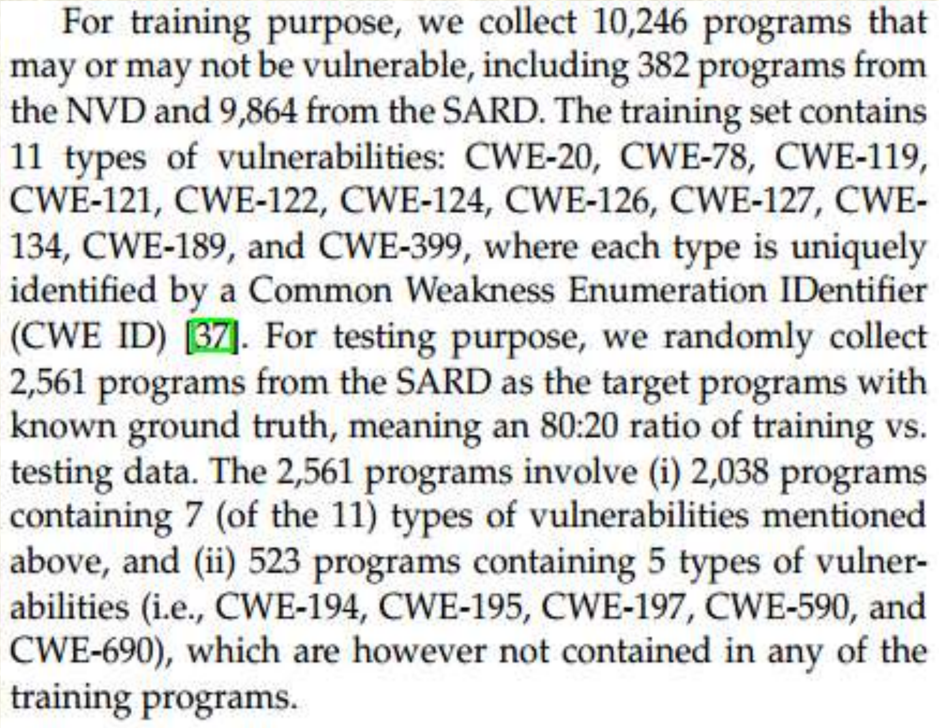
实验部分的写作原则

- 3.作者自行设计和创造的新方法，应详细描述



6.3 Preparing the Input to VulDeeLocator

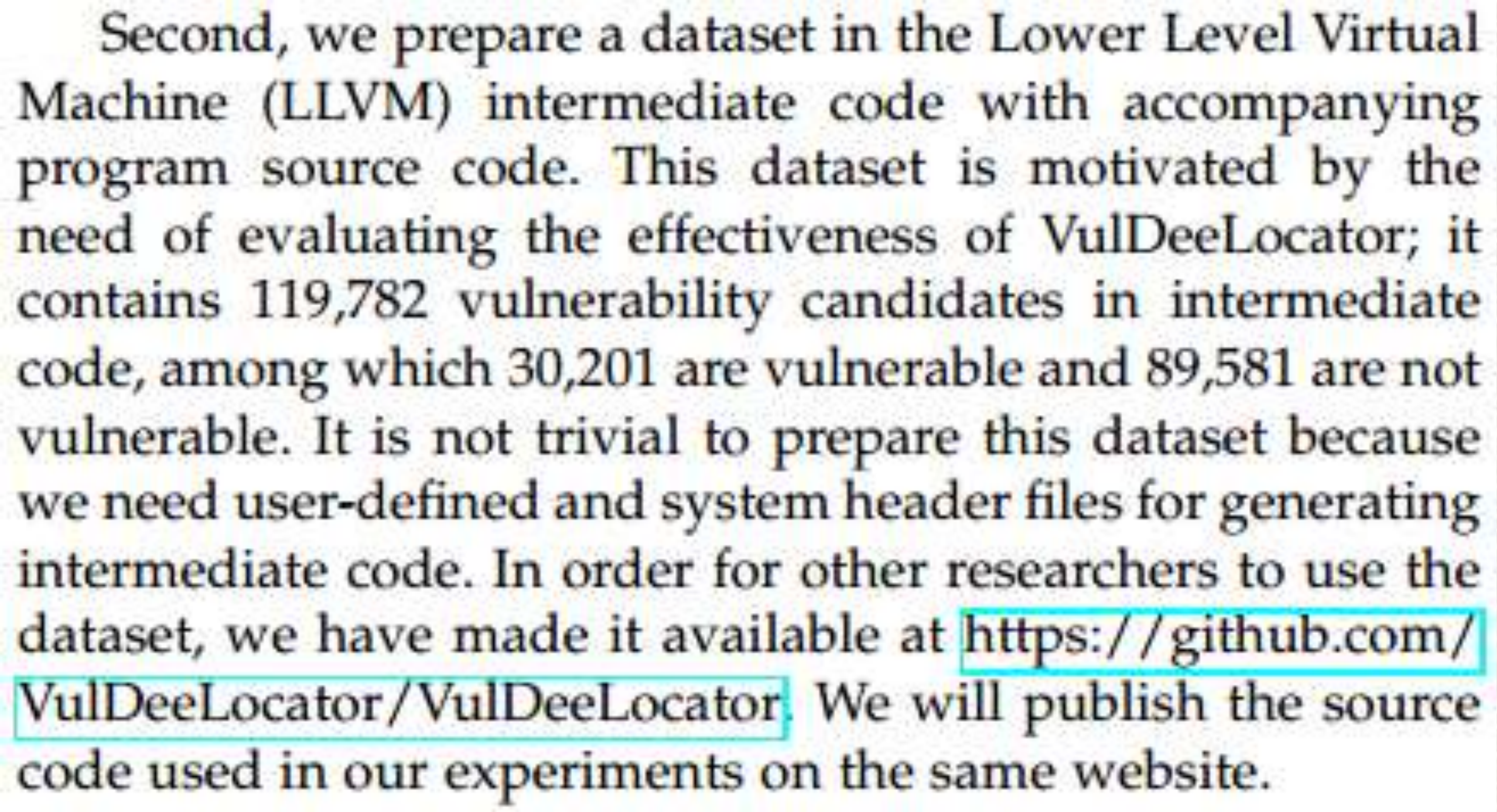
We collect the source code of C programs from two vulnerability sources: the NVD [25] and the Software Assurance Reference Dataset (SARD) [36]. The programs collected from the NVD are accompanied by their *diff* files, which describe the difference between the programs before and after patching the vulnerabilities in question. The programs collected from the SARD are accompanied by labels, which indicate whether they are vulnerable or not. Note that SARD contains production, synthetic, and academic programs (i.e., test cases). We filter out the programs that cannot be compiled into the LLVM intermediate code. We also filter out those programs whose length is less than 500 lines of code, which are not so useful for our purposes because they are mainly simple synthetic programs that contain limited functionalities.



For training purpose, we collect 10,246 programs that may or may not be vulnerable, including 382 programs from the NVD and 9,864 from the SARD. The training set contains 11 types of vulnerabilities: CWE-20, CWE-78, CWE-119, CWE-121, CWE-122, CWE-124, CWE-126, CWE-127, CWE-134, CWE-189, and CWE-399, where each type is uniquely identified by a Common Weakness Enumeration IDentifier (CWE ID) [37]. For testing purpose, we randomly collect 2,561 programs from the SARD as the target programs with known ground truth, meaning an 80:20 ratio of training vs. testing data. The 2,561 programs involve (i) 2,038 programs containing 7 (of the 11) types of vulnerabilities mentioned above, and (ii) 523 programs containing 5 types of vulnerabilities (i.e., CWE-194, CWE-195, CWE-197, CWE-590, and CWE-690), which are however not contained in any of the training programs.

实验部分的写作原则

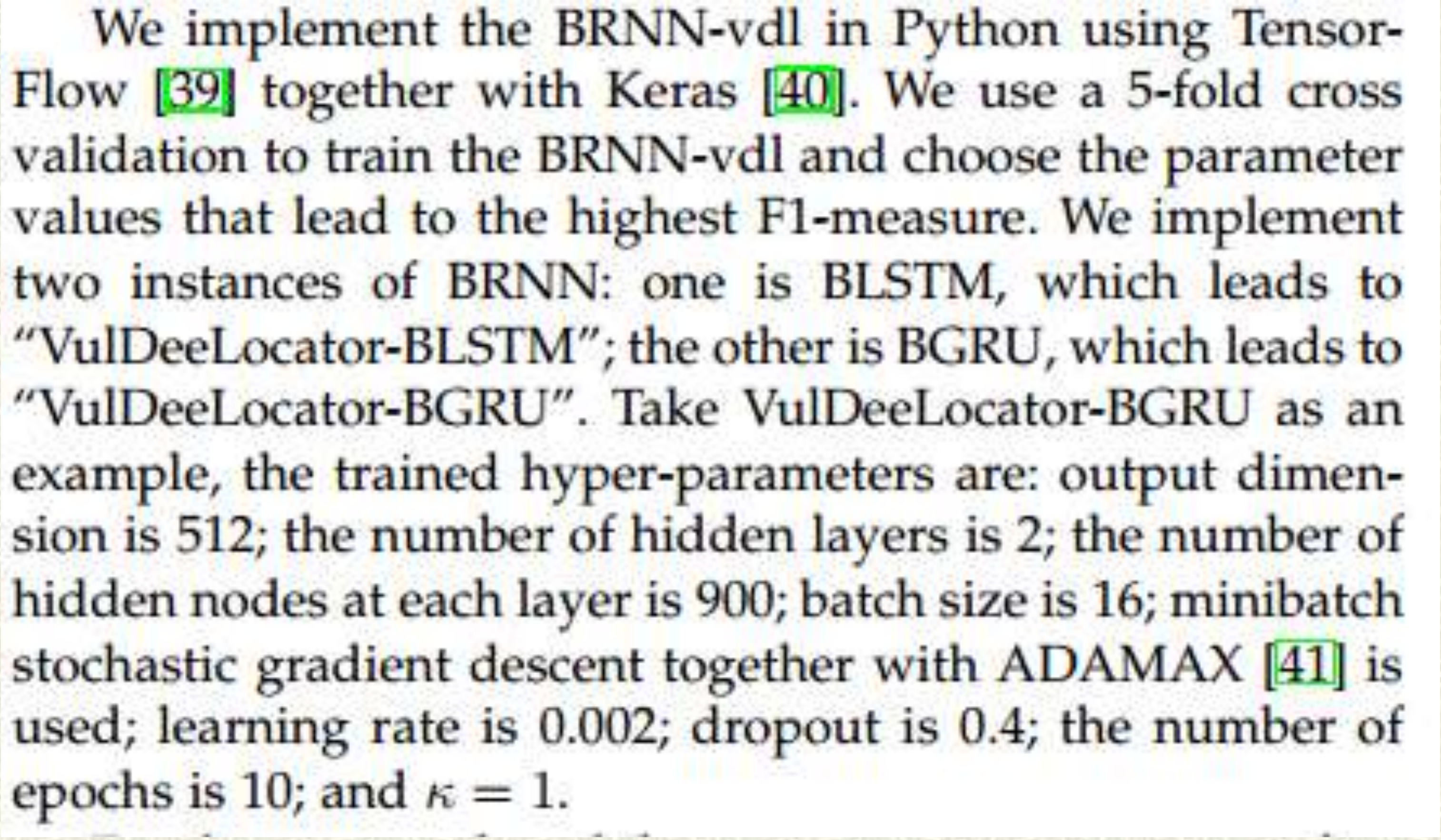
- **4.公开数据集和代码，确保实验可复现**



Second, we prepare a dataset in the Lower Level Virtual Machine (LLVM) intermediate code with accompanying program source code. This dataset is motivated by the need of evaluating the effectiveness of VulDeeLocator; it contains 119,782 vulnerability candidates in intermediate code, among which 30,201 are vulnerable and 89,581 are not vulnerable. It is not trivial to prepare this dataset because we need user-defined and system header files for generating intermediate code. In order for other researchers to use the dataset, we have made it available at https://github.com/VulDeeLocator/VulDeeLocator. We will publish the source code used in our experiments on the same website.

实验部分的写作原则

5.给出实验细节和参数设置，确保实验可复现



We implement the BRNN-vdl in Python using TensorFlow [39] together with Keras [40]. We use a 5-fold cross validation to train the BRNN-vdl and choose the parameter values that lead to the highest F1-measure. We implement two instances of BRNN: one is BLSTM, which leads to "VulDeeLocator-BLSTM"; the other is BGRU, which leads to "VulDeeLocator-BGRU". Take VulDeeLocator-BGRU as an example, the trained hyper-parameters are: output dimension is 512; the number of hidden layers is 2; the number of hidden nodes at each layer is 900; batch size is 16; minibatch stochastic gradient descent together with ADAMAX [41] is used; learning rate is 0.002; dropout is 0.4; the number of epochs is 10; and $\kappa = 1$.

实验部分的写作原则

⌗ 6.实验结果的展示

- 以文字、图表等形式，表达与论文有关的实验数据和结果

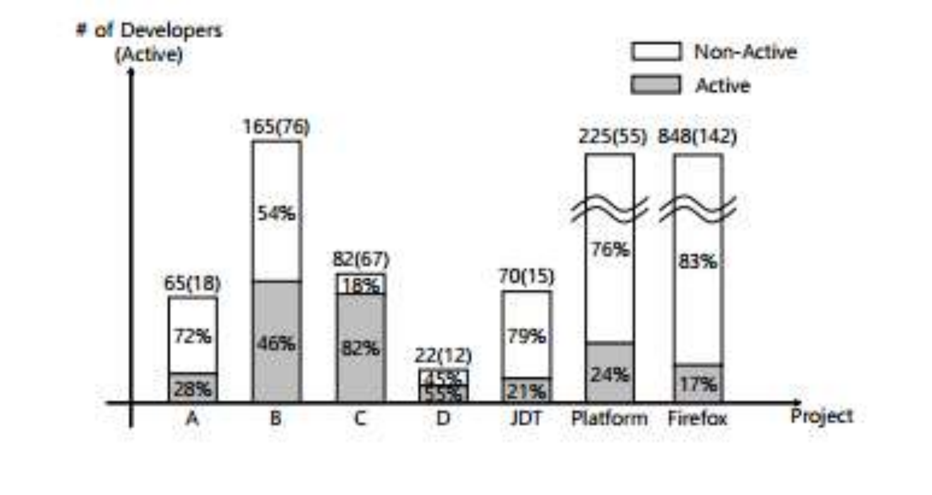


Figure 2: Percentage of active developers in each project

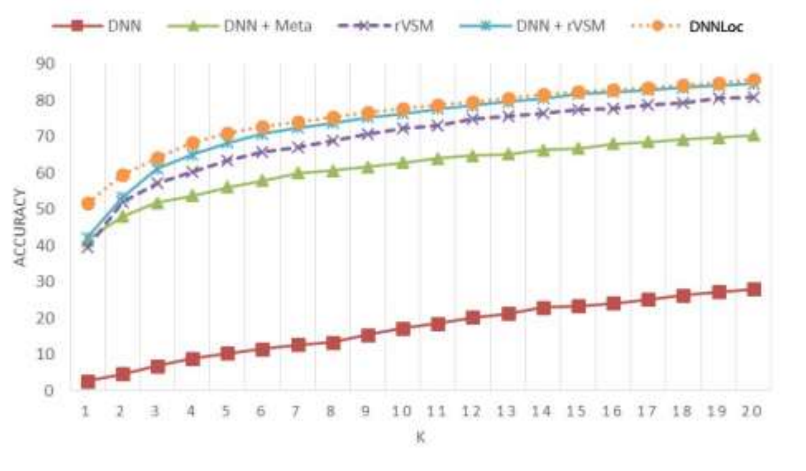


Fig. 7: Top-*k* Accuracy with Different Components

实验部分的写作原则

7.图表要有自明性，相对独立

- Figures and their captions are the first thing the reader will see!
- Make them self-contained, with extremely concise and clear captions, saying what they mean and their conclusion.
  - When there's a paper you like, take literally notes, and try to understand why you liked reading it!

具有自明性的图例

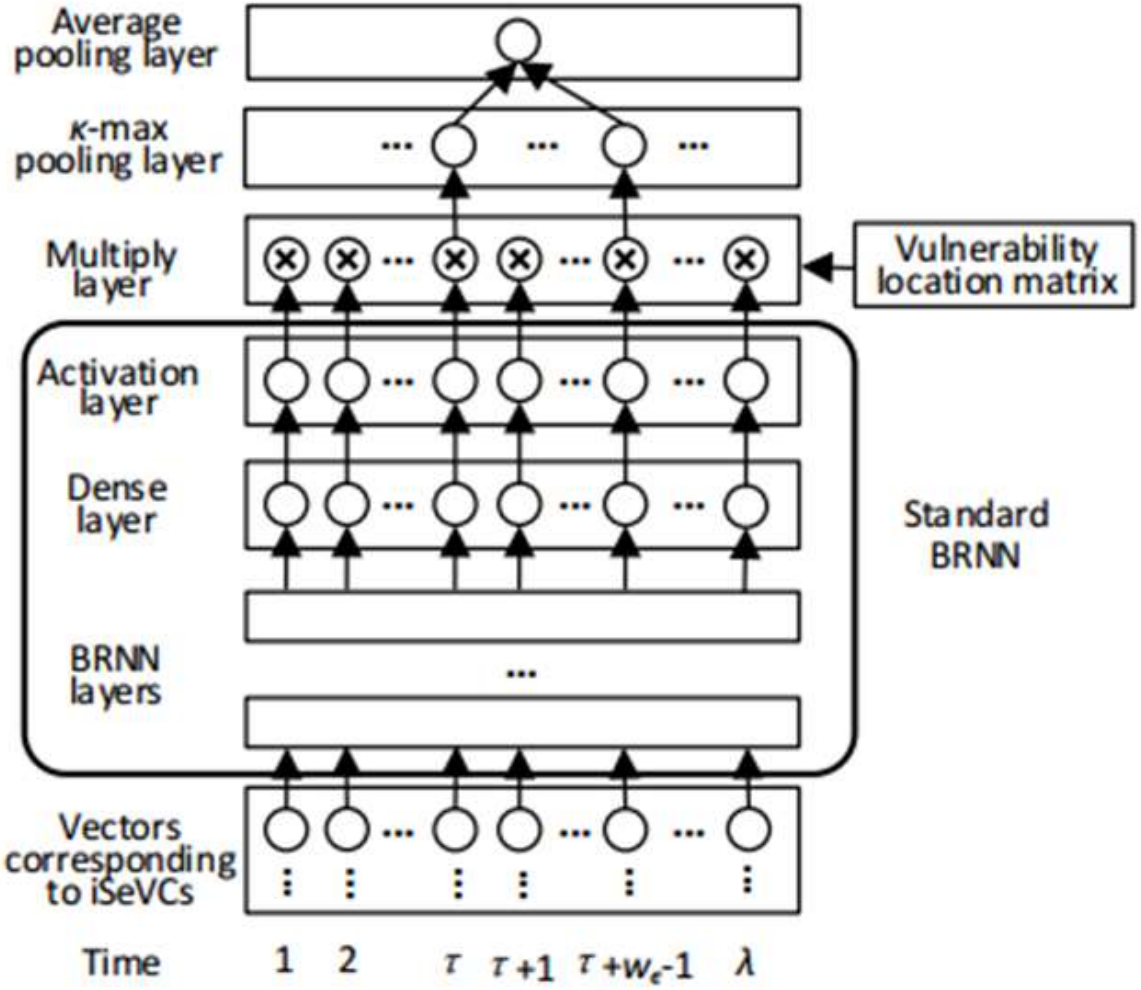


Fig. 5. BRNN-vdl extends BRNN with three extra layers (i.e., the *multiply*, *κ-max pooling*, and *average pooling* layers) that formulate the "vdl" part to achieve three desired properties.

具有自明性的图例

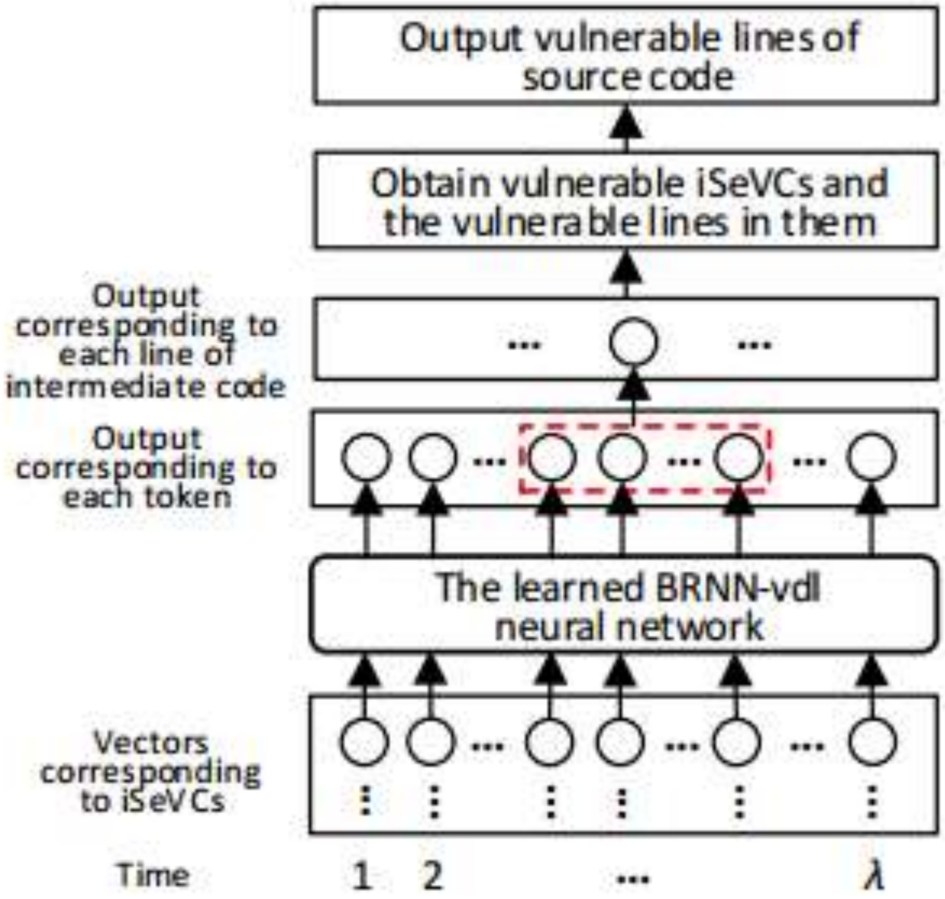

Output vulnerable lines of source code
Obtain vulnerable iSeVCs and the vulnerable lines in them
Output corresponding to each line of intermediate code
Output corresponding to each token
The learned BRNN-vdl neural network
Vectors corresponding to iSeVCs
Time
1
2
...
λ

Fig. 6. Using the learned BRNN-vdl to detect vulnerabilities in target programs, where the dashed box highlights the tokens extracted from a line of code.

具有自明性的表例

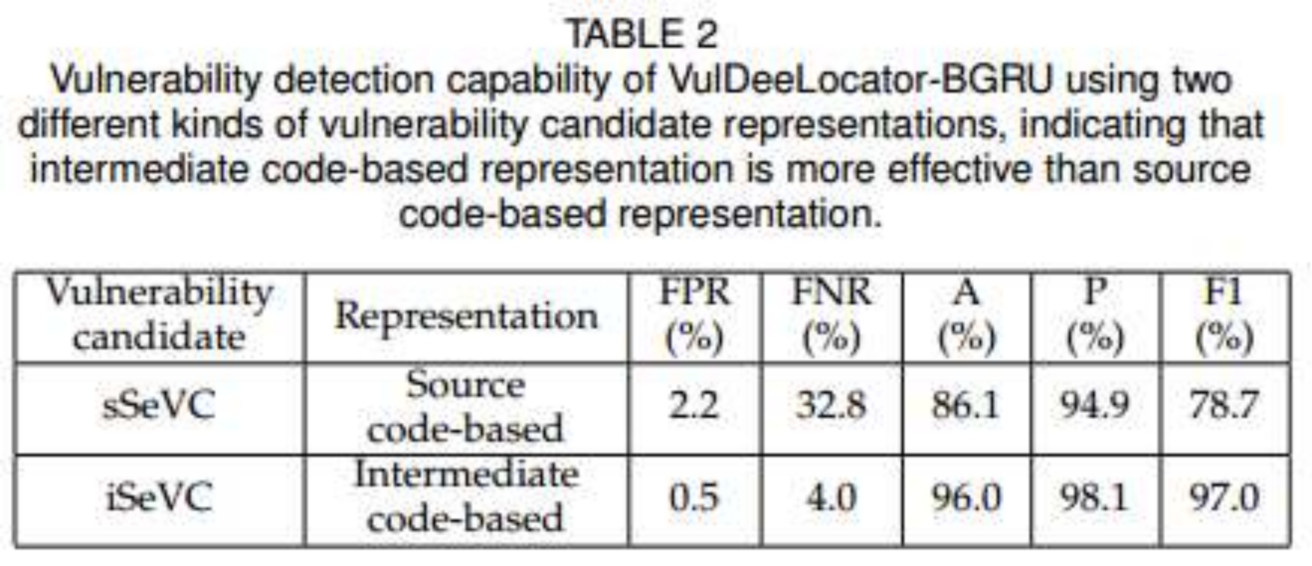


TABLE 2
Vulnerability detection capability of VulDeeLocator-BGRU using two different kinds of vulnerability candidate representations, indicating that intermediate code-based representation is more effective than source code-based representation.

Vulnerability candidate	Representation	FPR (%)	FNR (%)	A (%)	P (%)	F1 (%)
sSeVC	Source code-based	2.2	32.8	86.1	94.9	78.7
iSeVC	Intermediate code-based	0.5	4.0	96.0	98.1	97.0

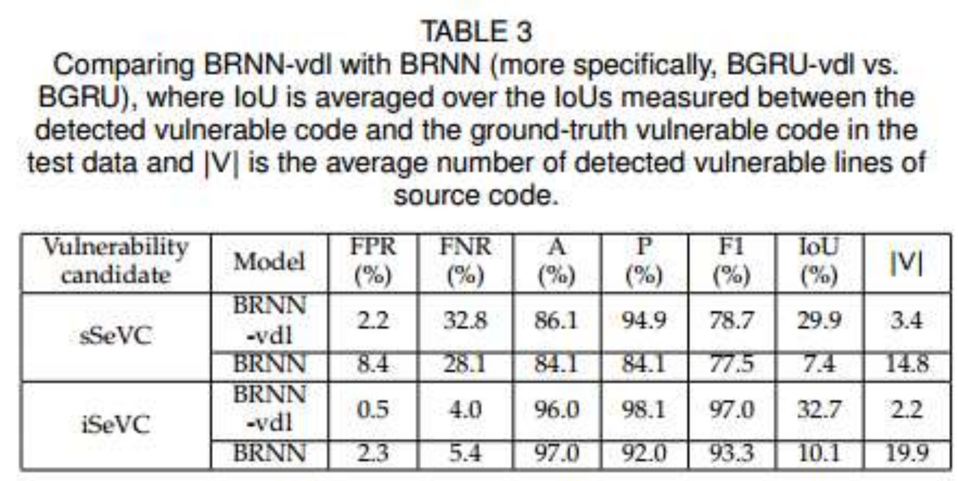


TABLE 3
Comparing BRNN-vdl with BRNN (more specifically, BGRU-vdl vs. BGRU), where IoU is averaged over the IoUs measured between the detected vulnerable code and the ground-truth vulnerable code in the test data and |V| is the average number of detected vulnerable lines of source code.

Vulnerability candidate	Model	FPR (%)	FNR (%)	A (%)	P (%)	F1 (%)	IoU (%)	|V|
sSeVC	BRNN -vdl	2.2	32.8	86.1	94.9	78.7	29.9	3.4
	BRNN	8.4	28.1	84.1	84.1	77.5	7.4	14.8
iSeVC	BRNN -vdl	0.5	4.0	96.0	98.1	97.0	32.7	2.2
	BRNN	2.3	5.4	97.0	92.0	93.3	10.1	19.9

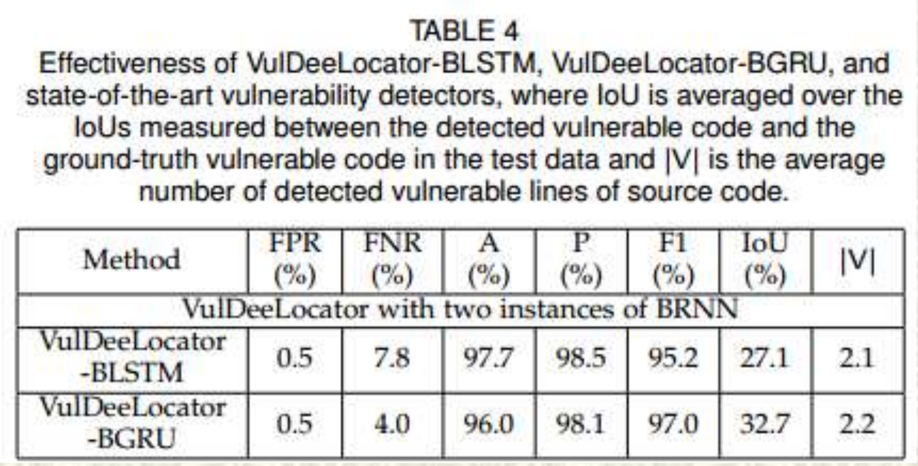


TABLE 4
Effectiveness of VulDeeLocator-BLSTM, VulDeeLocator-BGRU, and state-of-the-art vulnerability detectors, where IoU is averaged over the IoUs measured between the detected vulnerable code and the ground-truth vulnerable code in the test data and |V| is the average number of detected vulnerable lines of source code.

Method	FPR (%)	FNR (%)	A (%)	P (%)	F1 (%)	IoU (%)	|V|
VulDeeLocator with two instances of BRNN							
VulDeeLocator -BLSTM	0.5	7.8	97.7	98.5	95.2	27.1	2.1
VulDeeLocator -BGRU	0.5	4.0	96.0	98.1	97.0	32.7	2.2

具有自明性的公式

- 一图值千言
  - 善于利用图表来生动阐述学术内容
  - 避免过多的文字说明，且效果也更好

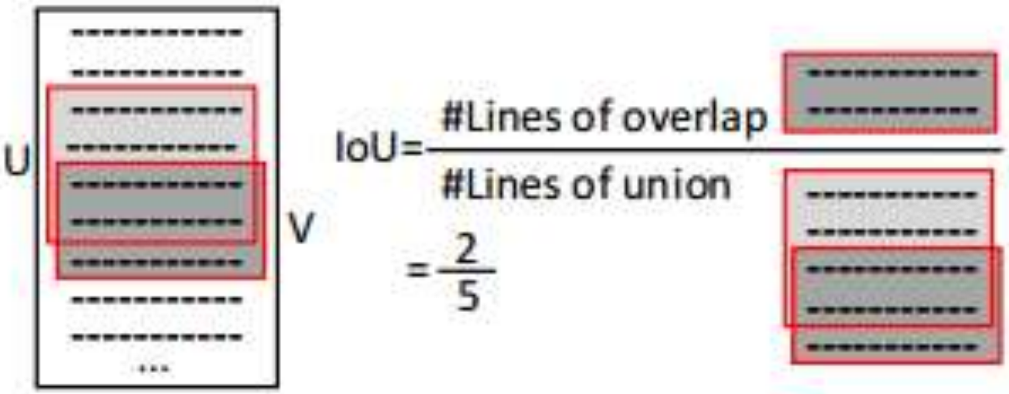
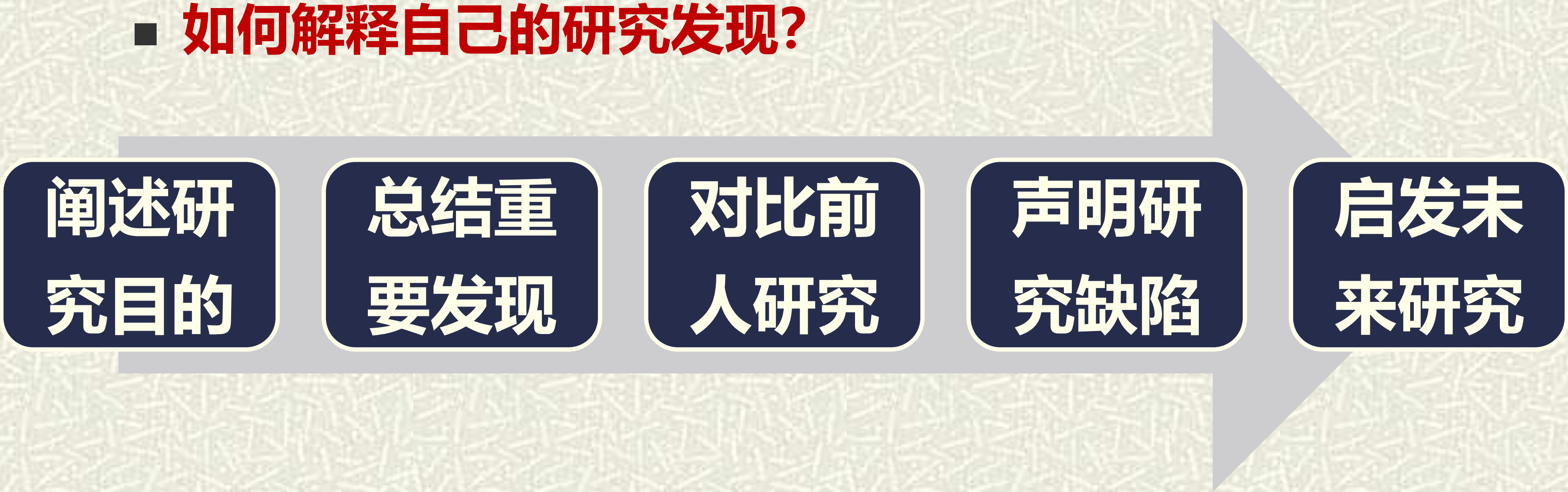


Fig. 7. An example illustrating the meaning of IoU, where a dashed line represents a program statement.

讨论

讨论部分写什么?

- 围绕回答下面几个问题来展开：
  - data，result，finding，逐层深入
  - 与别人比有什么优劣？优劣的原因是什么？
  - 为什么会出现这样的结果？出现这样的结果意味着什么？说明了什么？
  - 如何解释自己的研究发现？



阐述研究目的

总结重要发现

对比前人研究

声明研究缺陷

启发未来研究

讨论的写作范式

- Patterns
  - Cause/effect（因果)
  - Comparison/contrast（对比)
  - Concentration pattern（集中)
  - Parallel pattern（平行)
  - Progressive pattern（渐进)

讨论部分的写作范式（英）

- Discussion
  - Purpose
    - The relationship between your work and some very related works
  - Style
    - Work A
      - Why it is very related
      - Difference to your work
    - Work B
      - Why it is very related
      - Difference to your work
    - …

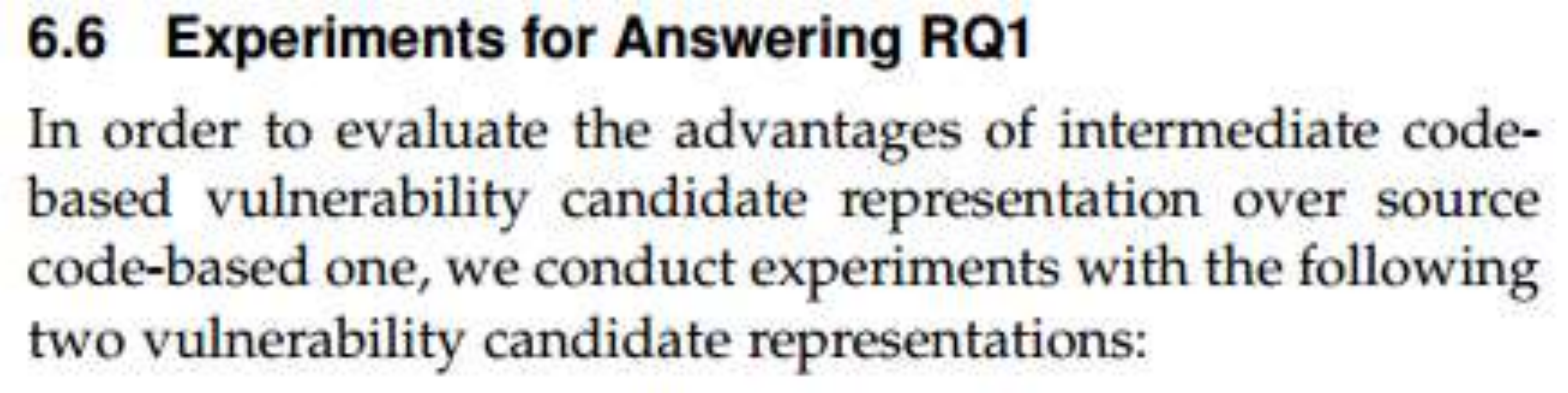
- Threats To Validity （limitations）

讨论部分的写作范式（中）

- 对实验结果的解释，以现有的证据为依据加以说明
  - 第一段
    - 用2-3句话，归纳主要实验结果
  - 随后各段
    - 详细说明与现有文献的相似之处和不同之处
    - 结果分析
  - 接着在另一段中
    - 讨论优缺点
  - 最后
    - 以不超过一到两句话的结论作为结尾

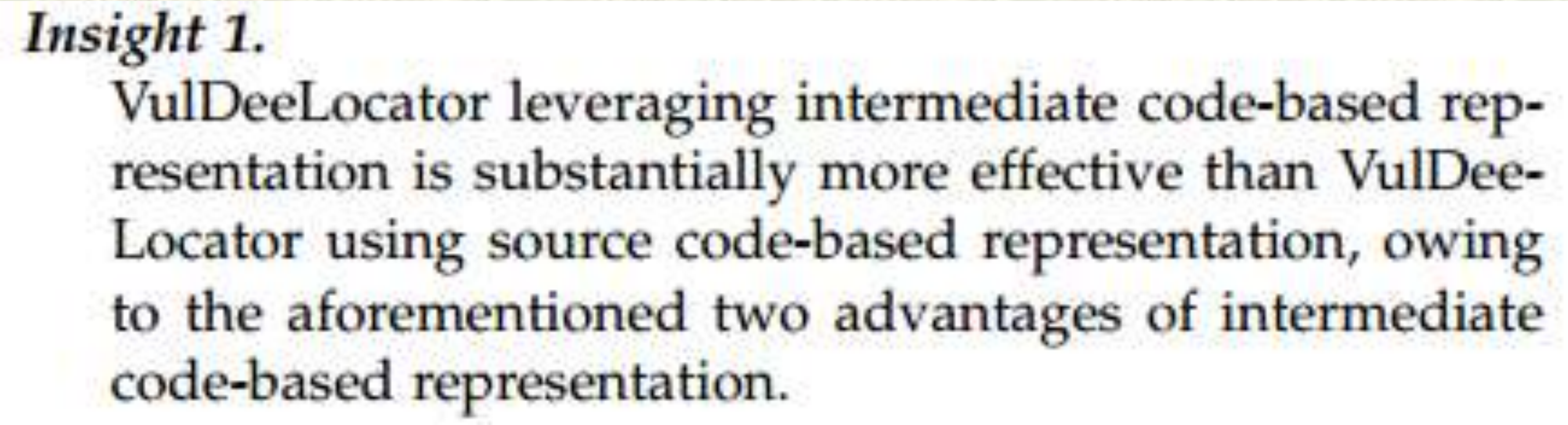
讨论部分的写作原则

- **1. 分小标题，围绕需要回答的问题（RQs）展开讨论，并逐条给出洞察结论**



6.6 Experiments for Answering RQ1

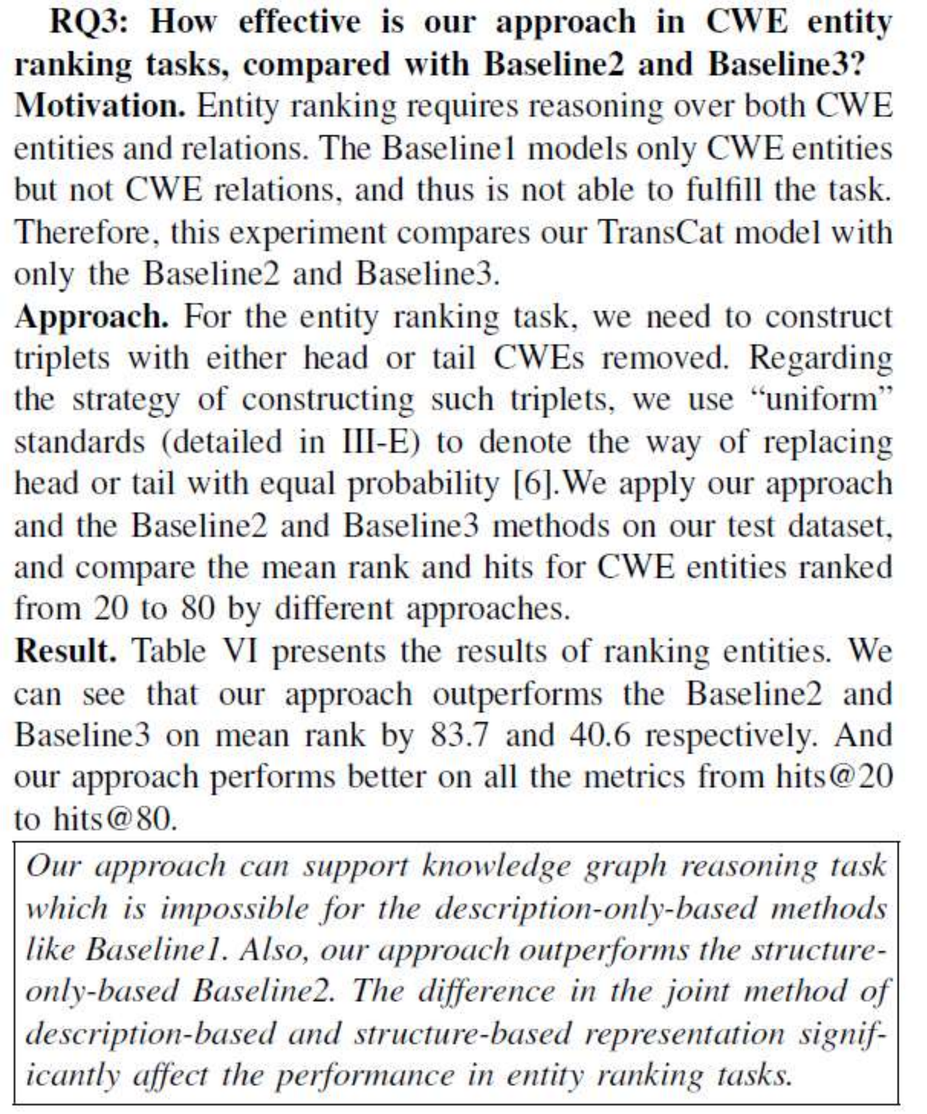
In order to evaluate the advantages of intermediate code-based vulnerability candidate representation over source code-based one, we conduct experiments with the following two vulnerability candidate representations:



Insight 1.

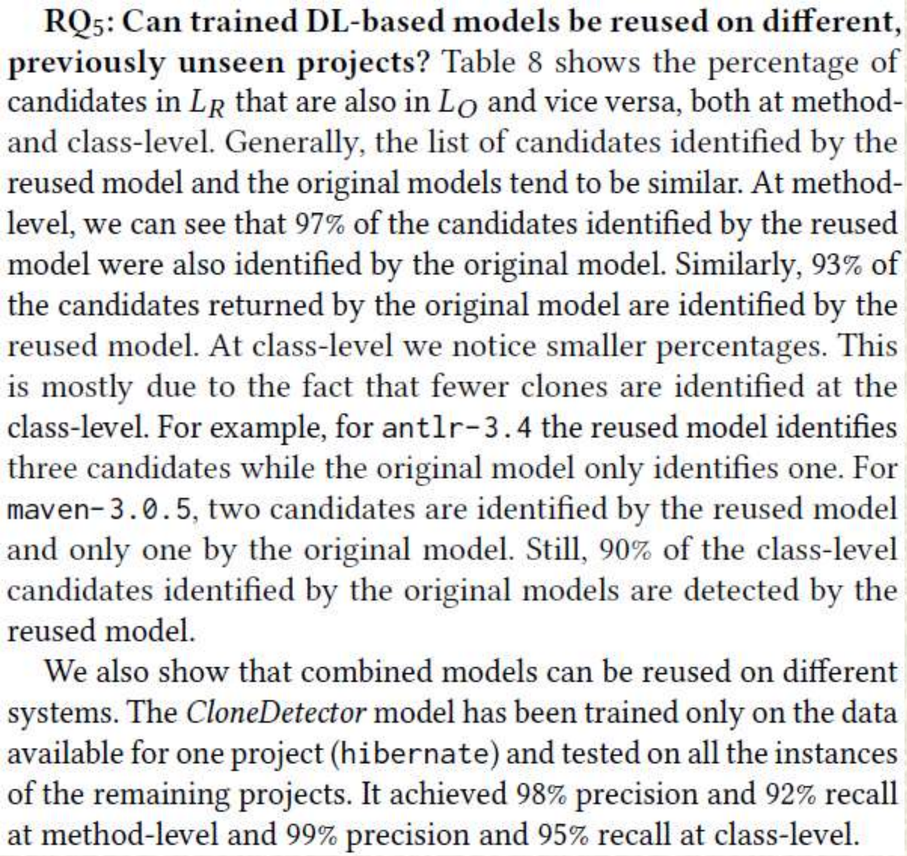
VulDeeLocator leveraging intermediate code-based representation is substantially more effective than VulDeeLocator using source code-based representation, owing to the aforementioned two advantages of intermediate code-based representation.

讨论部分的示例



RQ3: How effective is our approach in CWE entity ranking tasks, compared with Baseline2 and Baseline3?
Motivation. Entity ranking requires reasoning over both CWE entities and relations. The Baseline1 models only CWE entities but not CWE relations, and thus is not able to fulfill the task. Therefore, this experiment compares our TransCat model with only the Baseline2 and Baseline3.
Approach. For the entity ranking task, we need to construct triplets with either head or tail CWEs removed. Regarding the strategy of constructing such triplets, we use "uniform" standards (detailed in III-E) to denote the way of replacing head or tail with equal probability [6].We apply our approach and the Baseline2 and Baseline3 methods on our test dataset, and compare the mean rank and hits for CWE entities ranked from 20 to 80 by different approaches.
Result. Table VI presents the results of ranking entities. We can see that our approach outperforms the Baseline2 and Baseline3 on mean rank by 83.7 and 40.6 respectively. And our approach performs better on all the metrics from hits@20 to hits@80.

Our approach can support knowledge graph reasoning task which is impossible for the description-only-based methods like Baseline1. Also, our approach outperforms the structure-only-based Baseline2. The difference in the joint method of description-based and structure-based representation significantly affect the performance in entity ranking tasks.



RQ$_5$: Can trained DL-based models be reused on different, previously unseen projects? Table 8 shows the percentage of candidates in L_R that are also in L_O and vice versa, both at method- and class-level. Generally, the list of candidates identified by the reused model and the original models tend to be similar. At method-level, we can see that 97% of the candidates identified by the reused model were also identified by the original model. Similarly, 93% of the candidates returned by the original model are identified by the reused model. At class-level we notice smaller percentages. This is mostly due to the fact that fewer clones are identified at the class-level. For example, for `antlr-3.4` the reused model identifies three candidates while the original model only identifies one. For `maven-3.0.5`, two candidates are identified by the reused model and only one by the original model. Still, 90% of the class-level candidates identified by the original models are detected by the reused model.

We also show that combined models can be reused on different systems. The *CloneDetector* model has been trained only on the data available for one project (`hibernate`) and tested on all the instances of the remaining projects. It achieved 98% precision and 92% recall at method-level and 99% precision and 95% recall at class-level.

讨论部分的写作原则

2. 分析解释导致结果的原因

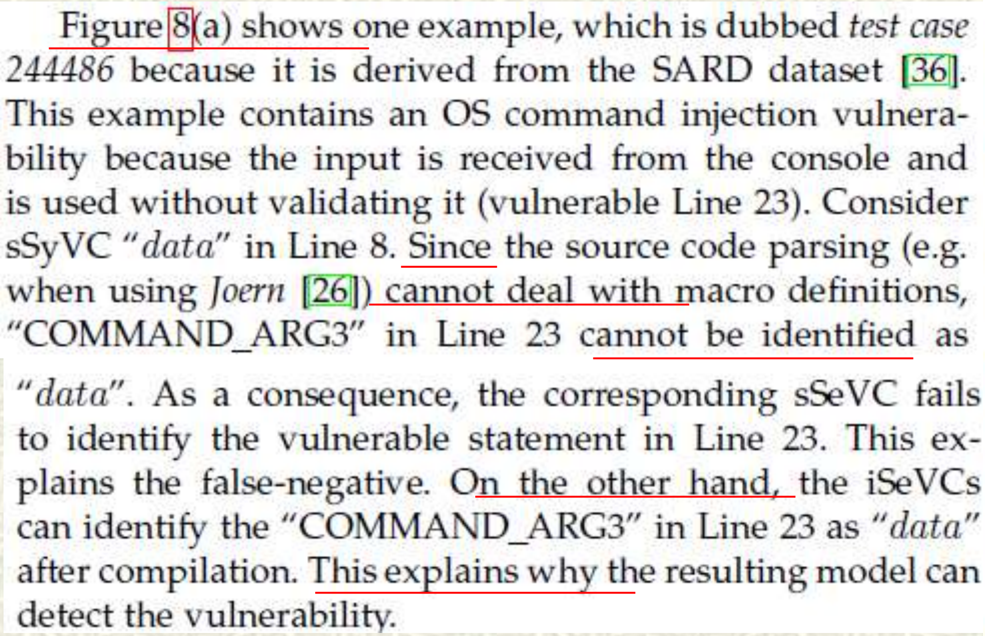


Figure 8(a) shows one example, which is dubbed *test case 244486* because it is derived from the SARD dataset [36]. This example contains an OS command injection vulnerability because the input is received from the console and is used without validating it (vulnerable Line 23). Consider sSyVC "*data*" in Line 8. Since the source code parsing (e.g. when using *Joern* [26]) cannot deal with macro definitions, "COMMAND_ARG3" in Line 23 cannot be identified as "*data*". As a consequence, the corresponding sSeVC fails to identify the vulnerable statement in Line 23. This explains the false-negative. On the other hand, the iSeVCs can identify the "COMMAND_ARG3" in Line 23 as "*data*" after compilation. This explains why the resulting model can detect the vulnerability.

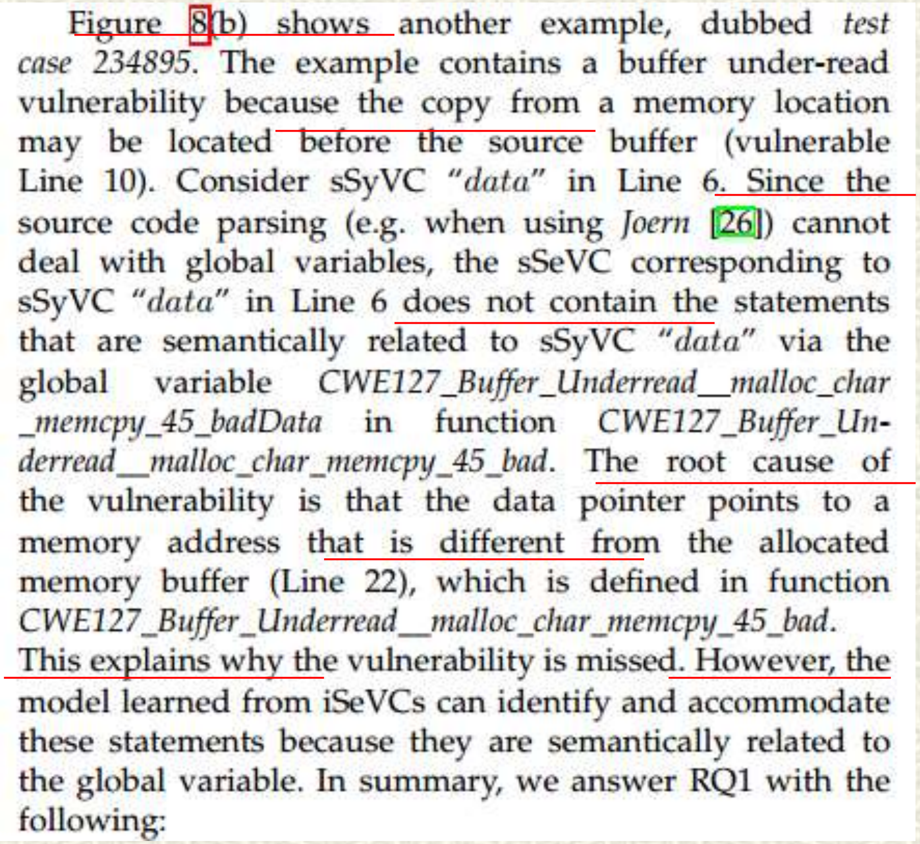


Figure 8(b) shows another example, dubbed *test case 234895*. The example contains a buffer under-read vulnerability because the copy from a memory location may be located before the source buffer (vulnerable Line 10). Consider sSyVC "*data*" in Line 6. Since the source code parsing (e.g. when using *Joern* [26]) cannot deal with global variables, the sSeVC corresponding to sSyVC "*data*" in Line 6 does not contain the statements that are semantically related to sSyVC "*data*" via the global variable *CWE127_Buffer_Underread__malloc_char_memcpy_45_badData* in function *CWE127_Buffer_Underread__malloc_char_memcpy_45_bad*. The root cause of the vulnerability is that the data pointer points to a memory address that is different from the allocated memory buffer (Line 22), which is defined in function *CWE127_Buffer_Underread__malloc_char_memcpy_45_bad*. This explains why the vulnerability is missed. However, the model learned from iSeVCs can identify and accommodate these statements because they are semantically related to the global variable. In summary, we answer RQ1 with the following:

对比sSeVC和iSeVCs

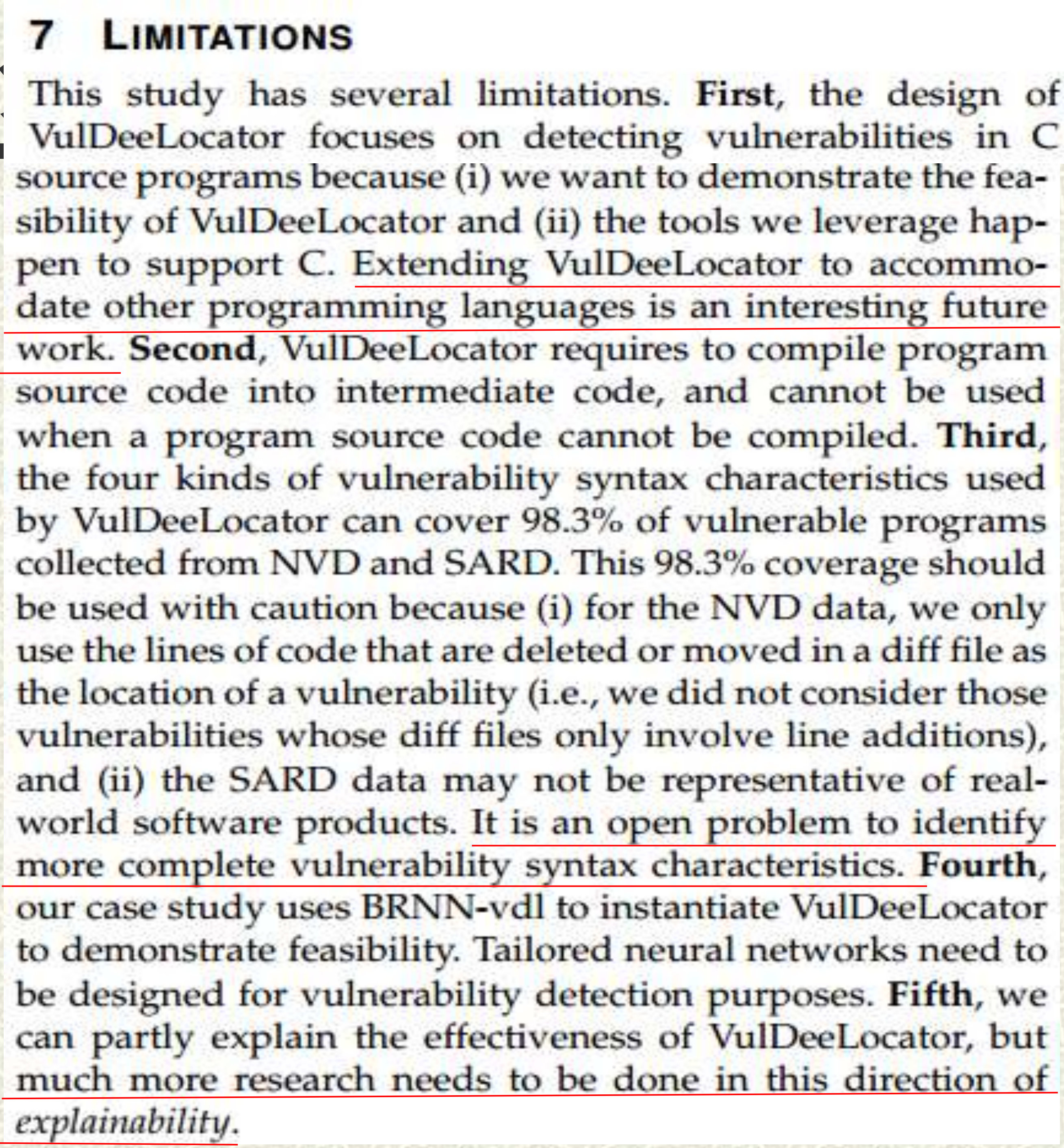
解释为什么后者可以检测

讨论部分的写作原则

- 3. 不要忽视负面结果和不足之处，切忌回避掩盖自己的不足
- Highlight problems and negative results
  - People will run into them.
  - Better to prepare them, and propose potential solutions.
  - These are avenues for further work.
  - Other readers will often figure out how to solve them, and your algorithm will be even better later.

讨论部

- **Limitations**

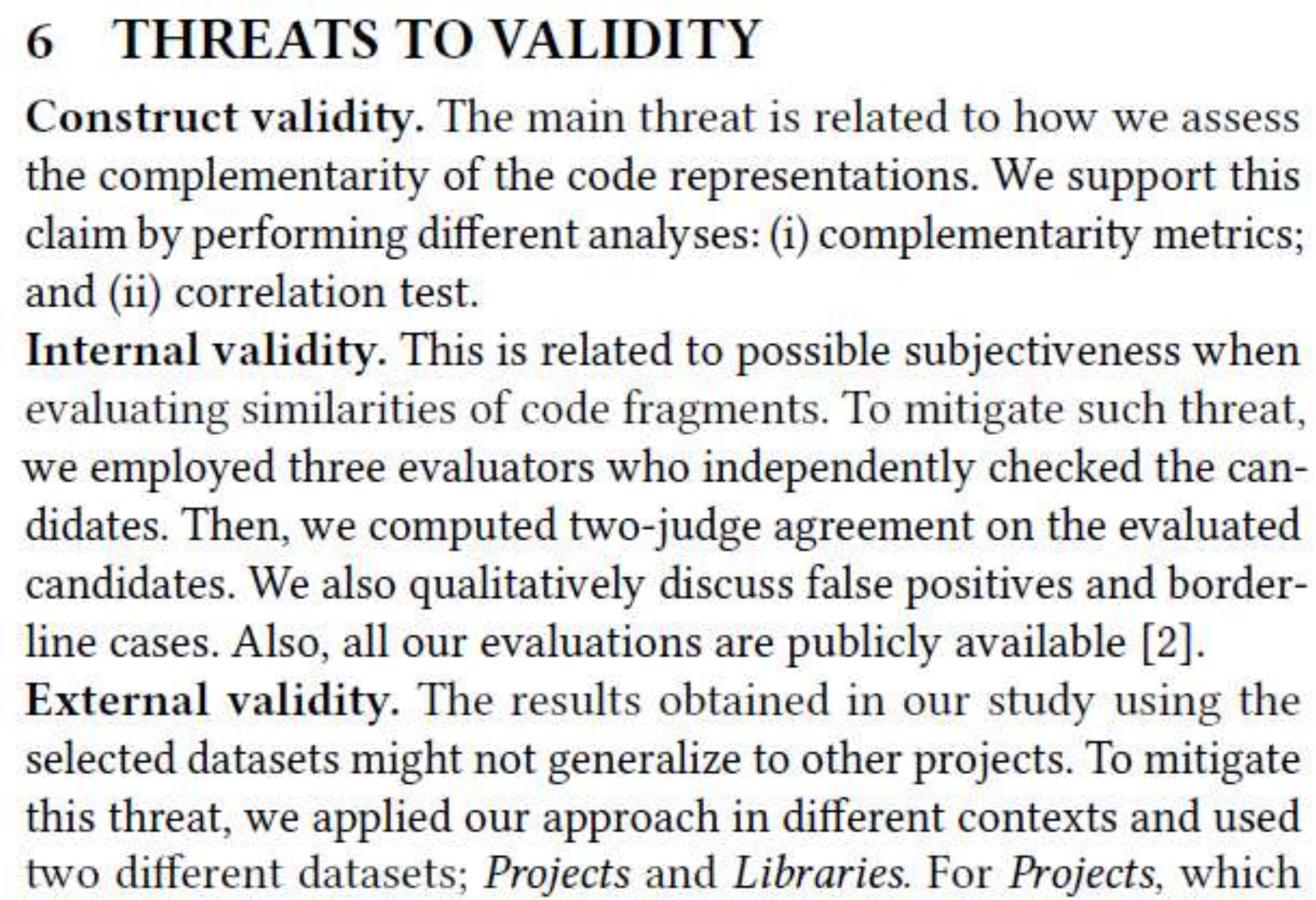


7 Limitations

This study has several limitations. **First**, the design of VulDeeLocator focuses on detecting vulnerabilities in C source programs because (i) we want to demonstrate the feasibility of VulDeeLocator and (ii) the tools we leverage happen to support C. Extending VulDeeLocator to accommodate other programming languages is an interesting future work. **Second**, VulDeeLocator requires to compile program source code into intermediate code, and cannot be used when a program source code cannot be compiled. **Third**, the four kinds of vulnerability syntax characteristics used by VulDeeLocator can cover 98.3% of vulnerable programs collected from NVD and SARD. This 98.3% coverage should be used with caution because (i) for the NVD data, we only use the lines of code that are deleted or moved in a diff file as the location of a vulnerability (i.e., we did not consider those vulnerabilities whose diff files only involve line additions), and (ii) the SARD data may not be representative of real-world software products. It is an open problem to identify more complete vulnerability syntax characteristics. **Fourth**, our case study uses BRNN-vdl to instantiate VulDeeLocator to demonstrate feasibility. Tailored neural networks need to be designed for vulnerability detection purposes. **Fifth**, we can partly explain the effectiveness of VulDeeLocator, but much more research needs to be done in this direction of *explainability*.

讨论部分的示例

- **THREATS TO VALIDITY**



6 THREATS TO VALIDITY

Construct validity. The main threat is related to how we assess the complementarity of the code representations. We support this claim by performing different analyses: (i) complementarity metrics; and (ii) correlation test.

Internal validity. This is related to possible subjectiveness when evaluating similarities of code fragments. To mitigate such threat, we employed three evaluators who independently checked the candidates. Then, we computed two-judge agreement on the evaluated candidates. We also qualitatively discuss false positives and borderline cases. Also, all our evaluations are publicly available [2].

External validity. The results obtained in our study using the selected datasets might not generalize to other projects. To mitigate this threat, we applied our approach in different contexts and used two different datasets; *Projects* and *Libraries*. For *Projects*, which

常见的问题

重复表述
实验数据
没有分析

无法佐证
创新点

分析停留在
表面，叶茂
而枝不繁

捕风捉影
夸大其词

结论

（Conclusion）

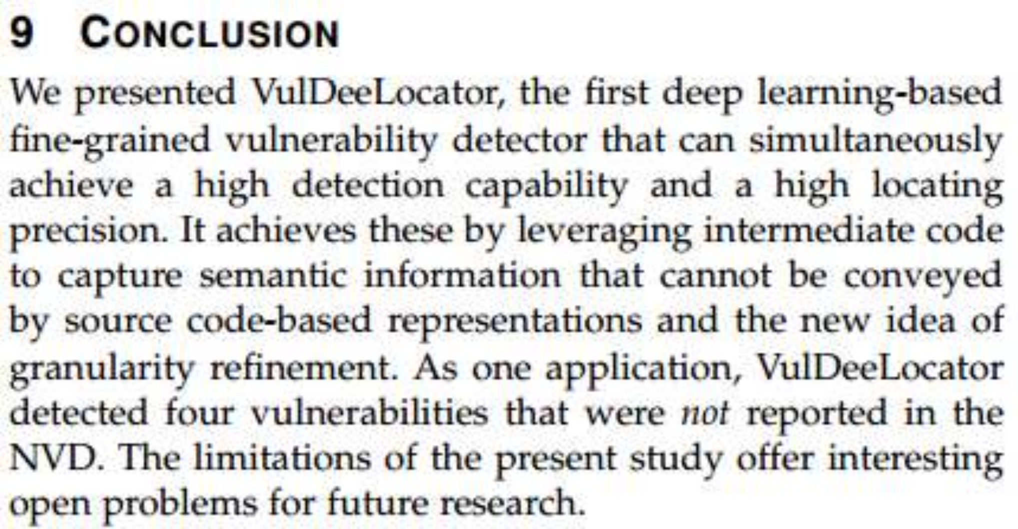
结论写什么?

- Conclusion
  - 总结归纳实验结果分析中得出的结论
  - 内容
    - a)快速简短的总结
    - b)未来工作的展望
    - c)结束全文

结论的写作范式（英）

Conclusion

- Purpose
  - Summary and future works
- Style
  - Restatement of purpose of the study or the research question
  - Summary of the most important findings
  - Possible interpretations of the findings
  - Comparison with expected results and other studies
  - Strengths and limitations of the study
  - Suggestions for future study

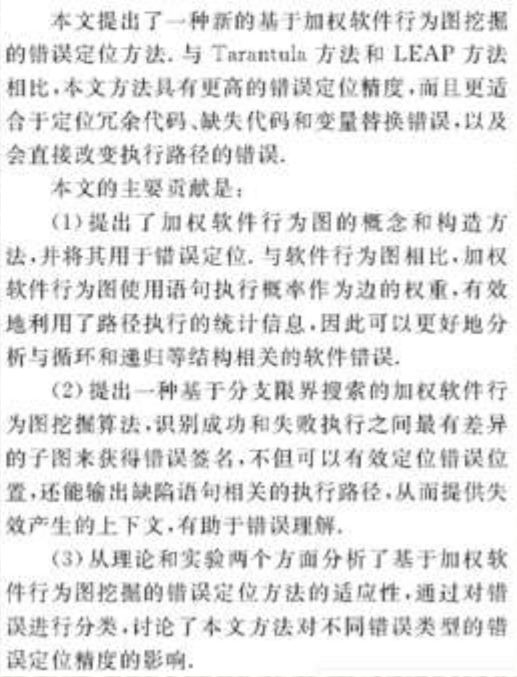


9 CONCLUSION

We presented VulDeeLocator, the first deep learning-based fine-grained vulnerability detector that can simultaneously achieve a high detection capability and a high locating precision. It achieves these by leveraging intermediate code to capture semantic information that cannot be conveyed by source code-based representations and the new idea of granularity refinement. As one application, VulDeeLocator detected four vulnerabilities that were *not* reported in the NVD. The limitations of the present study offer interesting open problems for future research.

结论的写作范式（中）

1. 采用的理论、方法、平台、工具
2. 论文的成果和创新点
3. 解决的主要问题，解决效果
4. 尚存在的问题与未来工作



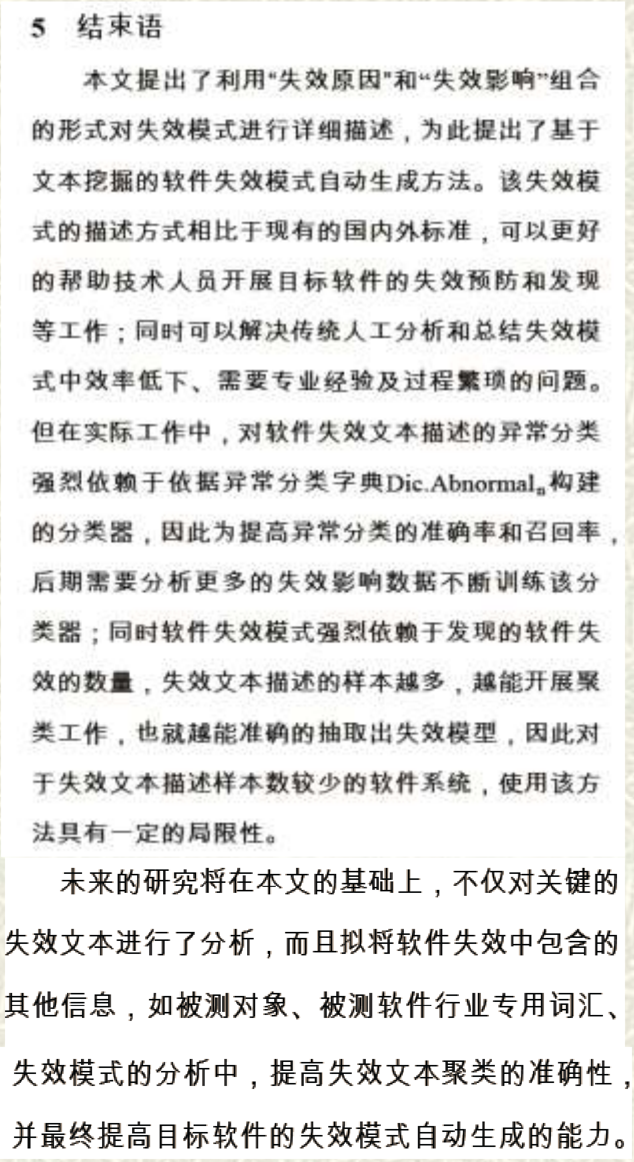
本文提出了一种新的基于加权软件行为图挖掘的错误定位方法. 与 Tarantula 方法和 LEAP 方法相比，本文方法具有更高的错误定位精度，而且更适合于定位冗余代码、缺失代码和变量替换错误，以及会直接改变执行路径的错误.

本文的主要贡献是：

(1) 提出了加权软件行为图的概念和构造方法，并将其用于错误定位. 与软件行为图相比，加权软件行为图使用语句执行概率作为边的权重，有效地利用了路径执行的统计信息，因此可以更好地分析与循环和递归等结构相关的软件错误.

(2) 提出一种基于分支限界搜索的加权软件行为图挖掘算法，识别成功和失败执行之间最有差异的子图来获得错误签名，不但可以有效定位错误位置，还能输出缺陷语句相关的执行路径，从而提供失效产生的上下文，有助于错误理解.

(3) 从理论和实验两个方面分析了基于加权软件行为图挖掘的错误定位方法的适应性，通过对错误进行分类，讨论了本文方法对不同错误类型的错误定位精度的影响.



5 结束语

本文提出了利用"失效原因"和"失效影响"组合的形式对失效模式进行详细描述，为此提出了基于文本挖掘的软件失效模式自动生成方法。该失效模式的描述方式相比于现有的国内外标准，可以更好的帮助技术人员开展目标软件的失效预防和发现等工作；同时可以解决传统人工分析和总结失效模式中效率低下、需要专业经验及过程繁琐的问题。但在实际工作中，对软件失效文本描述的异常分类强烈依赖于依据异常分类字典Dic.Abnormal$_a$构建的分类器，因此为提高异常分类的准确率和召回率，后期需要分析更多的失效影响数据不断训练该分类器；同时软件失效模式强烈依赖于发现的软件失效的数量，失效文本描述的样本越多，越能开展聚类工作，也就越能准确的抽取出失效模型，因此对于失效文本描述样本数较少的软件系统，使用该方法具有一定的局限性。

未来的研究将在本文的基础上，不仅对关键的失效文本进行了分析，而且拟将软件失效中包含的其他信息，如被测对象、被测软件行业专用词汇、失效模式的分析中，提高失效文本聚类的准确性，并最终提高目标软件的失效模式自动生成的能力。

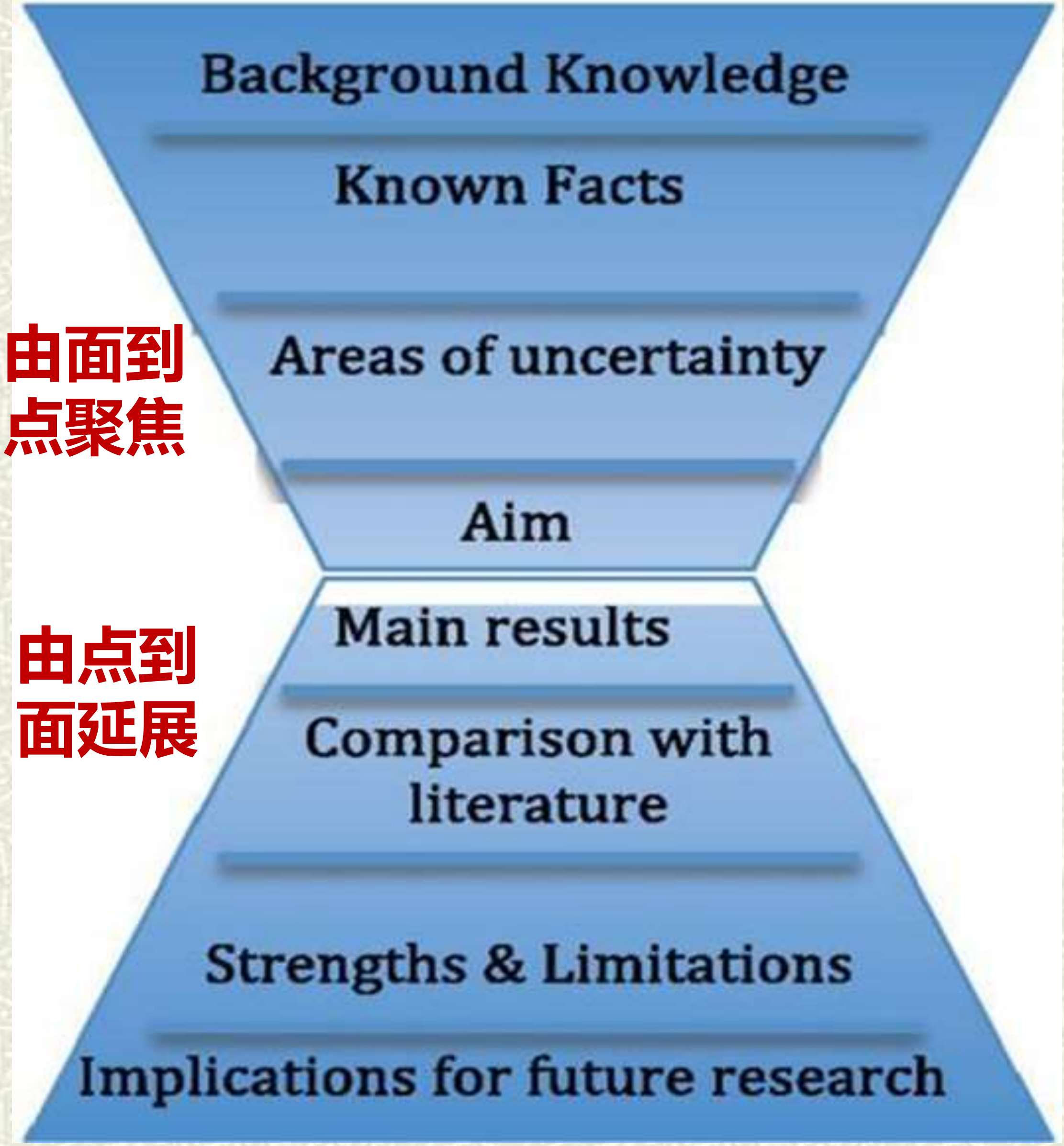
引言 vs 结论

- 漏斗型结构
  - Introduction and discussion sections together form a hourglass pattern

general to specific to general

由面到点聚焦

由点到面延展



Background Knowledge

Known Facts

Areas of uncertainty

Aim

Main results

Comparison with literature

Strengths & Limitations

Implications for future research

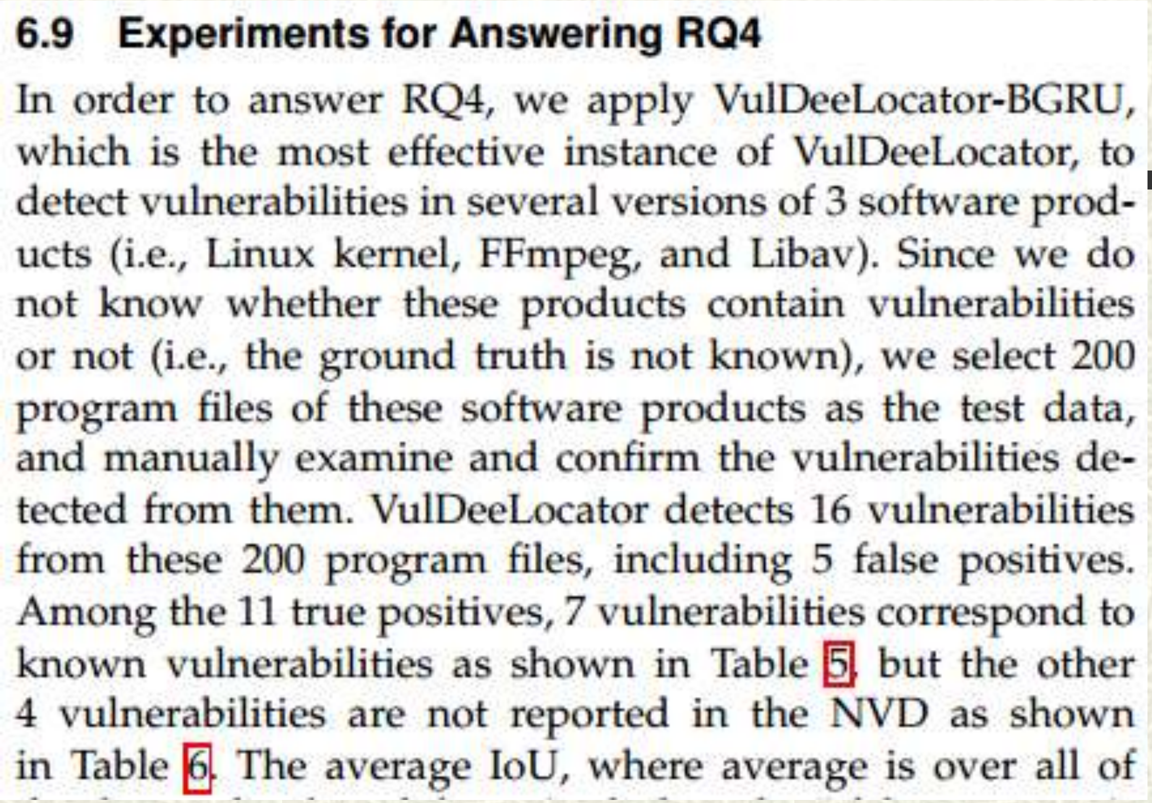
漏斗型结构适用于所有段落

- Write to enlighten (for the reader)
  - Academic writing is not like writing prose.
    - There are no set ups, no surprise, and no punch lines.
  - Doesn't mean it has to be dry, though!
  - This hour-glass structure(general to specific to general) works very well at the level of paragraphs, sections, and papers.

漏斗型结

由面到点
general to specific

由点到面
specific to general



6.9 Experiments for Answering RQ4

In order to answer RQ4, we apply VulDeeLocator-BGRU, which is the most effective instance of VulDeeLocator, to detect vulnerabilities in several versions of 3 software products (i.e., Linux kernel, FFmpeg, and Libav). Since we do not know whether these products contain vulnerabilities or not (i.e., the ground truth is not known), we select 200 program files of these software products as the test data, and manually examine and confirm the vulnerabilities detected from them. VulDeeLocator detects 16 vulnerabilities from these 200 program files, including 5 false positives. Among the 11 true positives, 7 vulnerabilities correspond to known vulnerabilities as shown in Table 5, but the other 4 vulnerabilities are not reported in the NVD as shown in Table 6. The average IoU, where average is over all of

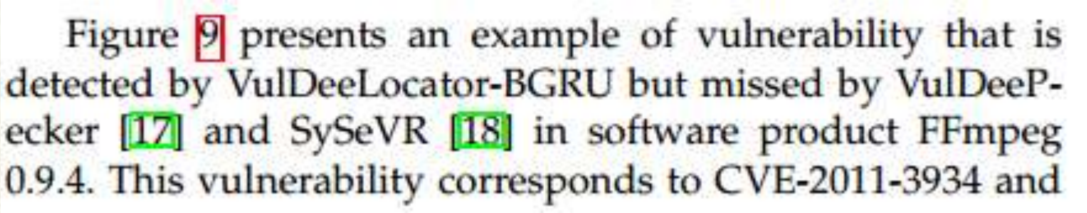


Figure 9 presents an example of vulnerability that is detected by VulDeeLocator-BGRU but missed by VulDeePecker [17] and SySeVR [18] in software product FFmpeg 0.9.4. This vulnerability corresponds to CVE-2011-3934 and

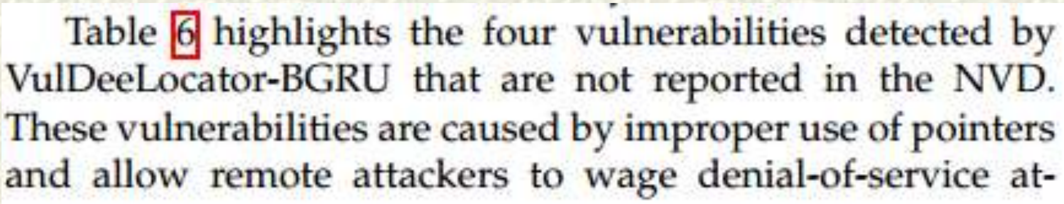
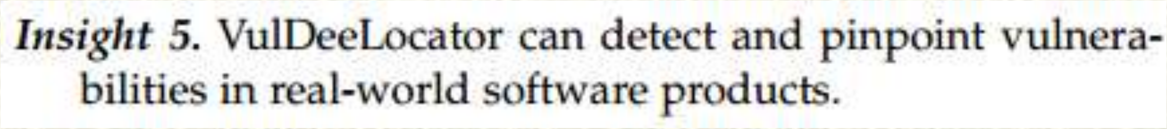


Table 6 highlights the four vulnerabilities detected by VulDeeLocator-BGRU that are not reported in the NVD. These vulnerabilities are caused by improper use of pointers and allow remote attackers to wage denial-of-service at-



Insight 5. VulDeeLocator can detect and pinpoint vulnerabilities in real-world software products.

结论的撰写原则

- 概括准确
- 措词严谨
  - 不用“大概”、“也许”、“可能是”
- 明确具体
- 简短精练
- 不作自我评价

致谢

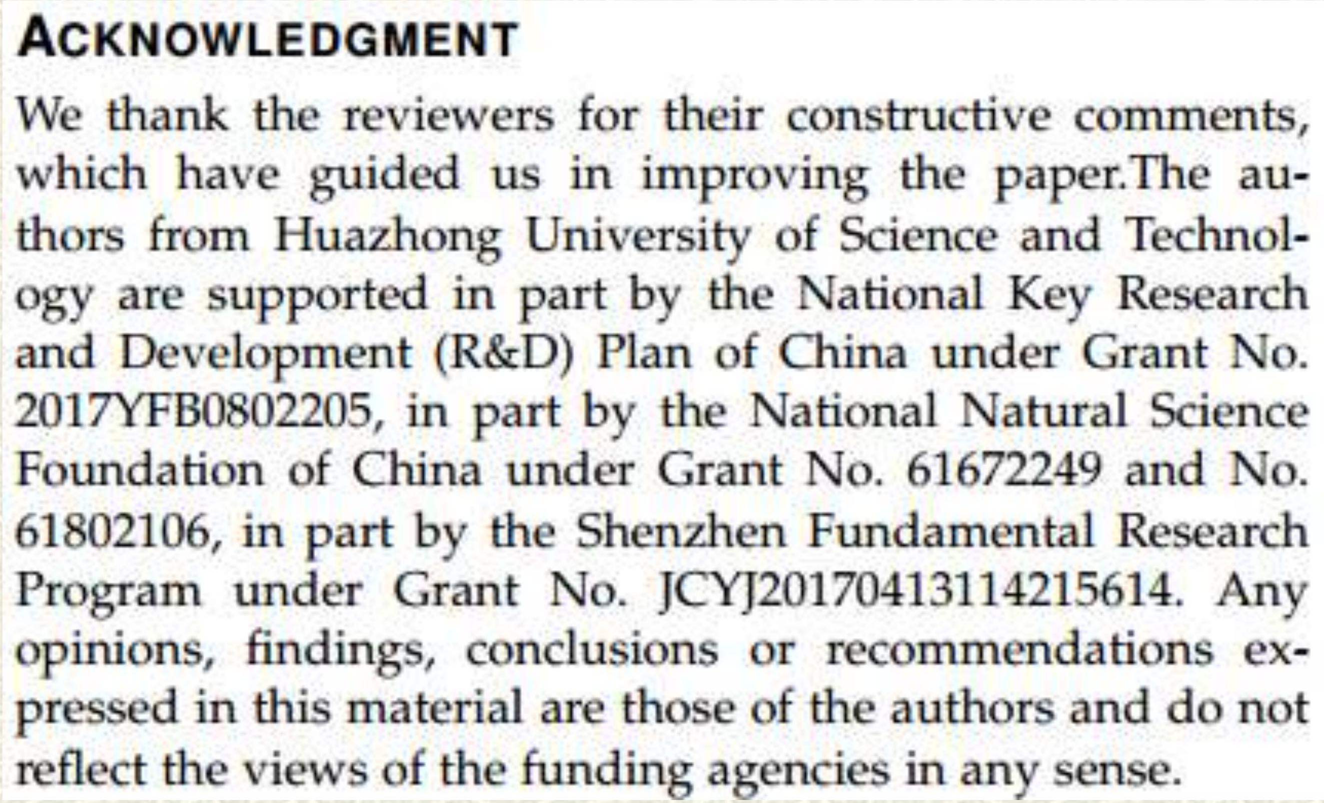
(Acknowledgement)

致谢

- 一项研究工作往往不是一个人能单独完成的
- 论文公开发表后
  - 相当于用书面形式记载了你的科研成果
  - 同时也记下了你的科研道德
- 在第二次世界大战结束后，科学界出现了一次极不公平的诺贝尔奖事件
  - 物理学家莱丝•梅特娜
  - 奥托•哈恩

致谢的对象

- 对本研究直接提供过资金、设备、人力以及文献资料等支持和帮助的团体和个人
- 对论文有贡献、有帮助的人员、机构或项目



ACKNOWLEDGMENT

We thank the reviewers for their constructive comments, which have guided us in improving the paper.The authors from Huazhong University of Science and Technology are supported in part by the National Key Research and Development (R&D) Plan of China under Grant No. 2017YFB0802205, in part by the National Natural Science Foundation of China under Grant No. 61672249 and No. 61802106, in part by the Shenzhen Fundamental Research Program under Grant No. JCYJ20170413114215614. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of the funding agencies in any sense.

参考文献

（References）

哪些文献要列入参考文献著录?

- 参考文献著录
- 下列文献均应列出
  - 说明当时的研究所达到的水平
  - 在研究工作开展中，受哪些文献资料的启发
  - 从哪些论文中获得了教益，促进了研究进度

列写参考文献著录的注意事项

- 知而不引
- 断章取义
- 引而未读
- 引而不确
- 来源不实
- 盲目自引

参考文献著录的目的与作用

- 论文的依据、起点和深度
- 索引作用
- 节省论文篇幅
- 反映了作者的科学态度，关系到论文的可信度和作者的声誉

参考文献——一个容易被忽视的角落

- 不认真严肃对待参考文献，会怎样?
  - 读者、编辑和审稿人可能会看作是一种不良学风
- 一个值得学习的实例：“孟德尔定律”的故事

附录

（Appendix）

附录

- 附录
  - 论文主体的补充项目
  - 写入正文有可能有损于行文的条理性、逻辑性和精练性
- 添加附录的目的
  - 体现整篇论文材料上的完整性

哪些材料需要采用附录形式?

- 比正文更为详尽的理论根据
- 研究方法和技术要点更深入的叙述
- 建议可以阅读的参考文献题录,
- 对了解正文内容有用的补充信息等
- 因篇幅过长或取材于复制品，而不宜写入正文的资料
- 不便于写入正文的罕见珍贵资料
- 一般读者并非必要阅读, 但对本专业同行很有参考价值的资料
- 某些重要的原始数据、数学推导、计算程序、框图、结构图、统计表、计算机打印输出件等

科技写作的三种境界

1 把事情讲明白

2 有逻辑，有条理

3 有情感，打动人

有思想的写作

将写作变为一种享受和艺术

Thank you for your attention !

SuXiaoHong