软件工程综述简介

南京大学 软件学院 iSE实验室

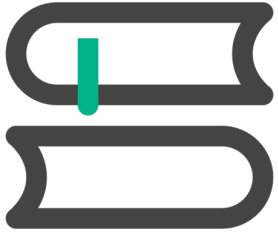
目录

01

What is survey?

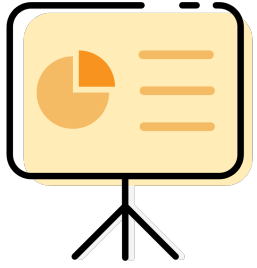
综述的定位

- 综述：针对阅读、写作和表达能力的复合挑战
  - RWPH：SE研究的四个维度



Reading

Writing



Presenting

Hacking

综述的定义

- 综述（Survey） vs. 文献回顾（Literature Review）
  - Survey ⇄ **Systematic** Review
  - “综述是一种以新颖的方式总结和组织最近的研究成果、整合并添加对该领域工作的理解和认识的研究工作。一篇综述一般对现有文献进行分类，同时评估领域的发展趋势、提出独到的看法”—— **ACM Computing Survey**

综述的定义

- 综述（Survey） vs. 文献回顾（Literature Review）
  - Survey ⇄ **Systematic** Review
  - “综述的对象是文献；综述文章应当全面回顾选定领域的发展。” —— **IEEE Communications Surveys & Tutorials journal**
  - “综述应当对相关文献进行批判性审查” —— **Elsevier journal of Computer Science Review**

综述的定义

- 综述（Survey） vs. 文献回顾（Literature Review）
  - Survey ⇄ **Systematic** Review
  - **系统文献回顾**（Systematic Literature Review），简称文献回顾（Literature Review）是一种**识别**、**评估**和**解释**与特定**研究问题（Research Question）**、**领域主题(Topic Area)**或**有趣现象（Phenomenon of Interest）**相关的所有研究的**方法**。

综述的定义

- 初级研究与二级研究
  - **初级研究**（Primary Study）：为系统综述提供帮助和资料来源的研究工作。如技术类工作（Technical Paper）和经验研究（Empirical Study）
  - **二级研究（Secondary Study）：**在现有的一系列工作的基础上进行整理、分类、分析的研究工作。系统文献回顾/综述就是二级研究

02

Why survey?

综述研究的动机

- 开展一次汇报
- 开启全新的研究方向
- 回顾特定领域的发展
- 完成毕业论文

……

- **完成作业并获得学分**

综述研究的目标

- **总结有关技术的现有进展。**例如，总结一下特定模糊测试技术优点和局限性，（甚至）给出相关的经验证据（Empirical Evidence）→ 技术发展（Technology development）
- **找出当前某个研究领域的不足。**例如，分析一下现有变异测试技术存在的问题 → 提升空间（Room for Improvement）

综述研究的目标

- **提供研究框架/背景**，放置新的研究方向 → 研究方向（New Direction）
- **检查经验证据在多大程度上支持/反对理论假设，**甚至在前人的基础上衍生得到全新的假设。→ 假设验证（Hypotheses Examination）

03

How to survey?

综述研究流程

- 三个环节，五个步骤
  - 三个环节：规划（Plan）、实施（Conduct）、报告（Report）
  - 五个步骤
    - **框架搭建**：制定研究框架、设计研究问题
    - **文献检索**：检索目标领域近5~10年的工作
    - **文献阅读**：阅读文献，了解文章内容
    - **文献分类**：按照一定的规则，将收集到的文献划分成3~6个正交（Orthogonal）类别
    - **文献分析**：提取文献共性、甄别文献特点、得出研究方向

规划综述

- 确定综述需求
- 委托综述需求
- 确定研究问题
- 搭建综述框架/协议（Review Protocol）
- 评估综述结果

规划综述

- 确定综述需求
  - 需求方：研究人员，研究团队
  - 来源：总结现有信息、归纳已有现象、评价近期工作
  - 要求：彻底（Thorough）、系统（Systematic）、公正（Unbiased）

规划综述

- 委托综述需求
  - 某些时候，一些机构或者组织拥有一些面向特定领域的综述需求，但却没有时间或者相关领域的专家实施综述，这时就会产生综述需求的委托。

规划综述

- 确定研究问题
  - **研究问题**（Research Question，RQ）：一个或一组研究聚焦的核心问题，是研究工作的出发点和驱动力，表达了研究者的研究兴趣和研究热情
  - 作用：为综述研究一整套系统方法论提供引导
  - 地位：综述研究中最为重要的部分

规划综述

- 确定研究问题
  - 良好的研究问题应当具备的特点
    - 清晰（Clear）：让读者能清楚的知道问题的含义
    - 集中（Focused）：关注研究工作中极小的方面
    - 简洁（Concise）：用尽可能少的词汇来表达
    - 复杂（Complex）：需要通过分析才能得出结论
    - 有争议（Arguable）：问题是开放的，有可能会得出相反结论

规划综述

Table 1. Research Questions

Ref#	Question
	General questions
RQ_1	Which methods, algorithms and data sources have been used for automated query reformulations targeting code search in the literature?
RQ_2	Which methods, metrics or subject systems have been used to evaluate and validate the researches on automated query reformulations?
RQ_3	What are the major challenges of automated query reformulations intended for code search? How many of them have been solved to date by the literature?
	Statistical questions
RQ_4	How much activities of research on automated query reformulations have been performed to date? What are the venues that these researches got published at?
	Focused questions
RQ_5	What are the differences and similarities between query reformulations for local code search and query reformulations for Internet-scale code search?
RQ_6	What are the scopes for future work in the area of automated query reformulation targeting the code search?

A Systematic Literature Review of Automated Query Reformulations in Source Code Search[1]

[1] Rahman M M, Roy C K. A Systematic Literature Review of Automated Query Reformulations in Source Code Search[J]. arXiv preprint arXiv:2108.09646, 2021.

规划综述

- 搭建综述框架/协议
  - **综述协议**（Review Protocol）规定了进行特定领域综述研究所采用的方法；一个预先定义好的综述协议是必要的，它能够减少研究人员在综述过程中产生偏误的可能性

规划综述

- 评估综述结果
  - 研究者必须为综述协议制定评估标准。综述协议是综述研究中最关键的部分，因此所有的研究者必须要在综述协议的评估上达成一致
  - 内容：确定评估的方法、流程、指标

实施综述

- 划定文献范围
- 选取初级研究
- 评估研究质量
- 数据提取（Data Extraction）
- 数据合成（Data Synthesis）

实施综述

- 划定文献范围：基本流程
  - 确定主题和出版物、检索文献
  - 确定要探索的研究领域；调查目标领域的主要出版物，搜集大量的初级研究（Primary Study）
  - 检索：使用**无偏见搜索策略（Unbiased Search Strategy）**
  - 出版物参考：中国计算机学会推荐国际学术会议和期刊目录（第五版）[1，2]

[1] 中国计算机学会推荐国际学术会议和期刊目录（第五版），2019年修订， https://ccf.atom.im/
[2] CCF推荐期刊会议下载链接： https://www.ccf.org.cn/ccf/contentcore/resource/download?ID=99185

实施综述

- 划定文献范围：基本流程

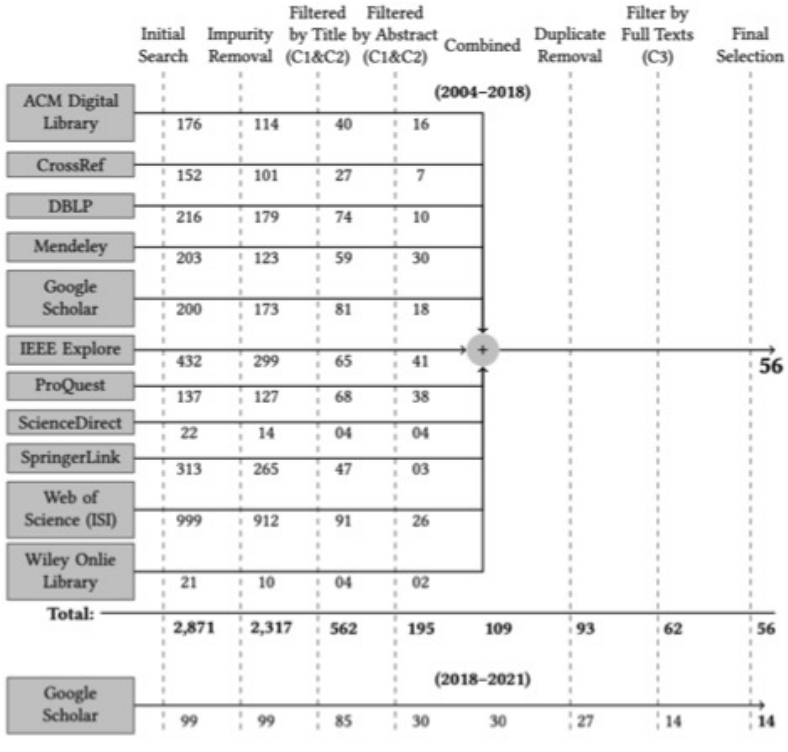


Fig. 4. Selection of primary studies

Sources to be Searched

Source	Responsible
Information and Software Technology (IST)	Kitchenham
Journal of Systems and Software	Kitchenham
IEEE Transactions on Software Engineering	Kitchenham
IEEE Software	Kitchenham
Communications of the ACM (CACM)	Brereton
ACM Surveys	Brereton
Transactions on Software Engineering Methods (TOSEM)	Brereton
Software Practice and Experience	Budgen & Kitchenham
Empirical Software Engineering Journal (ESEM)	Budgen
IEE Proceedings Software (now IET Software)	Kitchenham
Proceedings International Conference on Software Engineering (ICSE 04, 05, 06, 07)	Linkman & Kitchenham & Brereton
Proceedings International Seminar of Software Metrics (Metrics04, Metrics05)	Kitchenham & Brereton
Proceedings International Seminar on Empirical Software Engineering (ISESE 04, 05, 06)	Kitchenham & Brereton

实施综述

- 划定文献范围：搜索文献
  - 常用学术搜索引擎：Google Scholar[1], DBLP[2]
  - **检索方式**
    - 搜索关键词（Keywords）：Fuzzing，Fuzz testing, Fuzzing IoT，Greybox Fuzzing
    - 搜索相关（Relevant）领域：Fuzzing & Test Generation，Fuzzing & Differential Testing
    - 搜索引用（Reference）：从某篇文章的引用出发查找文献

[1] 谷歌学术：https://scholar.google.com/
[2] DBLP, Computer Science bibliography, https://dblp.org/

实施综述

- 选取初级研究 & 评估研究质量
  - 初级研究的质量直接影响到最终综述的质量，应当把眼光放在经典、优质、有影响力的论文上
  - 出版物：选取被优秀会议和期刊收录的论文 → 论文的出身
  - 相关性：选取真正相关的论文 → 论文的内容
  - 引用次数：选择高引用的初级研究

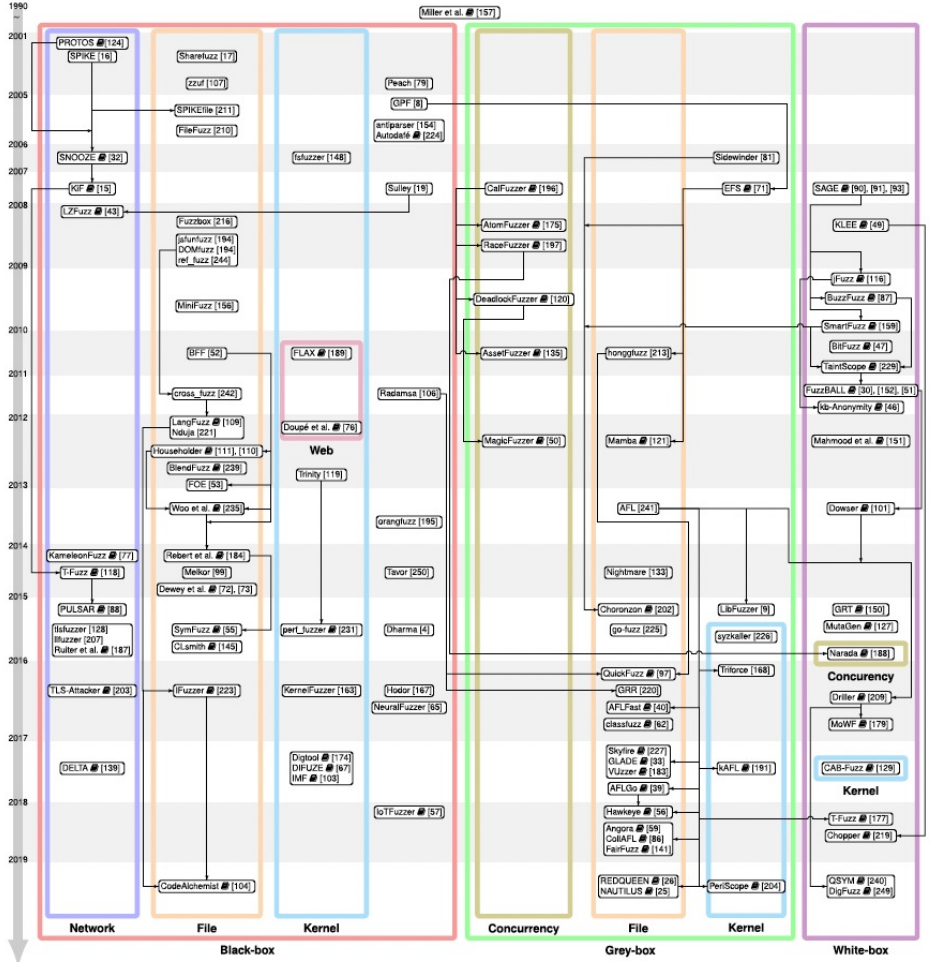
实施综述

- 数据提取 & 数据合成
  - 提取：设计并按照一定的格式来准确有序地记录从初级研究中获得的数据
  - 合成：对从初级研究中提取得到的数据进行分析、归纳和总结，并围绕处理后的数据进行讨论和衍生，以产生**观察**（Observation）、提出**建议**（Recommendation）、给出**结论**（Conclusion）、进行**展望**（Future Work）

04

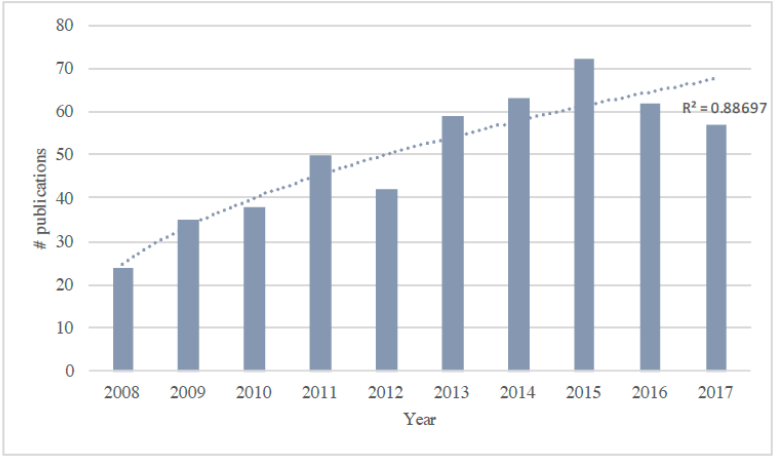
Example & Reference

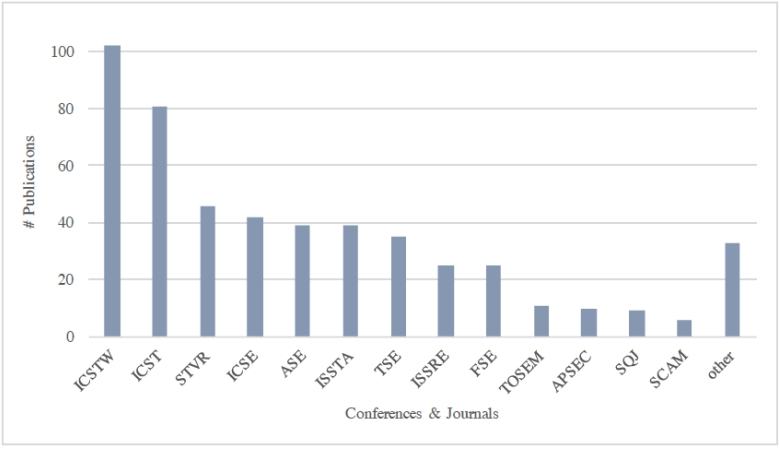
发展历程



TSE'19-The Art, Science, and Engineering of Fuzzing: A Survey

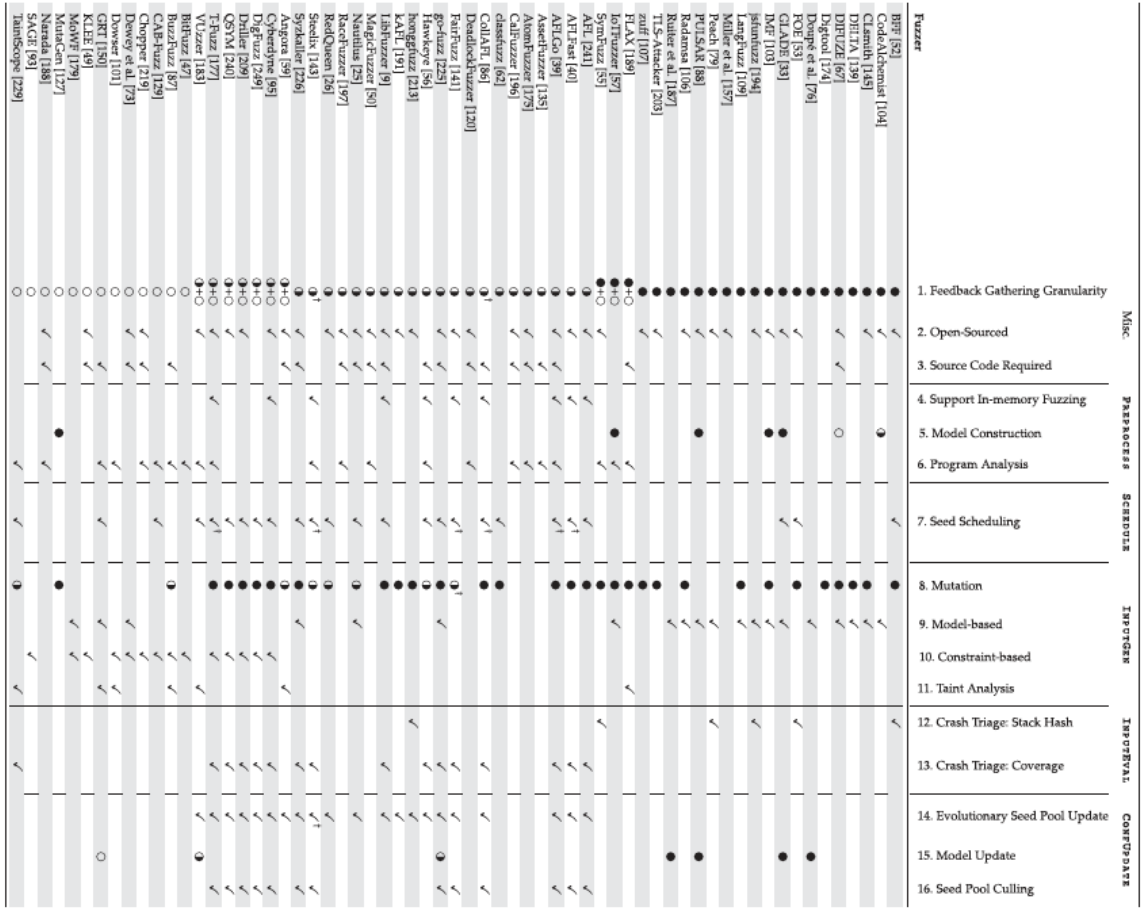
发展历程





Elsevier,AIC'19-Mutation Testing Advances: An Analysis and Survey

文献分类



TSE'19-The Art, Science, and Engineering of Fuzzing: A Survey

文献分类

Year	Fuzzer	Solution(Process)	Fitness By	Target App/Bug	Input		Runtime Info.
					Muta. -based	Gene. -based	
2006	Sidewinder [62]	MC(seed.) + GA(rete.)	block transition	general	✓		◐
2007	RANDOOP [141]	GA(rete.)	legality	object-oriented	✓		●
2013	FuzzSim [192]	WCCP(seed.)	#bugs	general	✓		●
2014	COVERSET [158]	MSCP(set.)	code coverage	general		✓	◐
		ILP(seed.)	#bugs		✓		●+◐+○
2015	Joeri *et al.* [55]	GA(rete.)	state machine	protocol	✓		●
2016	AFLFast [23]	MC(seed.) + GA(rete.)	path transition	general	✓		◐
2016	classfuzz [42]	MH(mutation.) + GA(rete.)	code coverage	JVM	✓		◐
2017	VUzzer [157]	MC(seed.) + GA(rete.)	block transition	general	✓		◐
2017	AFLGo [22]	SA(seed.) + GA(rete.)	path transition	general	✓		◐
2017	NEZHA [151]	GA(rete.)	asymmetry	semantic bugs	✓		●+◐
2017	DeepXplore [148]	GA(rete.)	neuron coverage	deep learning	✓		◐
2018	STADS [17]	Species(seed.)∗	state discovery	general	✓		◐
2018	CollAFL [66]	GA(rete.)	△code coverage	general	✓		◐
2018	Angora [37]	GD(byte.) + GA(rete.)	△code coverage	general	✓		◐
2019	DigFuzz [210]	MC(seed.) + GA(rete.)	block transition	general	✓		○
2019	MOPT [116]	PSO(mutation.) + GA(rete.)	code coverage	general	✓		◐
2019	NEUZZ [172]	NN(byte.) + GA(rete.)	branch behavior	general	✓		◐
2019	Cerebro [105]	MOO(seed.) + GA(rete.)	△code coverage	general	✓		◐
2019	DifFuzz [138]	GA(rete.)	asymmetry	side-channel	✓		●
2020	AFLNET [154]	GA(rete.)	state machine	protocol	✓		◐
2020	EcoFuzz [203]	VAMAB(seed.) + GA(rete.)	path transition	general	✓		◐
2020	Entropic [21]	Shannon(seed.) + GA(rete.)	state discovery	general	✓		◐
2020	MTFuzz [171]	MTNN(byte.) + GA(rete.)	△branch behavior	general	✓		◐
2020	Ankou [118]	GA(rete.)	△code coverage	general	✓		◐
2020	FIFUZZ [87]	GA(rete.)	△code coverage	error-handling	✓		◐
2020	IJON [6]	GA(rete.)	△code coverage	general	✓		◐
2020	Krace [194]	GA(rete.)	alias coverage	data race	✓		◐
2021	AFL-HIER [88]	UCB1(seed.) + GA(rete.)	△path transition	general	✓		◐
2021	PGFUZZ [82]	GA(rete.)	safety policy	robotic vehicle	✓		●
2021	Yousra *et al.* [1]	GA(rete.)	validation log	SmartTV	✓		●
2021	AFLChurn [214]	SA(seed.) + ACO(byte.) + GA(rete.)	path transition + commit history	general	✓		◐

MC: Markov Chain; **MSCP**: Minimal Set Cover Problem; **ILP**: Integer Linear Programming Problem; **WCCP**: Weighted Coupon Collector's Problem; **VAMAB**: Variant of Adversarial Multi-Armed Bandit; **UCB1**: Upper Confidence Bound, version one; **MH**: Metropolis-Hastings; **PSO**: Particle Swarm Optimization; **Shannon**: Shannon's entropy; **Species∗**: Models of Species Discovery; **ACO**: Ant Colony Optimization; **SA**: Simulated Annealing; **NN**: Neural Network; **MTNN**: Multi-task Neural Networks; **GA**: Genetic Algorithm; **GD**: Gradient Descent; **MOO**: Multi-objective Optimization; **R**: Random.
set.: Seed Set Selection; seed.: Seed Schedule; byte.: Byte Schedule; mutation.: Mutation Schedule; rete.: Seed Retention;
○: Whitebox Fuzzing; ◐: Greybox Fuzzing; ●: Blackbox Fuzzing.
△: More sensitive code coverage.

CUSR'22-Fuzzing: A Survey for Roadmap

文献分类

Table 2: Summary of the main studies concerned with the relationship of test criteria and faults.

Author(s) [Reference]	Year	Test Criterion	Summary of Primary Scientific Findings
Inozemtseva & Holmes [61]	'14	statement, branch, modified condition	There is a correlation between coverage and test effectiveness when ignoring the influence of test suite size. This is low when test size is controlled.
Just *et al.* [43]	'14	statement, mutation	Both mutation and statement coverage correlate with fault detection, with mutants having higher correlation.
Gopinath *et al.* [62]	'14	statement, branch, block, path	There is a correlation between coverage and test effectiveness. Statement coverage predicts best the quality of test suites.
Ahmed *et al.* [63]	'16	statement, mutation	There is a weak correlation between coverage and number of bug-fixes
Ramler *et al.* [64]	'17	strong mutation	Mutation testing provides valuable guidance towards improving the test suites of a safety-critical industrial software system
Chekam *et al.* [6]	'17	statement, branch, weak & strong mutation	There is a strong connection between coverage attainment and fault revelation for strong mutation but weak for statement, branch and weak mutation. Fault revelation improves significantly at higher coverage levels.
Papadakis *et al.* [41]	'18	mutation	Mutation score and test suite size correlate with fault detection rates, but often the individual (and joint) correlations are weak. Test suites of very high mutation score levels enjoy significant benefits over those with lower score levels.

Table 2: Summary of the main studies concerned with the relationship of test criteria and faults.

Author(s) [Reference]	Year	Test Criterion	Summary of Primary Scientific Findings
Chen *et al.* [50]	01	Block	coverage can be used for predicting the software failures in operation.
Andrews *et al.* [42]	'06	block, c-uses, p-uses, branch	Block, c-uses, p-uses and branch coverage criteria correlate with test effectiveness.
Namin & Andrews [51]	'09	block, c-uses, p-uses, branch	Both test suite size and coverage influence (independently) the test effectiveness
Li *et al.* [52]	'09	prime path, branch, all-uses, mutation	Mutation testing finds more faults than prime path, branch and all-uses.
Papadakis & Malevris [53]	'10	Mutant sampling, 1^{st} & 2^{nd} order mutation	1^{st} order mutation is more effective than 2^{nd} order and mutant sampling. There are significantly less equivalent 2nd order mutants than 1^{st} order ones.
Ciupa *et al.* [54]	'11	Random testing	Random testing is effective and has predictable performance.
Kakarla *et al.* [55]	'11	mutation	Mutation-based experiments are vulnerable to threats caused by the choice of mutant operators, test suite size and programming language.
Wei *et al.* [56]	'12	Branch	Branch coverage has a weak correlates with test effectiveness.
Baker & Habli [57]	'13	statement, branch, MC/DC, mutation, code review	Mutation testing helps improving the test suites of two safety-critical systems by identifying shortfalls where traditional structural criteria and manual review failed.

研究框架

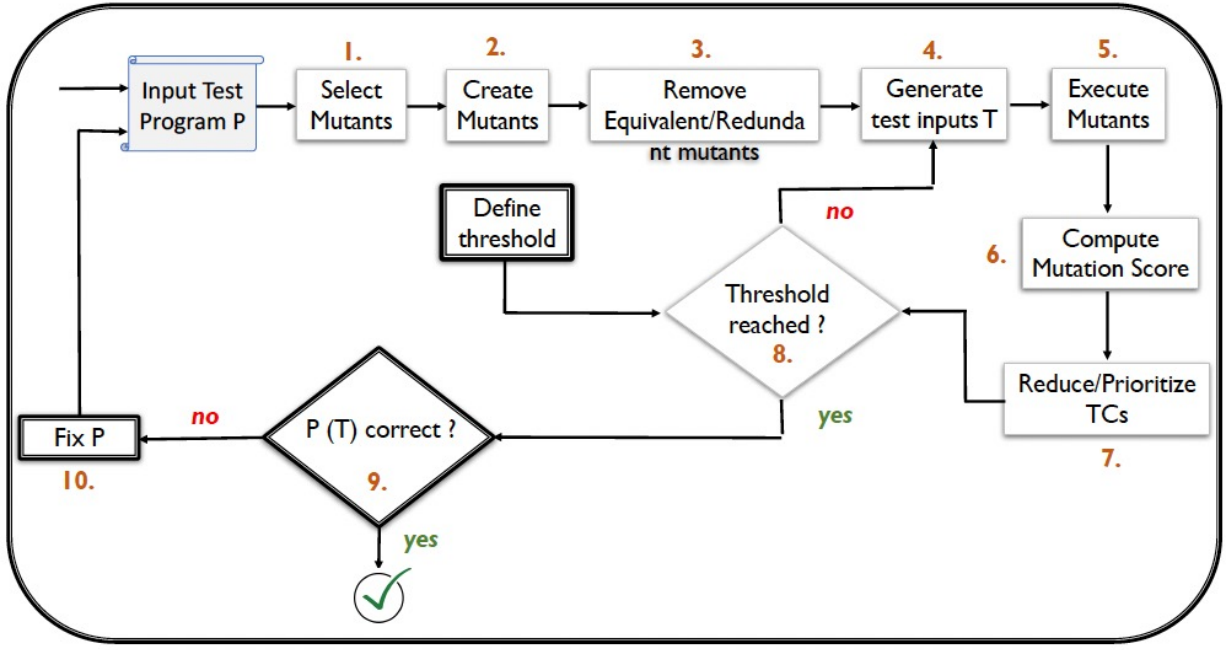


Figure 3: Modern Mutation Testing Process. The process forms an adaptation of the Offutt's and Untch's proposition [26] based on the latest advances in the area. Bold boxes represent steps where human intervention is mandatory.

Elsevier,AIC'19-Mutation Testing Advances: An Analysis and Survey

研究框架

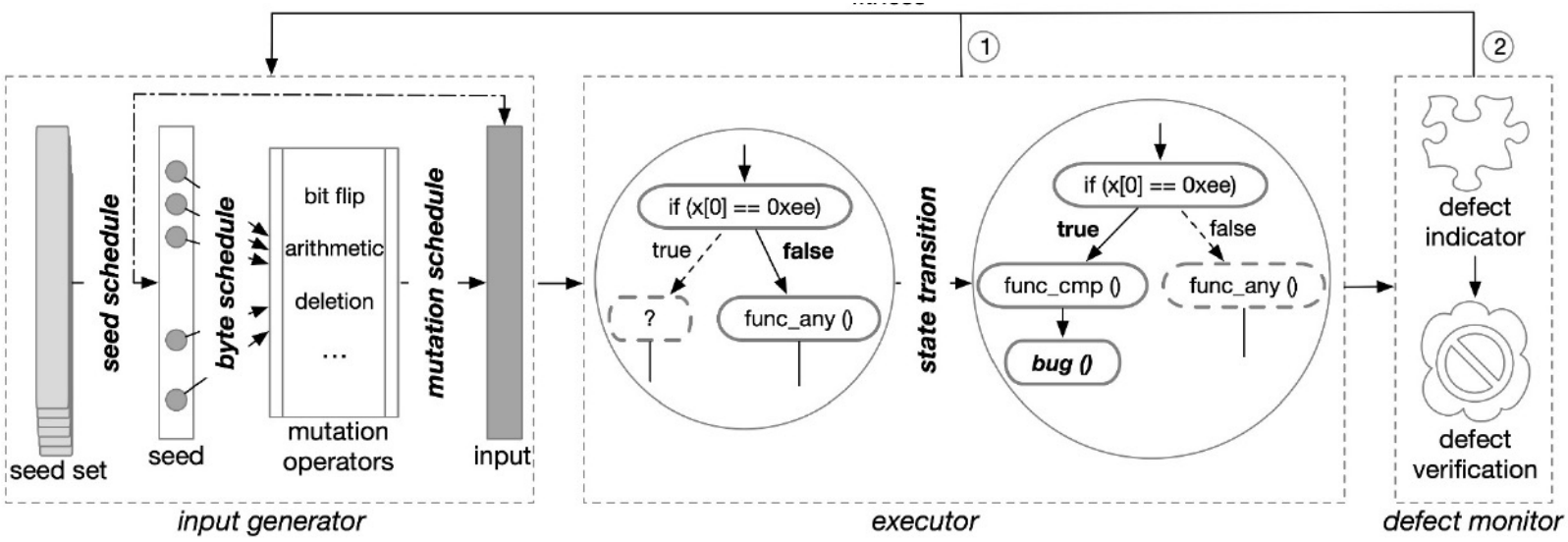


Fig. 2. General workflow of fuzzing. Essentially, fuzzing consists of three components: input generator, executor, and defect monitor.

CUSR'22-Fuzzing: A Survey for Roadmap

研究框架

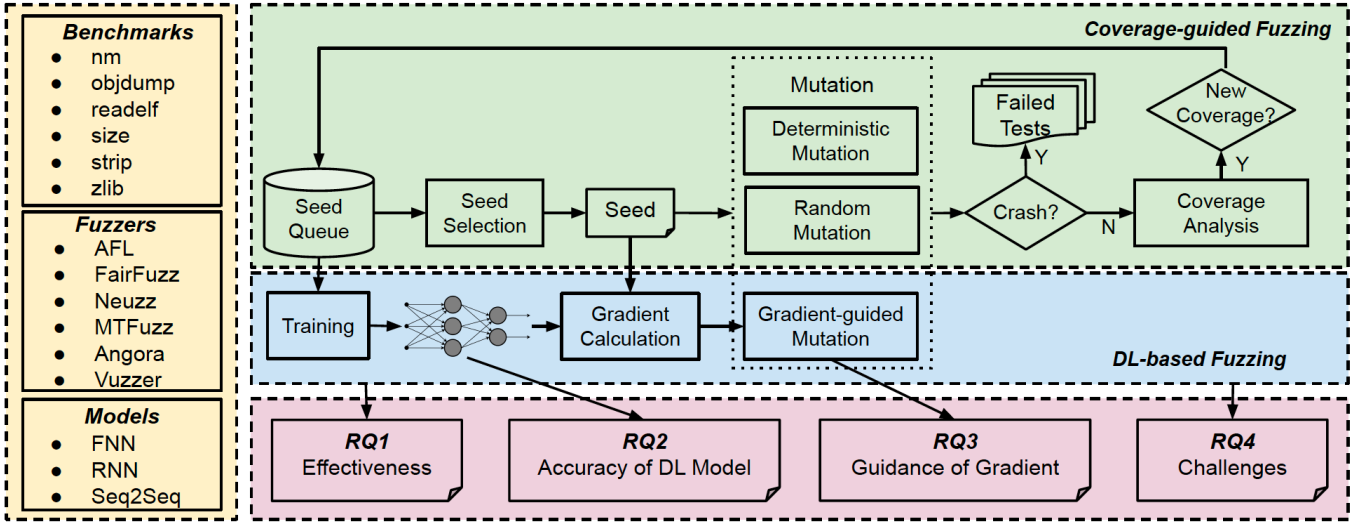


Fig. 1: Overview of our study. DL-based fuzzing can be regarded as a "plugin" into the coverage-guided fuzzing (gree region), which takes runtime coverage information as the dataset to train a deep learning model, and use the model gradients to guide mutation (blue region). In this study, we ask research questions (RQ1-4) revolving around different aspects of DL-based fuzzing (purple region), i.e., overall effectiveness on coverage (RQ1), the model prediction accuracy (RQ2), the effectiveness of gradient to provide feedback (RQ3), and the summarized challenges (RQ4).

TDSC'22-Deep Learning for Coverage-Guided Fuzzing: How Far Are We?

zychen@nju.edu.cn
fangchunrong@nju.edu.cn

Thank you!