CS 4301

Machine Learning in Cyber Security

Wei Yang

Department of Computer Science
University of Texas at Dallas

Outline

- Overview
- Course Content
- Sample presentation by the instructor

Outline

- Overview
- Course Content
- Sample presentation by the instructor

Instructor

- Instructor: Wei Yang
- Office: ECSS 4.225
- Email: wei.yang@utdallas.edu
- Homepage: http://youngwei.com/

Background


SHANGHAI JIAO TONG UNIVERSITY
1896


THE UNIVERSITY OF CALIFORNIA BERKELEY
LET THERE BE LIGHT
1868


THE UNIVERSITY OF TEXAS AT DALLAS
UTD
EST. 1969

Research Interests

- **MobileSecurity** (WHYPER, Pluto, AppContext[1][2][3], Telemade, MRV, CLAP[1], EnMobile, MalScan)
- **Automated Testing** (ORBIT, WCTester[1][2], NMTtest[1], REINAM)
- **SE/Security for Machine Learning** (PerInv, MRV, Telemade, NMTtest)
- **ML/NLP for SE/Security** (WHYPER, Pluto, CLAP, SemRegex[1], REINAM)
- **IoT Security** (iRuler)

SE and Security in UT Dallas

Rank institutions in the USA by publications from 2010 to 2020

All Areas [off | on]

AI [off | on]

- ▶ Artificial intelligence
- ▶ Computer vision
- ▶ Machine learning & data mining
- ▶ Natural language processing
- ▶ The Web & information retrieval

Systems [off | on]

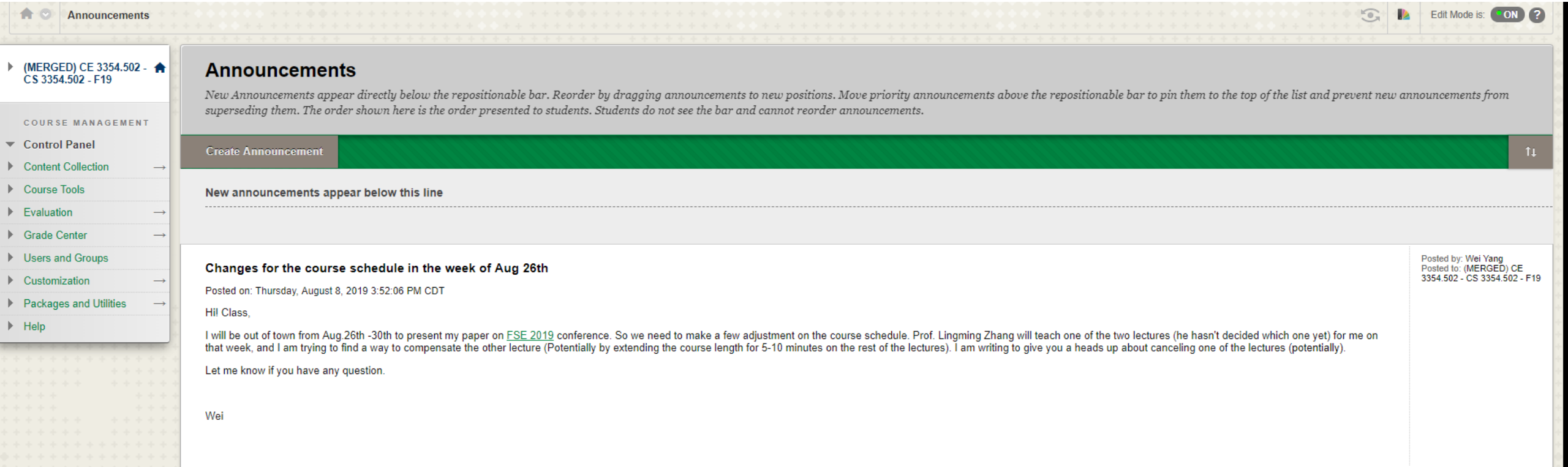
- ▶ Computer architecture
- ▶ Computer networks
- ▶ Computer security ☑
- ▶ Databases
- ▶ Design automation
- ▶ Embedded & real-time systems
- ▶ High-performance computing
- ▶ Mobile computing
- ▶ Measurement & perf. analysis
- ▶ Operating systems
- ▶ Programming languages
- ▶ Software engineering ☑

#	Institution	Count	Faculty
1	▶ Carnegie Mellon University	20.0	25
2	▶ Univ. of Illinois at Urbana-Champaign	14.1	29
3	▶ Purdue University	12.3	14
4	▶ Georgia Institute of Technology	12.0	21
5	▶ University of California - Irvine	11.7	15
6	▶ University of California - Berkeley	11.6	19
7	▶ North Carolina State University	9.8	14
8	▶ University of California - Santa Barbara	9.6	7
9	▶ Columbia University	9.0	14
9	▶ University of Texas at Dallas	9.0	12
11	▶ Pennsylvania State University	8.6	11
11	▶ University of California - San Diego	8.6	20
13	▶ Northeastern University	8.1	18
13	▶ University of Michigan	8.1	21
15	▶ University of Washington	7.9	15
16	▶ Massachusetts Institute of Technology	7.7	14

Electronic communication

- https://elearning.utdallas.edu

- Announcements, questions, answers



Textbook and Readings

- Textbook: https://d2l.ai/
- Some (text)books recommended
  - See Course Syllabus

- Required reading linked from Schedule

Grading

- Paper Reading and Presentation: 60%
- Project: 30%
- Class Participation: 10%

- For fairness, we REPORT all cheating

Paper Reading and Presentation

- Select a topic from the list of topics

- Students need to select the at least three topics as presenter, and another three topics as reviewer.

- Each topic has two or three presenters and two or three reviewers

Roles

Presenter (70%)

- Presenters need to work in group and present the papers as an inherent topics instead of just individual papers.
- The content of the presentation should mention a wide range of related work on the topic instead of just the papers listed.
- The selected papers are most recent ones. Presenter should at least introduce some of the selected papers in detail.
- Most of the selected papers have slides or presentation videos from paper authors. Presenters should go through the video or slides first.
- Presenters need to submit their slides three days before the presentation for the instructor to review.
- Presenters need to revise their slides based on the comments received from the instructor before class and after the first class.
- Each presentation will be divided into two lectures, at the end of the first lecture and at the beginning of the second lecture.

Roles

Reviewer (20%)

- Reviewers should be able to answer the questions from other students and the instructor. The instructor will direct some of the questions to the reviewers instead of presenters to inspire discussion.

- In order to answer the questions, the reviewers need to:
  - Read the papers in detail
  - View the presenters' slides and paper authors' video before class

Roles

Other students (10%)

- Other students need to submit at least one question about the topic to be presented before class.
- Students should submit their questions one day before class.
- If students forget to submit the question, they can compensate by asking questions on class.
- For students that have asked questions on class, find the instructor after class to log their question asking status.

Format of Lecture

- Each Lecture is divided into three parts (except the first two).
- 10:00 – 10:20 Q&A on prior presentation
- 10:20 – 11:40 Teaching on the next topic by Instructor
- 11:45 – 12:15 Presentation on the next topic

Outline

- Overview
- Course Content
- Sample presentation by the instructor

Machine Learning in Physical World

Autonomous Driving

Healthcare

Smart City

Malware Classification

Fraud Detection

Biometrics Recognition

Security & Privacy Problems

Sections

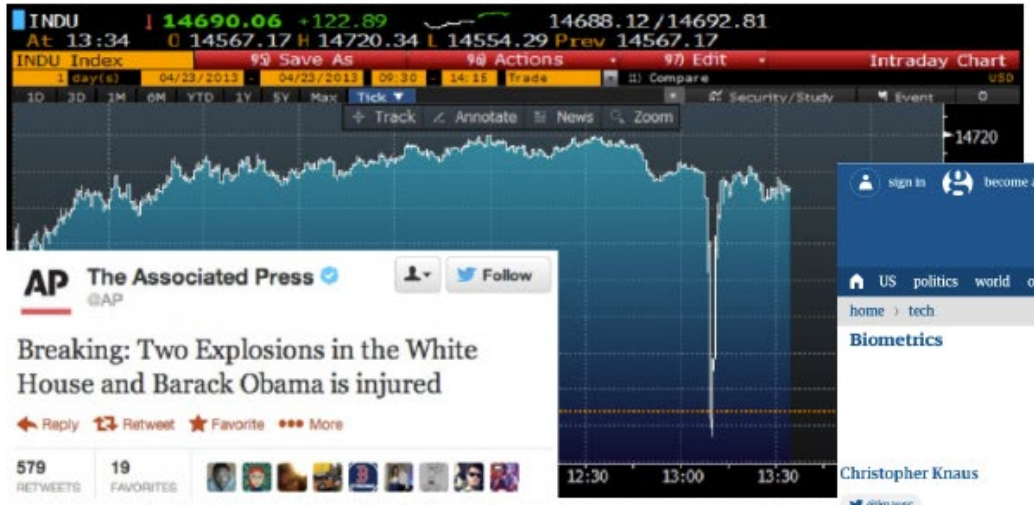
The Washington Post

WorldViews

Syrian hackers claim AP hack that tipped stock market by $136 billion. Is it terrorism?

By Max Fisher April 23, 2013

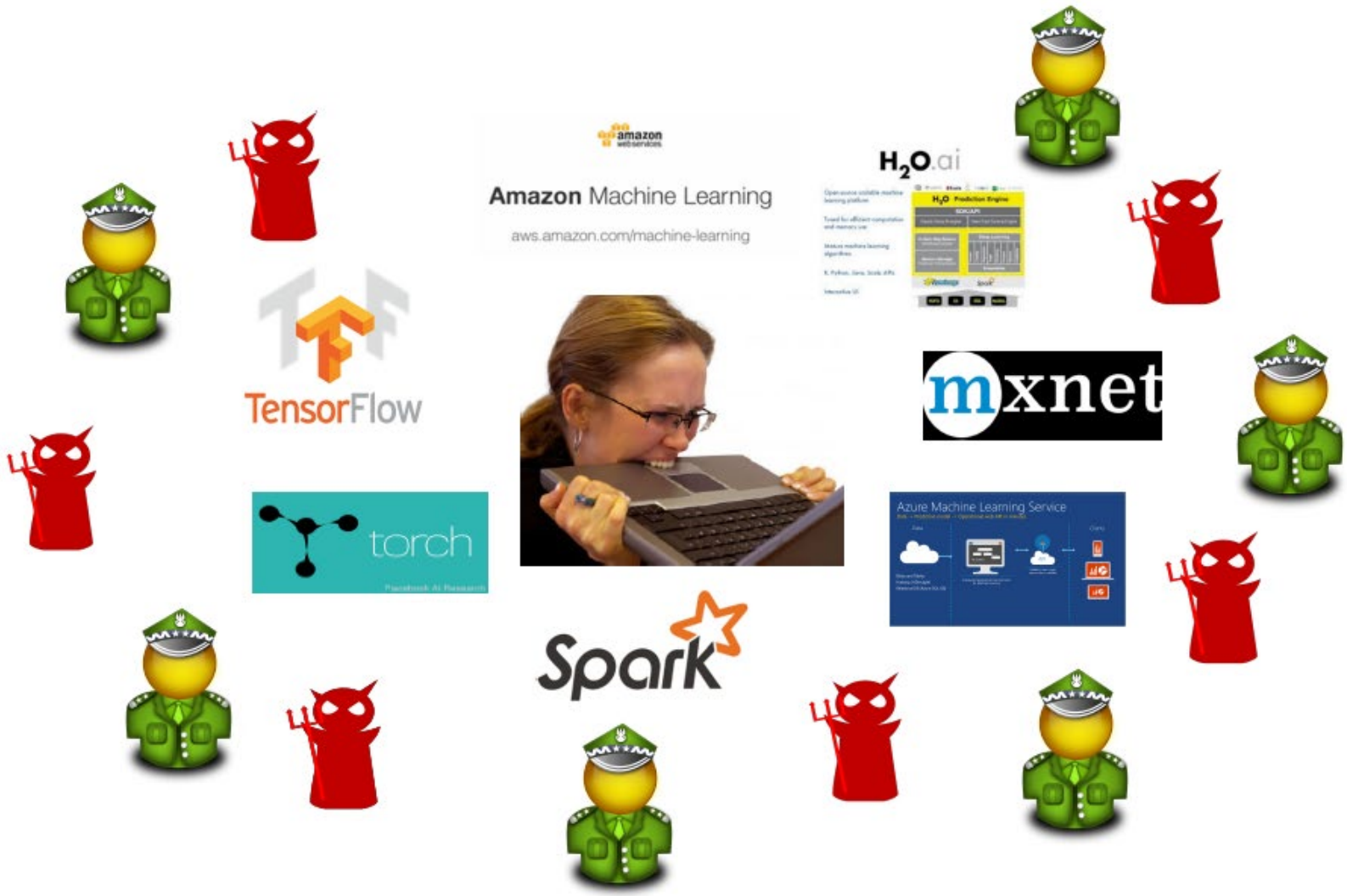
Security Problems



This chart shows the Dow Jones Industrial Average during Tuesday afternoon's drop, caused by a fake A

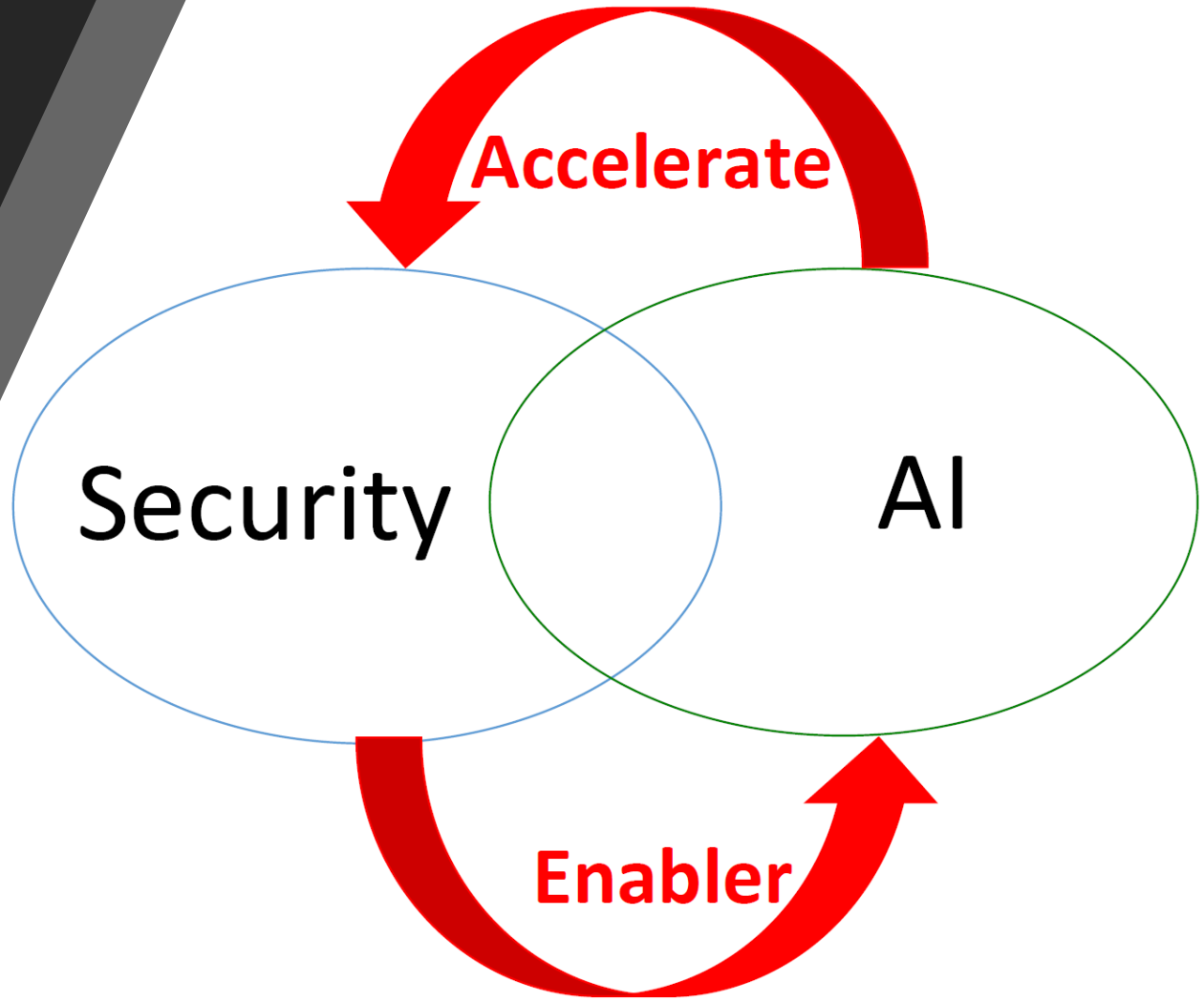
Privacy Concerns

We Are in Adversarial Environments



Vision: AI and Security

- Security enables better AI
  - Integrity
  - Confidentiality/Privacy
- AI accelerate Security Evolution

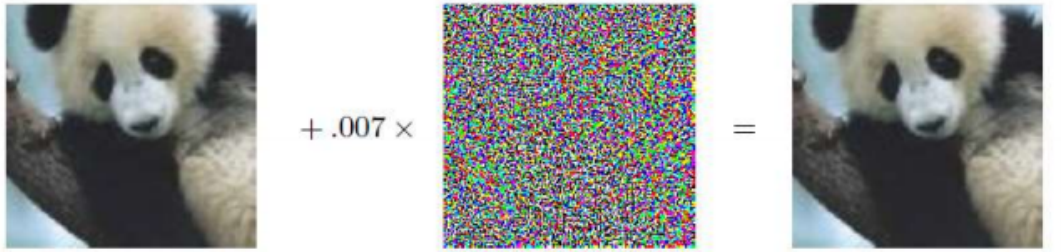


Course Contents

- The **general goal** is to understand the state-of-the-art.
- Security Issues in Machine Learning
  - Adversarial Machine Learning (Evasion Attack, Poisoning Attack)
  - Privacy in Machine Learning Models (Inference Attack)
  - Interpretability of Machine Learning Models
  - Fairness of Machine Learning
- Using Machine Learning to Address Security Problems
  - Probabilistic Inference to Ensure Security (Spire, DP-Finder, Bayonet)
  - Machine Learning models for security applications (DeGuard, JSNice, and DeBin)
  - Secure Machine Learning vs. Blockchain (Ethereum, Solidity, Securify)
  - Automated Deep Verification Systems (Dagger)

Security Issues in Machine Learning

Noisy attack: vision system thinks we now have a gibbon...



x

"panda"
57.7% confidence

$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"nematode"
8.2% confidence

$x + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"gibbon"
99.3 % confidence

Explaining and Harnessing Adversarial Examples, ICLR '15

Tape pieces make network predict a 45mph sign

Robust Physical-World Attacks on Deep Learning Visual Classification, CVPR'18

Self-driving car: in each picture one of the 3 networks makes a mistake...

DRV_C1: right DRV_C2: right DRV_C3: right

DeepXplore: Automated Whitebox Testing of Deep Learning Systems, SOSP'17

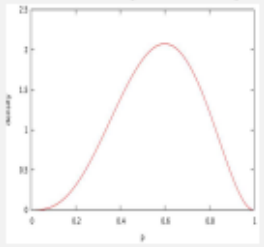
Machine Learning for Security Problems

Probabilistic Programming [psisolver.org]

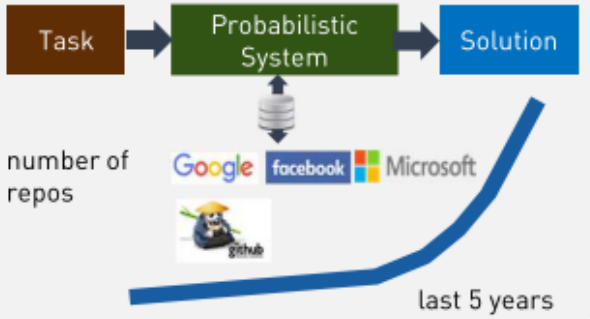
Probabilistic Program

```
def main() {
 p := Uniform(0,1);
 r := [1,1,0,1,0];
 for  i in [0..r.len]
  observe(Bern(p) == r[i]);
 return p;
}
```

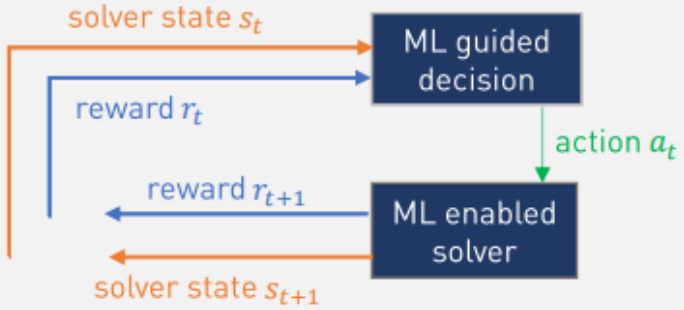
Probability Density



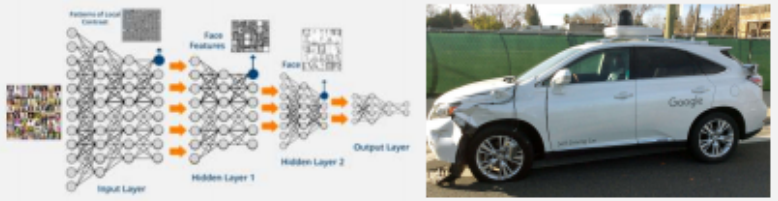
ML for Big Code [deepcode.ai]



ML-guided Solvers [fastsmt.ethz.ch/]

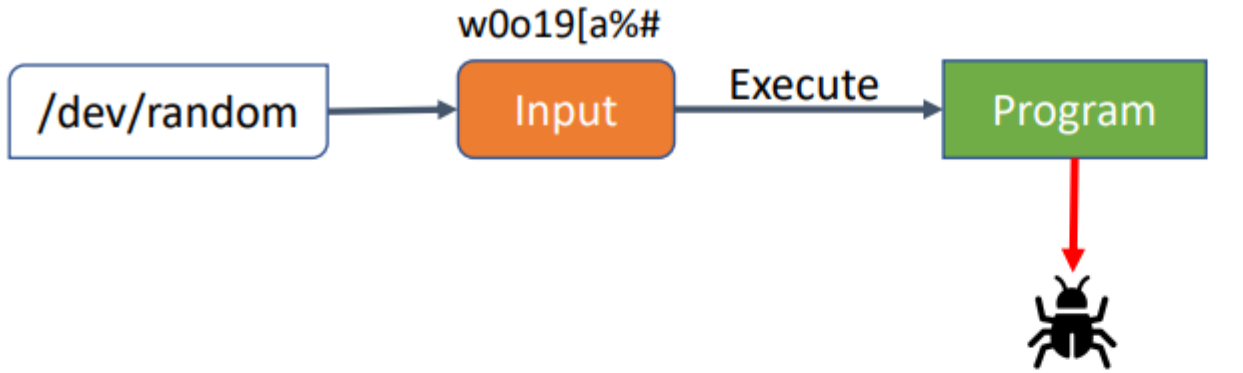


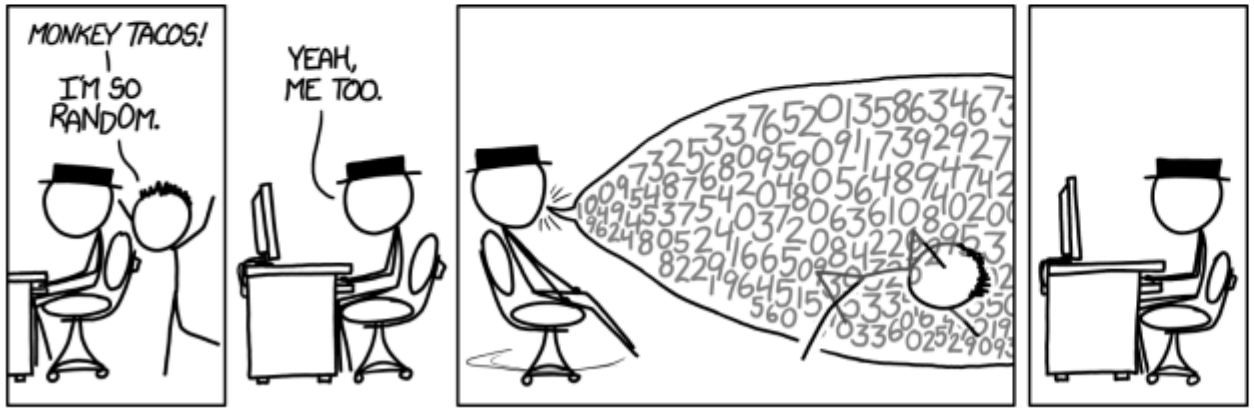
Trusted Artificial Intelligence [safeai.ethz.ch]



List of topics

Fuzzing





Bugs = Crashes (segfaults, aborts, etc.)

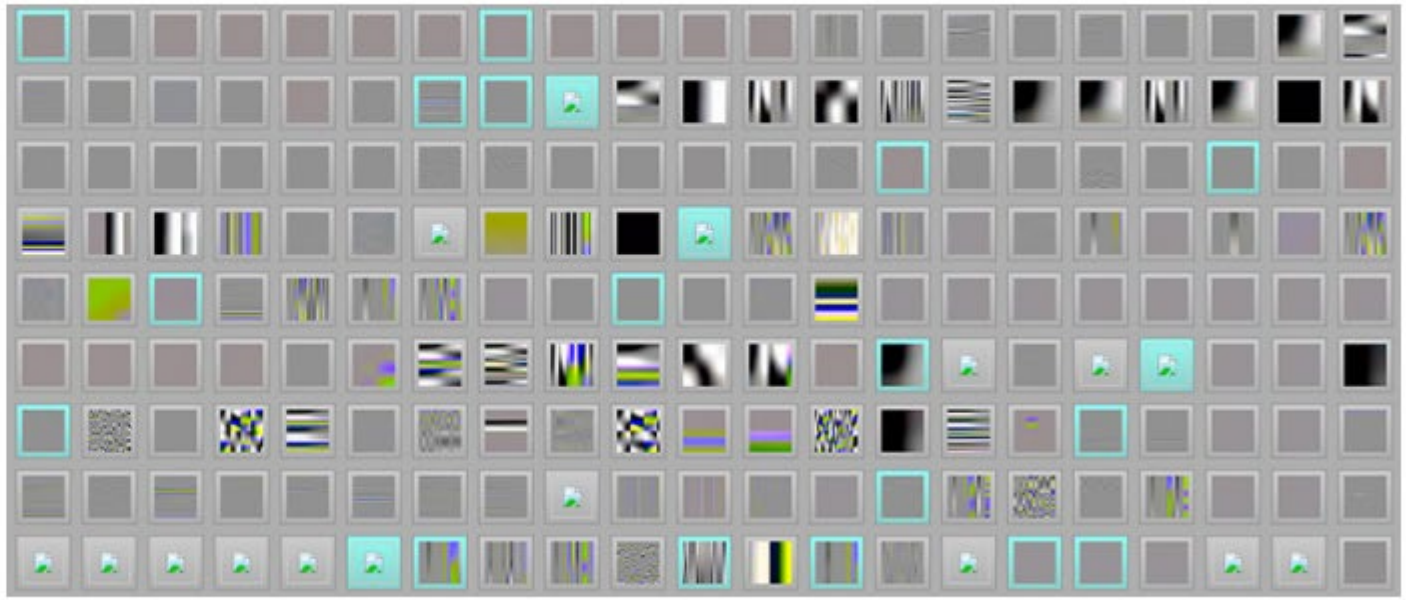
Fuzzing

November 07, 2014

Pulling JPEGs out of thin air

This is an interesting demonstration of the capabilities of afl; I was actually pretty surprised that it worked!

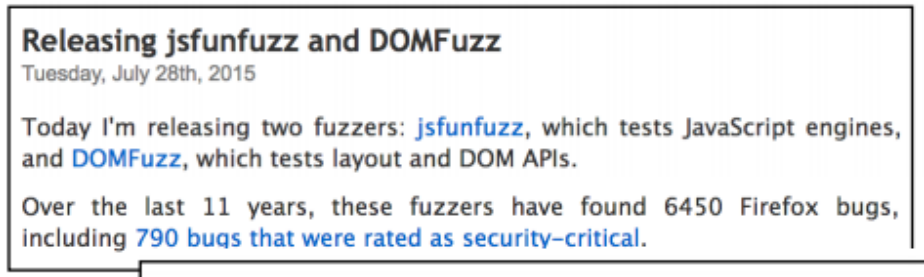
```
$ mkdir in_dir
$ echo 'hello' >in_dir/hello
$ ./afl-fuzz -i in_dir -o out_dir ./jpeg-9a/djpeg
```

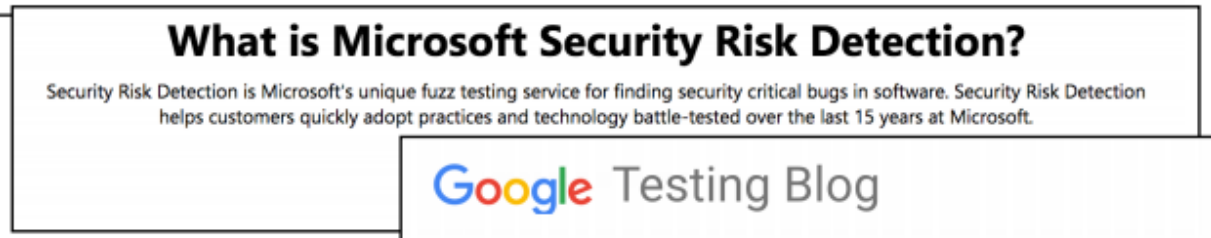


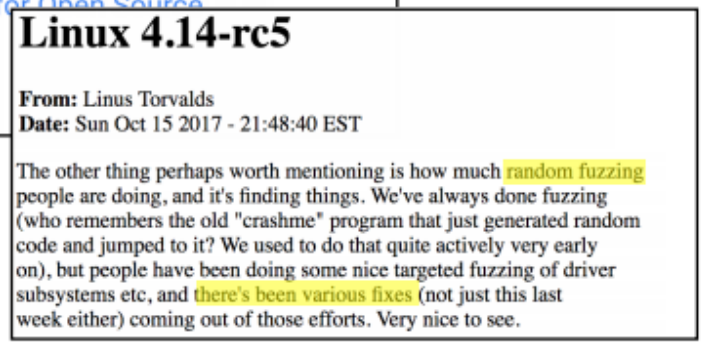
Fuzzers

- AFLFast [CCS 2016]
- Driller [NDSS 2016]
- AFLGo [CCS 2017]
- Vuzzer [NDSS 2017]
- Steelix [FSE 2017]
- SlowFuzz [CCS 2017]
- PerfFuzz [ISSTA 2018]
- FairFuzz [ASE 2018]
- Angora [IEEE S&P 2018]
- T-Fuzz [IEEE S&P 2018]
- NEUZZ [IEEE S&P 2019]
- Nautilus [NDSS 2019]
- Redqueen [NDSS 2019]
- Superion [ICSE 2019]
- MOPT [Usenix Sec 2019]
- GRIMOIRE [Usenix Sec 2019]
- MemFuzz [ICST 2019]
- Zest [ISSTA 2019]
- DifFuzz [ICSE 2019]
- AFLSmart [IEEE TSE 2019]
- FuzzChick [OOPSLA 2019]
- …

Fuzzing for Security







CVE-2014-6277: “ShellShock” bug in Bash

CVE-2014-0160: “Heartbleed” bug in OpenSSL

CVE-2015-1606
CVE-2015-1607
CVE-2014-9087
CVE-2014-6355
CVE-2015-0061
CVE-2015-7855
CVE-2016-7434
CVE-2015-7941
CVE-2015-8035
CVE-2015-8241
CVE-2015-8242
CVE-2015-8317
CVE-2016-4658
CVE-2016-5131
CVE-2015-5309
CVE-2015-5311
CVE-2015-0232
CVE-2017-5340
CVE-2015-2158
CVE-2015-0860
CVE-2015-8380
CVE-2016-1925

Uncertainty in ML

Photo-Essay

BIRTH:

September 4, 1936, Tel Aviv.

EDUCATION:

B.S., Electrical Engineering (Technion, 1960); M.S., Electronics (Newark College of Engineering, 1961); M.S., Physics (Rutgers University, 1965); Ph.D., Electrical Engineering (Polytechnic Institute of Brooklyn, 1965).

EXPERIENCE:

Research Engineer, New York University Medical School (1960–1961); Instructor, Newark College of Engineering (1961); Member of Technical Staff, RCA

JUDEA PEARL

United States – 2011

CITATION

For fundamental contributions to artificial intelligence through the development of a calculus for probabilistic and causal reasoning.

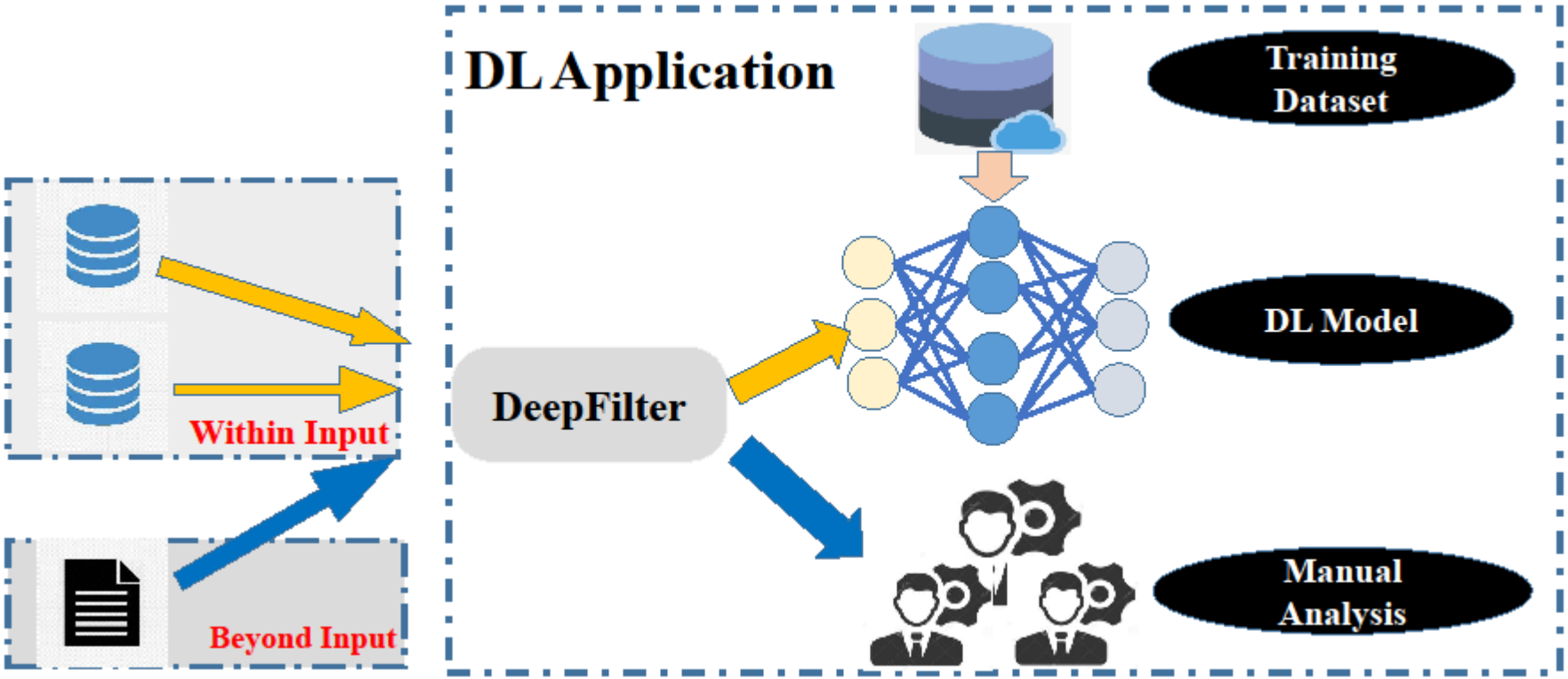
SHORT ANNOTATED BIBLIOGRAPHY | ACM TURING AWARD LECTURE VIDEO | RESEARCH SUBJECTS | ADDITIONAL MATERIALS | VIDEO INTERVIEW

Judea Pearl created the representational and computational foundation for the processing of information under uncertainty.

He is credited with the invention of *Bayesian networks*, a mathematical formalism for defining complex probability models, as well as the principal algorithms used for inference in these models. This work not only revolutionized the field of artificial intelligence but also became an important tool for many other branches of engineering and the natural sciences. He later created a mathematical framework for causal inference that has had significant impact in the social sciences.

Judea Pearl was born on September 4, 1936, in Tel Aviv, which was at that time administered under the British Mandate for Palestine. He grew up in Bnei Brak, a Biblical town his grandfather went to reestablish in 1924. In 1956, after serving in the Israeli army and joining a Kibbutz, Judea decided to study engineering. He attended the Technion, where he met his wife, Ruth, and received a B.S. degree in Electrical Engineering in 1960. Recalling the Technion faculty members in a 2012 interview in the *Technion Magazine*, he emphasized the thrill of discovery:

Uncertainty in ML



Responses from Computing Researchers to HUD's Implementation of the Fair Housing Act's Disparate Impact Standard

January 8th, 2020 / in **Announcements**, **CCC**, **policy**, **research horizons**, **Research News** / by **Helen Wright**

The following blog post is from Computing Community Consortium (CCC) Vice Chair Elizabeth Bradley (University of Colorado Boulder) and CCC Executive Council member Suresh Venkatasubramanian (University of Utah).

Algorithmic bias can be insidious, making it all but impossible to pinpoint factors that contribute to discrimination. This is particularly concerning in the context of high-stakes decisions. The new Department of Housing and Urban Development (HUD) guidelines around the use of algorithms to aid in housing decisions are an example of this. This HUD proposal acknowledges the existence of algorithmic bias but would shift much of the burden of proof to demonstrate discriminatory behavior back onto the plaintiffs, using standards for algorithmic transparency and explainability that seem unmoored from extant science about what we can hope to extract from algorithmic decision pipelines. Among other things, this would allow landlords and lenders to deflect lawsuits with an overly naive statistical approach, looking at individual factors rather than taking them in combination and thereby ignoring the potential collective effect of many lenders using the same third-party algorithm. Writing in Forbes, Elizabeth Fernandez suggests that this could undermine the Fair Housing Act.

Computing researchers who study these issues have submitted formal responses to the public call for comments regarding these new guidelines. These included a coordinated response by members of the GRAIL network, a new initiative led by the Center for Democracy and Technology (CDT) and the R Street Initiative. GRAIL's goal is to connect technical and policy experts to inform discussions around technology policy in Washington and provide deep, rapid responses to questions of tech policy. Their response, which was led by Natasha Duarte at CDT and involved CCC Council member Suresh Venkatasubramanian, details how the different components of the

Fairness – Regulation & Rules

- https://www.regulations.gov/document?D=HUD-2019-0067-0001

PR FR-6111-P-02 HUD's Implementation of the Fair Housing Act's Di

This Proposed Rule document was issued by the **Department of Housing and Urban Development** (HUD)

For related information, **Open Docket Folder**

Action

Proposed rule.

Summary

Title VIII of the Civil Rights Act of 1968, as amended (Fair Housing Act or Act), prohibits discrimination in the sale, rental, or financing of dwellings and in other housing-related activities on the basis of race, color, religion, sex, disability, familial status, or national origin. HUD has long interpreted the Act to create liability for practices with an unjustified discriminatory effect, even if those practices were not motivated by discriminatory intent. This rule proposes to amend HUD's interpretation of the Fair Housing Act's disparate impact standard to better reflect the Supreme Court's 2015 ruling in *Texas Department of Housing and Community Affairs* v. *Inclusive Communities Project, Inc.,* and to provide clarification regarding the application of the standard to State laws governing the business of insurance. This rule follows a June 20, 2018, advance notice of proposed rulemaking, in which HUD solicited comments on the disparate impact standard set forth in HUD's 2013 final rule, including the disparate impact rule's burden-shifting approach, definitions, and causation standard, and whether it required amendment to align with the decision of the Supreme Court in *Inclusive Communities Project, Inc.*

Course Project / Empirical Study

- Grading:
  - Proposal (5%)
  - Midterm Report (5%)
  - Midterm Demo (10%)
  - Final Report (20%)
  - Final Demo (10%)
  - Project Document (10%)
    - Project Website, README, Example subjects
  - Code Evaluation (10%)
    - Readability, Reusability
  - Effectiveness Evaluation (30%)
    - Evaluate based on the metrics proposed in the proposal.
    - Level of difficulty will be taken into consideration (e.g., achieving or exceeding the state of the art).

Course Project / Empirical Study

- Topic: Propose your own projects (Feel free to talk to the instructor about the proposed topic). I will provide a list of papers for you to get ideas.

- Important deadline
  - Proposal, Feb 1st.
  - Midterm Report & Demo, TBD.
  - Final Report & Demo, TBD.

- Content
  - Proposal: Introduction and related work
  - Midterm report: Motivating example, problem formulation, draft of approach and evaluation
  - Final report: Complete draft

How to select a good topic?

- The topic will be evaluated based on the potential impact of the project.
- There are different types of impacts: research, industrial, societal/social, …
- Research impact, e.g., impact on research colleagues in various forms -- citations, inspiration, opening a new field/direction, ...
- General, fundamental, conceptual ideas (beyond a tool, implementation, infrastructure, study..)
- Overreaching contributions conveyed as insights.

Two fundamental success element of your project

- Two main elements
  - Interesting idea(s) accompanying interesting claim(s)
  - claim(s) well validated with evidence
- Then how to define “interesting”?
  - Really depend on the readers’ taste but there may be general taste for a community
  - Ex: being the first in X, being non-trivial, contradicting conventional wisdoms, …
  - Can be along problem or solution space; in security (or software engineering field), being the first to point out a refreshing and practical problem would be much valued – Uniqueness, elegance, significance?

Detailed Desirable Characteristics I

- Crosscutting characteristics
- Interesting work, e.g., intriguing, unpredictable, surprising/unexpected
  - Ask interesting questions
  - Have interesting ideas in solution
  - Have interesting findings in evaluation
- Novel work, e.g., being the first
  - New problem
  - New solution
  - New findings

Detailed Desirable Characteristics II

- Inspiring work
  - General ideas (produced w/ research generalization, see later slides)
  - Problem formulation: general/abstract problem definition that could describe other concrete problems
  - Solution formulation: a general idea that could be used elsewhere
- Impactful work
  - Impactful problem: a real problem with
    - high severity level: impact an case seriously
    - large scope level: impact many cases
  - Impactful solution: an effective/efficient solution to well address the problem
    - E.g., many/high percentage (serious) (previously-undetected) vulnerabilities your approach finds
    - E.g., N man-hours that your approach saves
    - Great if having evidence of adoption in practice (You can ask your classmate for help to conduct empirical studies).

Detailed Desirable Characteristics III

Rigorous/accurate description

- Clear problem definition (no matter formalized or not)
  - inputs/outputs of the approach
  - requirements on the output
- Clear solution description with both algorithms and examples (don't use only examples!)
  - reach the level of reproducible (others could reimplement your approach with enough high-level design information)

Detailed Desirable Characteristics IV

Significant work (e.g., not easy problem to solve)

- Technical challenges (see later slides)
  - problem level
  - solution level
- Pose intellectual “stress” for whoever wants to address the problem

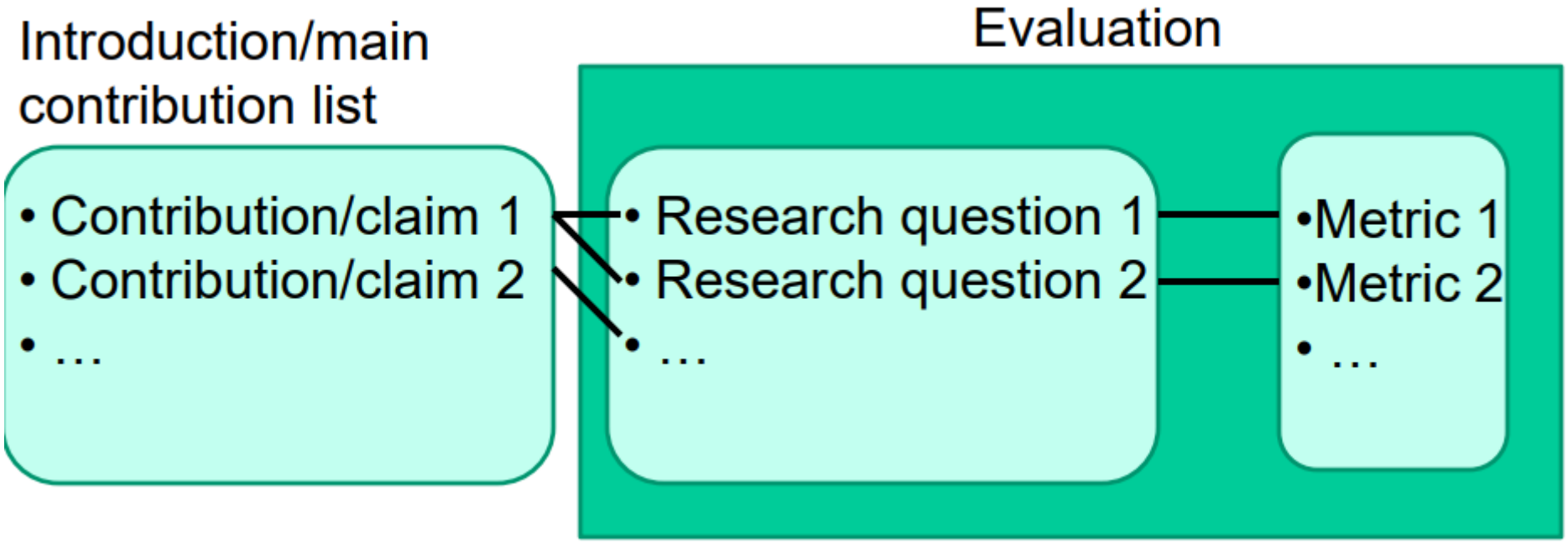
Validated work (Very important for our course project!!!)

- Clear and strong (empirical) evidence to validate/justify the claims

Key Questions to Double Check Your Project

- Is the research problem significant/important?
  - NOT: a problem created/imagined by you and no one else cares about it
  - YES: a problem that people care (evidenced by concrete statistics or examples)
- Is your research solution significant or addressing technical challenges? (may be less critical for some type of work)
  - NOT: a solution that is incremental over previous work
  - NOT: a solution that is straightforward/trivial (e.g., simple adoption or slight adaption of an existing technique is not significant enough, even when you are the first one in doing so)
- Is your evaluation justifying the claimed contributions or benefits of your solution? (e.g., faster, detecting more faults, …than existing techniques if any)
  - Double check by making traceability from your claims listed in your contributions to your research questions to investigate in your evaluation

Traceability Links (Important!)

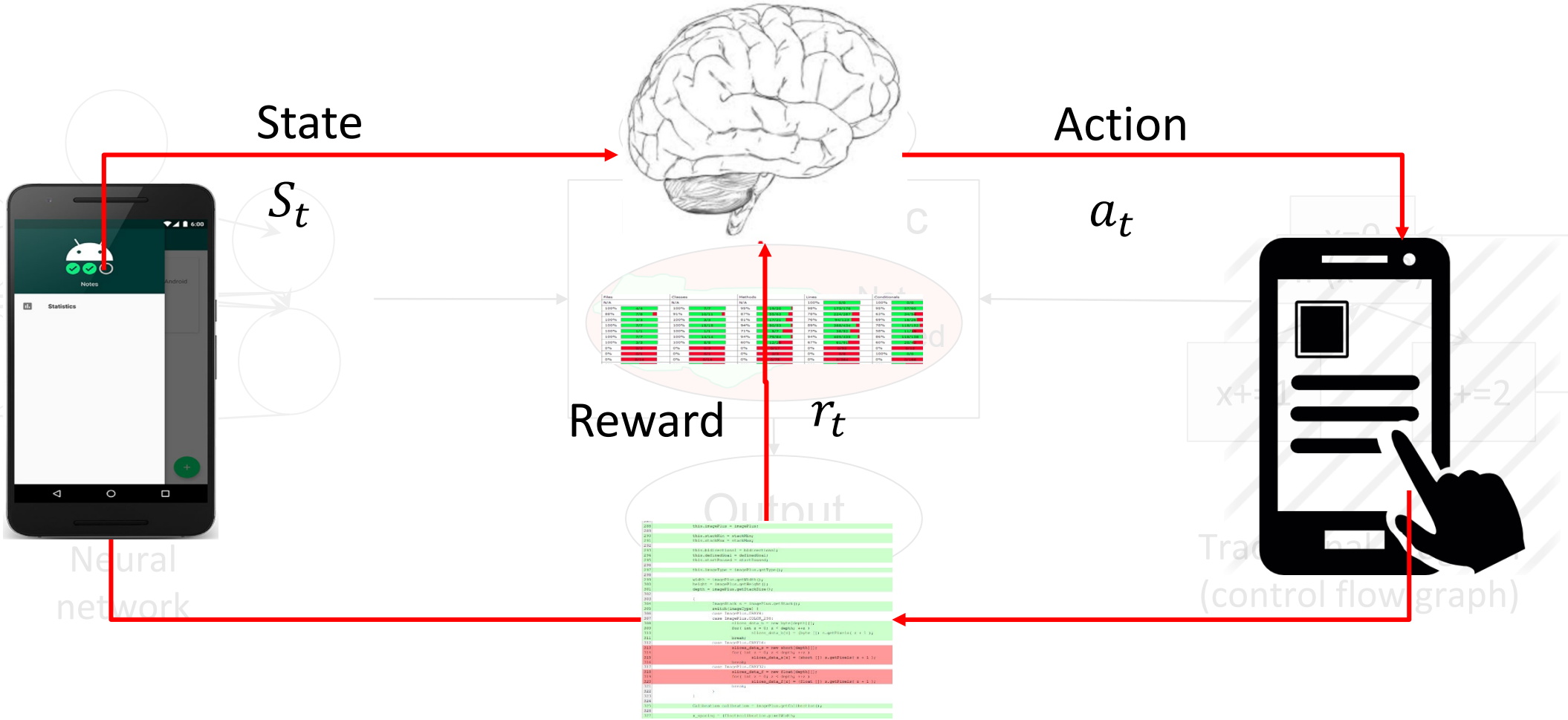


- Make sure each contribution/claim is translated to (appropriate) research question(s) -> no unsubstantiated claims
- Make sure each question is answered with help of (appropriate) metric(s)

Justify Your Choices

- Pitfall: In your proposal, you describe that you propose a way of solutions (e.g., dynamic analysis) to address your stated problem, BUT you never discuss why alternative way of solutions (e.g., static analysis) would not be chosen.
- Pitfall: When describing your approach, you describe that you use a technique (e.g., hierarchical clustering) to address a subproblem in your approach, BUT you never discuss why alterative way of techniques (e.g., partitional clustering) would not be chosen
- Pitfall: In your evaluation, you don't compare the results of including or not including an important technique (e.g., filtering) claimed to be a major contribution
- Pitfall: in your evaluation, you don't justify why you choose the experimental subjects or a subset of subjects used by previous work

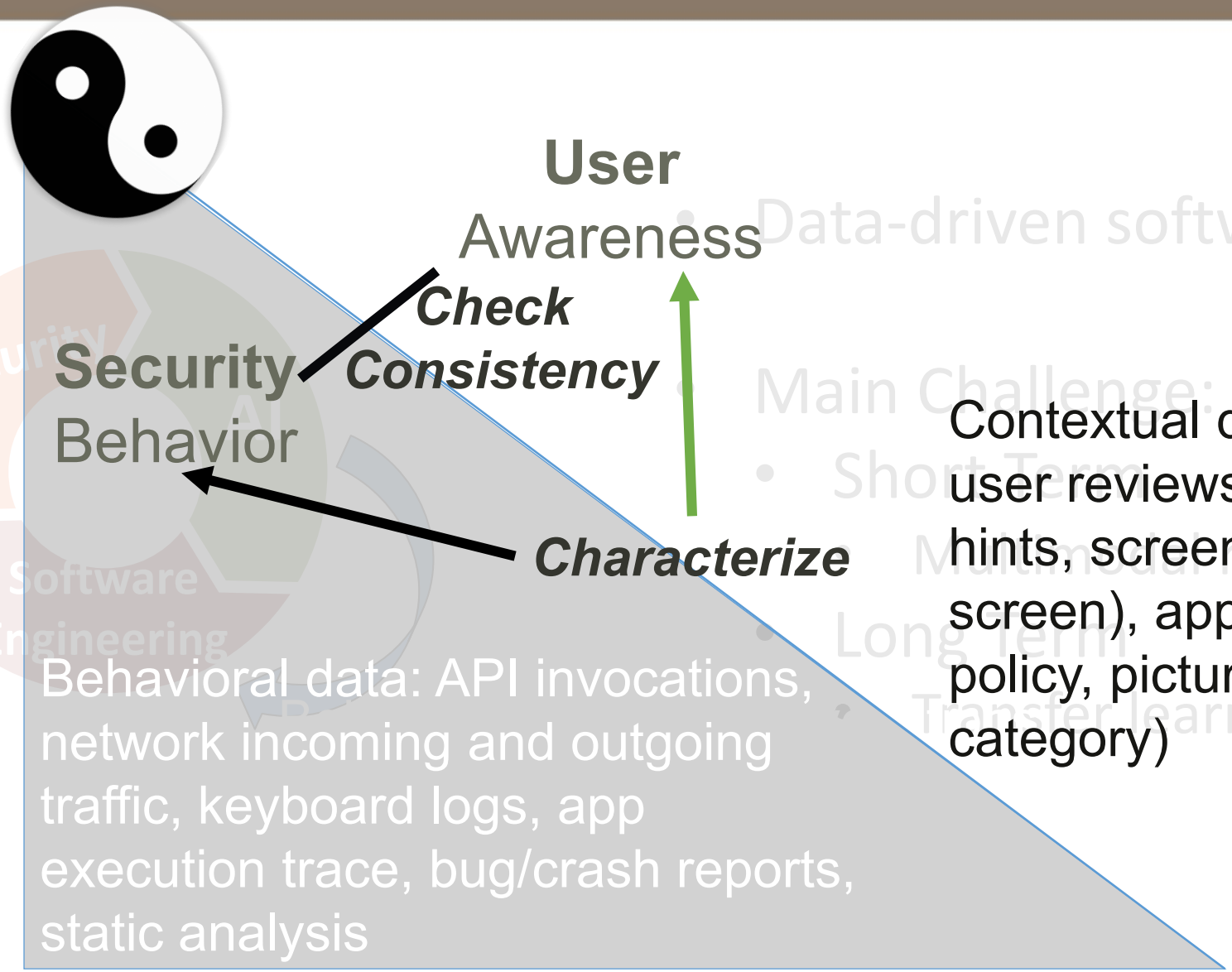
Potential Topics ——
UI testing agent with reinforcement learning



An Empirical Study of Android Test Generation Tools in Industrial Cases

Wang et al. ASE 2018

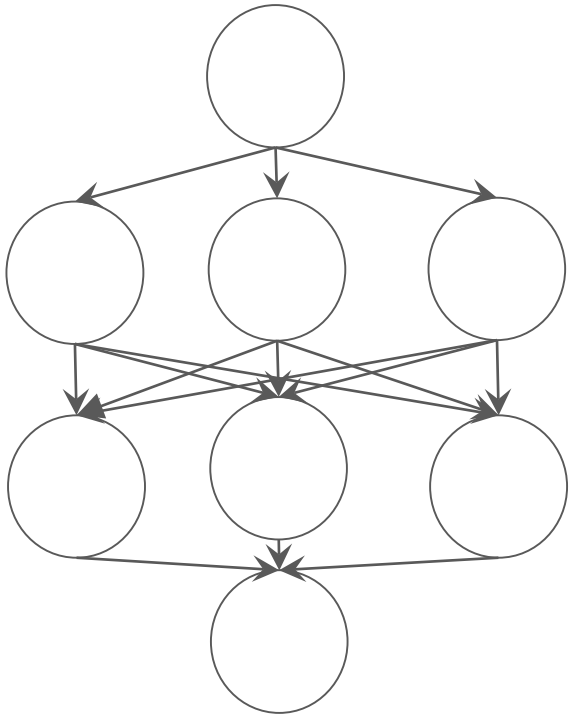
Potential Topic——
Yin-Yang view of data-driven app analysis



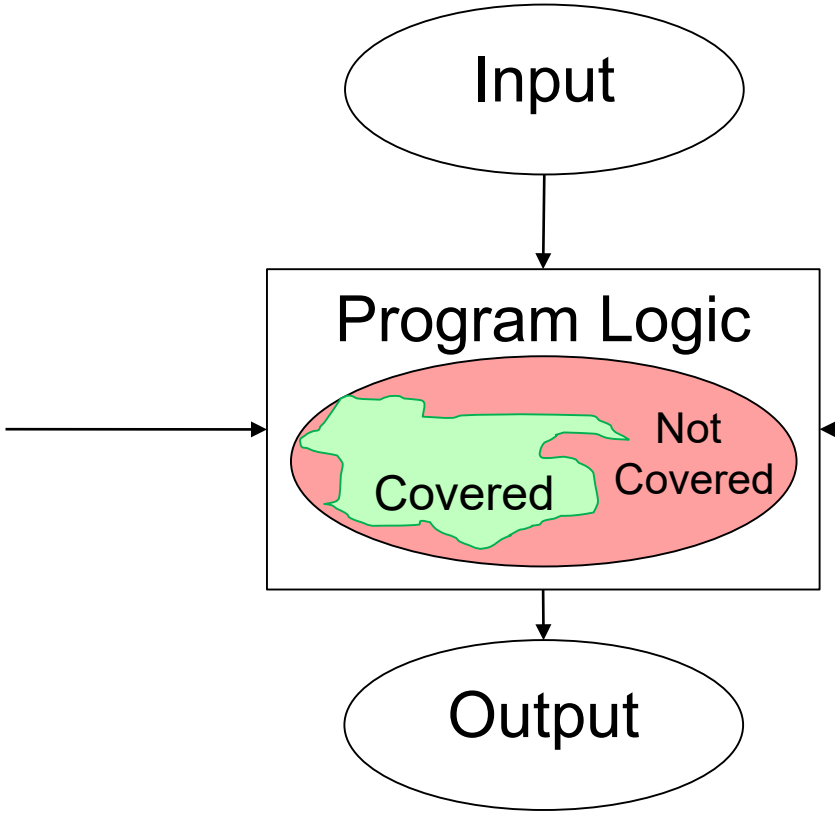
Contextual data: user usage data, user reviews, UI screen (labels, hints, screen, buttons, sequence of screen), app descriptions, privacy policy, pictures/videos, tags (app category)

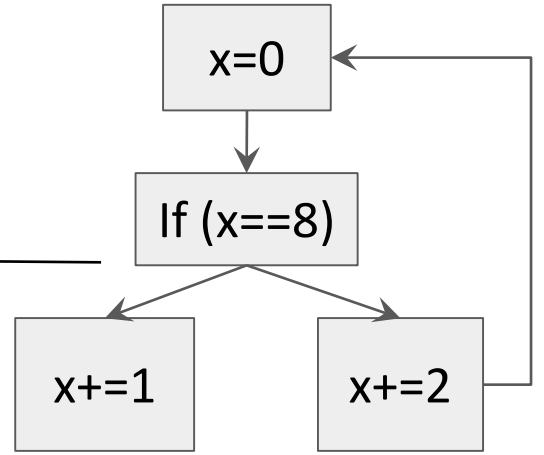
Behavioral data: API invocations, network incoming and outgoing traffic, keyboard logs, app execution trace, bug/crash reports, static analysis

Potential Topic —— Testing criterion



Neural network





Traditional program (control flow graph)

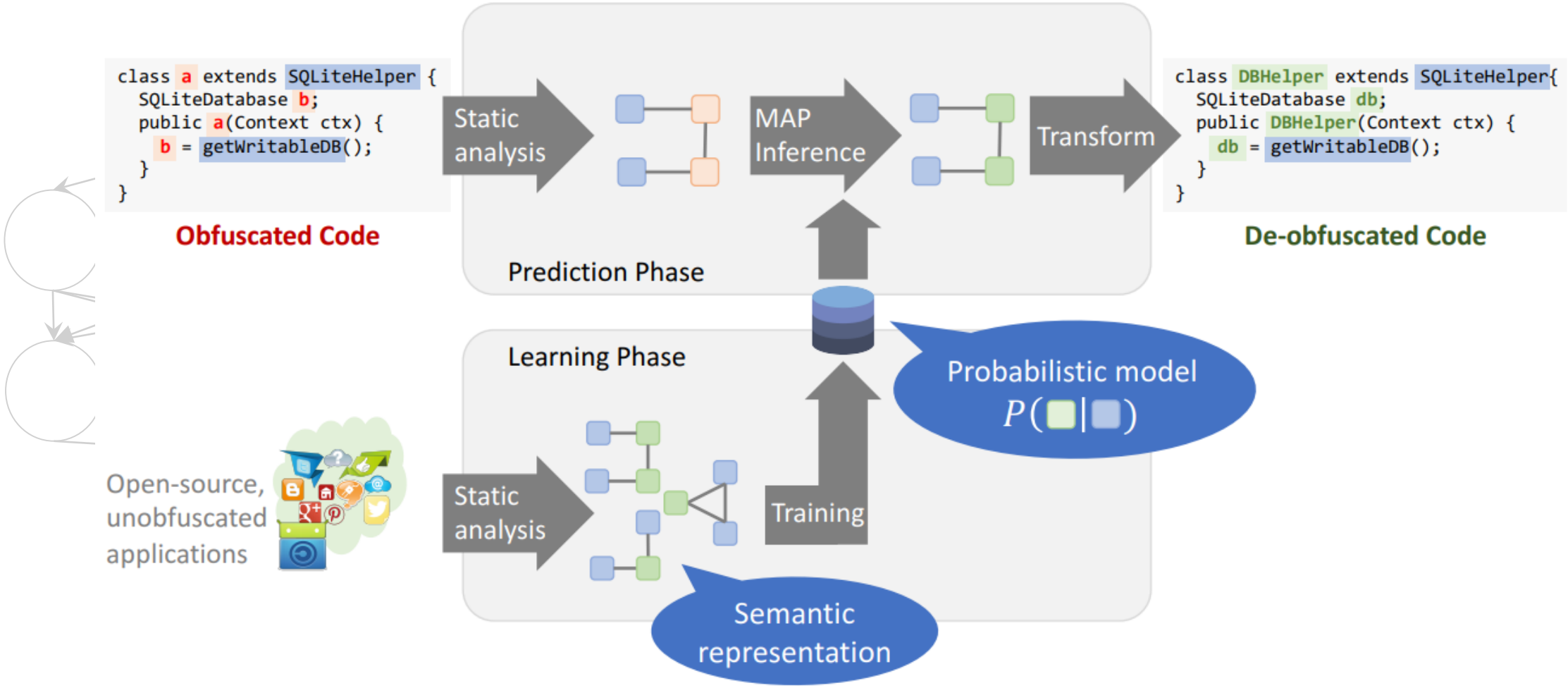
Potential Topic —— Testing criterion

Automatic Detection of Under- and Over-Translation in Neural Machine Translation

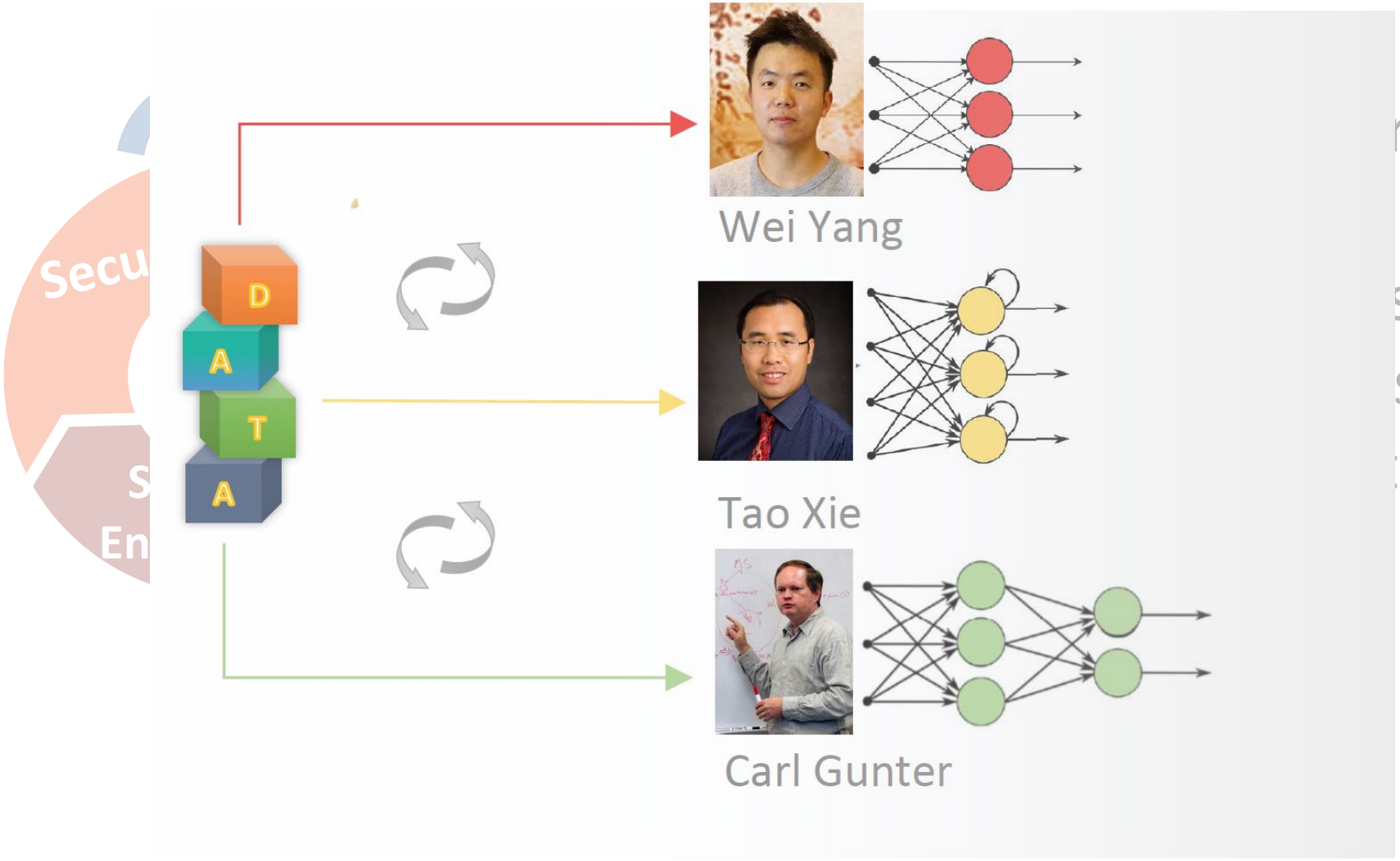
Peng et al.

Under submission

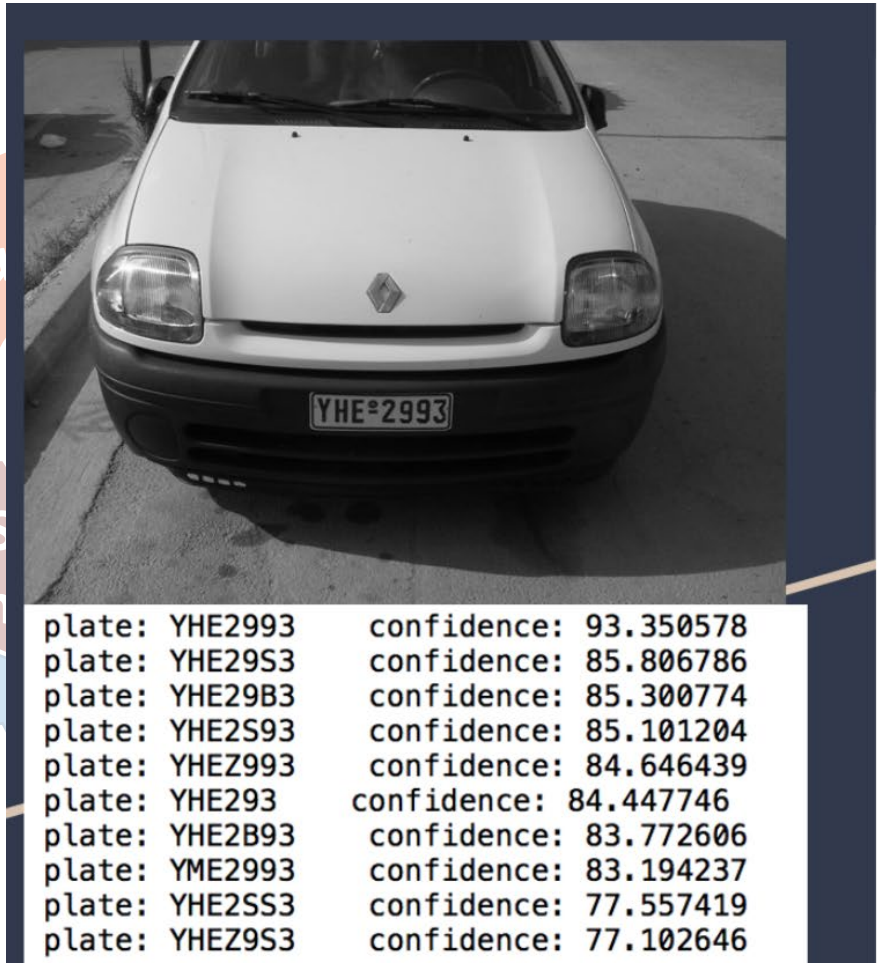
Potential Topic——
Code Obfuscation/De-obfuscation/Transformation



Future work – Defense by diversity



Future work——
Physical attacks to smart cities



Reference

- https://securify.chainsecurity.com/
- https://www.probfuzz.com/
- Rest of slides are detailed description about how to write a good report. Please read these slides offline.

Typical Report Structure

- Title/Abstract (**Proposal**)
- Introduction (**Proposal**)
- Background (**Proposal**)
- Related Work (**Proposal**)
- Formal Problem Definition (**Mid-term**)
- Example (**Mid-term**)
- Approach/Framework
- Implementation
- Evaluation
- Experiment/Case Studies/Experiences/Examples
- Discussion
- Conclusions (and Future work)

Proposal - Introduction Structure

- Long motivation, problem to be solved, why existing solutions are not sufficient (sometimes examples help)
- Need show the problem is significant (desirable to use concrete statistics, concrete examples, or citations)
- Proposed solution (inputs/outputs) and key ideas (steps)
- Optional: brief mention of related work if it is very related and explain differences
- Evaluation and evaluation results need to be added in final report.
- Optional: “The paper makes the following main contributions: + bulleted items”
  - Easy for reviewers to spot out major contributions
  - Being of the “first” in something is desirable as a contribution
- In the introduction section of your final report, list the structure layout of the paper (you want to give readers high level ideas how different parts are related to each other)
  - Similar principle applied throughout the paper for subsections

Proposal - Introduction Structure

- Don’t overclaim (even throughout the paper)!
  - But it is good to put your work in a bigger picture and a larger background
  - But it is important for you emphasize the significance of the problem and your solution (esp in intro)
- Similarly don’t over-criticize other’s work (even throughout the paper)!
- If you want to claim some unjustified points, it is better to put them in conclusion or discussion section
- Even if so, be careful on wording
  - X “Our approach provides a foundation for this new field.”
  - “We believe our approach can provide a foundation…”
  - “We believe our approach has a good potential for providing a foundation …”

Proposal - Introduction Structure

- Another example: be careful on wording
  - X “Our/X’s approach is the only/first one on ….”
  - “With the best of our knowledge, our/X’s approach is the only one/first on …”
  - “Our/X’s approach is one of the/a few approaches …”
  - “Our/X’s approach is a major/representative approach …”

- Some reviewers don’t like you to claim your own approach to be “novel” (at least don’t put “novel” in your paper title!) – they said novelty is to be judged by them not to be claimed by you
  - “TestEra: A Novel Framework for Automated Testing of Java Programs”
  - “TestEra: Specification-based Testing of Java Programs Using SAT”

Proposal – Writing Tips

- Iterate and improve the abstract and introduction in a small discussion group (e.g., read to others)

- Pay attention to the logical transitions in sentences in abstract and paragraphs in introduction section (e.g., using Mind Map: http://freemind.sourceforge.net/)

- Double check that earlier stated characteristics are satisfied
  - Ex. The target problem is significant/important
  - Ex. Your solution is significant/addressing non-trivial technical challenges, and is well validated

Proposal - Background

- Differences between background and related work
- You can organize related work with subsections or group them in several categories
- Background sometimes called Preliminaries
  - Includes notation, terminology, others' or your previous techniques that are not part of the contributions of this paper

Proposal - Related Work

- Don’t simply list related work without RELATING to your own work! – keywords to use: whereas, in contrast, but, however, …
  - “excuses” to use: “does not require specs”, “focus on different problems”, “complement with each other”, …
  - you can describe several similar related approaches together and compare them at once with yours
- Don’t just discuss the differences between your work with related work only in the solution space
  - Need to relate back to the effect/impact on the problem space
  - E.g., You may argue that your work uses dynamic analysis and related work uses static analysis --- but how would these two analysis types impact the problem you are addressing? Static analysis produces too many false warnings? … You need to compare them in terms of observable differences from the approaches’ user’s point of view in the problem space

Proposal - Related Work

- Don’t make unjustified unobvious criticisms on related work if you don’t have experimental results to back you up.
  - But you can cite others’ experiments to back you up.

- Don’t overclaim your work without justification

- Don’t intentionally leave out very related previous papers, including your own work (reviewers can find them out easily)
  - maybe even need to mention them in Introduction section and explain why the new work is different
  - reviewers often try to identify a marginal/incremental paper or a “least publishable unit (LPU)” (Google this term!)

Proposal - Related Work

- Where to put the related work section
  - After the introduction/example section
  - Before the conclusion section

- After the introduction/example section
  - Pros: Immediately clear out reviewers' wonder on how the work differs from previous work
  - Cons: hard to let readers to know what you are talking about before showing the approach details
    - But it may be ok to put it after the example section (see next slide)

- Before the conclusion section
  - Pros: Now reviewers' know what your approach is about
  - Cons: reviewers keep wondering how the work differs from previous work till this point
    - But for very closely related work, you should have pointed out the differences in the introduction section

Midterm Report - Problem Definition

- If your paper proposes a new problem or addresses a formalizable problem, it is good to have a section on problem definition

- Examples
  - Section 3.1 http://security.ece.cmu.edu/aeg/aeg-current.pdf
  - Section 4 http://youngwei.com/pdf/AppContext.pdf

- Such a section is useful to clearly describe the problem being addressed by the paper

Midterm Report - Problem Definition

- Define the problem that your approach intends to address
- Can be put in a section after intro/example section, serve the purpose of the example section as described later
  - When you formalize your problem, readers can have better grasp on what you are trying to address
- There you can also formally define some important concepts referred to in your approach (either in the problem space or solution space)
- Problem formalization can be a new contribution in the contribution list

Midterm Report - Technical Challenges

- Add to your intro or problem definition section in your midterm report.

- Why list challenges?
  - If your solution is so obvious and easy, you cannot impress readers/reviewers and justify significance

- Challenges from two levels (you can describe challenges at one or both levels)
- Problem-level challenges
  - Independently of any solution to the problem (e.g., static vs dynamic analysis), what are the challenges of addressing the problem?
- Solution-level challenges
  - For the style/direction that you will commit to (e.g., static in contrast to dynamic analysis; of cz, you need to justify why static not dynamic already here), what are the challenges of carrying out the solution to address the problem?

Simple vs. Sophisticated Solutions

- Don’t ignore simple (basic, straightforward) solutions while hunting for sophisticated solutions
  - At least try simple ones out, only when they don’t work, use the challenges/difficulties faced there to drive the hunting of more sophisticated solutions
  - Simple ones serve as baseline base in evaluation

- Often the time, students may be too proud of some clever “tricks” that they came up and had tendency of losing sight of easier, simpler solutions

“Make things as simple as possible, but not simpler.” - Einstein

Midterm Report - Technical Challenges

- Challenges -> Contribution Points
- Normal structure of main contribution list
  - The overall approach
  - A list of specific techniques in the approach
  - Implementation and evaluation
  - Evaluation results
- For each specific technique in your contribution list, you shall have at least one corresponding clearly articulated technical challenge
  - If your solution/technique is so obvious and easy, you cannot impress readers/reviewers and justify significance
- Alternatively, you may articulate technical challenges just for the overall approach

Midterm Report - Example

- A simple example
  - Include: where it comes from; a figure listing source code; brief description – Throughout the paper, it is important to have illustrating examples for those places that contain “dry” descriptions of your approach
  - If you use several examples throughout the paper, you may not need a separate Example section.

- Optional/important part of the section: high level description of applying your approach on the example
  - describe inputs/outputs of your approach without getting into too much detail
  - very important if the later approach description involves heavy hard-to-understand formalisms

Final Report - Approach

- Generalize your work in an abstraction level, e.g., positioning it as a framework or algorithm rather than a tool
  - What you develop should be beyond your own implementation
  - Then you are in a better position when you discuss limitations of your work: Inherent limitation of the framework? Or limitation of your current particular implementation of the framework?
  - A workflow diagram is useful for explaining your framework
- Try to separate the ideas from (a particular) concrete implementation
  - But sometimes you have to mention it a bit and refer the readers to the implementation section.
- Explain some details with examples (even if you have illustrated your high level ideas in the example section)
  - Often still need to provide algorithm descriptions to precisely describe your approach instead of using ONLY examples to explain it

Final Report - Implementation

- What libraries you used in your tool
  - e.g., BCEL, Daikon frontend, Soot
- Detailed implementations of each step in your framework
- List complications of implementing a certain idea and how you get around them
  - if some complications are important and general, you may move them to the framework section.
- Applicable to both approach/implementation
  - Don’t detail the entire story of how you arrived at your approach/implementation/results, unless they provide useful lessons learned to readers (even so, put them in discussion section)

Final Report - Evaluation

- (Controlled) Experiment: good for tools that don't involve human interactions within the approach experiment writing structure:
  - Hypotheses/Questions to be answered
    - Double check your questions. Ex. "Can our approach perform better than a previous related approach?" "How much better can our approach perform than …"
  - Measures you use to answer these questions (higher better?)
  - Experiment setup: a good number of subjects, some scripts, some third-party tools or reimplemented tools for comparison
  - Independent variables + dependent variables -> metrics
  - Experimental results
    - Illustrate how to read your table/diagrams (columns, x/y axis, etc.)
    - Explain what does the curve or data mean, e.g., "We observed that …", "The experimental results show …"
    - Summarize your findings, remember to get back to answer the hypotheses and questions; it is ok to have an undecisive or negative answer based on the experimental results
    - Optional: discussion subsection; or you can put it as a separate section – Sometimes you may not include cost (time/memory) in your experimental results but you need to at least discuss the analysis cost – Threats to validity: internal, external

Final Report - Evaluation

- Need explain evaluation results or describe your insights from the observed results rather than just describing the results
  - E.g., if some subjects' results are especially favorable or unfavorable, explain the reasons or even your hypothesis (wordings: "We suspect that …" "We hypothesize that …"). You may leave confirmation of these hypotheses to future work (e.g., on more experiments)

- Need describe "Experiment Designs"
  - E.g., factors (independent variables), treatments (one factor multiple treatments or one factor one treatment)

- Need hypothesis testing, t-testing especially if you want to say "A result is **significantly** better than B result"; statistically significant vs. practically significant

- Measure both mean and variance/deviation, not just mean

Final Report - Evaluation

- In evaluation (experiments or case studies), we write

Research question (first)

Hypotheses (then) [Optional]

- Research questions
  - Abstract, general, high level
- Hypotheses
  - Concrete, specific, often answers to the research questions

- In the experimental results, need describe how the results relate back to which hypotheses and how hypotheses relate back to which research questions

Final Report – Evaluation for Empirical Studies

- Case studies, experiences, and examples are often good for
  - approaches with human involvements [experiments can also involve humans though]
  - approaches whose results are hard to quantify with numbers
  - approaches you don't have a good enough number of subjects for controlled experiments
- Case studies
  - uncontrolled but just observe – lessons learned

- Feasibility studies: not directly assess or apply the approach on the real environment but give hints on feasibility

- Experiences/Examples

Empirical Studies

- Some guidelines on doing/writing experiments
  - "Experimental program analysis: A new program analysis paradigm." ISSTA 06 http://esquared.unl.edu/articles/downloadArticle.php?id=208 http://esquared.unl.edu/wikka.php?wakka=ExperimentalProgramAnaly sis
  - http://wwwusers.cs.umn.edu/~heimdahl/ase08ds/AndrewsEvaluation.pdf
  - http://www.acm.org/crossroads/xrds7-4/empirical.html – http://www-static.cc.gatech.edu/~harrold/8803/Classnotes/
    - Notes of Weeks 18, 19, 20, and 21
- Some relevant papers/examples of doing/writing various types of evaluation
  - http://www.cs.washington.edu/education/courses/590n/04sp/
- Experiments vs. Case Studies
  - "Evaluating emerging software development technologies: lessons learned from assessing aspect-oriented programming" by Murphy et al. http://ieeexplore.ieee.org/search/wrapper.jsp?arnumber=799936
- A good book on case study research in general – "Case Study Research : Design and Methods" by Robert K. Yin – http://www.amazon.com/gp/product/0761925538/104-9365607- 2004707?v=glance&n=283155

Empirical Studies

- Better Empirical Science for Software Engineering, Basili and Elbaum, ICSE 06
  - http://csce.unl.edu/~elbaum/talks/PresentedICSE2006.ppt
- Preliminary guidelines for empirical research in software engineering, Kitchenham et al. TSE 02
  - http://csdl.ics.hawaii.edu/techreports/05-06/doc/Kitchenham2002.pdf
- FOSE 07: The Future of Empirical Methods in Software Engineering Research
  - http://www.simula.no/research/engineering/publications/Simula.SE.13
- Hints for Reviewing Empirical Work in Software Engineering Tichy ESE 00
  - http://www.springerlink.com/content/rr70j282h2k01960/
- Readings in Empirical Evaluation for Budding Software Engineering Researchers
  - http://csdl.ics.hawaii.edu/techreports/05-06/05-06.html
- Courses
  - http://www.cs.toronto.edu/~sme/CSC2130/index.html
  - http://www.cs.tut.fi/~pselonen/OHJ-1860/

Final Report - Discussion

- Limitations and issues your approach/implementation currently cannot address
  - Optional: how are you going to address them in future work
- Other caveats (scope of your approach)
- It is often a good idea to list (obvious) limitations and discuss possible solutions for them rather than hiding them
  - Reviewers can often identify obvious limitations even if you don't state them; then they will criticize your work on these limitations (you often don't have a rebuttal against these criticisms in conference reviews).
  - If your paper discusses these obvious limitations as well as their potential solutions, the situation can be alleviated (it is like you have a rebuttal in your paper already before being criticized!).
- Possible applications of your approach that you haven't validated but are convincingly feasible or effective.

Final Report - Conclusion

- Often easy to write conclusions
  - nothing here should surprise readers; simply summarize your contributions and findings
  - In the introduction, “We propose a new approach …” vs. In the conclusions, “We have proposed a new approach …”
- You can state the broader impacts of your approach and your vision
- You can optionally describe limitations and future work here if you don’t have a discussion section for them and propose future work
- May mark your territory of your future work by saying “We are currently doing X..., and preliminary results are promising.” (http://infolab.stanford.edu/~widom/paper-writing.html)

Project Website

- Construct a project web including the evaluation subjects, evaluation results …
  - If tool is releasable, release your tool here (even binary form)
  - If a demo video is available, put it up here (e.g., http://osl.cs.uiuc.edu/~ksen/cute/demo.htm)
- Why? Building trust from reviewers in your work and your results