Improved Labeling of Security Defects in Code Review by Active Learning with LLMs

Johannes Härtel
Vrije Universiteit Amsterdam
Netherlands

Showcase

Code reviews provide interesting data on the **code's security.**

```
+
+     @Override
+     public Map<String, String> extractHeadersToLog(Map<String, String> headers) {
+         return headersPropagator.extract(headers);
```

piotrrzysko on Jun 27, 2022 Member

I'm not sure if this should be the default behavior. What if someone propagates headers with sensitive data? In my opinion, we should either return an empty map here or give users a parameter where they can specify which headers they want to log.

👍 1

Showcase

Code reviews provide interesting data on the **code's security.**

```
+
+    @Override
+    public Map<String, String> extractHeadersToLog(Map<String, String> headers) {
+        return headersPropagator.extract(headers);
```

piotrrzysko on Jun 27, 2022 — Member

I'm not sure if this should be the default behavior. What if someone propagates headers with sensitive data? In my opinion, we should either return an empty map here or give users a parameter where they can specify which headers they want to log.

👍 1

A review discussing a potential security defects.

Johannes Härtel, Vrije Universiteit Amsterdam, j.a.hartel@vu.nl

Showcase

Code reviews provide interesting data on the **code's security.**

```
13 +
14 +     @Override
15 +     public Map<String, String> extractHeadersToLog(Map<String, String> headers) {
16 +         return headersPropagator.extract(headers);
```

Security Defect (label)

piotrrzysko on Jun 27, 2022 Member

I'm not sure if this should be the default behavior. What if someone propagates headers with sensitive data? In my opinion, we should either return an empty map here or give users a parameter where they can specify which headers they want to log.

👍 1

A review discussing a potential security defects.

Showcase

Code reviews provide interesting data on the **code's security.**

```
+
+     @Override
+     public Map<String, String> extractHeadersToLog(Map<String, String> headers) {
+         return headersPropagator.extract(headers);
```

Security Defect (label)

piotrrzysko on Jun 27, 2022 Member

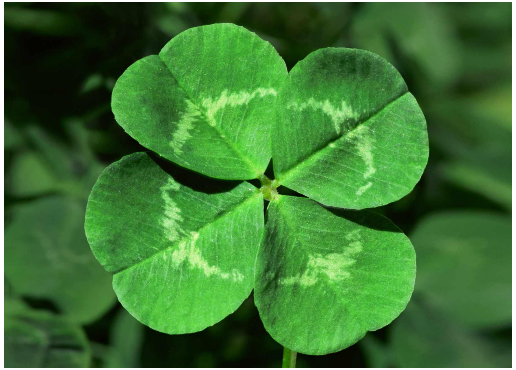
I'm not sure if this should be the default behavior. What if someone propagates headers with sensitive data? In my opinion, we should either return an empty map here or give users a parameter where they can specify which headers they want to log.

👍 1

We have over 5 million unlabeled reviews

A review discussing a potential security defects.

They are **rare!**



- Rare things and corner cases are often more interesting for understanding our world.
- Nether random samples nor exhaustiveness might be a suitable solution.

Related Work

Johannes Härtel, Vrije Universiteit Amsterdam, j.a.hartel@vu.nl

Related Work

Rare classes or labels are a very common to cause problems in labeling.

*“[RQ1 asks for the proportion of security related reviews.]
Being a manual effort, we could **not inspect the entire initial dataset**, rather we proceeded selecting statistically significant sample sets […]
[result is approximating 1%]*

*[this sample is] not large enough to have even an initial answer to our RQ2 [asking for a taxonomy]. Extending this set of [by random sampling ...] would have been a **time-consuming** and, more importantly, **error-prone** approach.” (direct citation)*

It’s a bit old, but this problem pops up over and over again.

Source: Marco di Biase, Magiel Bruntink, and Alberto Bacchelli. 2016. **A Security Perspective on Code Review: The Case of Chromium**. In SCAM. IEEE, 21–30.

Johannes Härtel, Vrije Universiteit Amsterdam, j.a.hartel@vu.nl

Related Work

The typical solution.

*“We then used these **keywords to retrieve** security related review comments.*

Our keyword list is made by the following terms: buffer, cast, command, cookie, crypto, emismatch, exception, exec, form, field, heap, injection, integer, ondelete, out of memory, overflow, password, printf, privilege, race, random, sanitize, security, sensitive, sql, URL, use-after-free, vulnerability, xhttp, xml. We also used regular expressions and stemming based on [...]“ (direct citation)

It's always some classifier/model used for candidate selection!

Source: Marco di Biase, Magiel Bruntink, and Alberto Bacchelli. 2016. **A Security Perspective on Code Review: The Case of Chromium**. In SCAM. IEEE, 21–30.

Johannes Härtel, Vrije Universiteit Amsterdam, j.a.hartel@vu.nl

Active Learning

Johannes Härtel, Vrije Universiteit Amsterdam, j.a.hartel@vu.nl

Stereotypical Solution

We cannot sample randomly.

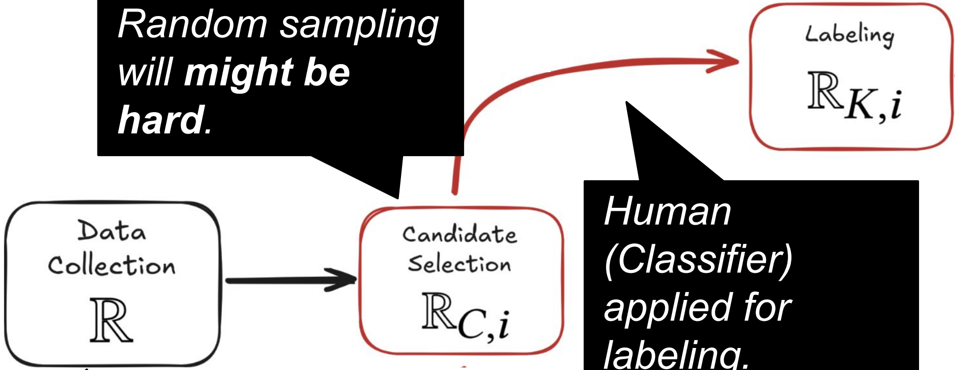
Huge unlabeled dataset.

Johannes Härtel, Vrije Universiteit Amsterdam, j.a.hartel@vu.nl

Stereotypical Solution

We cannot sample randomly.

Random sampling will ***might be hard***.

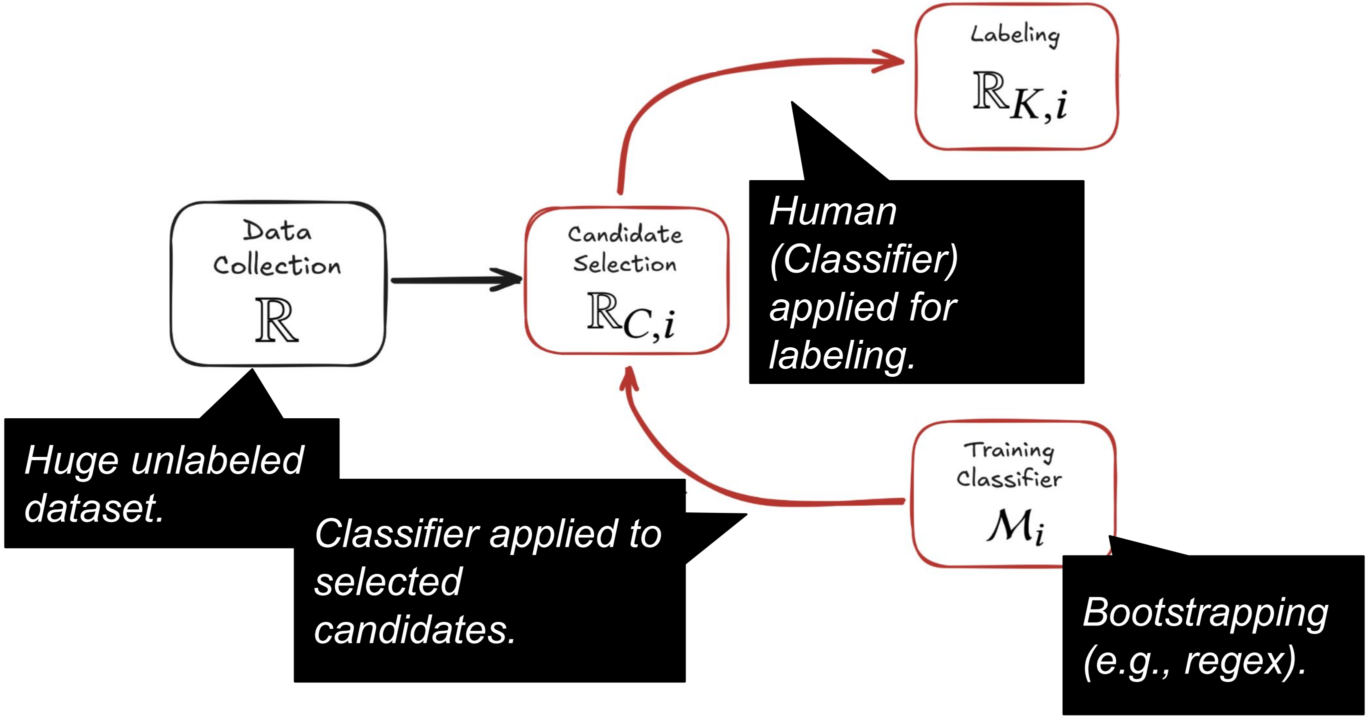


Human (Classifier) applied for labeling.

Huge unlabeled dataset.

Stereotypical Solution

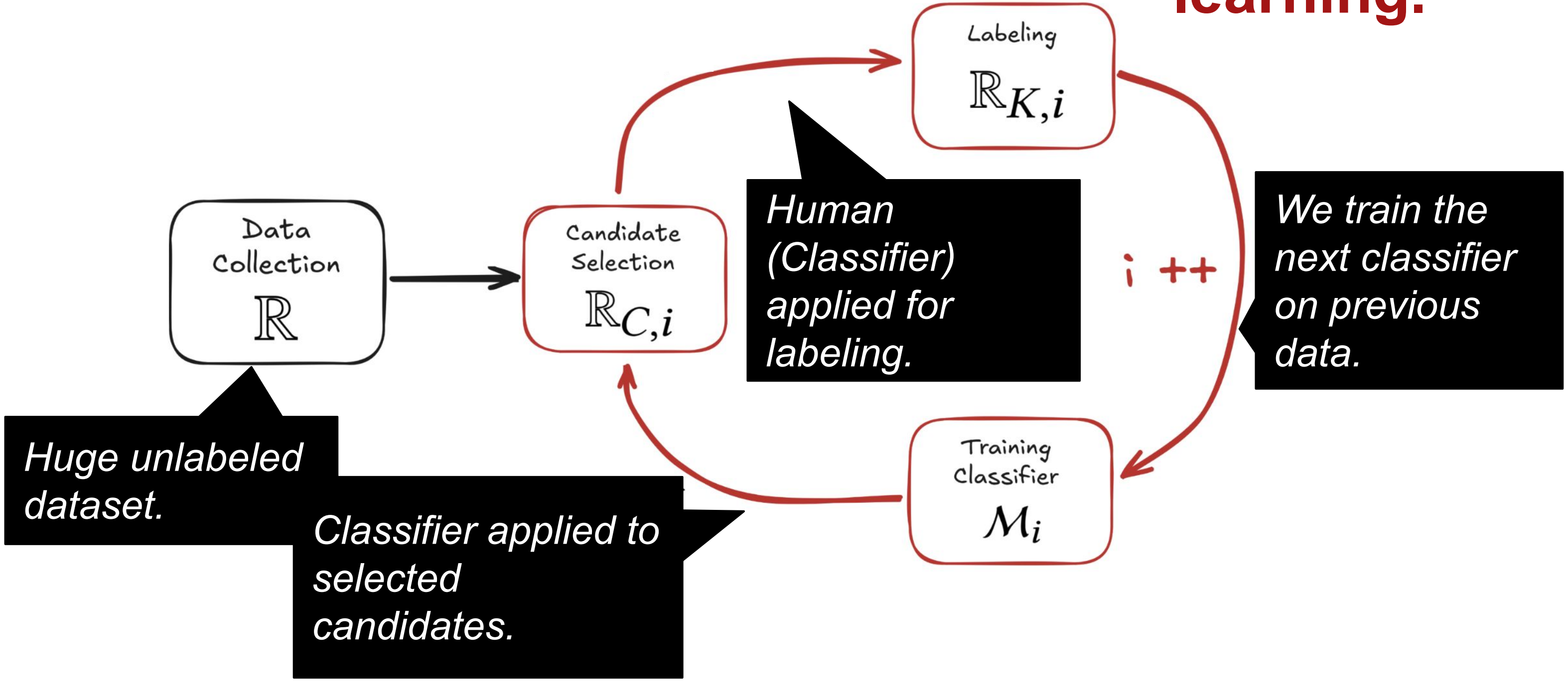
We use a second classifier/mode to do candidate selection.



Stereotypical Solution

We learn this classifier/model by active learning in iterations.

Now it's active learning.



Stereotypical Solution

We learn this classifier/model by active learning in iterations.

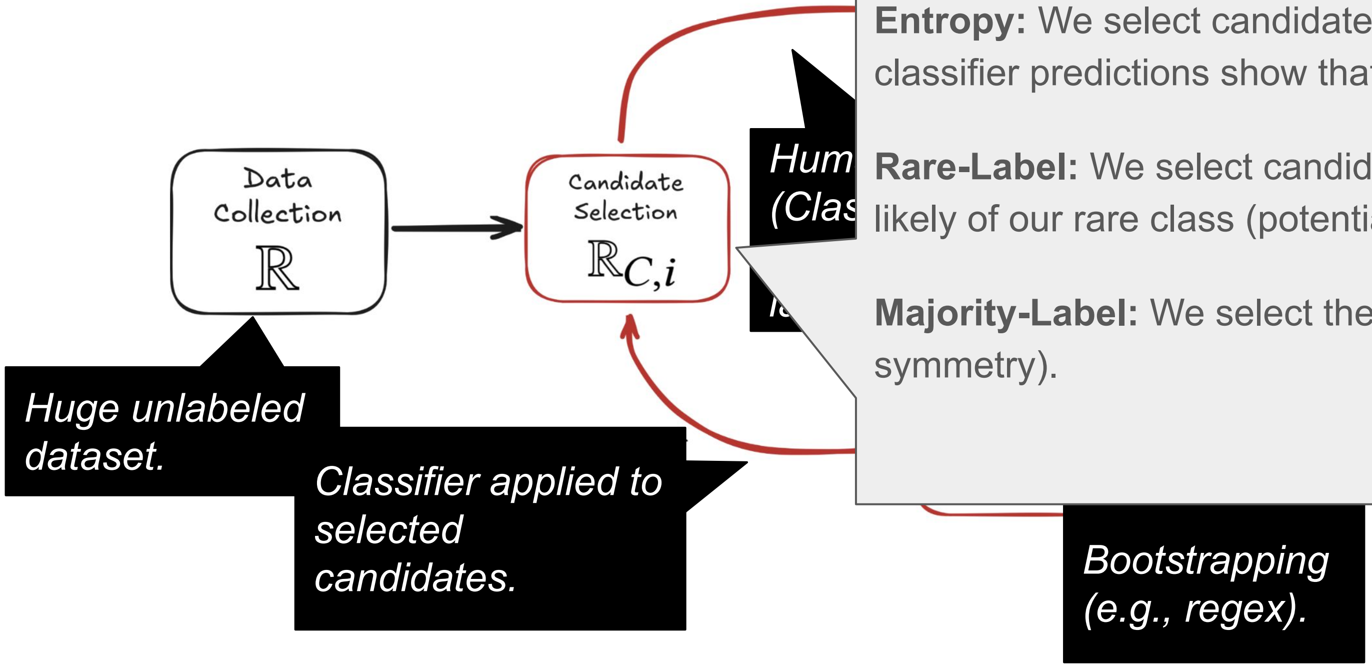
Now it's active

Random: We select candidates randomly.

Entropy: We select candidates where previous classifier predictions show that highest entropy.

Rare-Label: We select candidates that are mostly likely of our rare class (potential security defect).

Majority-Label: We select the opposite (just for symmetry).



Our Contributions

On active learning and fine-tuned LLMs

- An **empirical study** on labeling pull request reviews from GitHub using active learning with fine-tuned LLMs for labeling.
- An **simulation study** on labeling artificial data using active learning with LLM’s head only for labeling.

Johannes Härtel, Vrije Universiteit Amsterdam, j.a.hartel@vu.nl

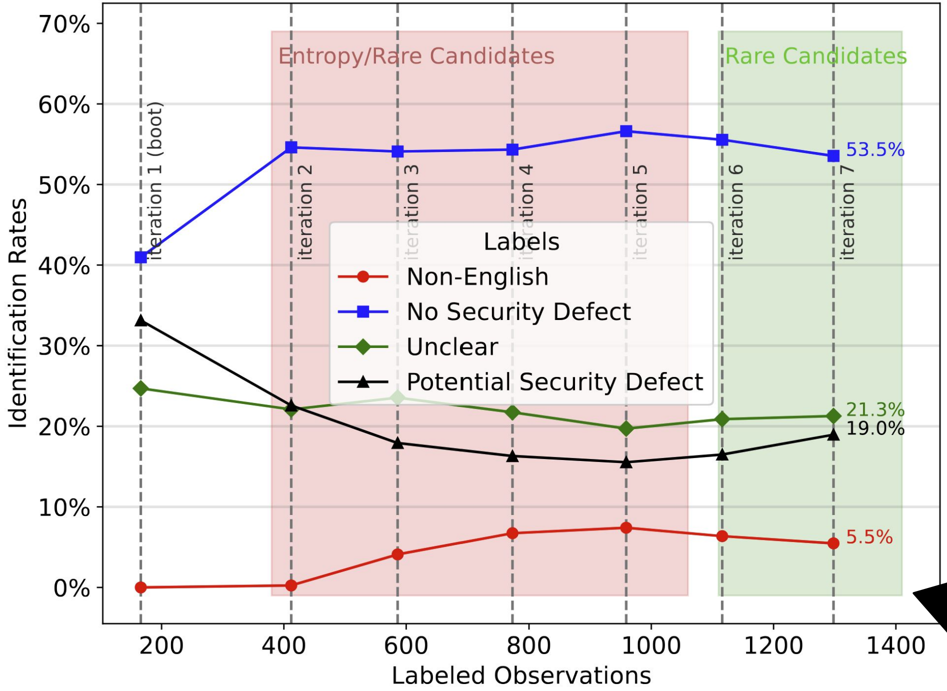
Empirical Study

Johannes Härtel, Vrije Universiteit Amsterdam, j.a.hartel@vu.nl

Empirical Study

We apply active learning with LLM and finally found 246 reviews on potential security defects **(19%)** after 7 iterations.

(fractions in dataset)



Random sampling lies below 3%.

Johannes Härtel, Vrije Universiteit Amsterdam, j.a.hartel@vu.nl

Empirical Study

The evolution of the LLM over the iterations 1 to 7.

Review Text	**Iterations LLM Classifier** $\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$	$\mathcal{M}_4$	$\mathcal{M}_5$	$\mathcal{M}_6$	$\mathcal{M}_7$
	(Does the review discuss a pot. security defect?)						
R0: Can we refactor this code to make it maintainable?	0%	7%	2%	0%	0%	0%	0%
R1: This code enables an attacker to get access to our data.	0%	45%	**89%**	6%	**95%**	**84%**	**93%**
R2: This code might provide access to sensitive data.	0%	34%	**76%**	4%	**95%**	**81%**	**91%**
R3: This might allow someone to run denial of service.	0%	**53%**	23%	2%	11%	**77%**	**80%**
R4: We might expose credentials.	0%	**58%**	**92%**	1%	42%	**74%**	**89%**

Johannes Härtel, Vrije Universiteit Amsterdam, j.a.hartel@vu.nl

Simulation Study

Why a simulation study?

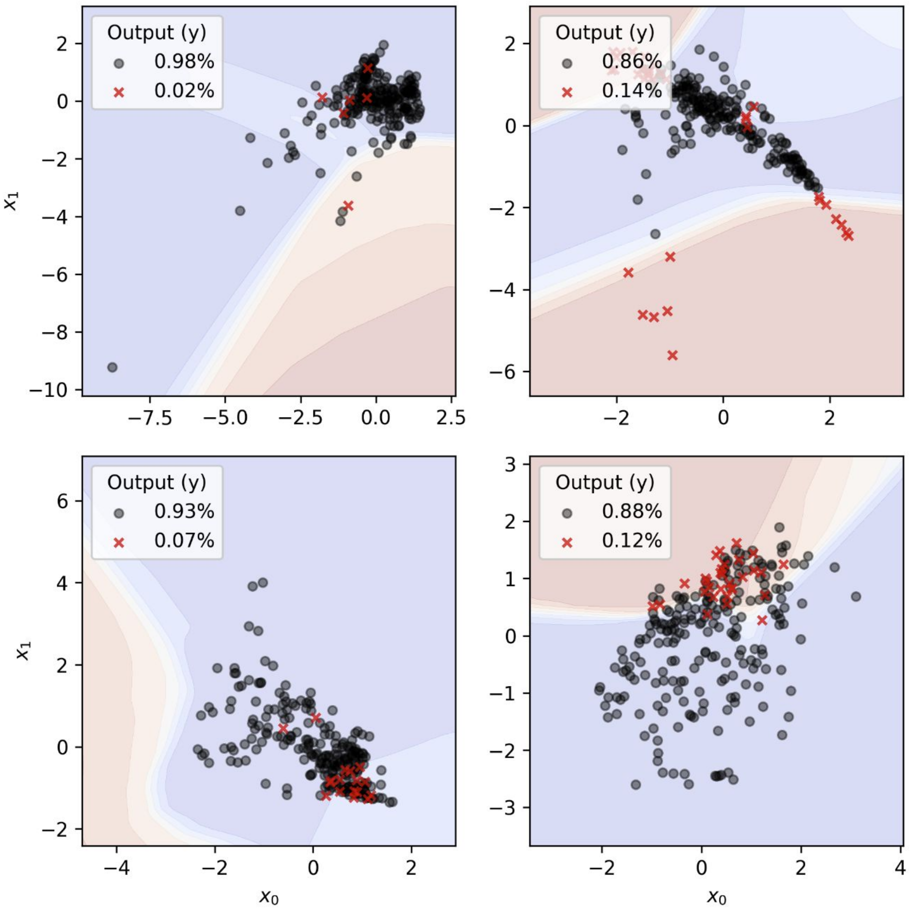
A simulated **data generating process (DGP)** is handy to evaluate a method.

- Simulating a DGP is **transparent**: We know how data has been produced, since we know the code.
- Simulating a DGP is **controllable**: We can modify (hyper-) parameters specific to the DGP (the problems), not just those of the method.
- Simulating a DGP is **repeatable**: We can re-run a method on data from a fundamentally new DGP, and not only on a fresh split of the same data by the same DGP.

Johannes Härtel, Vrije Universiteit Amsterdam, j.a.hartel@vu.nl

Example

Simulated data for a random categorical problem with strong imbalance.

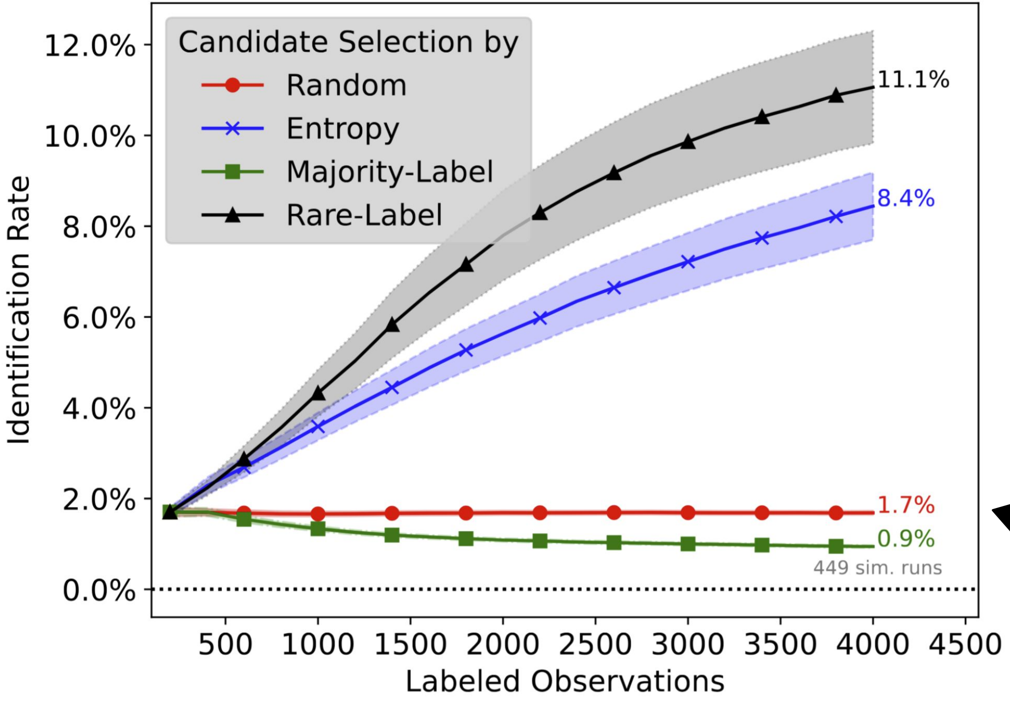


Johannes Härtel, Vrije Universiteit Amsterdam, j.a.hartel@vu.nl

What is the impact of candidate selection?

Our simulations show that candidate selection is a driving hyperparameter, preferably we select candidates that our model predicts are “rare-label”.

(fractions in dataset)



Overall, simulations targeted 1.7% are rare.

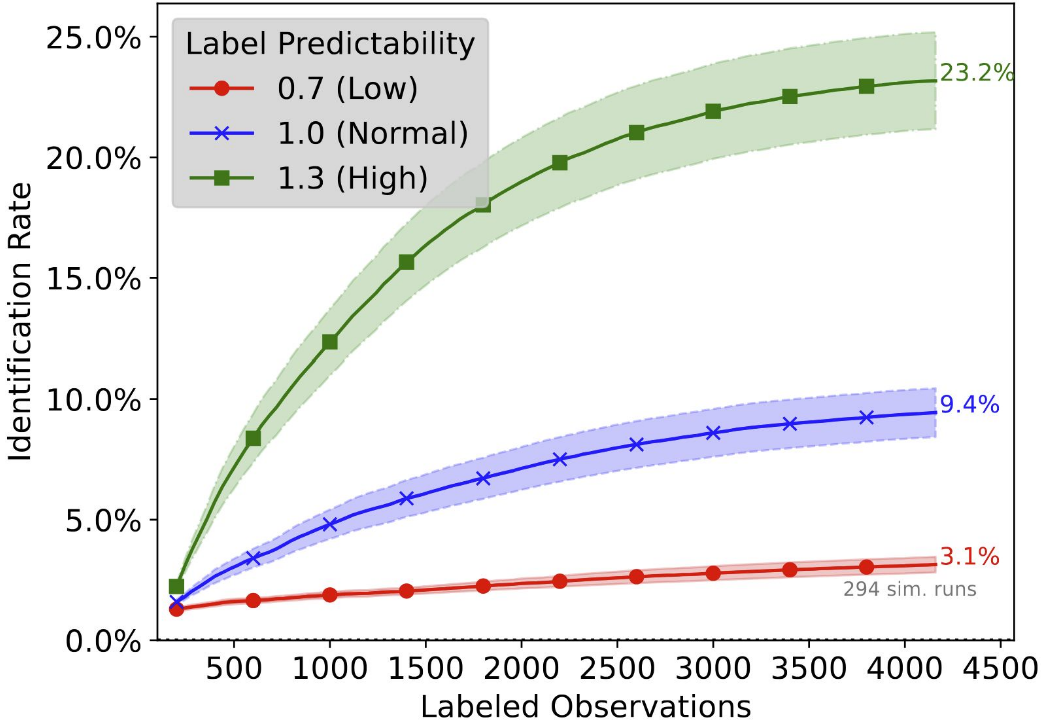
Shaded area depicts 95% confidence over different data generating processes

Johannes Härtel, Vrije Universiteit Amsterdam, j.a.hartel@vu.nl

What is the impact of label predictability?

Our simulations show that the problem specific hyperparameter strongly effects active learning. Very predictable labels are best.

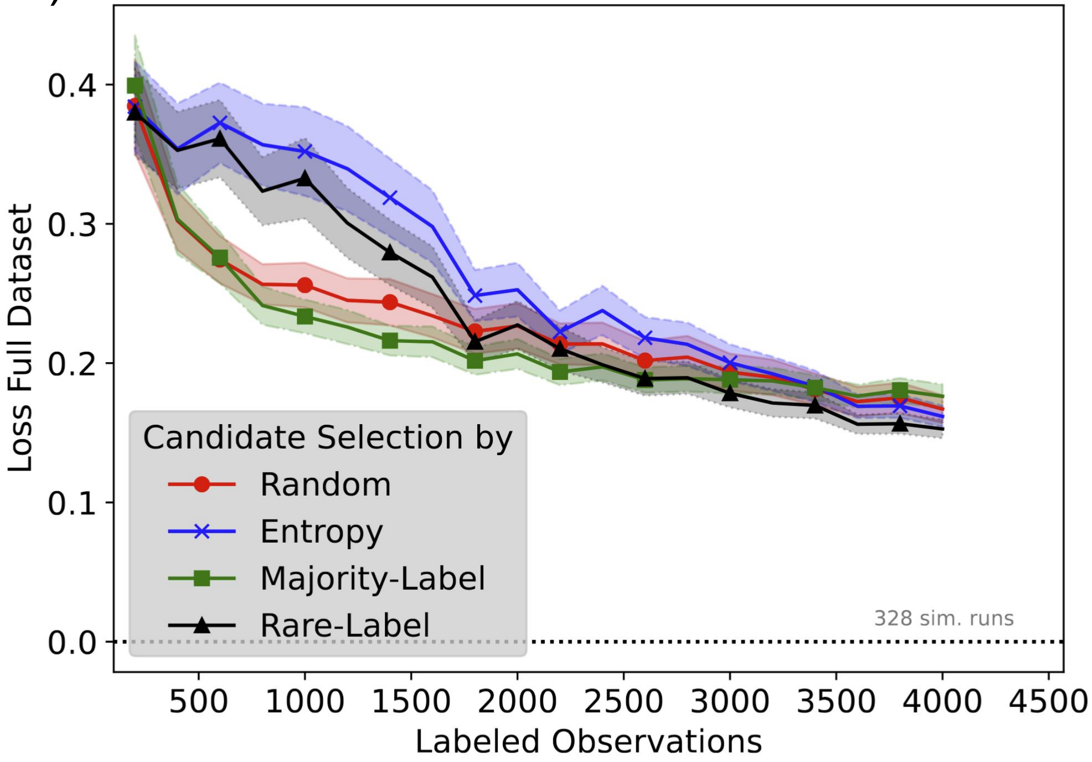
(fractions in dataset)



Shaded area depicts 95% confidence over different data generating processes

What is the impact on the overall loss?

It appears that we can also get the overall loss down faster (on the full, mostly unseen dataset).



… but caution, it is no differential analysis, it is yet a simple average, not maxing out hyperparameter options.

Shaded area depicts 95% confidence over different data generating processes

Summary: Active learning is an interesting solution

- It is a very basic.
- It includes a feedback loop (model ↔ human).
- It appears to solve the rare class problem.

Backup

Empirical Study

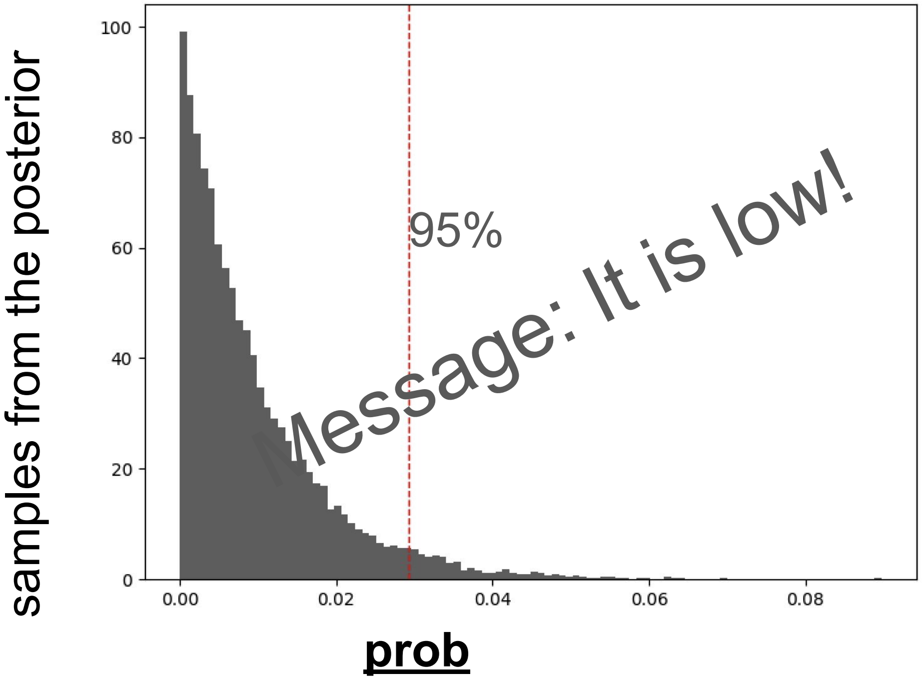
We focused on ~ 5 million pull requests reviews, manually labeling 1298 of them.

Label	Description
Potential Security Defect	The review discusses a potential security defect of code or other artifacts.
No Security Defect	The review does not discuss a security defect of code or other artifacts.
Unclear	The relation to a security defect is unclear.
Broken	The review is a broken link. This is a technical problem with some of our input data.
Bot	The review is written by a bot.
Non-English	The review is not written in English.

Johannes Härtel, Vrije Universiteit Amsterdam, j.a.hartel@vu.nl

Empirical Study

A small experiment with **100 random instances** labeled says the **prob**ability of potential security defects it is **below 3%.**



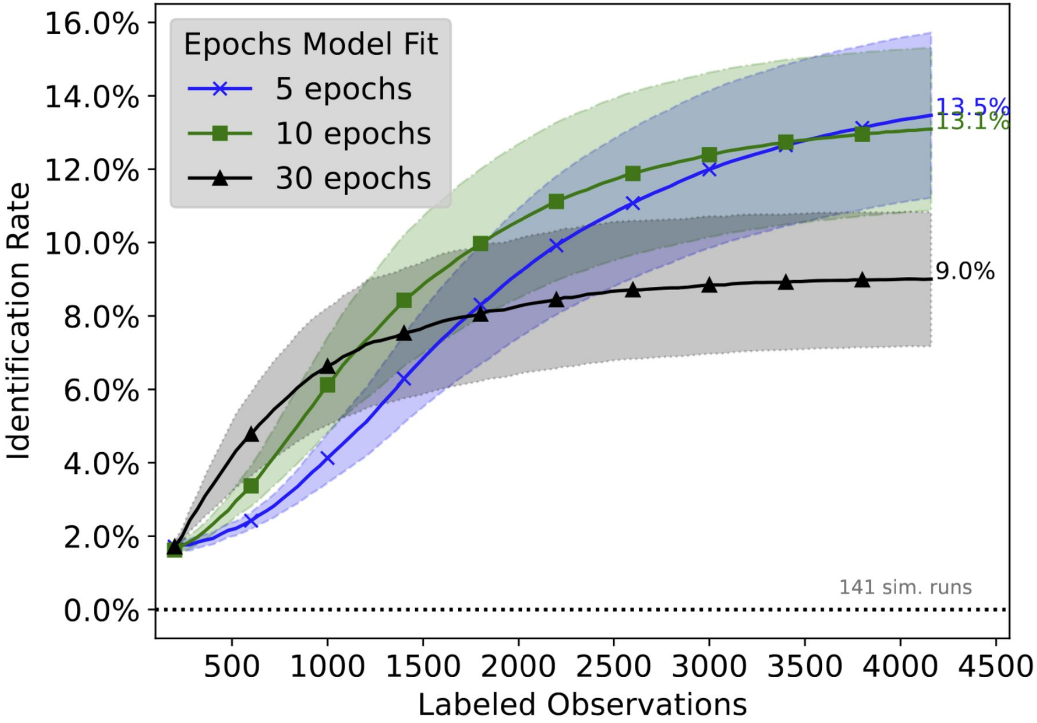

```
model {
    // Prior for prob, which is neutral.
    prob ~ beta(1, 1);

    // Likelihood.
    y ~ binomial(1, prob);
}
```


Johannes Härtel, Vrije Universiteit Amsterdam, j.a.hartel@vu.nl

What is the impact of epochs?

In our simulations, the number of epochs used to fit the model (early stopping) influences the performance. Simply speaking: Start high and then go down.

(fractions in dataset)



Shaded area depicts 95% confidence over different data generating processes

Thanks for the attention.

https://github.com/johanneshaertel/EASE_2025_active_learning_LLM

https://doi.org/10.6084/m9.figshare.28303904

Johannes Härtel, Vrije Universiteit Amsterdam, j.a.hartel@vu.nl