Cross-project alignment of dependency updates

Dylan Hu | Supervisors: Sebastian Proksch, Cathrine Paulsen
EEMCS, Delft University of Technology | CSE3000 Research Project

1. The question

When 50 projects each get the same dependency-update PR (e.g. `lodash 4.17.20 → 4.17.21`), do they decide alike?

- Prior work measures merge rates *one project at a time*
- Most studies pre-filter to Dependabot only
- Outcome read from PR state alone → every closed PR looks rejected
- **Nobody compares decisions across projects on the same version pair**

RQ1. How do update PRs distribute across five dimensions, and how does that differ across Maven, npm, Cargo, pip, Go?

RQ2. For shared (*package*, v_1, v_2) triples, how aligned are decisions, and does alignment vary by ecosystem, semver, security, or source?

⇒ **If projects agree, "others merged this" is usable evidence. If not, the signal misleads.**

2. The instrument

Discover-then-filter pipeline − keeps *all* PR sources, not Dependabot only.

- **Discover** arbitrary PRs from GH Archive (BigQuery)
- **Filter** to PRs whose diff touches *only* manifest/lockfiles
- **Parse** per-ecosystem adapters (5 ecosystems)
- **Classify** 5 dimensions: source, outcome, security, semver tier, direct/transitive
- **Align** per shared triple at $k_{\text{decided}} \geq 5$

⇒ **Validated at macro-F1 ≥ 0.85 on all five dimensions (500-PR / 1,000-change gold set).**

3. Dataset

19 GH Archive days, Jan–May 2026:

- 1,297,514 candidate PRs
- 143,111 enriched via GraphQL
- 30,630 passed manifest-only filter
- 19,911 parsed PRs → 274,126 change rows

Change rows by ecosystem:

npm 246k | pip 10.1k | Go 9.4k
Cargo 6.7k | Maven 1.6k

Shared triples at $k_{\text{decided}} \geq 5$: 1,898

npm 1,634 | Cargo 102 | pip 96 | Go 52 | Maven 14

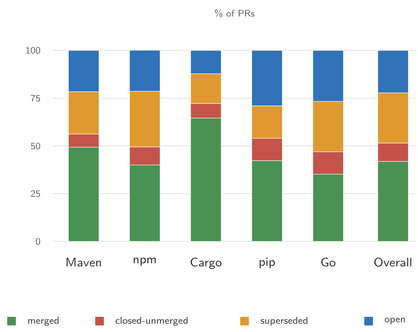
⇒ **Open code + dataset: `github.com/dylanhu-code/deptracker`; Zenodo DOI 10.5281/zenodo.20787811**

4. Finding 1 — most "rejections" are bot bookkeeping

Q: If a bot PR is closed-unmerged, was it rejected? A: Usually not.

Bots close their own earlier PRs when superseding or regrouping them, visible only in closing comments.

- **5,226 of 7,141** bot closures are supersession, not decisions
- State-only: **64.1%** of decided PRs look closed-unmerged; after separating supersession, **18.7%**



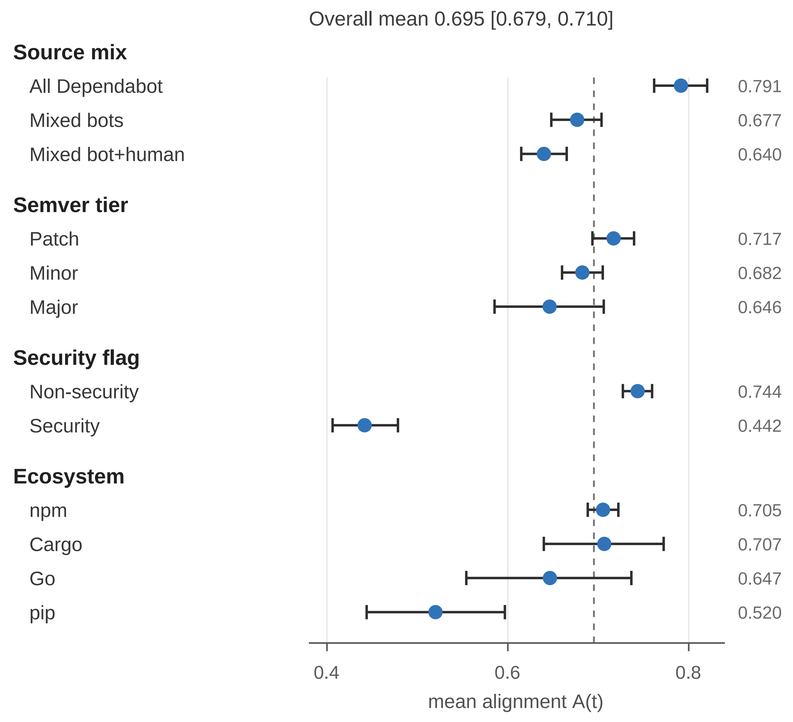
⇒ **Merge/close ratios in prior bot-PR studies mix real decisions with bot workflow (≈1:3 here).**

5. Finding 2 — projects agree, except on security

Q: How aligned are decisions on a shared upgrade?

$A(t) = 1 - H(d_1, \ldots, d_k)/\log 2$ over genuine decisions (merged vs. closed). $A = 1$ unanimous, $A = 0$ evenly split.

A: High overall − mean $A(t) = 0.70$, median 0.86.



The exception is security: flagged updates align **0.44** vs. **0.74** (non-overlapping CIs).

*Source mix is directional (novel): all-Dependabot triples agree by **closing**, mixed bot+human triples agree by **merging**.*

⇒ **Projects agree on security updates the *least*, not the most; crowd signals are weakest where the stakes are highest.**

6. Limitations

- **npm dominates RQ2.** 86% of $k \geq 5$ triples are npm; small-ecosystem claims are tentative (Maven only 14 triples)
- **Manifest-only filter** biases toward bot-like PRs, excluding PRs that mix code with dependency changes
- **Supersession detection** is a lower bound, so the all-Dependabot closure share is an upper bound on real rejection
- **Security flag** marks a detected advisory link, not proof of a real vulnerability fix

7. Why it matters

Practitioners: security updates need *more* human review, not less. Tooling should suppress crowd/compatibility signals on security PRs − they mislead exactly where stakes are highest.

Researchers: a closed bot PR is often not a rejection. Classify outcome from closing comments, not PR state alone.

Next: regress $A(t)$ on advisory and package features to explain *why* security updates split projects.