Escaping the Time Pit: Pitfalls and Guidelines for Using Time-based Git Data

Samuel W. Flint

Jigyasa Chauhan

Robert Dyer

Department of Computer Science and Engineering
University of Nebraska–Lincoln

2021 Mining Software Repositories Conference

Time Data is Everywhere!

Almost any repository data we touch has a time component:

- Commits
- Issues
- Pull Requests
- Forge metadata
- Logs
- *Etc*.

But like any data, we need to be careful handling it.

(https://latestaesthetics.com/products/melting-dali-clock)

Research Questions

1. How many MSR papers rely on time-based data?
2. What kinds of data include time?
3. What filtering or cleaning techniques are used?
4. Is bad time-based Git data common?

Selection Criteria

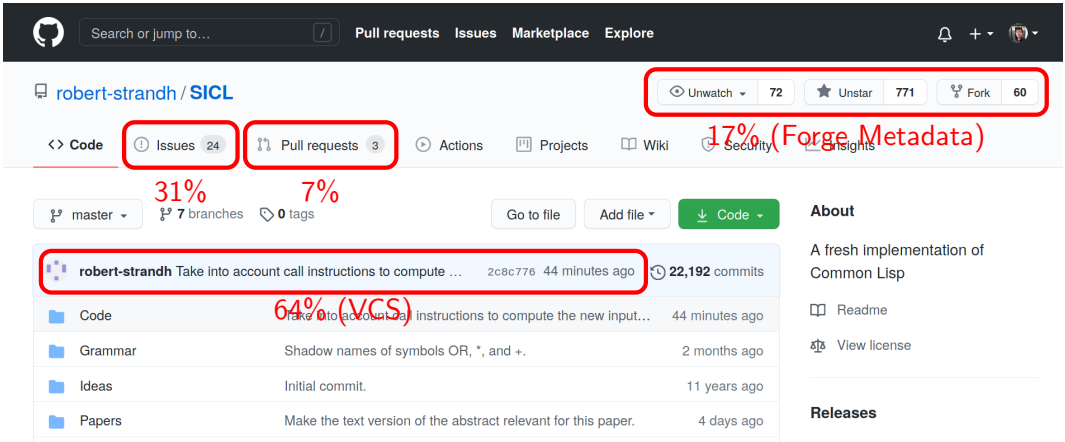
- **Mining Software Repositories (MSR)** conference
- Technical track & Data Showcase papers
- 2004–2020
- 580 technical + 110 data showcase = **690 papers**
- Filtered by time-related keywords, *e.g.*, `minute`, `hour`, `epoch`, `interval`

Selection Criteria

- 282 matched the keyword search
- 43 papers were removed after manual inspection
- **RQ1 result: 239 papers (35% of 690)** were retained

Survey Results

RQ2 result: Kinds of Data



Survey Results

Data Sources

49 (21%)

19 (8%)

19 (8%)

17 (7%)

Torrent
14 (6%)

13 (5%)

9 (4%)

8 (3%)
And others. . .

7 (3%)

7 (3%)

Survey Results

RQ3 result: Filtering and Cleaning Techniques

- None explicitly mentioned (146, 62%)
- Selecting within a time window (24, 10%)
- Describing a cut-off date (15, 6%)
- Using a Custom Condition (7, 3%)
- Coalescing Changesets (5, 2%)
- Using an Analysis Tool (4, 2%)
- Correcting Date Formats (3, 1%)

Survey Results

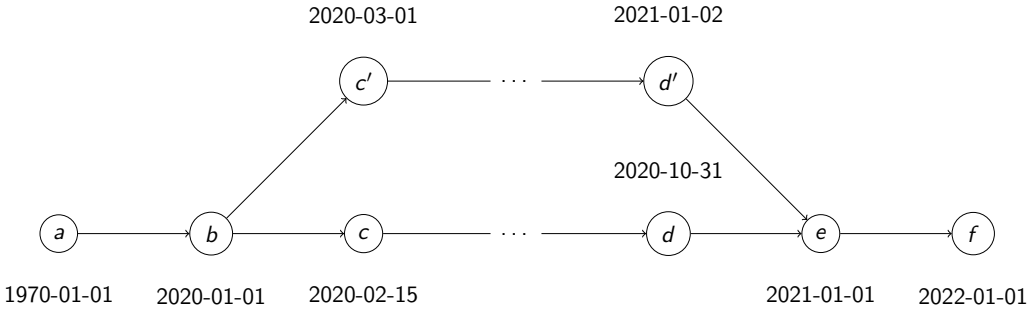
RQ3 result: Filtering and Cleaning Techniques

- **None explicitly mentioned (146, 62%)**
- Selecting within a time window (24, 10%)
- Describing a cut-off date (15, 6%)
- Using a Custom Condition (7, 3%)
- Coalescing Changesets (5, 2%)
- Using an Analysis Tool (4, 2%)
- Correcting Date Formats (3, 1%)

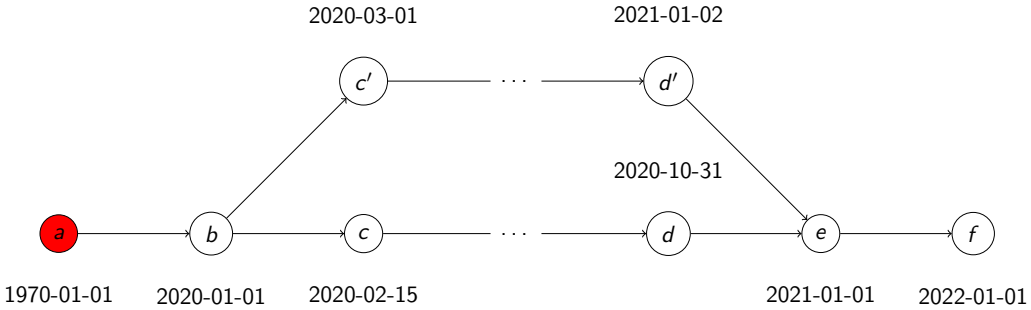
Studying Git

- GitHub is the most common data source
- VCS data is the most studied kind of data
- What would a time problem look like? How frequently are they found?

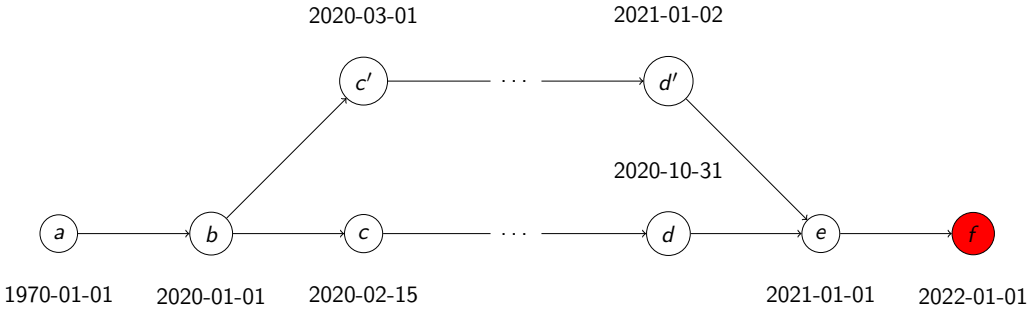
Kinds of Time Problems



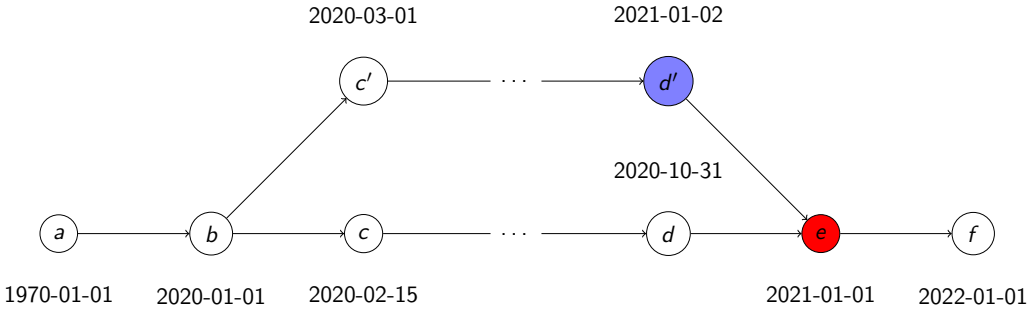
Kinds of Time Problems



Kinds of Time Problems



Kinds of Time Problems



Observed Dataset Problems

- From Boa's 2019 GitHub Large Dataset
- 26,252 suspicious due to being out of order, from 4,744 projects
- 4,735 suspicious due to being too old, from 82 projects
- 11 suspicious due to being in the "future", from 3 projects

Proposed Mitigation

- Filter by Project
- Filter by Date
- Remove projects modified by `git-svn`

Conclusion

Literature

- **RQ1:** At least 35% of MSR studies consider time-based data
- **RQ2:** VCS is the most common, most often from GitHub
- **RQ3:** Filtering & cleaning of time-based data isn't commonly described

RQ4: Mining GitHub

- Out-of-order commits most common error, 26,252 from 4,744 projects
- For "too old" commits, `git-svn` seems connected
- Causes likely due to tools, misconfiguration or user error

Replication Package: https://doi.org/10.5281/zenodo.4625288