Modelling Risk in Open-Source Software

Róisín Ní Bhriain - 23269640
Sneha Dechamma Mallengada Suresh - 23262168
Course: MCM Computing
Supervisor: Darragh O'Brien

Introduction

- Popularity of Open-Source Software has been increasing
  - 3.6 million repos depend on the top 50 open-source projects
- Comes with the risk of software vulnerabilities
  - Attackers can exploit these
- Prediction of risk can be used to explore risk
  - Software metrics
  - Project Activity
  - Vulnerability Data

Research Question

Can we provide developers with an effective risk measure when choosing between multiple open-source components?

Key Terms

- ➔ Maven
  - ◆ Open-source build automation & project management tool
- ➔ Time series analysis
  - ◆ Fitted model based on past data can predict future values
- ➔ ARIMA
  - ◆ Autoregressive integrated moving average
  - ◆ p, d, q parameters
- ➔ AutoARIMA
  - ◆ Tests parameter combinations to determine the best model

Background Research

Vulnerability Propagation

- Only 1.2% directly use vulnerable code
- Small packages can affect 375, 607 packages in the Maven ecosystem
  - jackson-databind, netty-codec-http
  - CVEs can affect a large number of projects
- The more dependencies the more complex it is to locate where the project pulls in vulnerable code

Project Metadata Analysis

- ➔ Project activity level is an indicator of survival
  - Number of commits, time-to-fix
  - Long-time Contributors
- ➔ Less than 50% of abandoned projects find new contributors
- ➔ Standard measurement for commits is:
  - Number of commits per month

Vulnerability CVE Data Analysis

- Predicting number of vulnerabilities per month is useful
- ARIMA is a very popular method of prediction
  - Seasonality not a factor
- Datasets from the NVD were used in every study we analysed
  - Prediction of CVEs
- Most models fall flat at the three month mark

Case Studies

Log4j Vulnerability

- Discovered in November 2021 in popular logging library
- Gateway to gain control of the machine
- Many projects unaware they were affected

Heartbleed Vulnerability

- Discovered in April 2014 in popular cryptography library
- Receive private information after crafting similar messages
- Many servers unaware they were affected

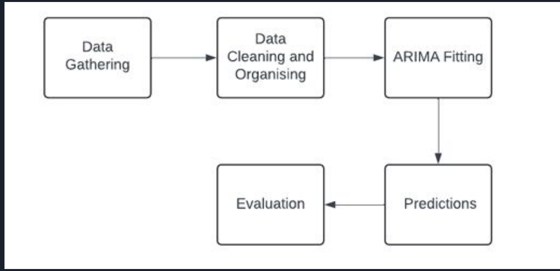
Methodology

Data & Dependencies

- ➔ Different GitHub projects to test the algorithm
- ➔ GitHub API for project activity
- ➔ NVD API for vulnerability Data
- ➔ Each Maven Dependency as a node
- ➔ NetworkX graph
- ➔ Keywords and libraries extracted for prediction
- ➔ Colour-coded

Data Source	Purpose	Type
GitHub projects	Find Dependencies	Maven-dependency trees
GitHub API	Project Meta-data	API
NVD API	Vulnerability Prediction	CVE API

Predictions



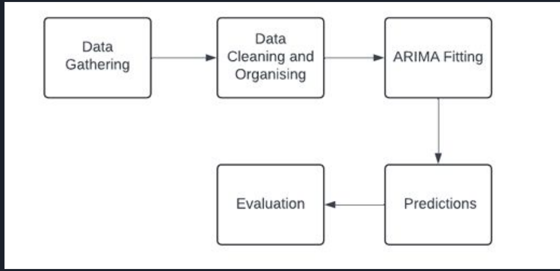
Project Activity

- ➔ GitHub API
  - ◆ All available data gathered
- ➔ Commits, average time-to-fix issues or both
- ➔ AutoARIMA model
- ➔ Predicted value for the next month returned

```
Hamcrest Dependency in Maven Project:
org.hamcrest:hamcrest-core:jar:1.3:test

Keyword for Hamcrest Dependency:
hamcrest-core
```

Predictions



Vulnerabilities

- ➔ NVD API
  - ◆ Call made for each keyword
- ➔ Number of vulnerabilities per month
- ➔ AutoARIMA model
- ➔ Predicted value for the next month returned

```
JUnit4 Dependency in Maven Project:
junit:junit:jar:4.11:test

Array of Keywords in JUnit Dependency:
['junit', 'junit4', 'junit jar']
```

Risk Calculations

$$projectActivityScore = (x/numDaysToFixIssues) * 10$$

$$projectActivityScore = (numCommits/x) * 10$$

$$vulnerabilityScore = (x/vulsCountPerMonth) * 10$$

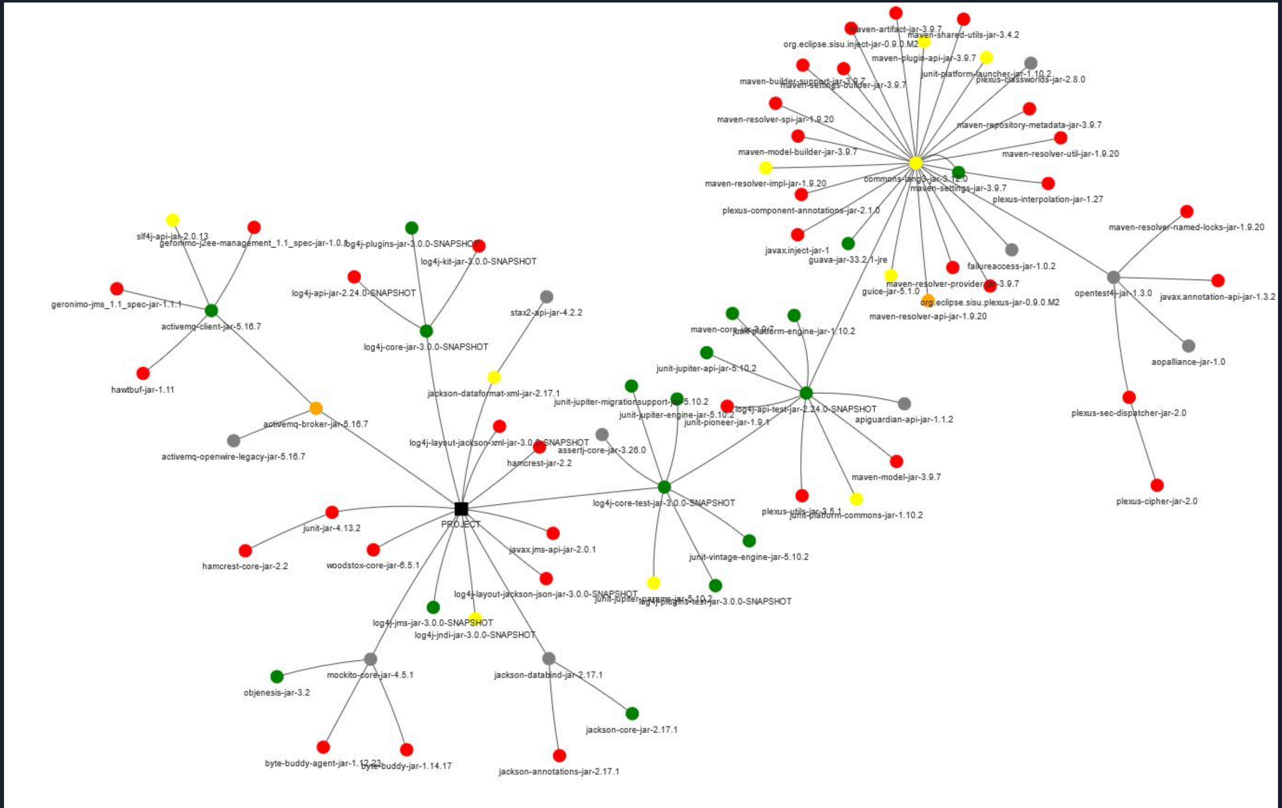
$$overallScore = (projectActivityScore + vulnerabilityScore)/2$$

- ➔ User-decided acceptable levels
- ➔ Options for project activity score
- ➔ Combination of project activity and vulnerability for overall score

Score	Risk Level
<0	Not Enough Data
0 - 2.5	Low Risk
2.5-5	Medium Risk
5-7.5	High Risk
>7.5	Severe Risk

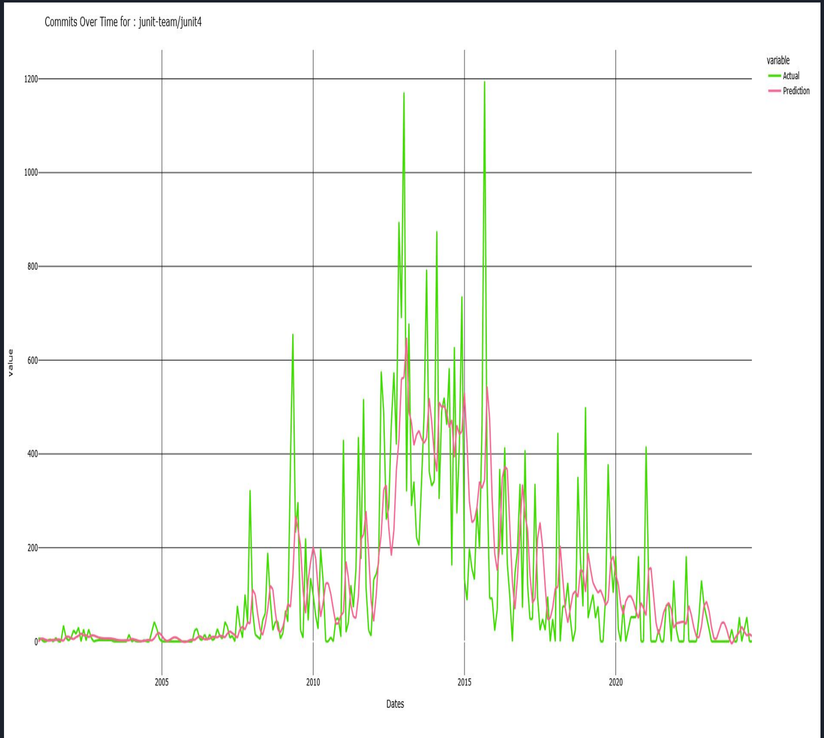
Results

Example Final Graph



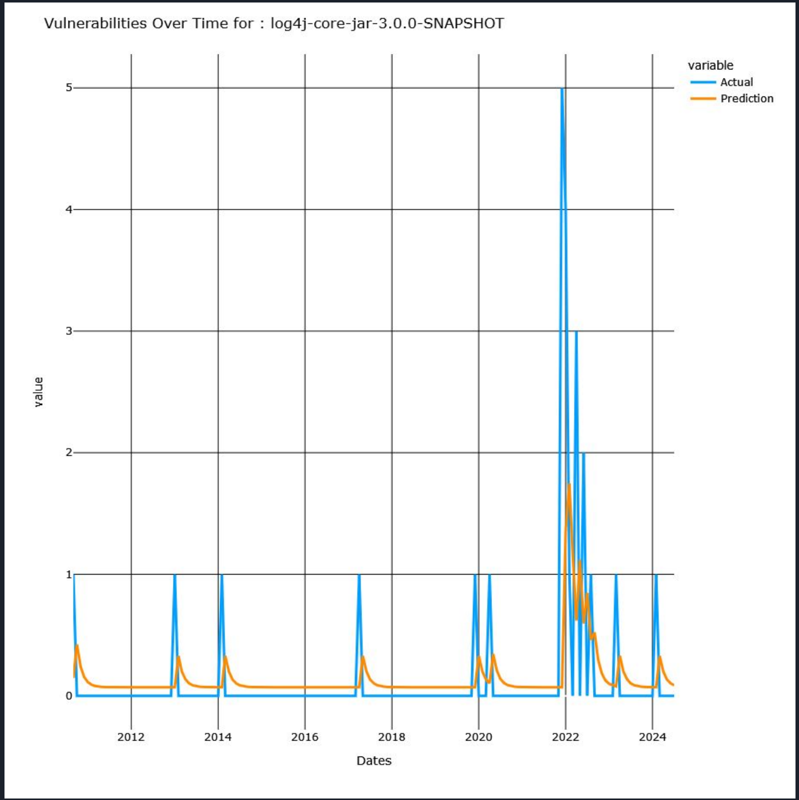
Example Graph of Project Activity Model

- ➔ JUnit4 commits over time
- ➔ Activity has declined in recent years
- ➔ 2012 to 2017 were high-commit years
  - ◆ At its peak over 200 commits per month
- ➔ Fitted values are fairly accurate as can be seen in the graph
- ➔ Predicted value for the next month
  - ◆ 10.9951



Example Graph of Vulnerability Data Model

- ➔ Log4j vulnerabilities over time
- ➔ Many were released in 2021
  - ◆ 5 released in one month
- ➔ Reasonably accurate fitted values in subsequent months
- ➔ Predicted value for the next month
  - ◆ 0.07898



Evaluations

Prediction	MAE	RMSE
Commits per month	123.18	213.96
Vulnerabilities per month	0.145	0.476

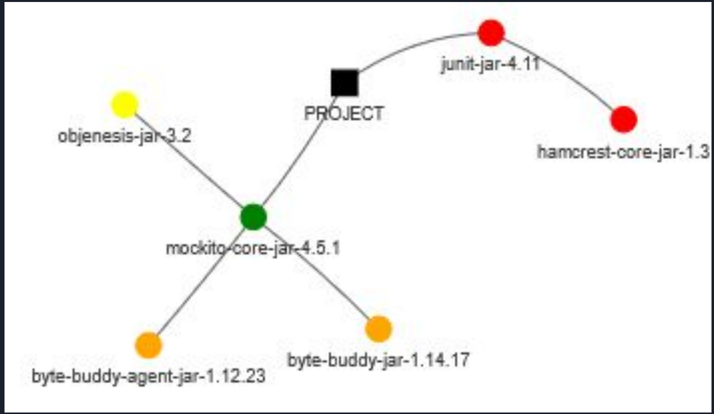
- ➔ Log4j sample project
- ➔ Commonly used metrics
  - ◆ Mean Absolute Error (MAE)
  - ◆ Root Mean Squared Error (RMSE)
- ➔ Commits have larger standard deviation
  - ◆ 152.79
- ➔ Vulnerabilities have lower standard deviation
  - ◆ 0.432

Technical Challenges

- ➔ Graphing dependencies and dependencies of dependencies
- ➔ Choosing of ARIMA parameters
  - ◆ Decided on AutoARIMA
  - ◆ PACF and ACF graphs
  - ◆ Differencing data until ADF test below 0.05
- ➔ Connecting dependencies to GitHub URLs
  - ◆ Done using a mapping file

Tool Uses

- ➔ Comparing similar dependencies
  - ◆ JUnit vs **Mockito**
- ➔ Exploring risk in current dependencies
  - ◆ Finding risky modules



Student Contribution

Róisín

- Gitlab setup + Assigning Issues
- Literature Review
  - Writing + read 42 papers
- Software
  - All code
- Practicum Paper
  - All writing

Sneha

- Read around 15 papers
- Attempted to create graphs
- Attempted to create keyword function.

Conclusion

- ➔ Growing reliance on OSS
  - ◆ Challenges regarding security
- ➔ Better overview of dependency risk
- ➔ Focused on Maven dependency trees
- ➔ Compromised on accuracy to predict many dependencies
- ➔ Tool could aid developers in
  - ◆ Choosing less risky modules
  - ◆ Discovering already used risky dependencies

References

A. A. Ariyo, A. O. Adewumi, and C. K. Ayo (2014) 'Stock Price Prediction Using the ARIMA Model', in 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation. 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, pp. 106–112. Available at: https://doi.org/10.1109/UKSim.2014.67.

A. M. Mir, M. Keshani, and S. Proksch (2023) 'On the Effect of Transitivity and Granularity on Vulnerability Propagation in the Maven Ecosystem', in 2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). 2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), pp. 201–211. Available at: https://doi.org/10.1109/SANER56733.2023.00028.

C. Liu et al. (2022) 'Demystifying the Vulnerability Propagation and Its Evolution via Dependency Trees in the NPM Ecosystem', in 2022 IEEE/ACM 44th International Conference on Software Engineering (ICSE). 2022 IEEE/ACM 44th International Conference on Software Engineering (ICSE), pp. 672–684. Available at: https://doi.org/10.1145/3510003.3510142.

Chahal, K. and Saini, M. (2016a) 'Fuzzy Analysis and Prediction of Commit Activity in Open Source Software Projects', IET Software, 10. Available at: https://doi.org/10.1049/iet-sen.2015.0087.

Chahal, K. and Saini, M. (2016b) 'Fuzzy Analysis and Prediction of Commit Activity in Open Source Software Projects', IET Software, 10. Available at: https://doi.org/10.1049/iet-sen.2015.0087.

CISA Open Source Software Security Roadmap | CISA (2023). Available at: https://www.cisa.gov/resources-tools/resources/cisa-open-source-software-security-roadmap (Accessed: 19 July 2024).

Decan, A. et al. (2020) 'GAP: Forecasting Commit Activity in git Projects', Journal of Systems and Software, 165, p. 110573. Available at: https://doi.org/10.1016/j.jss.2020.110573.

F. Maulana et al. (2023) 'Unmasking log4j's Vulnerability: Protecting Systems against Exploitation through Ethical Hacking and Cyberlaw Perspectives', in 2023 9th International Conference on Computer and Communication Engineering (ICCCE). 2023 9th International Conference on Computer and Communication Engineering (ICCCE), pp. 311–316. Available at: https://doi.org/10.1109/ICCCE58854.2023.10246082.

GitHub REST API documentation (no date) GitHub Docs. Available at: https://docs.github.com/_next/data/mjcX-Kg16OuchVhOk_10Y/en/free-pro-team@latest/rest.json?apiVersion=2022-11-28&versionId=free-pro-team%40latest&productId=rest (Accessed: 9 July 2024).

H. Gupta, A. Chaudhary, and A. Kumar (2022) 'Identification and Analysis of Log4j Vulnerability', in 2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART). 2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART), pp. 1580–1583. Available at: https://doi.org/10.1109/SMART55829.2022.10047372.

Leverett, E., Rhode, M. and Wedgbury, A. (2021) 'Vulnerability Forecasting: Theory and Practice', Digital Threats: Research and Practice, 3. Available at: https://doi.org/10.1145/3492328.

Maven – Introduction (no date). Available at: https://maven.apache.org/what-is-maven.html (Accessed: 21 July 2024).

Ohm, M. et al. (2020) 'Backstabber's Knife Collection: A Review of Open Source Software Supply Chain Attacks', CoRR, abs/2005.09535. Available at: https://arxiv.org/abs/2005.09535.

References

'pmdarima: Python's forecast::auto.arima equivalent' (no date). Available at: http://alkaline-ml.com/pmdarima (Accessed: 10 July 2024).

Pokhrel, N.R., Rodrigo, H. and Tsokos, C. (2017) 'Cybersecurity: Time Series Predictive Modeling of Vulnerabilities of Desktop Operating System Using Linear and Non-Linear Approach', Journal of Information Security, 08, pp. 362–382. Available at: https://doi.org/10.4236/jis.2017.84023.

Roumani, Y., Nwankpa, J.K. and Roumani, Y.F. (2015) 'Time series modeling of vulnerabilities', Computers & Security, 51, pp. 32–40. Available at: https://doi.org/10.1016/j.cose.2015.03.003.

S. Kyatam, A. Alhayajneh, and T. Hayajneh (2017) 'Heartbleed attacks implementation and vulnerability', in 2017 IEEE Long Island Systems, Applications and Technology Conference (LISAT). 2017 IEEE Long Island Systems, Applications and Technology Conference (LISAT), pp. 1–6. Available at: https://doi.org/10.1109/LISAT.2017.8001980.

Sen, R., Singh, S.S. and Borle, S. (2012a) 'Open source software success: Measures and analysis', Decision Support Systems, 52(2), pp. 364–372. Available at: https://doi.org/10.1016/j.dss.2011.09.003.

Sen, R., Singh, S.S. and Borle, S. (2012b) 'Open source software success: Measures and analysis', Decision Support Systems, 52(2), pp. 364–372. Available at: https://doi.org/10.1016/j.dss.2011.09.003.

Subramanian, V.N. (2020) 'An empirical study of the first contributions of developers to open source projects on GitHub', in Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings. New York, NY, USA: Association for Computing Machinery (ICSE '20), pp. 116–118. Available at: https://doi.org/10.1145/3377812.3382165.

Vulnerability APIs (no date). Available at: https://nvd.nist.gov/developers/vulnerabilities (Accessed: 9 July 2024).

Xia, T. et al. (2022) 'Predicting health indicators for open source projects (using hyperparameter optimization)', Empirical Softw. Engg., 27(6). Available at: https://doi.org/10.1007/s10664-022-10171-0.

Zajdel, S., Costa, D.E. and Mili, H. (2022) 'Open source software: an approach to controlling usage and risk in application ecosystems', in Proceedings of the 26th ACM International Systems and Software Product Line Conference - Volume A. New York, NY, USA: Association for Computing Machinery (SPLC '22), pp. 154–163. Available at: https://doi.org/10.1145/3546932.3547000.

Thank You For Listening!

Questions?