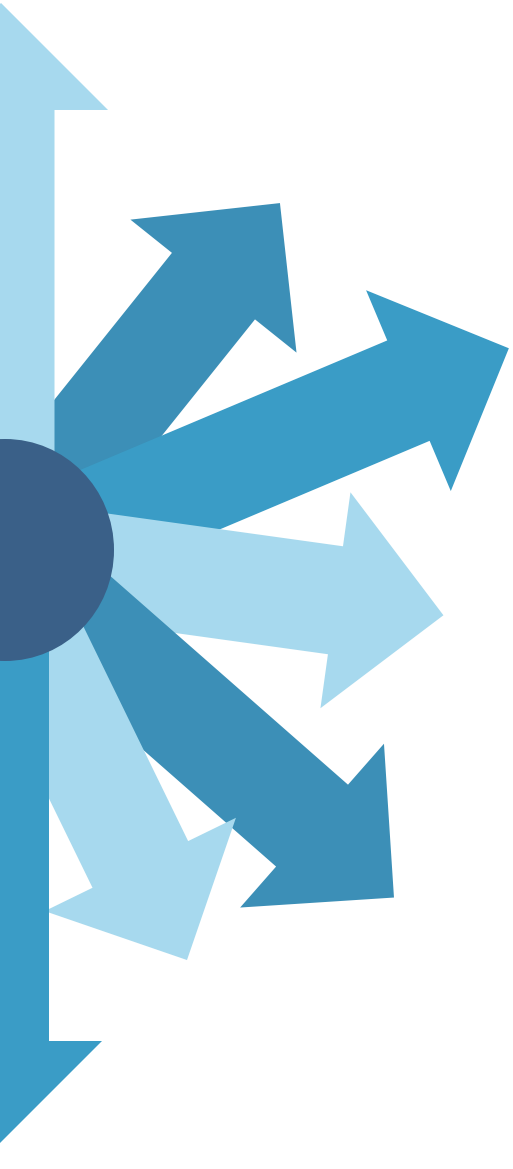
An analysis of LLM fine tuning and few-shot learning for flaky test detection and classification

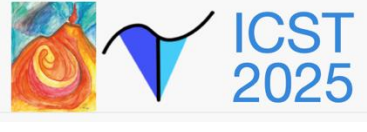
Riddhi More, Jeremy S. Bradbury

Software Engineering & Education Research Lab
Ontario Tech University, Oshawa, Canada

http://www.seerlab.ca



SEER LAB | OntarioTech UNIVERSITY



LLMs for Software

- Task automation
- Defect Prediction
- Code review
- Bug fixing
- Test Case generation
- Etc..

LLMs for Software

- Task automation
- Defect Prediction
- Code review
- Bug fixing
- Test Case generation
- Etc..

- Computational resources?
- Sustainability?
- Carbon footprint?
- Reproducibility?
- Etc..

What are Flaky Tests?

*Flaky tests exhibit **non-deterministic** behavior during execution, and they may **pass** or **fail** without any changes to the program under test.*

Datasets

- International Dataset of Flaky Tests (IDoFT) [1]
  - Flaky (3195) & non-flaky tests (618)
  - Tests organized by open-source projects
- FlakyCat Dataset [2]
  - Only flaky tests (369)
  - Diverse data from multiple open-source projects

TABLE I: IDoFT and FlakyCat datasets

IDoFT - Flaky vs. Non-Flaky	**# Tests**
Flaky tests	3195
Non-Flaky tests	618
Total	**3813**
IDoFT - Flaky Test Categories	
Non-deterministic-order-dependent (NDOD)	84
Non-order-dependent (NOD)	226
Order-dependent ()OD)	932
Non-idempotent-outcome (NIO)	196
Implementation-dependent (ID)	1617
Unknown-dependency (UD)	140
Total	**3195**
FlakyCat - Flaky Test Categories	
Async wait (Asyn.)	125
Concurrency (Conc.)	48
Time	42
Test Order Dependency (OD)	103
Unordered Collections (UC)	51
Total	**369**

[1] "International dataset of flaky tests (IDoFT)," https://github.com/TestingResearchIllinois/idoft.
[2] A. Akli, G. Haben, S. Habchi, M. Papadakis, and Y. Le Traon, "Flakycat: predicting flaky tests categories using few-shot learning," in 2023 IEEE/ACM Int. Conf. on Automation of Software Test (AST 2023), pp. 140–151.

Our research is motivated by the need for an improved understanding of the trade-offs between ***fine-tuning*** and ***few-shot learning (FSL)*** techniques in addressing flaky test challenges.

Fine-tuning vs FSL

Fine-Tuning:

- Adapts pre-trained models to general task-based datasets.
- Requires large labeled datasets for precision.
- Ideal for more global usage
- Enhances model accuracy and robustness
- Very resource-intensive.

Fine-tuning vs FSL

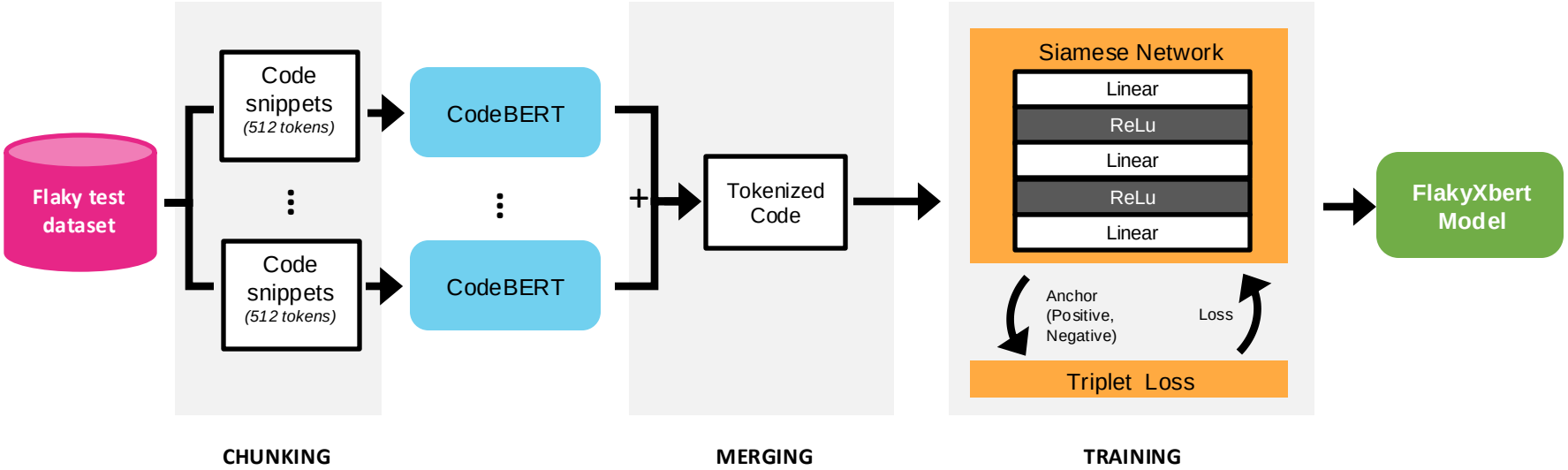
Fine-Tuning:

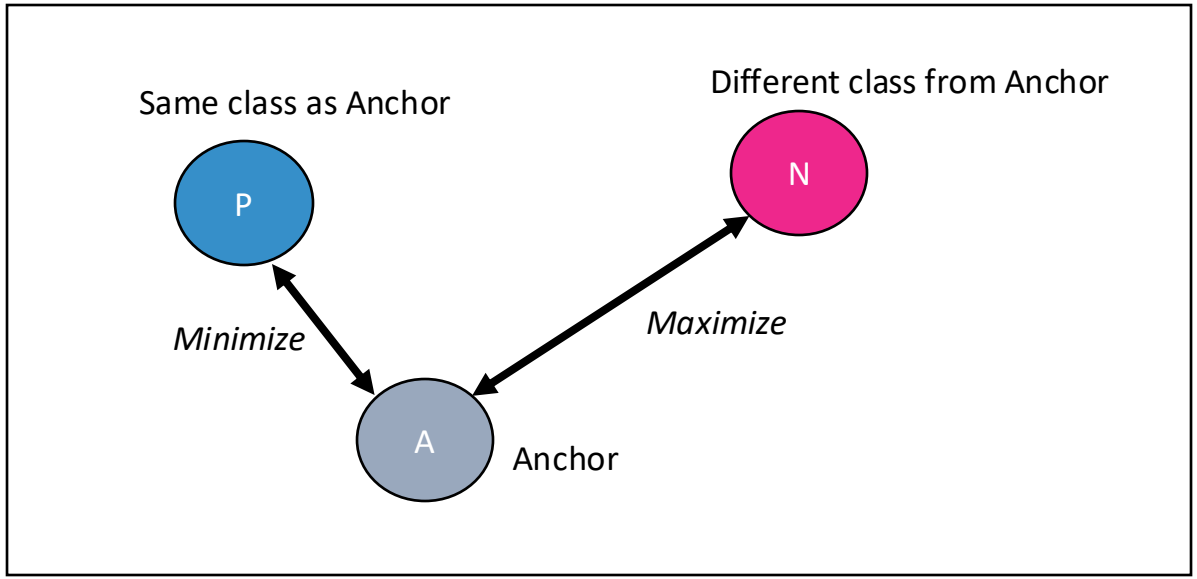
- Adapts pre-trained models to general task-based datasets.
- Requires large labeled datasets for precision.
- Ideal for more global usage
- Enhances model accuracy and robustness
- Very resource-intensive.

Few-Shot Learning (FSL):

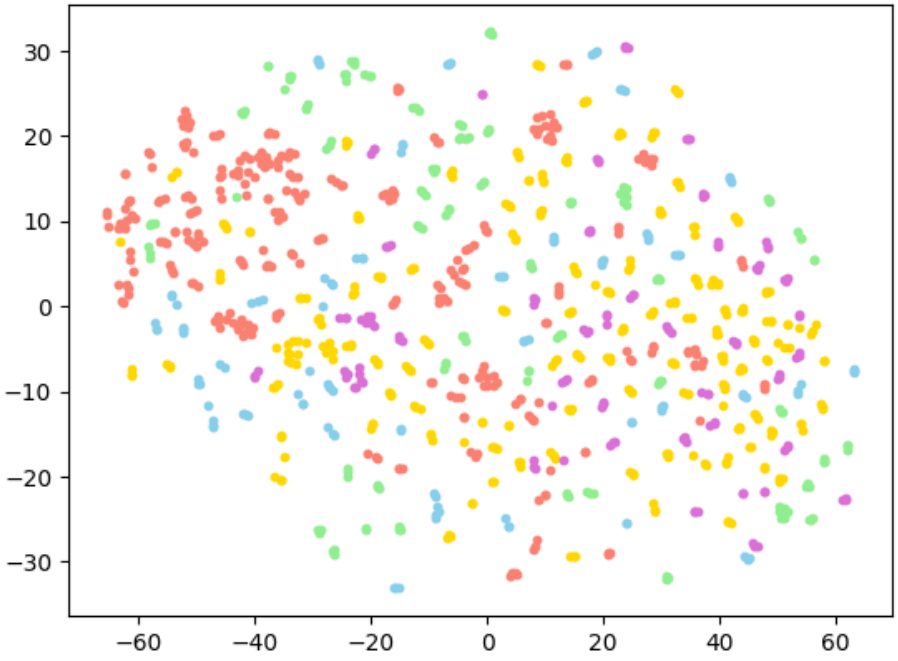
- Adapts model to hyper-specific context
- Works with minimal data.
- Ideal for data-constrained environments or rapid deployment.
- Faster and more flexible
- Does not generalise well.

FlakyXbert: Architecture

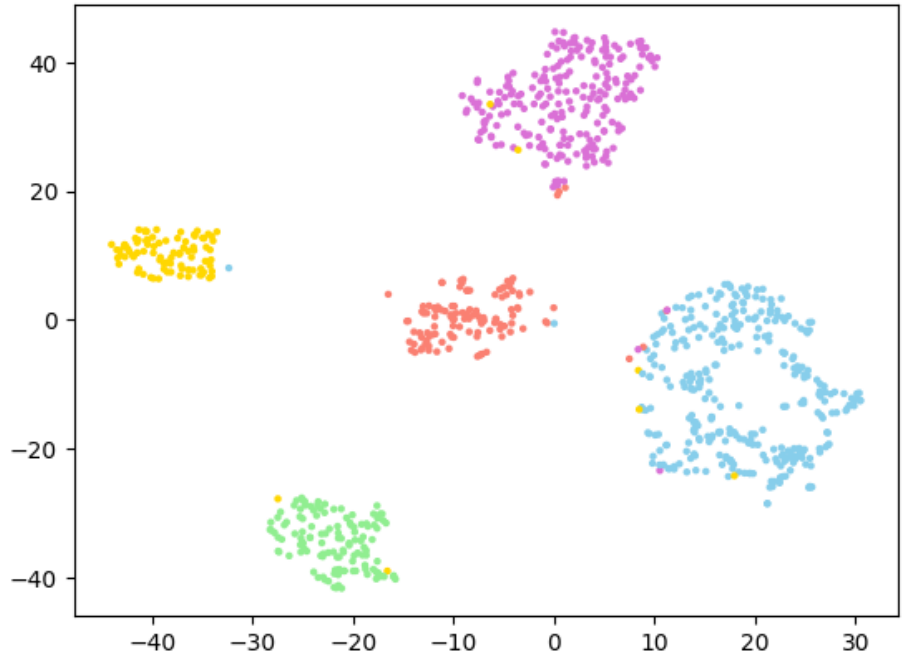




Triplet Loss function



Before FlakyXbert



After FlakyXbert

Research Questions

- **RQ1:** How does the performance of FSL and fine-tuning compare for flaky test detection and classification across different data scenarios?
  - **RQ1.1:** What is the performance of FSL compared to fine-tuning on small per-project data? (IDoFT)
  - **RQ1.2:** What is the performance of FSL compared to fine-tuning with a diverse data set? (FlakyCat)

- **RQ2:** What is the cost of FSL vs. fine-tuning?

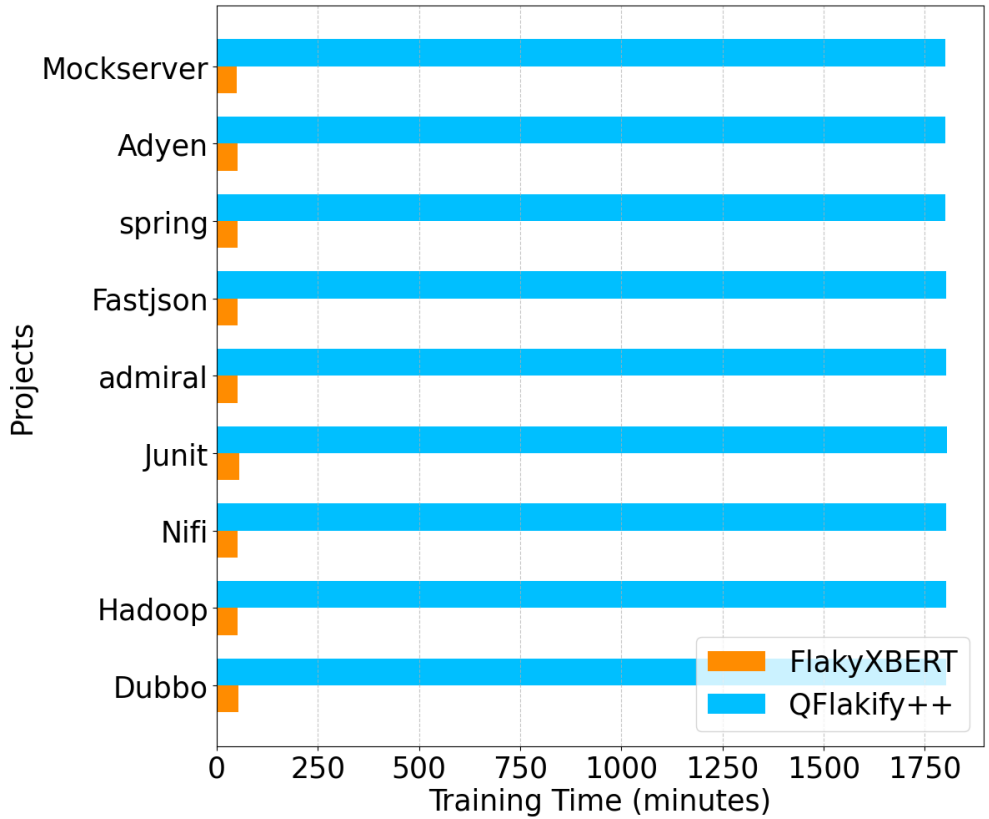
Results: IDoFT Detection – F1-Score

Project	Support	FlakyXbert	Flakify++	Q-Flakify++	FlakyQ_RF
Dubbo	186	88.7	87.0	91.0	88.0
Hadoop	149	95.0	99.0	100.0	100.0
Nifi	146	91.5	99.0	100.0	100.0
Junit	250	94.0	99.0	99.0	99.0
Admiral	113	91.3	99.0	99.0	99.0
Fastjson	109	91.3	91.0	93.0	93.0
spring	68	100.0	100.0	100.0	100.0
Adyen	89	30.0	43.0	52.0	45.0
Mockserver	39	100.0	100.0	100.0	100.0
Total/ Weighted Avg.	2105	95.1	95.6	96.0	95.4

Note: To see the full version, please refer to the original paper [2].

[2] S. Rahman, A. Baz, S. Misailovic, and A. Shi, "Quantizing large- language models for predicting flaky tests," in *Proc. of the 17th IEEE Int. Conf. on Software Testing, Verification and Validation (ICST 2024)*.

Results: IDoFT Detection – Training Time



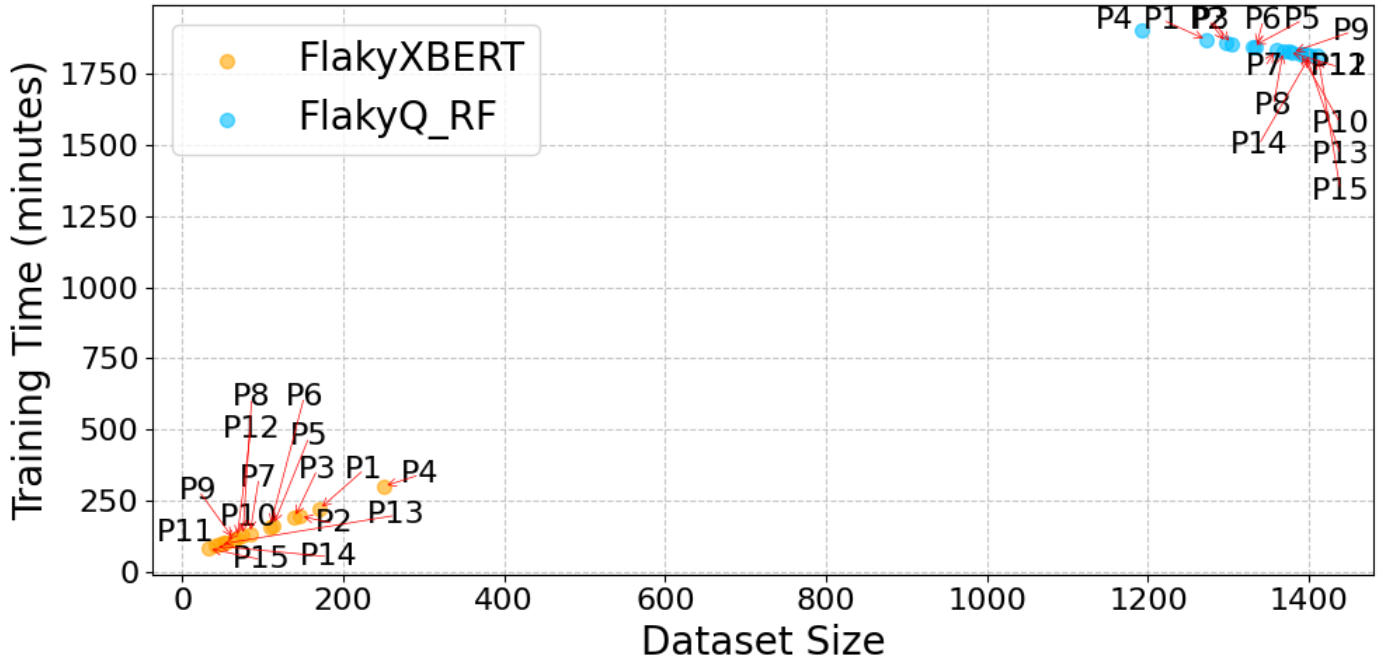
Results: IDoFT Classification – F1-Score

Project	Support	FlakyXbert	Flakify++	Q-Flakify++	FlakyQ_RF
Dubbo	170	71.0	77.0	77.0	73.0
Hadoop	146	51.0	90.0	88.0	91.0
Nifi	139	91.0	100.0	100.0	100.0
Junit	250	94.0	98.0	98.0	98.0
Ormlite	113	96.0	99.0	97.0	97.0
admiral	109	63.0	85.0	77.0	88.0
Wildfly	84	74.0	97.0	98.0	98.0
Mapper	75	100.0	93.0	80.0	100.0
Fastjson	64	82.0	91.0	88.0	94.0
Java	54	85.0	87.0	87.0	87.0
Biojava	51	91.0	19.0	16.0	32.0
spring	68	90.0	100.0	100.0	100.0
Hbase	47	76.0	98.0	95.0	98.0
hive	41	100.0	98.0	96.0	98.0
Nacos	32	96.0	100.0	97.0	97.0
Total/ Weighted Avg.	1810	76.5	90.2	93.0	94.8

Note: To see the full version, please refer to the original paper [2]

[2] S. Rahman, A. Baz, S. Misailovic, and A. Shi, "Quantizing large- language models for predicting flaky tests," in *Proc. of the 17th IEEE Int. Conf. on Software Testing, Verification and Validation (ICST 2024).*

Results: IDoFT Classification – Training Time vs. Dataset Size

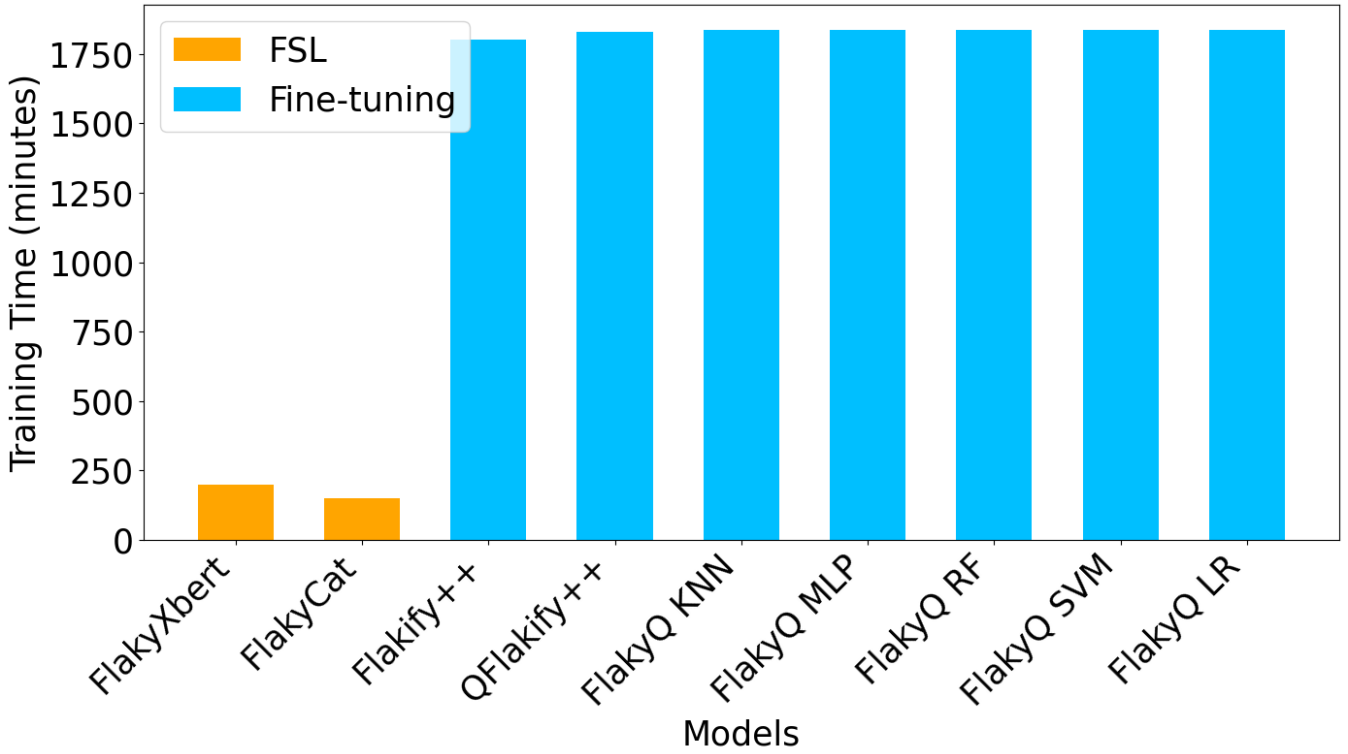


Results: FlakyCat Classification – F1-Score

Technique	Classifier	Asyn.	Conc.	Time	UC	OD	Weighted Avg.
Few-shot Learning (FSL)	FlakyXbert	98.0	90.0	93.0	97.0	99.0	96.0
	FlakyXbert (without augmentation)	52.0	80.0	36.0	43.0	78.0	60.0
	FlakyCat	72.0	36.0	75.0	72.0	73.0	67.5
Fine-tuning (FT)	Flakify++	94.8	93.3	96.9	96.1	97.1	95.6
	Q-Flakify++	92.6	87.1	96.9	95.0	95.8	93.6
	FlakyQ_KNN	93.1	90.7	95.5	95.0	96.3	94.2
	FlakyQ_MLP	94.0	89.7	95.5	94.8	96.6	94.5
	FlakyQ_RF	94.3	91.5	95.5	94.8	96.6	94.8
	FlakyQ_SVM	93.8	89.0	95.5	93.2	96.1	93.9
	FlakyQ_LR	92.7	89.8	95.5	94.8	96.1	93.9
Hybrid (FSL + FT)	FSL++	93.7	90.3	97.9	95.9	96.7	91.5

Note: Asyn. = Async Wait, Conc. = Concurrency, Time = Test Order Dependency, UC = Unordered Collections, OD = Other Dependencies

Results: FlakyCat Classification – Training Time



Conclusion

FSL (Few-Shot Learning) in the FlakyXbert model is effective in environments with sparse data.

- It leverages fewer labelled examples to classify and predict flakiness categories
- Small scale companies and research labs can benefit
- Retraining is required for a different project or use case

Conclusion

Fine-tuning, which requires more extensive data, excels by adapting to a broader range of features.

- It handles diversity better and typically offers more accurate predictions
- It demands greater computational resources and longer training time
- Minimal/no retraining is required when it comes to another project or use case

Conclusion

- The choice between ***FSL*** and ***Fine-tuning*** depends on balancing trade-offs between:
  - Data availability.
  - Computational efficiency.
  - Accuracy of Results
  - Adaptability to diverse flaky test characteristics.

Threats to Validity & Future Work

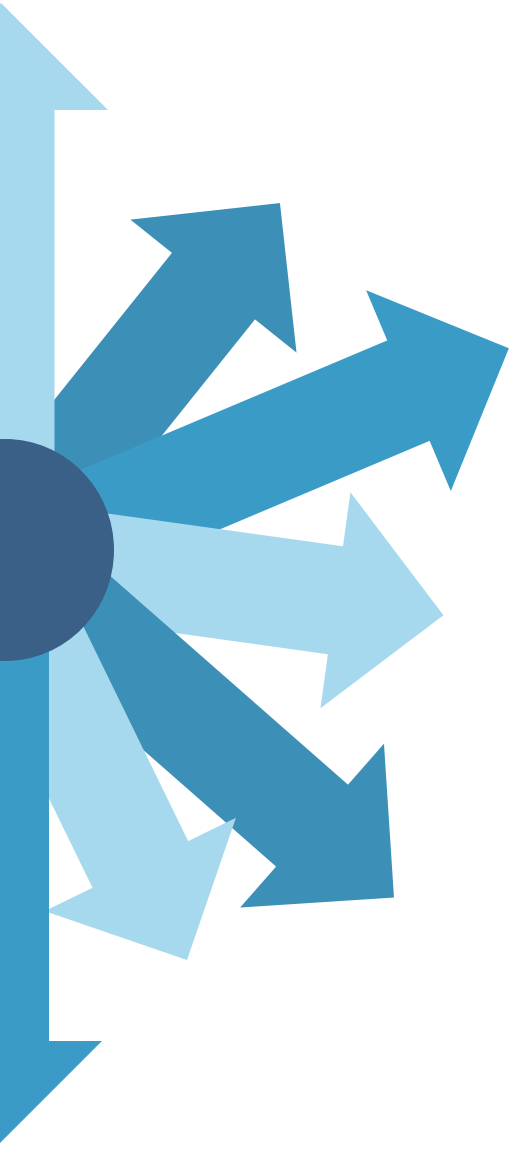
- We used publicly available datasets and consistent evaluation metrics across models
- Addressed class imbalances with augmentation – can lead to bias
- While further tuning could improve performance, variability in results across projects and uncertainty in generalizing our findings remain.
- More experimentation is needed to confirm broader applicability beyond flaky test detection.

An analysis of LLM fine tuning and few-shot learning for flaky test detection and classification

Riddhi More, Jeremy S. Bradbury

Software Engineering & Education Research Lab
Ontario Tech University, Oshawa, Canada

http://www.seerlab.ca



SEER LAB | OntarioTech UNIVERSITY