Use of AI for Log Analysis in CI/CD Pipelines

Bachelor Thesis - Defence

Maid Ališić

23 July 2025

FH Oberösterreich · Campus Hagenberg

Road map

Problem context

Research questions

Method

Architecture

Data & evaluation

Results

Impact

Problem context

Why do we care?

- CI/CD emits $\approx$ 10-20 GB of build, test & deploy logs *per day*.

Why do we care?

- CI/CD emits ≈ 10-20 GB of build, test & deploy logs *per day*.
- Manual `grep` slows the merge queue; critical faults slip through.

Why do we care?

- CI/CD emits $\approx$ 10-20 GB of build, test & deploy logs *per day*.
- Manual `grep` slows the merge queue; critical faults slip through.
- Business **SLO**: feedback within $\leq$ 200 ms per pipeline.

Why do we care?

- CI/CD emits $\approx$ 10-20 GB of build, test & deploy logs *per day*.
- Manual `grep` slows the merge queue; critical faults slip through.
- Business **SLO**: feedback within $\leq$ 200 ms per pipeline.
- Logs can leak customer IDs → **no SaaS export**.

Operational pain points

1. **Context-sensitivity** - identical tokens can be harmless or fatal.

Operational pain points

1. **Context-sensitivity** - identical tokens can be harmless or fatal.
2. **Concept drift** - each merge may rename tests or switches.

Operational pain points

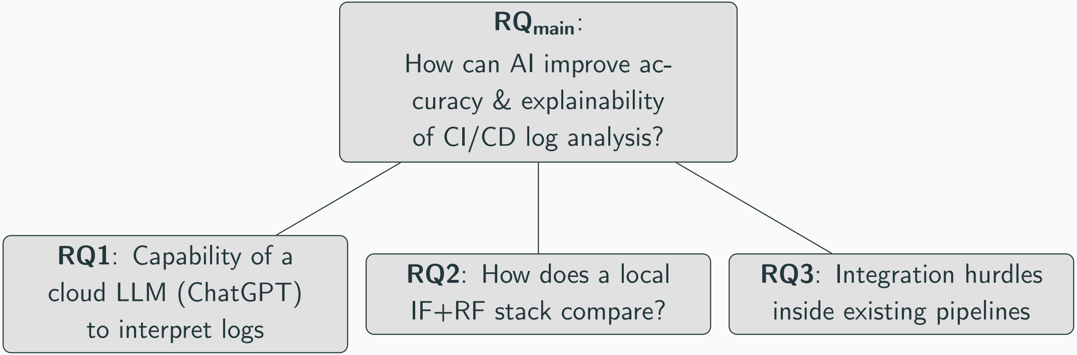
1. **Context-sensitivity** - identical tokens can be harmless or fatal.
2. **Concept drift** - each merge may rename tests or switches.
3. **Latency pressure** - analysis must finish before runner teardown.

Operational pain points

1. **Context-sensitivity** - identical tokens can be harmless or fatal.
2. **Concept drift** - each merge may rename tests or switches.
3. **Latency pressure** - analysis must finish before runner teardown.
4. **Alert fatigue** - regex rule sets grow without bound.

Research questions

Research questions



RQ$_{\text{main}}$: How can AI improve accuracy & explainability of CI/CD log analysis?

RQ1: Capability of a cloud LLM (ChatGPT) to interpret logs

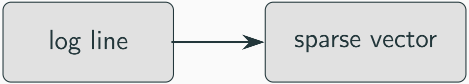
RQ2: How does a local IF+RF stack compare?

RQ3: Integration hurdles inside existing pipelines

Method

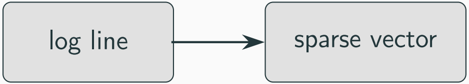
Vectorisation pipeline ①

1. **Normalise** - strip timestamps, colours, IDs.



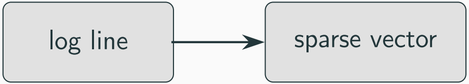
Vectorisation pipeline ①

1. **Normalise** - strip timestamps, colours, IDs.
2. **Tokenise** - 1-2-grams.



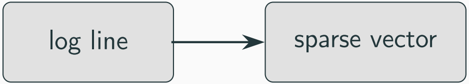
Vectorisation pipeline ①

1. **Normalise** - strip timestamps, colours, IDs.
2. **Tokenise** - 1-2-grams.
3. Weight with TF-IDF.



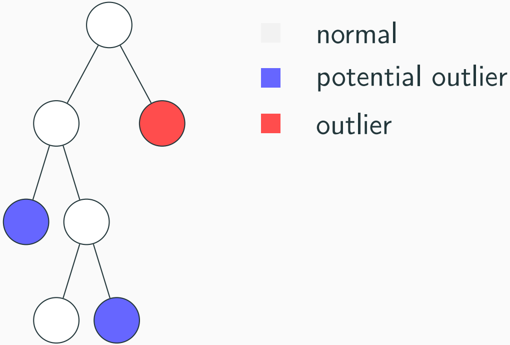
Vectorisation pipeline ①

1. **Normalise** - strip timestamps, colours, IDs.
2. **Tokenise** - 1-2-grams.
3. Weight with TF-IDF.
4. Produce 50 000-dim sparse vector; $> 10^5$ lines / s on one core.



Isolation Forest ② - intuition

- Random binary partitioning isolates unusual lines in fewer splits.



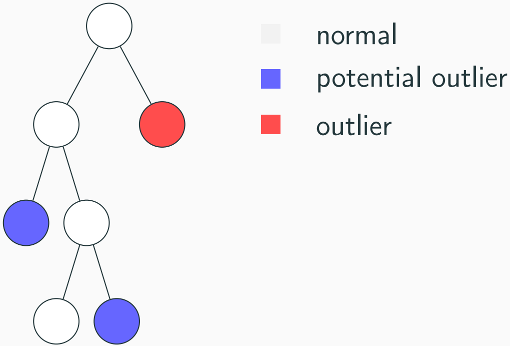
⬜ normal

🟦 potential outlier

🟥 outlier

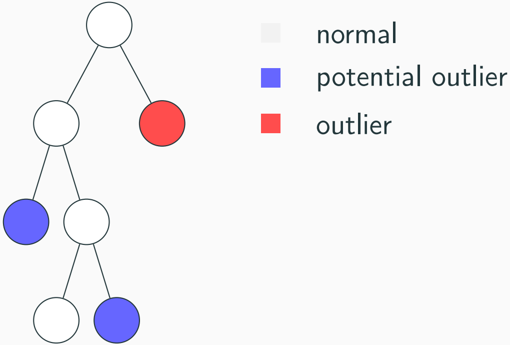
Isolation Forest ② - intuition

- Random binary partitioning isolates unusual lines in fewer splits.
- Score $s(x) = 2^{-h(x)/c(n)} \in [0,1]$ if high → outlier.



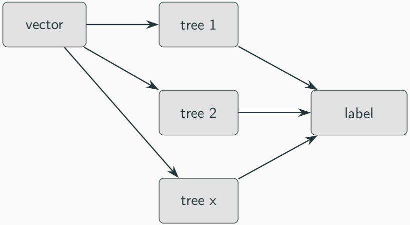
Isolation Forest ② - intuition

- Random binary partitioning isolates unusual lines in fewer splits.
- Score $s(x) = 2^{-h(x)/c(n)} \in [0,1]$ if high → outlier.
- CPU-only: $\approx 30\,\mu s$ per line.



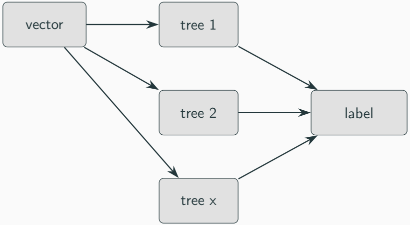
Random Forest ③ - error labelling

- Converts Isolation Forest-flags into 7 domain labels.



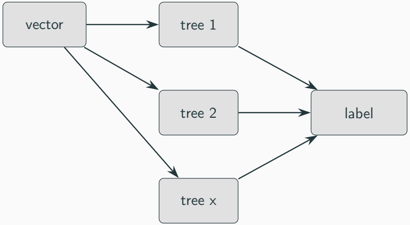
Random Forest ③ - error labelling

- Converts Isolation Forest-flags into 7 domain labels.
- Majority vote = deterministic, auditable output.



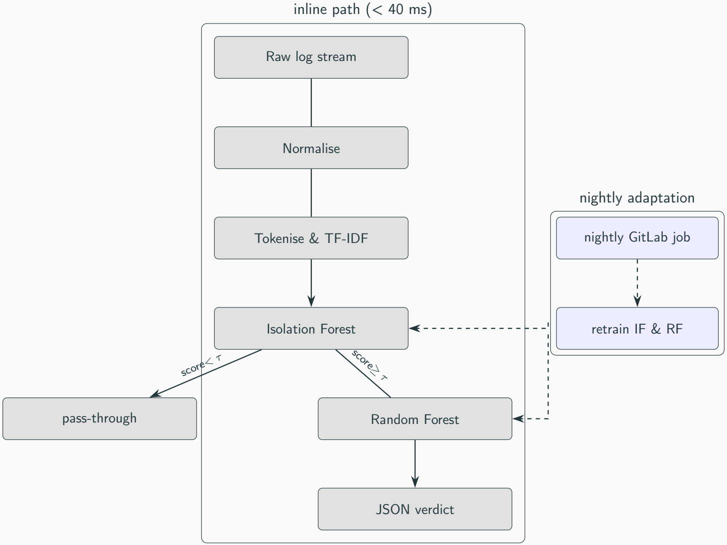
Random Forest ③ - error labelling

- Converts Isolation Forest-flags into 7 domain labels.
- Majority vote = deterministic, auditable output.
- Nightly retrain < 90 s; warm-start handles drift.



Architecture

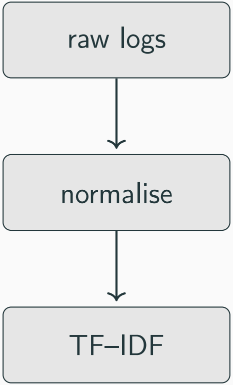
End-to-end pipeline (< 40 ms inline)



Data & evaluation

Datasets & metrics

- 117 k *macOS* logs + 655 k *OpenSSH* logs



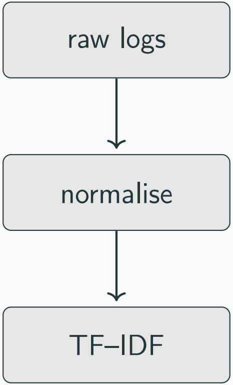
raw logs

normalise

TF–IDF

Datasets & metrics

- 117 k *macOS* logs + 655 k *OpenSSH* logs
- 504 labelled anomalies (≈1 : 200 skew)



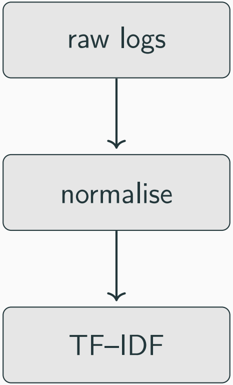
raw logs

normalise

TF–IDF

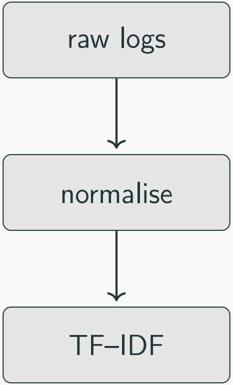
Datasets & metrics

- 117 k *macOS* logs + 655 k *OpenSSH* logs
- 504 labelled anomalies (≈1 : 200 skew)
- Split 70 / 15 / 15 % (train / val / test)



Datasets & metrics

- 117 k *macOS* logs + 655 k *OpenSSH* logs
- 504 labelled anomalies (≈1 : 200 skew)
- Split 70 / 15 / 15 % (train / val / test)
- Metrics: Macro-F_1, AUPRC, p99.9 latency



raw logs

normalise

TF–IDF

Results

Headline numbers

	Precision	Recall	F_1
Detection (Isolation Forest)	0.91	0.88	0.89
Classification (Random Forest)	0.99	0.99	0.99

Throughput: 45 000 lines/s | p99.9 latency: 37 ms

Impact

Operational impact

- **Latency**: minutes → **milliseconds** (inline verdict).

Operational impact

- **Latency**: minutes → **milliseconds** (inline verdict).
- **Cost-free**: 2.3 k lines of code, CPU-only, no token fees.

Operational impact

- **Latency**: minutes → **milliseconds** (inline verdict).
- **Cost-free**: 2.3 k lines of code, CPU-only, no token fees.
- **GDPR compliant**: logs never leave the VPN.

Wrap-up

Take-away

Light-weight on-prem ML matches AIOps SaaS

without latency, cost or privacy pain.

Questions welcome - thank you!