



# Flakiness Detection: An Extensive Analysis



**Prof. Fabio Palomba**  
**Dott. Valeria Pontillo**

**Angelo Afeltra**  
**Mtr: 0522501354**

# **Che cos'è il testing?**

**Il testing è il processo** di valutazione di un'applicazione software **per identificare difetti, errori o comportamenti indesiderati** con l'obiettivo di garantire un alta qualità.

# Quali vantaggi offre?



Fornisce fiducia agli utenti

Miglioramento della qualità

Individuazione dei difetti

Aumenta la sicurezza

Risparmio di tempi e costi

Riduce i danni finanziari

Riduzione del rischio

Conformità ai requisiti

Preserva la reputazione dell'azienda

Migliora l'esperienza utente



# Quali vantaggi offre?



Miglioramento  
della qualità

Individuazione dei  
difetti

Fornisce feedback  
agli utenti

Riduzione della  
sicurezza



**Il testing fornisce tali vantaggi solamente se i risultati dei test sono consistenti, caratteristica non valida per i test affetti dalla "flakiness".**

Riduzione del  
rischio

Conformità ai  
requisiti

Preserva la  
reputazione dell'  
azienda

Migliora  
l'esperienza  
utente



# Che cos'è la “Flakiness”?

I test **flaky**, sono test non deterministici che **producono un comportamento sia di pass che di failure in modo intermittente** quando vengono eseguiti senza aver apportato modifiche agli input, all'ambiente d'esecuzione o al codice sottoposto a test.

```
def CreateGapLease(self):  
    data_file = open('/leases/gap', 'w+')  
    data_file.write(contract_data)  
    data_file.close()  
  
def testCreateGapLease(self):  
    contract_writer.CreateGapLease()  
    self.assertEqual(ReadFileContents('/leases/gap'),  
                    contract_data)
```



# Cause e problemi

- Concorrenza
- Dipendenza da risorse esterna
- Stato iniziale non deterministico
- Ambiente di esecuzione variabile
- Ordine di esecuzione
- Ritardi temporali
- Scarsa qualità dei casi di test



**Debug Difficile**

**Calo di prestazioni  
del team**

**Perdita di fiducia**



# Cause e problemi

- Co
- Di
- Sta
- An
- Or
- Rit
- Sc

## Perdita di fiducia

*Il vero costo della flakiness è una mancanza di fiducia nei tuoi test.... Se non hai fiducia nei tuoi, test allora non sei in una posizione migliore di un team che ha zero test.*

# Come avviene la detection?



L'approccio più utilizzato per l'identificazione dei test flaky è quello della **ReRuns**, ovvero **rieseguire più volte la suite di test** ed osservare il comportamento dei singoli test.

# Come avviene la detection?



L'approccio più utilizzato per l'identificazione dei test **flaky** è quello della **ReRuns**, ovvero **rieseguire più volte la suite di test** ed osservare il comportamento dei singoli test.

## Limiti di tale approccio

1. Troppo costoso, specialmente per grandi suite di test
2. Non esiste un numero N di esecuzioni che ci assicura che un test non sia **flaky**.



# QUAL'È IL NOSTO OBIETTIVO?

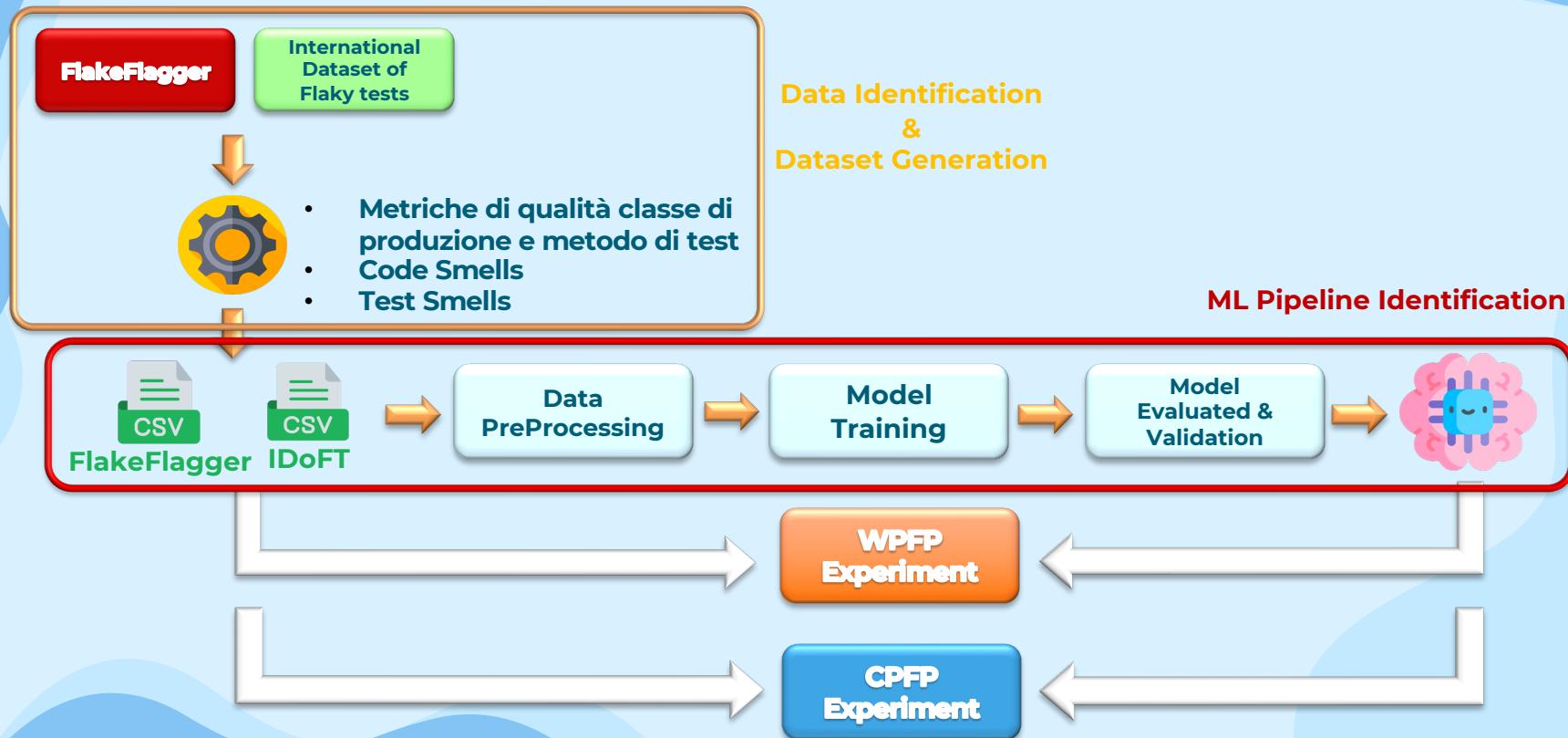
VERIFICARE SE IL MACHINE LEARNING È UNA VALIDA ALTERNATIVA PER LA DETECTION DEI TEST FLAKY

---

RQ1. Quanto è efficace un approccio basato sul machine learning per il rilevamento della flakiness, in una validazione *within-project*?

RQ2. Quanto è efficace un approccio basato sul machine learning per il rilevamento della flakiness, in una validazione *cross-project*?

# Cosa è stato fatto



# Pipeline Identification

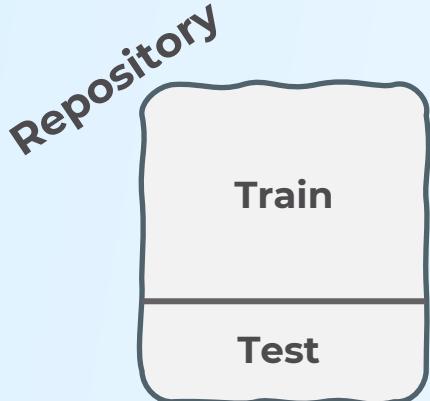




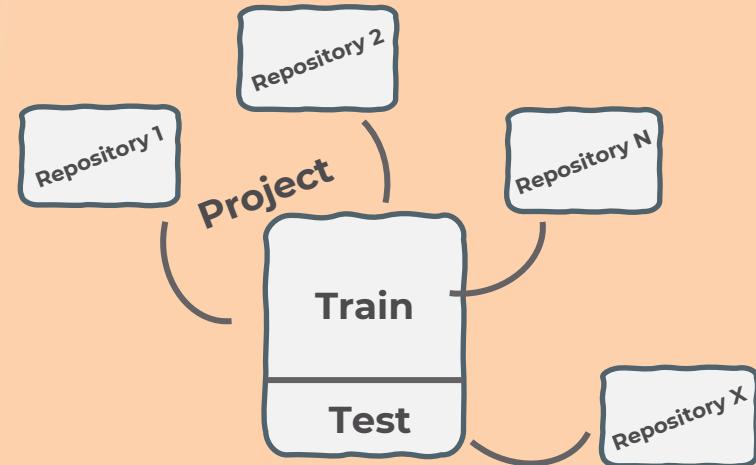
# Whitin/Cross - Project Flaky Test Prediction

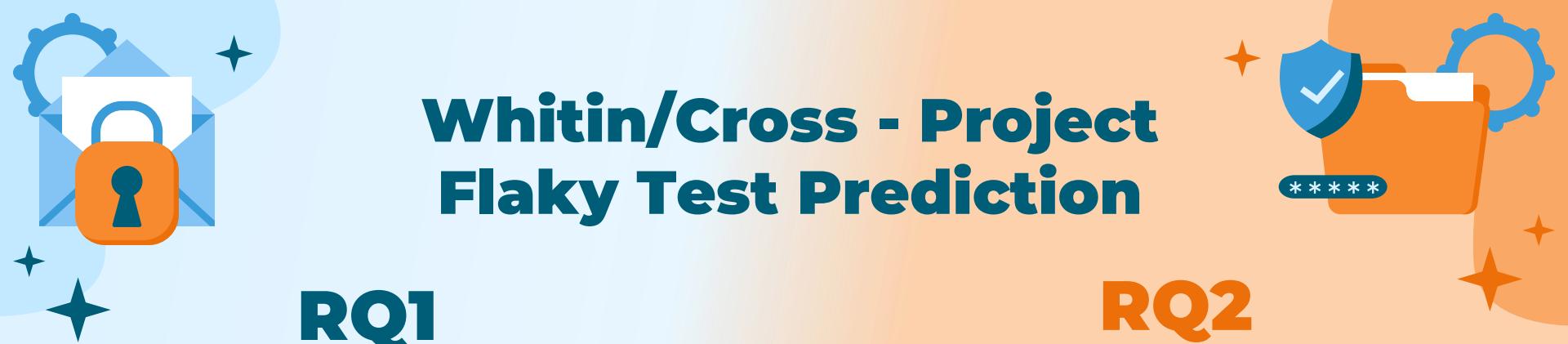


**RQ1**



**RQ2**





# Whitin/Cross - Project Flaky Test Prediction

**RQ1**



**RQ2**



# Perché il crollo di prestazioni in un contesto CPFP?

Feature	Mean Sorgente	Mean Target	STD Sorgente	STD Target	Feature Importance
tloc	11,1	6,7	12,5	5,3	0,104
lcom5	3,2	2,2	9,3	2,5	0,087
eagerTest	1,5	0,8	2,6	1,5	0,084
Assertion Roulette	1,5	0,4	3,5	1,0	0,073
mpc	106,2	29,2	385,3	26,8	0,069
cbo	22,9	16,8	44,8	11,2	0,067
...	...	...	...	...	...
testRunWar	0,0	0,0	0,0	0,0	0,0



# CPFP EXPERIMENTS

Metodi non supervisionati  
(Feature-Based)

Filtro di Burak

Modelli locali  
basati sul  
clustering

Modelli locali  
basati su  
repository

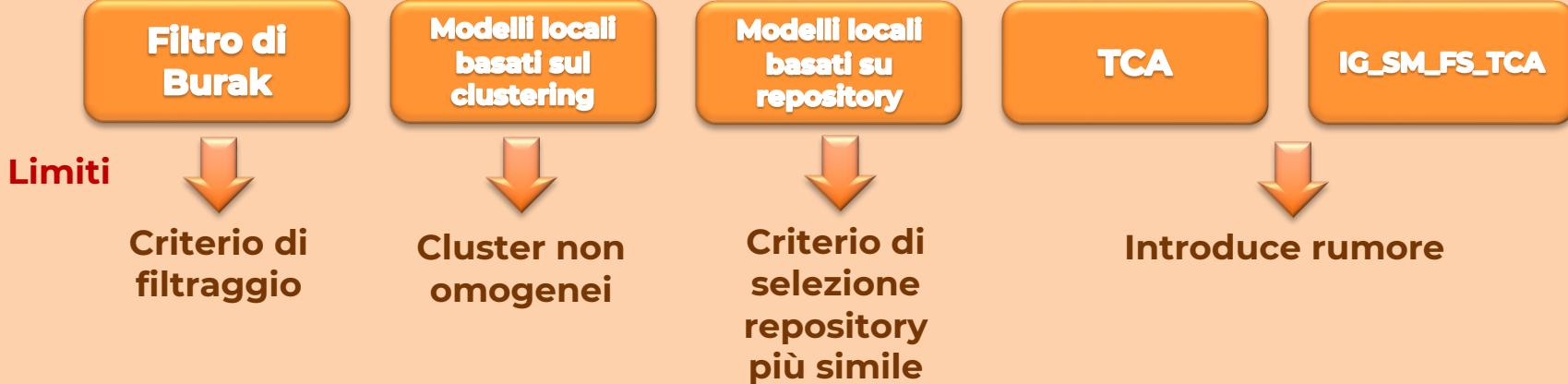
IG\_SM\_FS\_TCA

TCA

Metodi supervisionati  
(Instance-Based)

TrAdaBoost

# CPFP EXPERIMENTS



# CPFP EXPERIMENTS



## Strategia

Si assegnano dei pesi ai dati di training in modo tale da ridurre l'errore sui dati di test (dati etichettati della repository target).

TrAdaBoost

## Risultato

F1: 0% → F1: 87%

# Conclusioni

**RQ1.** Quanto è efficace un approccio basato sul machine learning per il rilevamento della flakiness, in una validazione within-project?

Ottime Prestazioni

Utilizzo Limitato

**RQ2.** Quanto è efficace un approccio basato sul machine learning per il rilevamento della flakiness, in una validazione cross-project?

Pipeline Inutilizzabile  
senza dati etichettati  
della repository target

# Conclusioni

**RQ1.** Quanto è efficace un approccio basato sul machine learning per il rilevamento della flakiness, in una validazione within-project?

Ottime Prestazioni

Utilizzo Limitato

**RQ2.** Quanto è efficace un approccio basato sul machine learning per il rilevamento della flakiness, in una validazione cross-project?

Pipeline Inutilizzabile  
senza dati etichettati  
della repository target

Sviluppi Futuri

NLP

Utilizzo metriche  
ottenute tramite  
analisi dinamica

# Flakiness Detection: An Extensive Analysis

## Che cos'è il testing?

Il testing è il processo di valutazione di un'applicazione software per identificare difetti, errori o comportamenti indesiderati con l'obiettivo di garantire un'alta qualità.

## Che cos'è la "Flakiness"?

I test flaky sono test non deterministici che producono un comportamento di pass o fail di failure in modo intermitente quando vengono eseguiti senza aver apportato alcuna modifica all'ambiente d'esecuzione o al codice sottoposto a test.

```
def test_crash():
    file = open('leaves/ppt', 'w')
    data_file.write('content')
    data_file.close()
    assert 'content' in file.read()
    content.write('content')
    content.close()
    assert 'content' in file.read()
```

## Come avviene la detection?

L'approccio più utilizzato per l'identificazione dei test flaky è quello della ReRuns, ovvero **rieseguire più volte** la suite di test ed osservare il comportamento dei singoli test.

### Limiti di tale approccio

1. Troppo costoso, specialmente per grandi suite di test.
2. Non esiste un numero N di esecuzioni che ci assicura che un test non sia flaky.

## QUAL'E IL NOSTO OBIETTIVO?

VERIFICARE SE IL MACHINE LEARNING È UNA VALIDA ALTERNATIVA A FLAKINESS DETECTION

- RQ1: Quante è efficace un approccio basato sul machine learning per il riconoscimento della flakiness, utilizzando la validazione cross-project?
- RQ2: Quante è efficace un approccio basato sul machine learning per il riconoscimento della flakiness, in una validazione cross-project?
- RQ3: Quante è efficace un approccio basato sul machine learning per il riconoscimento della flakiness, in una validazione cross-project?

## Cosa è stato fatto



## Pipeline Identification



## Within/Cross - Project Flaky Test Prediction

RQ1



RQ2



## Conclusioni

RQ1: Quante è efficace un approccio basato sul machine learning per il riconoscimento della flakiness, utilizzando la validazione cross-project?

RQ2: Quante è efficace un approccio basato sul machine learning per il riconoscimento della flakiness, in una validazione cross-project?

RQ3: Quante è efficace un approccio basato sul machine learning per il riconoscimento della flakiness, in una validazione cross-project?

Ottimo Prestazioni

Utile Lineare

Modelli personalizzati  
alla specifica applicazione

SVL

Modelli personalizzati  
alla specifica applicazione

MLP

Modelli personalizzati  
alla specifica applicazione

MLP

Prof. Fabio Palomba  
Dott. Valeria Pontillo



Grazie  
dell'attenzione.