



**Corso di Laurea Magistrale in Informatica  
Curriculum Software Engineering and IT Management**

# **Infrastructure-as-Code Defect Prediction using PDG metrics**

**Prof. Dario Di Nucci  
Dott. ssa Valeria Pontillo**

**Gerardo Iuliano  
Mat.: 0522501329**



## *Infrastructure as Code*

### **Cos'è:**

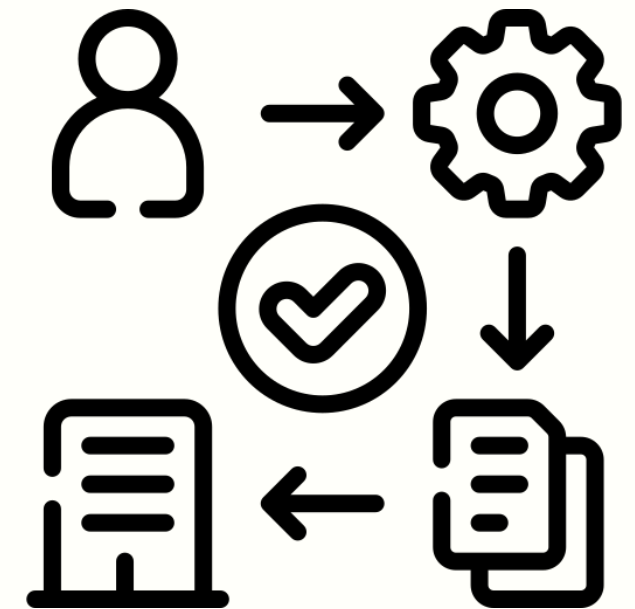
- Pratica DevOps che vede l'infrastruttura come codice
- Gestione automatizzata della configurazione dell'infrastruttura

### **Vantaggi:**

- Automazione
- Tracciabilità e versioning
- Ripetibilità

### **Tools:**

- Ansible, Chef e Puppet
- Kubernetes, Docker Swarm (container)
- Terraform, Cloudify (virtual machine)



## *Ansible*

### **Cos'è:**

- Strumento open source di automazione della configurazione e orchestrazione di sistemi IT

### **Vantaggi:**

- Agent-less
- Linguaggio dichiarativo
- Idempotenza

### **Struttura:**

- Playbook
- Task
- Roles
- Inventory



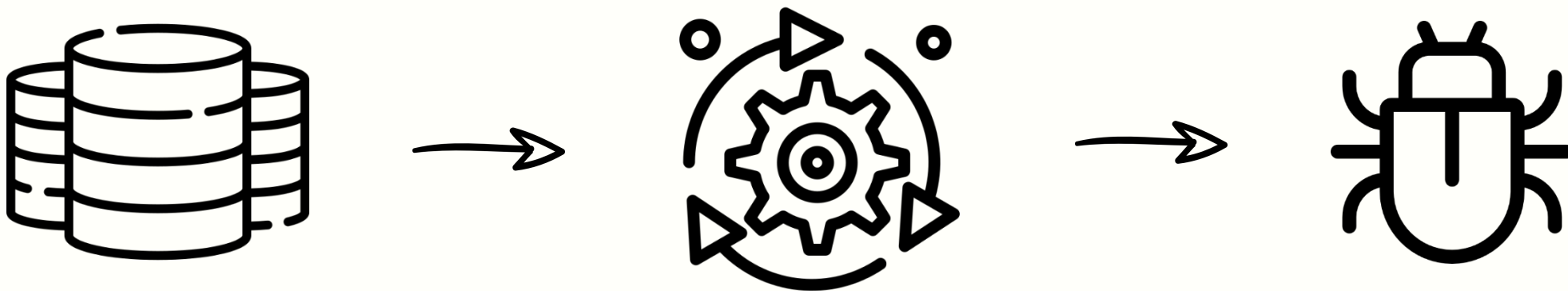
## *Defect Prediction*

### **Cos'è**

- È una pratica che mira a prevedere e identificare potenziali difetti o errori nel software

### **Come**

- Metriche strutturali
- Metriche di processo
- Metriche basate sugli sviluppatori
- Metriche basate su PDG



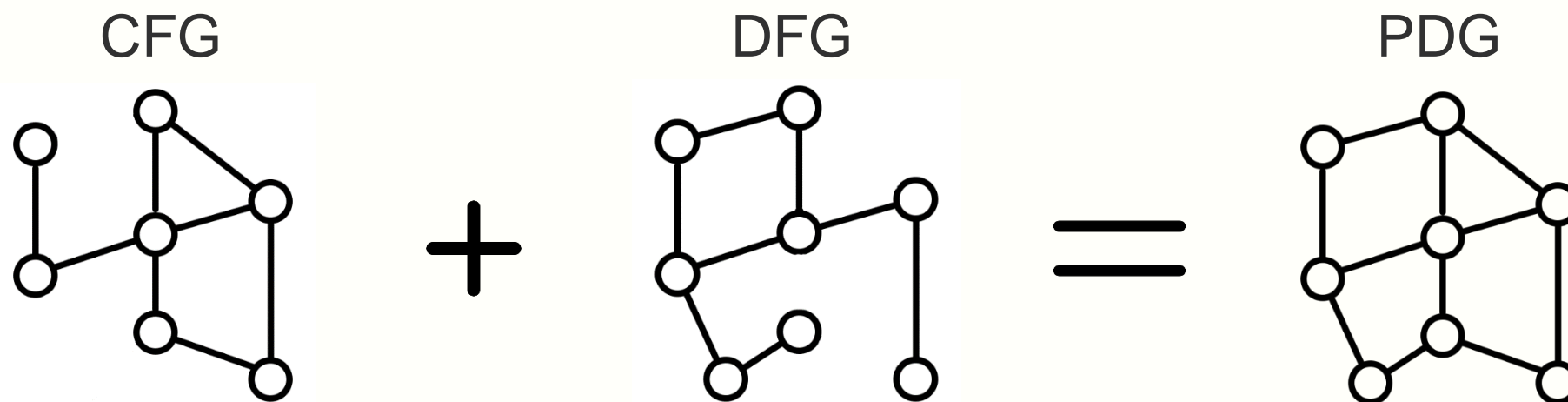
## *Program Dependence Graph*

### **Cos'è**

- Una rappresentazione tramite grafo delle dipendenze e delle relazioni nel codice
- Combinazione di CFG e DFG

### **Usi e vantaggi**

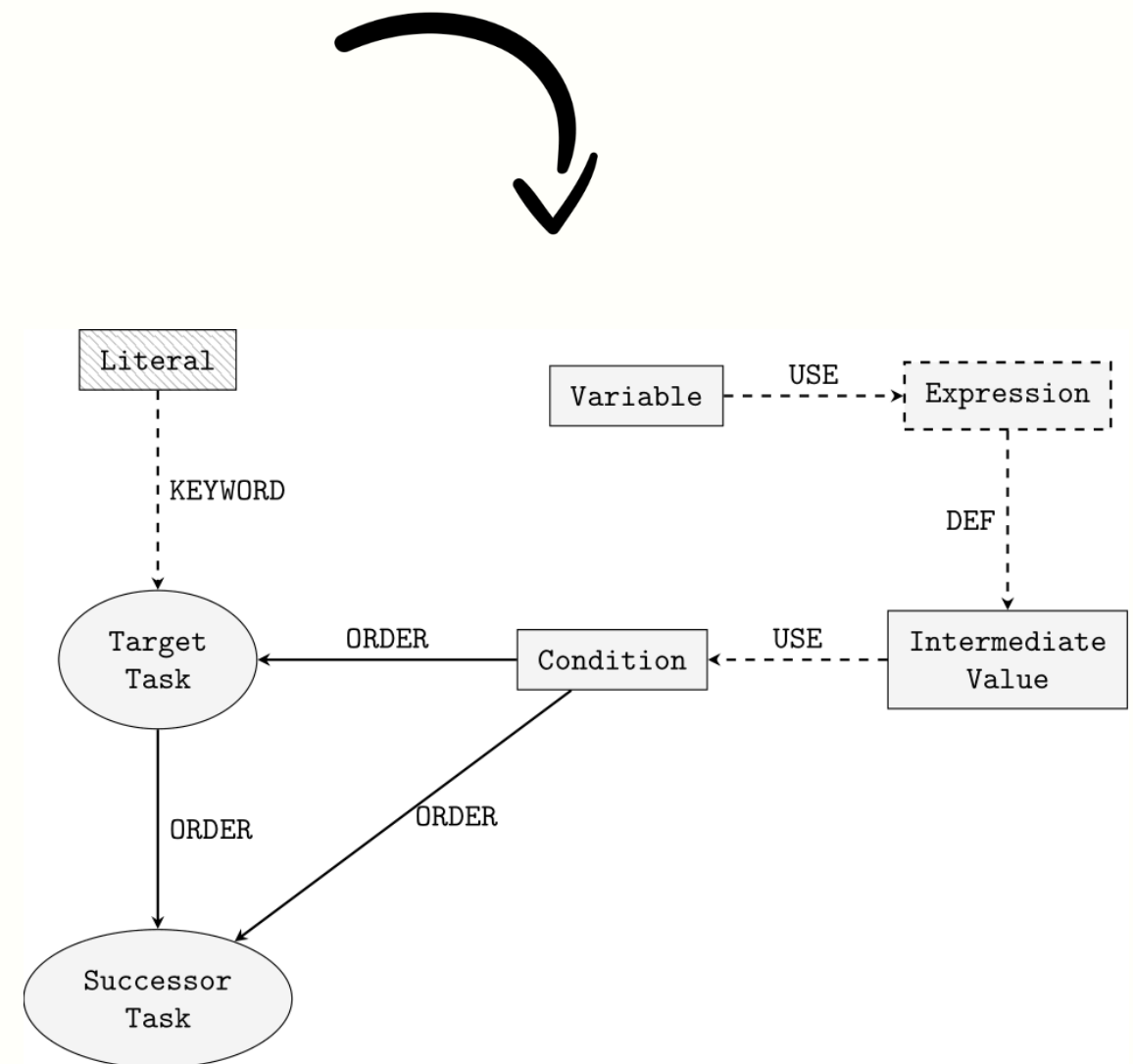
- Analisi del codice, identificazione dei problemi
- Comprensione del codice complesso



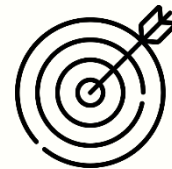
# Estrazione delle metriche

```
1  name: Gather Distribution Info
2  ansible.builtin.setup:
3    gather_subset: distribution!
4  when:
5    - ansible_distribution is not defined
```

<i><b>Metrica</b></i>	<i><b>Valore</b></i>
verticesCount	7
edgesCount	7
globalInput	2



## Obiettivo



L'obiettivo dello studio è valutare se le metriche estratte dal Program Dependence Graph sono adatte per i modelli di previsione dei difetti in un within-project scenario, con lo scopo di migliorare il rilevamento precoce dei difetti negli script IaC.

## *Research Questions*

**RQ1.** Quali metriche relative al Program Dependence Graph sono buoni predittori di difetti?

**RQ2.** Qual è il miglior modello di previsione dei difetti basato sulle metriche estratte da un PDG?

**RQ3.** In che misura un modello basato su metriche PDG è complementare ai modelli dello stato dell'arte?

**RQ4.** Una combinazione di metriche basate su PDG, strutturali e di processo migliora le prestazioni?



## Context Selection

**80** progetti Ansible open source che rispecchiano tali criteri:

- Almeno l'11% dei file presenti nel repository sono script lac
- Evidenzia una pratica di Continuous Integration
- Ha una frequenza di commit di almeno 2 al mese in media
- Ha una frequenza di issue di almeno 0,02 al mese in media
- Ha almeno 190 linee di codice sorgente
- Ha almeno due collaboratori di base



## *Empirical Study Variables*

**Variabile Dipendente:** è un valore binario che indica la presenza/assenza di un difetto.

**Variabili Indipendenti:** Insieme di metriche basate sull'analisi del Program Dependence Graph.

*maxPdgVertices, lackOfCohesion, verticesCount, edgesCount, edgesToVerticesRatio, globalInput, globalOutput, directFanIn, indirectFanIn, directFanOut, indirectFanOut*

## *Selecting Machine Learning Classifiers*

La selezione è stata guidata principalmente dalla nostra volontà di effettuare un confronto equo con lo stato dell'arte.

## *Configuration and Training*

### *Data balancing*

- No balancing
- Under-sampling
- Over-sampling

### *Data normalization*

- No normalization
- MinMaxScaler
- Standardization

## *Validation of the Approach*



***RQ1.** Quali metriche relative al Program Dependence Graph sono buoni predittori di difetti?*

**RFECV:** Recursive Feature Elimination Cross Validation.

L'insieme iniziale di features viene utilizzato per il primo addestramento ed ogni feature viene classificata in base al contributo che ha dato nell'addestramento.

Le features peggiori vengono eliminate dall'insieme corrente.

Questa procedura viene ripetuta ricorsivamente fino ad ottenere un numero ottimale di features.

**RQ1.** Quali metriche relative al Program Dependence Graph sono buoni predittori di difetti?

<b><i>Metrica</i></b>	<b><i>Occorrenze</i></b>	<b><i>Rank</i></b>
maxPdgVertices	57	2.21
verticesCount	50	2.92
edgesCount	41	3.88
edgesToVerticesRatio	42	3.95
globalInput	38	4.79
lackOfCohesion	24	13.58
...	...	...
globalOutput	2	310.50

**RQ2.** Qual è il miglior modello di previsione dei difetti basato sulle metriche estratte da un PDG?

Abbiamo sperimentato come le performance variano quando vengono incluse o escluse le operazioni di data balancing e data normalization.

Abbiamo calcolato metriche come *precision*, *recall*, *F-measure*, *MCC*, *AUC-PR*

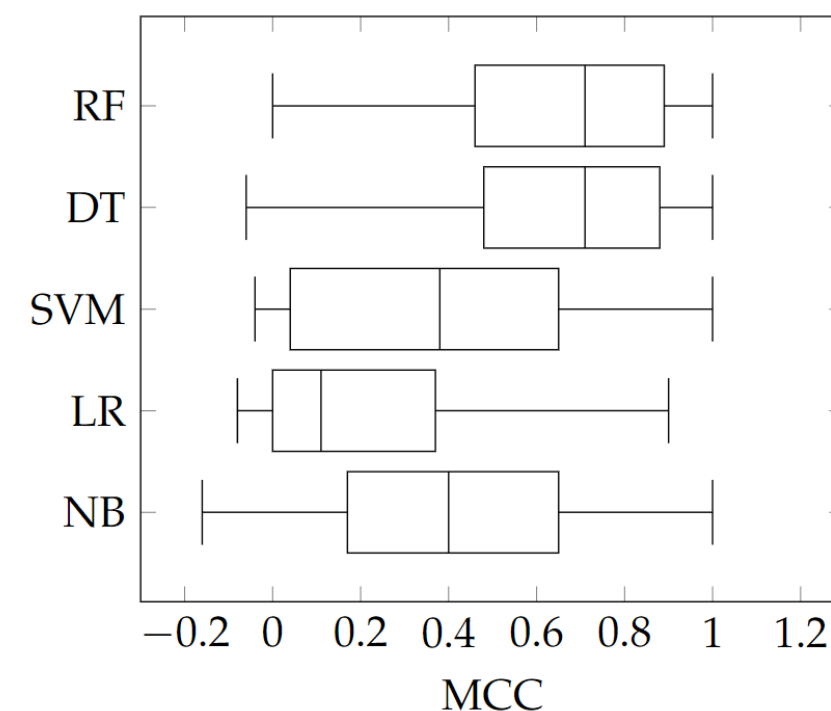
Abbiamo effettuato il test statistico di *Wilcoxon* applicando la correzione di *Bonferroni* e il test statistico di *Friedman* applicando il test post-hoc di *Nemenyi*

Abbiamo calcolato anche il coefficiente di *Cohen* per misurare le dimensioni dell'effetto tra le coppie di classificatori.

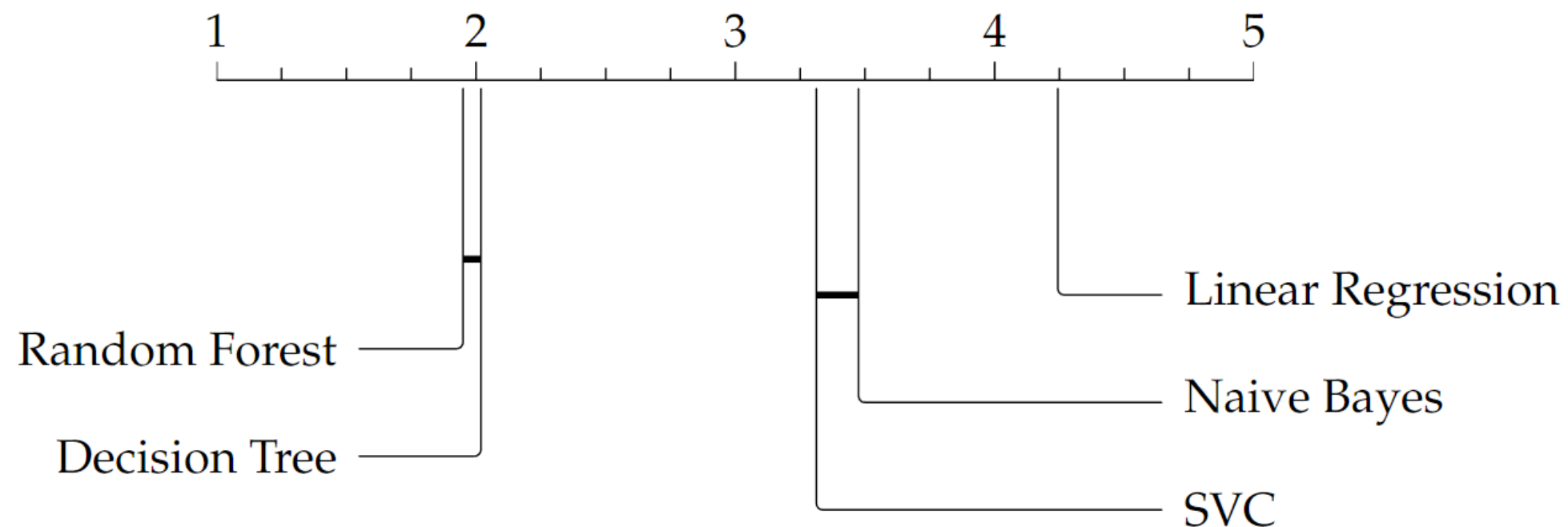
**RQ2.** Qual è il miglior modello di previsione dei difetti basato sulle metriche estratte da un PDG?

*Random Forest e Decision Tree* riportano un valore medio di MCC di 0.64 e 0.63 rispettivamente.

<b>Classificatore</b>	<b>Occorrenze</b>
Decision Tree	47
Random Forest	46
Naive Bayes	22
Linear Regression	15
Support Vector Machine	11



**RQ2.** Qual è il miglior modello di previsione dei difetti basato sulle metriche estratte da un PDG?



**Random Forest e Decision Tree** sono i migliori modelli basati sulle metriche estratte da un PDG.



**RQ3.** *In che misura un modello basato su metriche PDG è complementare ai modelli dello stato dell'arte?*

Dati i due modelli di previsione,  $m_i$  e  $m_j$  abbiamo calcolato:

- $m_i \cap m_j$ , ovvero il numero di bug correttamente previsti sia da  $m_i$  che da  $m_j$
- $m_i/m_j$  e  $m_j/m_i$ , ovvero il numero di bug correttamente previsti da  $m_i$  e mancati da  $m_j$  e viceversa.
- Il numero di bug mancati sia da  $m_i$  che da  $m_j$ .

**RQ3.** *In che misura un modello basato su metriche PDG è complementare ai modelli dello stato dell'arte?*

	$A \cap B$	$A \setminus B$	$B \setminus A$	$A \Delta B$
<b>PDG – Delta</b>	60,50%	23,80%	4,55%	5,15%
<b>PDG – Process</b>	69,16%	21,13%	4,75%	4,95%
<b>PDG – ICO</b>	88,70%	1,60%	6,92%	2,79%

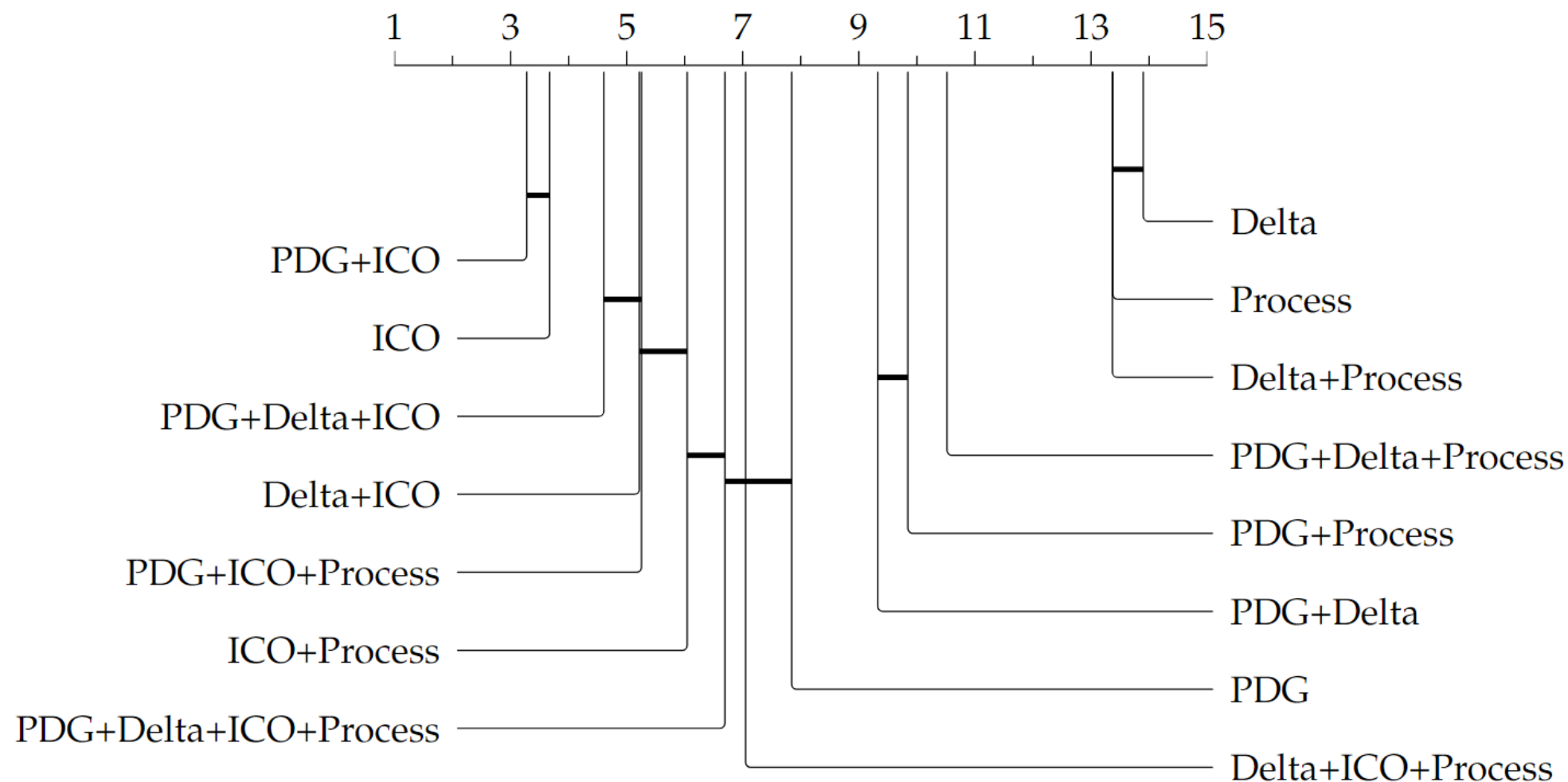
***RQ4.** Una combinazione di metriche basate su PDG, strutturali e di processo migliora le prestazioni?*



Abbiamo combinato 4 gruppi di metriche tra di loro.

Abbiamo generato 15 combinazioni di metriche e utilizzato la RFE per ottenere il sottoinsieme migliore da ogni combinazione.

Infine, le prestazioni sono state calcolate allo stesso modo della RQ2.

**RQ4.** *Una combinazione di metriche basate su PDG, strutturali e di processo migliora le prestazioni?*




Corso di Laurea Magistrale in Informatica  
Curriculum Software Engineering and IT Management


# Infrastructure-as-Code Defect Prediction using PDG metrics

Prof. Dario Di Nucci  
Dott. ssa Valeria Pontillo

Gerardo Iuliano  
Mat.: 0522501329

✉ g.iuliano29@studenti.unisa.it  
🌐 Gerardoluliano.github.io  
📧 @Gerardoluliano





## Introduzione e Background

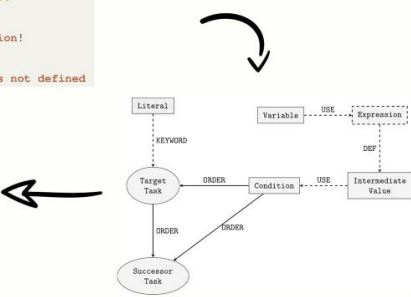
```

1 name: Gather Distribution Info
2 ansible.builtin.setup:
3   gather_subset: distribution!
4 when:
5   - ansible_distribution is not defined


```


Metriche

- VerticesCount: 7
- EdgesCount: 7
- GlobalInput: 2



✉ g.iuliano29@studenti.unisa.it  
🌐 Gerardoluliano.github.io  
📧 @Gerardoluliano





## Metodologia

### Configuration and Training


**Data balancing**

- No balancing
- Under-sampling
- Over-sampling

**Data normalization**


- No normalization
- Min max
- Standardization

### Validation of the Approach



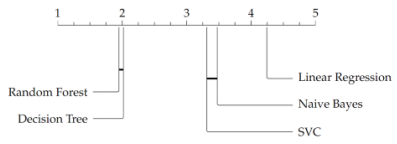
✉ g.iuliano29@studenti.unisa.it  
🌐 Gerardoluliano.github.io  
📧 @Gerardoluliano

laC Defect Prediction using PDG metrics  
Gerardo Iuliano  
Università degli Studi di Salerno



## Risultati

**RQ2. Qual è il miglior modello di previsione dei difetti basato sulle metriche estratte da un PDG?**



Il diagramma delle distanze di Nemenyi mostra una differenza non statisticamente significativa tra Random Forest e Decision Tree.

Pertanto, entrambi i classificatori sono considerati come i migliori modelli di previsioni dei difetti basati su PDG.

✉ g.iuliano29@studenti.unisa.it  
🌐 Gerardoluliano.github.io  
📧 @Gerardoluliano

laC Defect Prediction using PDG metrics  
Gerardo Iuliano  
Università degli Studi di Salerno

# Infrastructure-as-Code Defect Prediction using PDG metrics

Grazie!



Questa tesi ha contribuito a  
piantare un albero in Kenya



Gerardo Iuliano

g.iuliano29@studenti.unisa.it ✉  
Gerardoluliano.github.io 🌐  
@Gerardoluliano 📧