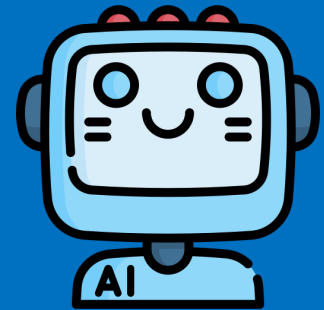


Corso di Laurea in Informatica

Identificazione a Grana Fine di Vulnerabilità Software in Codice Python

Prof. Fabio Palomba
Dott. Emanuele Iannone

Raffaele Aviello
Matricola: 0512110529

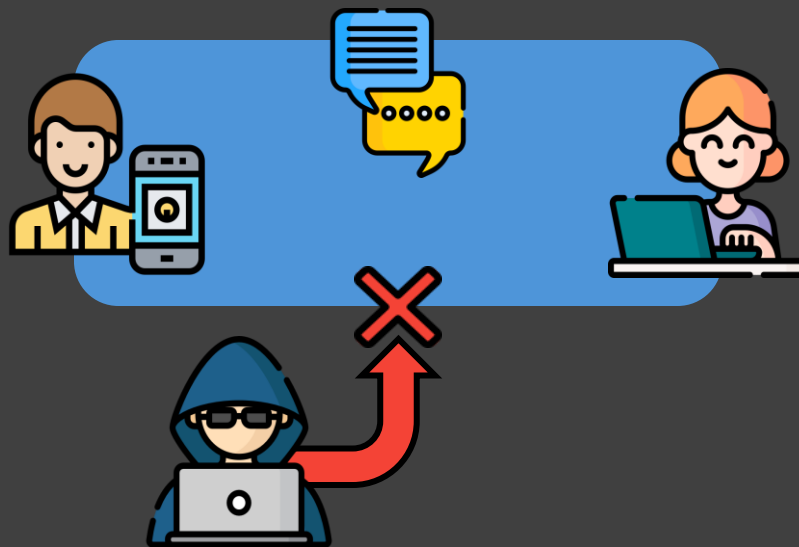


✉ r.aviello@studenti.unisa.it

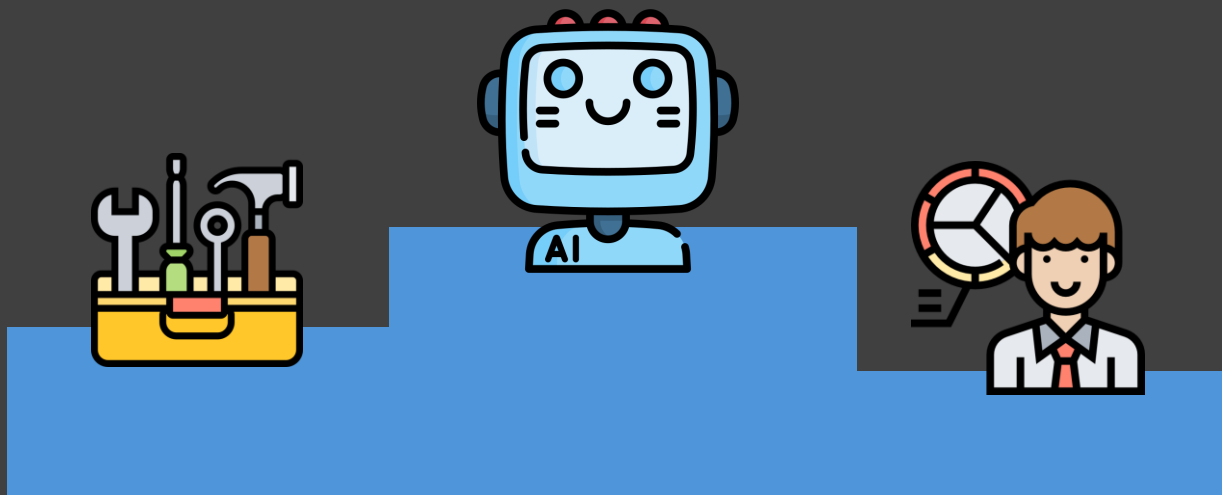
 www.linkedin.com/in/raffaele-aviello



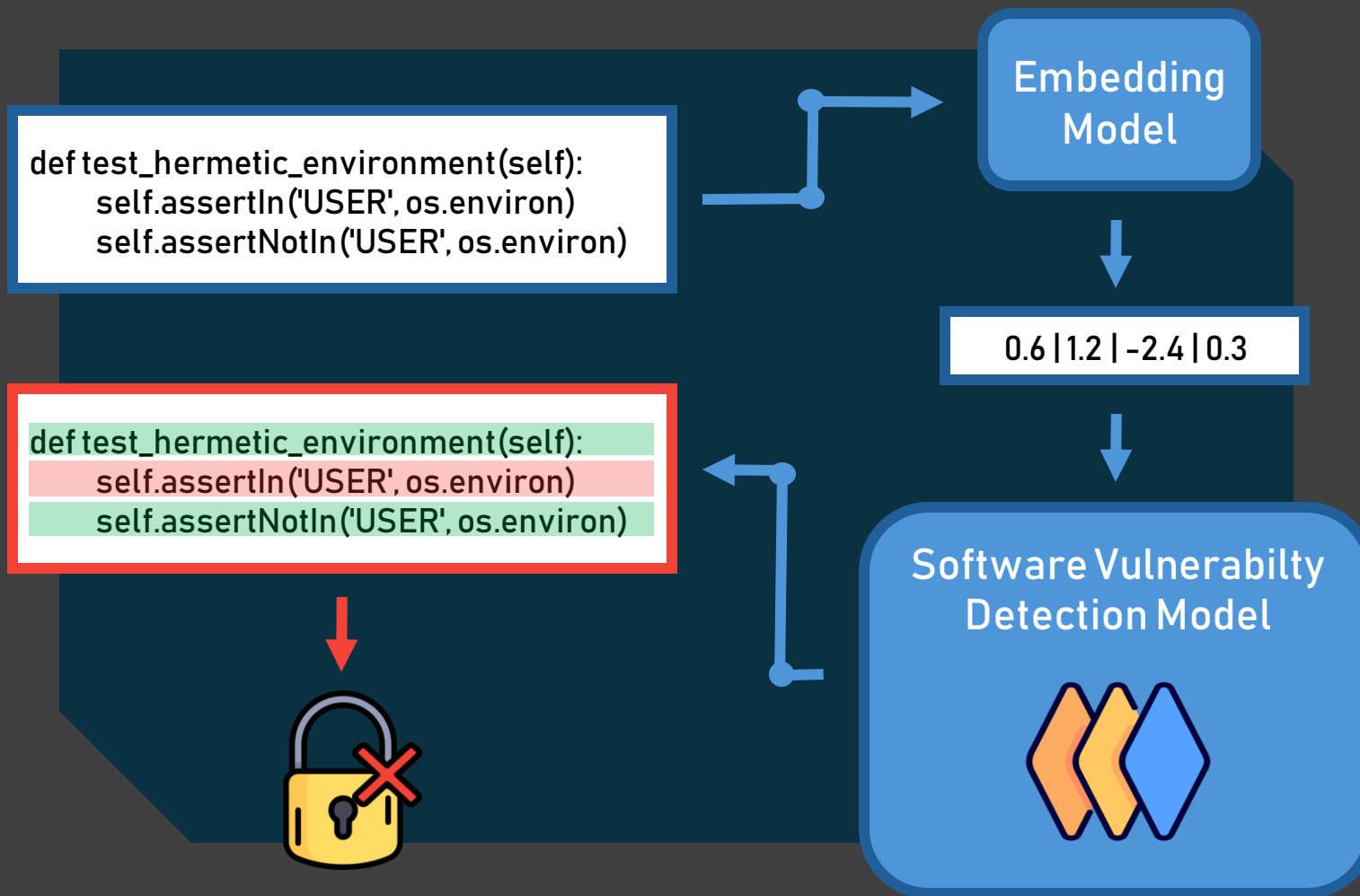
Il **software** è ovunque e viene usato
quotidianamente, per questo gli **utenti** vanno
tutelati da eventuali **attacchi**



Tra i diversi approcci, spiccano le tecniche a
grana fine che sfruttano **modelli** di
machine learning

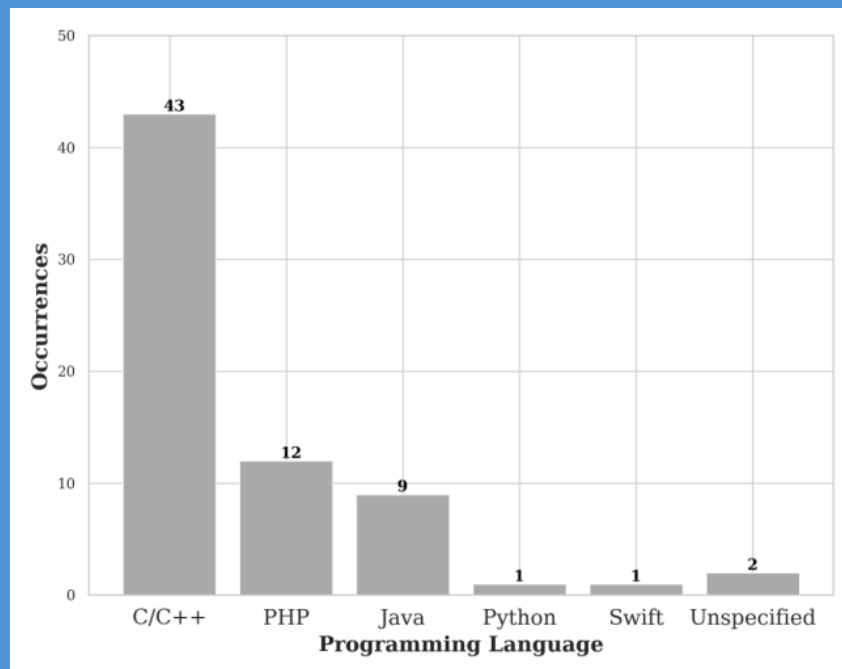


Introduzione e Background



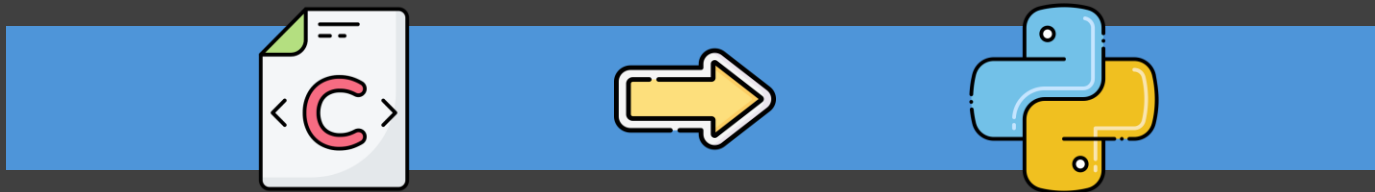
Introduzione e Background

La **maggior parte**
degli approcci opera
su codice **C**, sono
pochi quelli che
operano su codice
Python

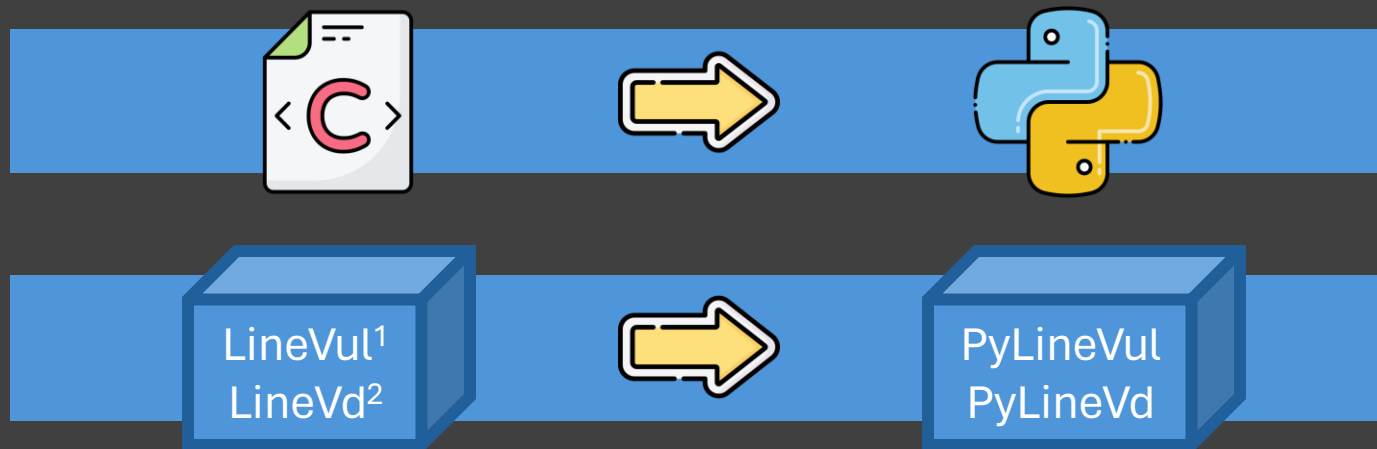


R. Croft, Y. Xie, and M. A. Babar, "Data preparation for software vulnerability prediction: A systematic literature review," arXiv:2109.05740v2, 2022.

L'obiettivo dello studio è **convertire** gli approcci
che operano su **codice C**, in tecniche che operano su
codice Python

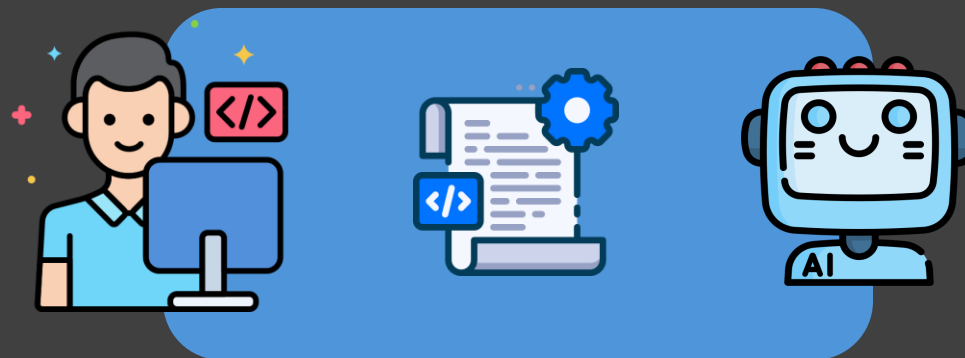


L'obiettivo dello studio è **convertire** gli approcci
che operano su **codice C**, in tecniche che operano su
codice Python

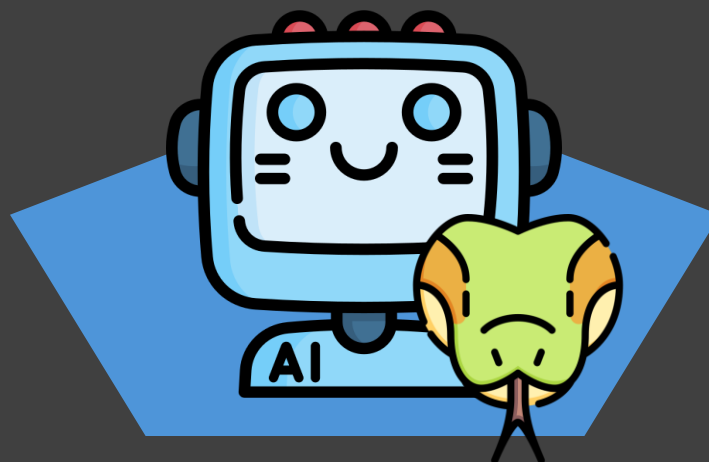


1) M. Fu and C. Tantithamthavorn, "Linevul: A transformer-based line-level vulnerability prediction", 2022 IEEE/ACM 19th International Conference on Mining Software Repositories, 2022
2) D. Hin, A. Kan, H. Chen, and M. A. Babar, "Linevd: Statement-level vulnerability detection using graph neural networks", 2022 Mining Software Repositories Conference, 2022

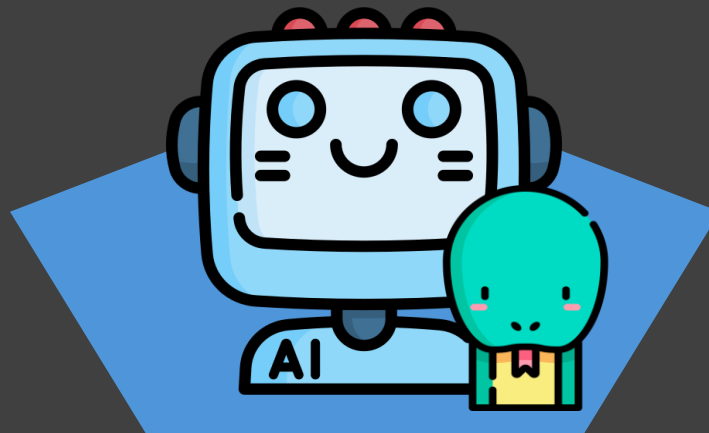
Per raggiungere l'**obiettivo**, sono stati
replicati e modificati gli **script** relativi
ai framework **LineVul** e **LineVd**



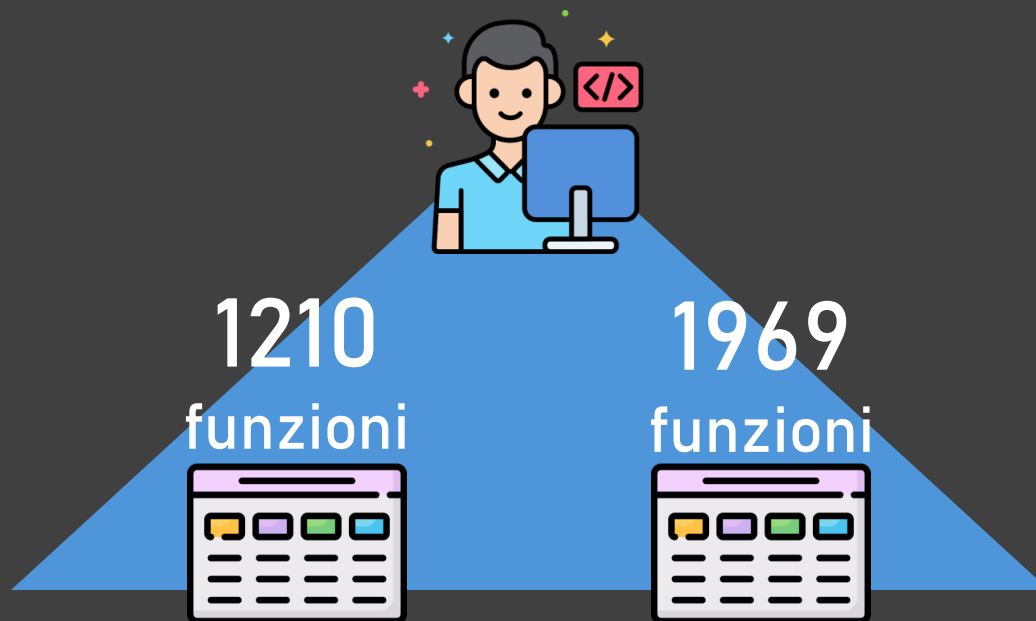
Per il framework **LineVul** è stato
allenato un **tokenizer Roberta** in grado di operare su
codice Python

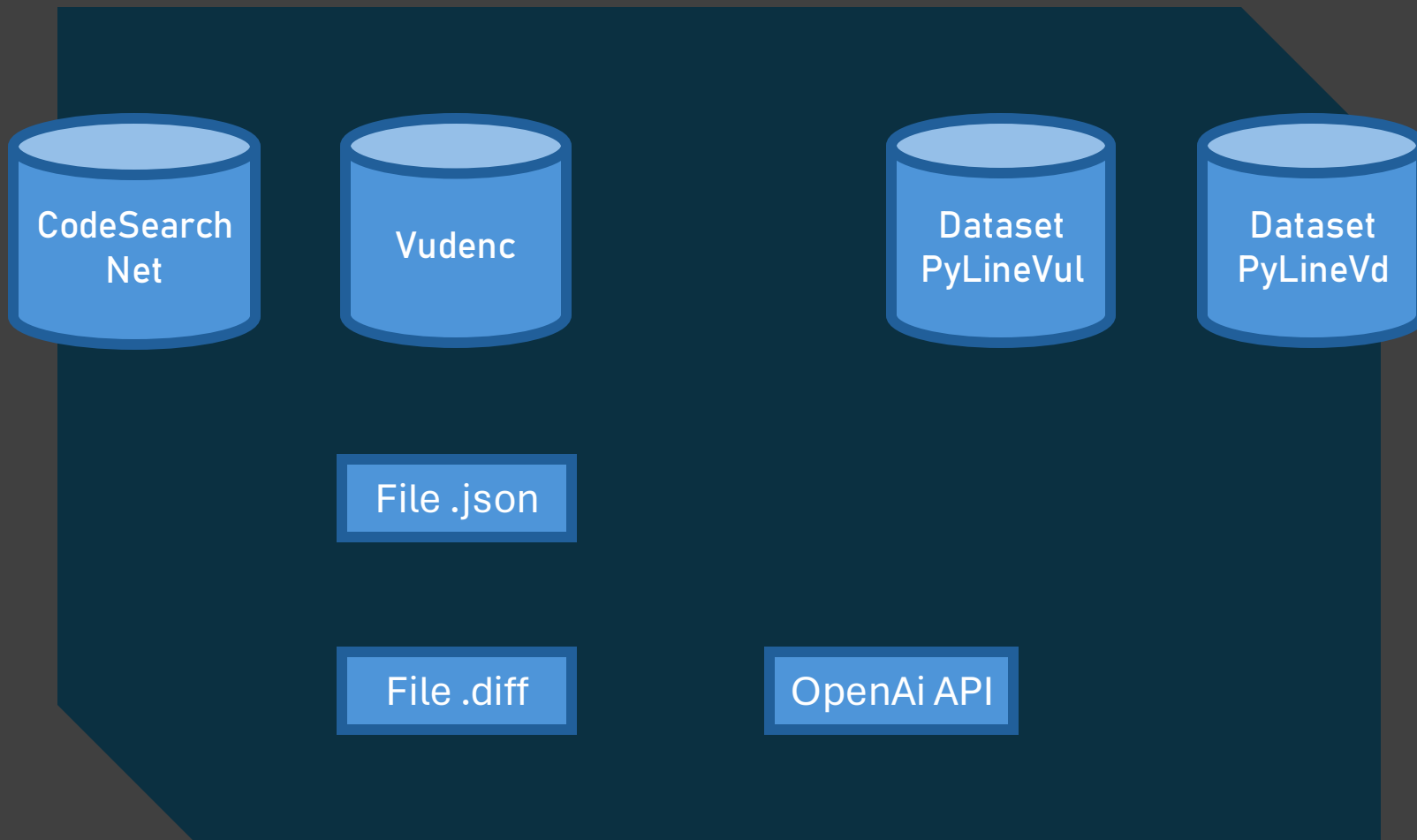


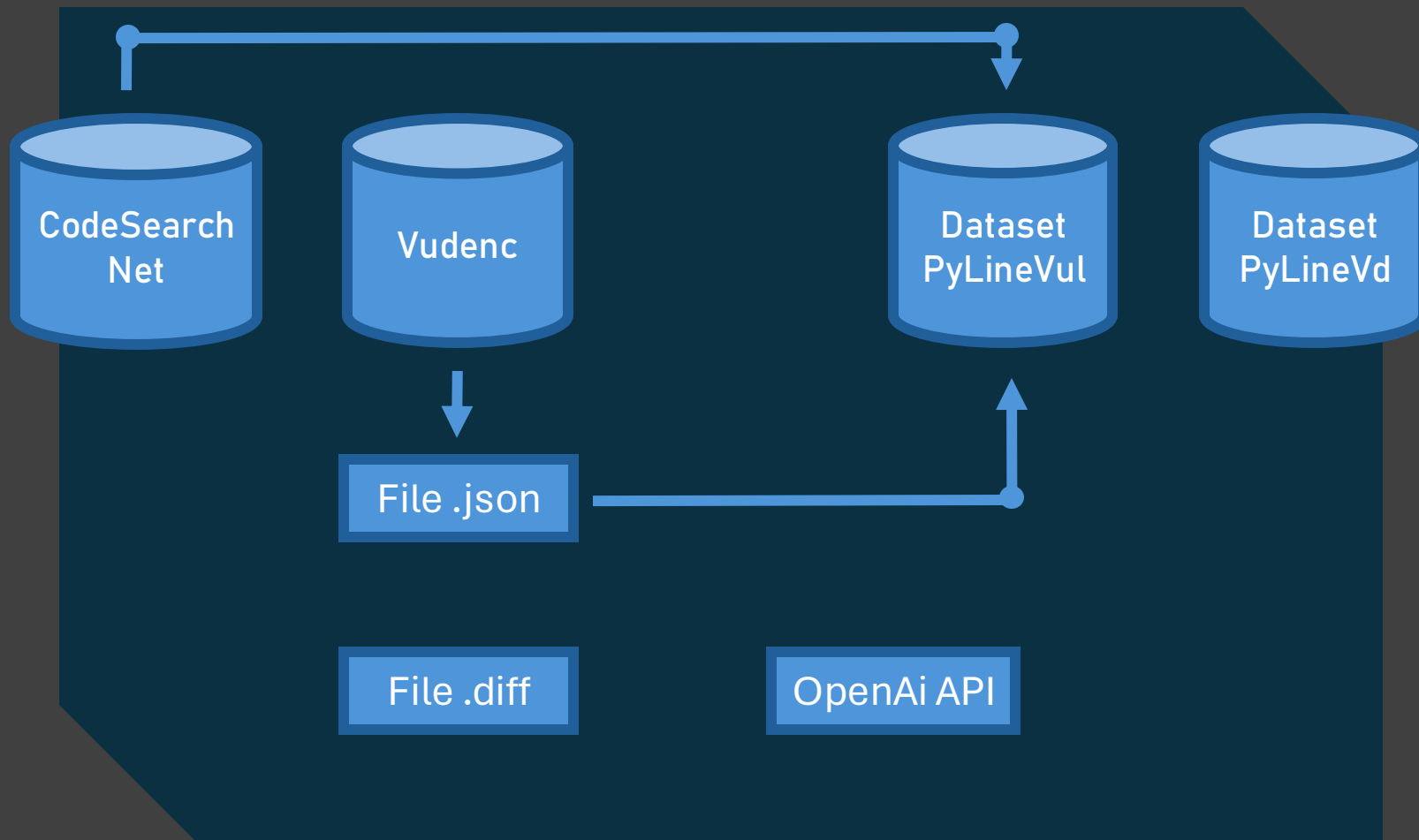
Per il framework **LineVd** sono stati modificati gli **script**
per il caricamento del **dataset** e per la
creazione dei **Code Property Graphs**

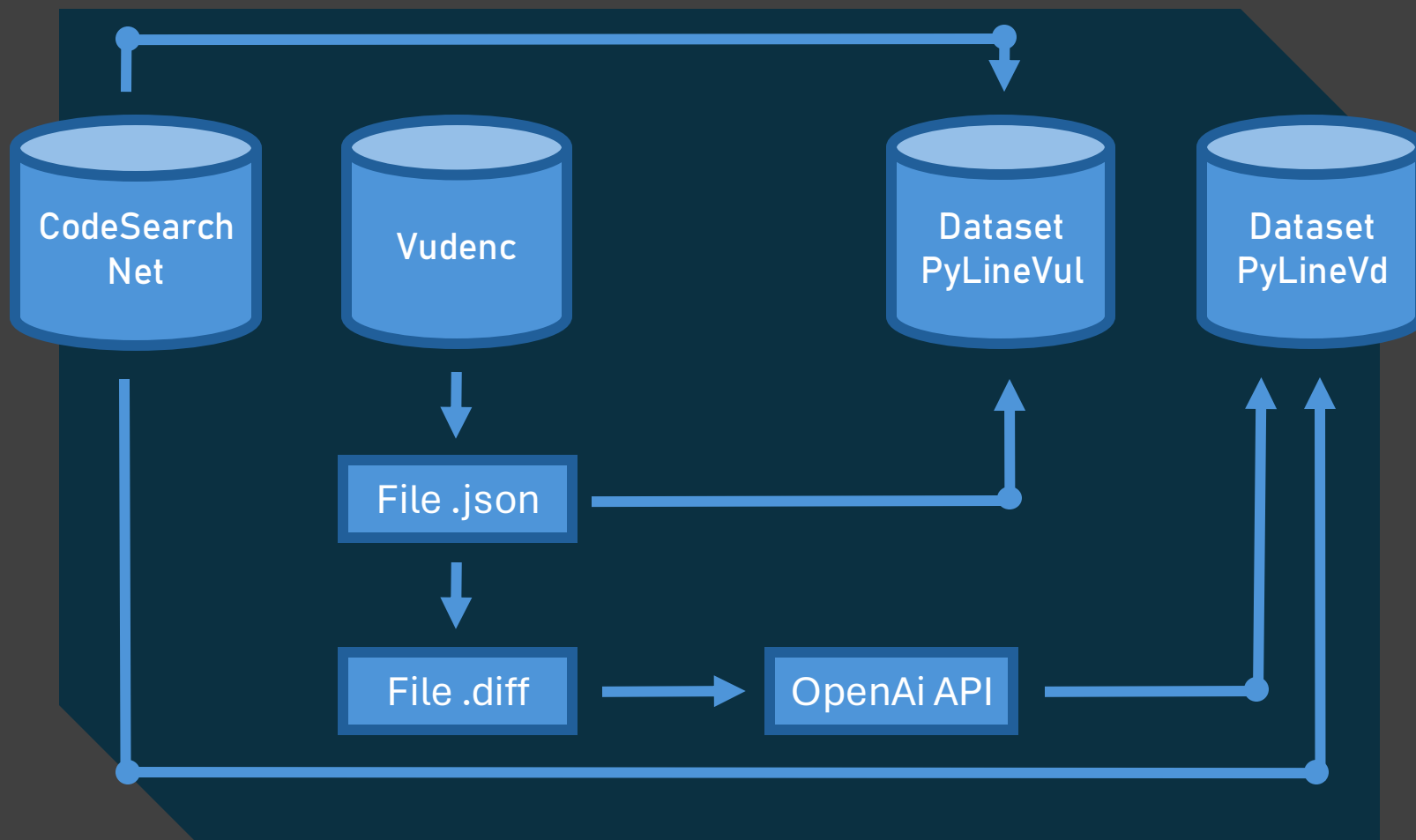


In seguito sono stati costruiti **due dataset** per
allenare i modelli **PyLineVul** e **PyLineVd**

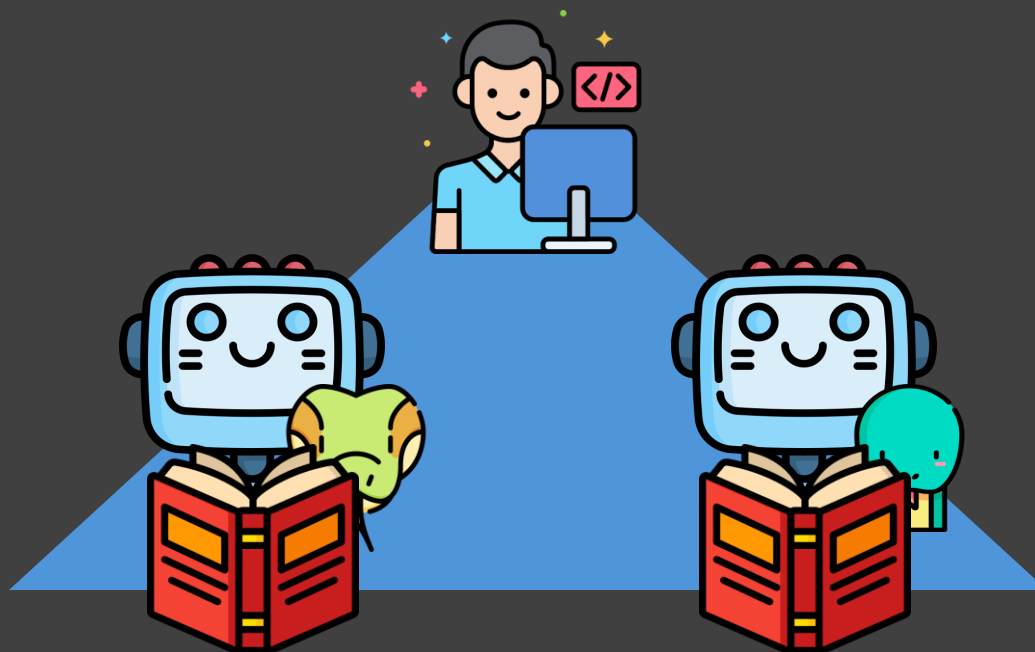








Dopo aver costruito i dataset e averli sottoposti a
split e balancing, sono stati
addestrati i modelli **PyLineVul** e **PyLineVd** ...



... che hanno ottenuto questi **risultati**

| Modello | Precision | Recall | F1 Score | Acc@5 | Acc@10 |
|-----------|-----------|--------|----------|-------|--------|
| PyLineVul | 0.27 | 0.91 | 0.42 | 0.20 | 0.30 |
| PyLineVd | 0.10 | 0.90 | 0.19 | 1.00 | 1.00 |

E dalla valutazione dei risultati si sono ottenuti
due **findings**



I modelli **PyLineVul** e **PyLineVd**, rispetto ai modelli originali, operano **peggio** a livello di **funzione** ...

| Modello | Precision | Recall | F1 Score | Acc@5 | Acc@10 |
|-----------|-----------|--------|----------|-------|--------|
| LineVul | 0.97 | 0.86 | 0.91 | | 0.65 |
| PyLineVul | 0.27 | 0.91 | 0.42 | 0.20 | 0.30 |
| LineVd | 0.88 | 0.92 | 0.89 | 0.90 | |
| PyLineVd | 0.10 | 0.90 | 0.19 | 1.00 | 1.00 |



... ma il modello **PyLineVd** supera il modello originale nella classificazione a livello di **riga**

| Modello | Precision | Recall | F1 Score | Acc@5 | Acc@10 |
|-----------|-----------|--------|----------|-------|--------|
| LineVul | 0.97 | 0.86 | 0.91 | | 0.65 |
| PyLineVul | 0.27 | 0.91 | 0.42 | 0.20 | 0.30 |
| LineVd | 0.88 | 0.92 | 0.89 | 0.90 | |
| PyLineVd | 0.10 | 0.90 | 0.19 | 1.00 | 1.00 |



I modelli **PyLineVul** e **PyLineVd** mantengono
una **proporzionalità**
nelle prestazioni, sia nella classificazione
a livello di funzione ...

| Modello | Precision | Recall | F1 Score | Acc@5 | Acc@10 |
|-----------|-----------|--------|----------|-------|--------|
| LineVul | 0.97 | 0.86 | 0.91 | | 0.65 |
| PyLineVul | 0.27 | 0.91 | 0.42 | 0.20 | 0.30 |
| LineVd | 0.88 | 0.92 | 0.89 | 0.90 | |
| PyLineVd | 0.10 | 0.90 | 0.19 | 1.00 | 1.00 |



... che nella classificazione
a livello di riga

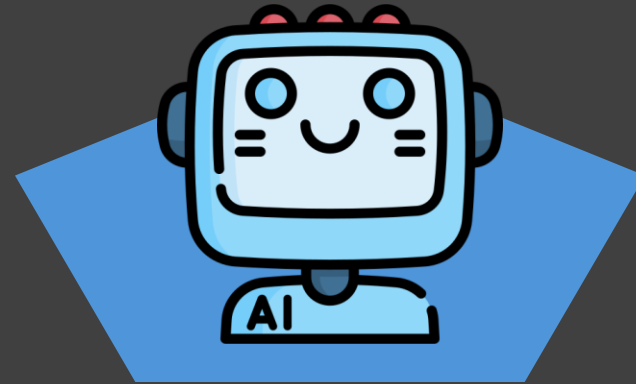
| Modello | Precision | Recall | F1 Score | Acc@5 | Acc@10 |
|-----------|-----------|--------|----------|-------|--------|
| LineVul | 0.97 | 0.86 | 0.91 | | 0.65 |
| PyLineVul | 0.27 | 0.91 | 0.42 | 0.20 | 0.30 |
| LineVd | 0.88 | 0.92 | 0.89 | 0.90 | |
| PyLineVd | 0.10 | 0.90 | 0.19 | 1.00 | 1.00 |



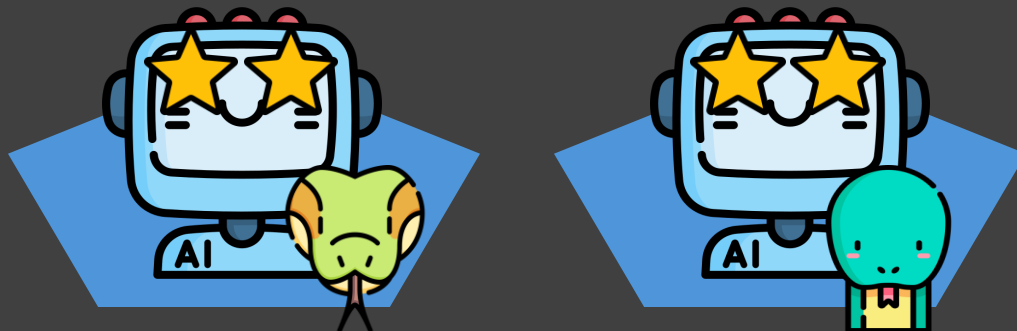
Addestrare i modelli
su un **dataset**
creato **ad hoc**



Addestrare i modelli
su un **dataset**
creato **ad hoc**



Creare un **modello** in
grado di **operare**
direttamente su
codice **Python**



Grazie per
l'attenzione!



Questa tesi ha contribuito a
piantare un albero in Kenya



Identificazione a
grana fine di
vulnerabilità
software
in codice
Python

Raffaele Aviello

r.aviello@studenti.unisa.it 

www.linkedin.com/in/raffaele-aviello 