VIRUS IMAGE CLASSIFIER

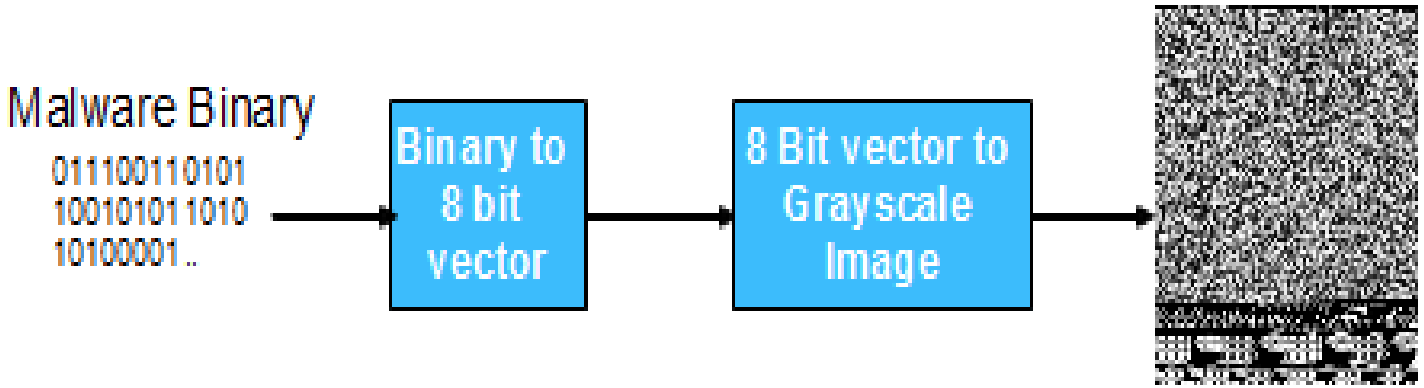
A cura di Fabio Parodi & Davide Caputo

ANALISI DEL PROBLEMA

Normalmente l'analisi di un malware viene eseguita in due modalità diverse: statica e dinamica.

1. Statica, consiste nell'analizzarlo senza avviarlo (offline), studiandone il codice e le funzioni per determinarne il comportamento , ed individuare a quale tipologia appartiene.
2. Dinamica, il malware viene fatto eseguire in una sandbox per studiarne il comportamento dal "vivo".

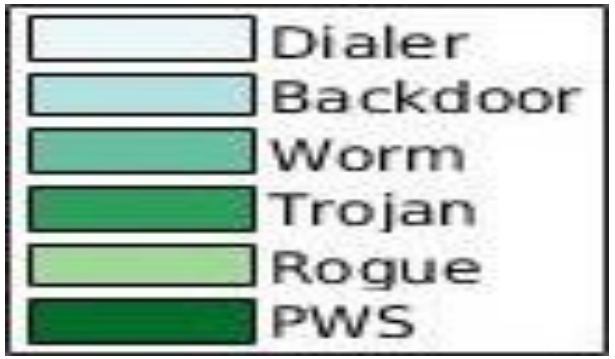
Il metodo proposto, invece di analizzare le singole righe di codice, analizza la sua rappresentazione visiva (immagine).

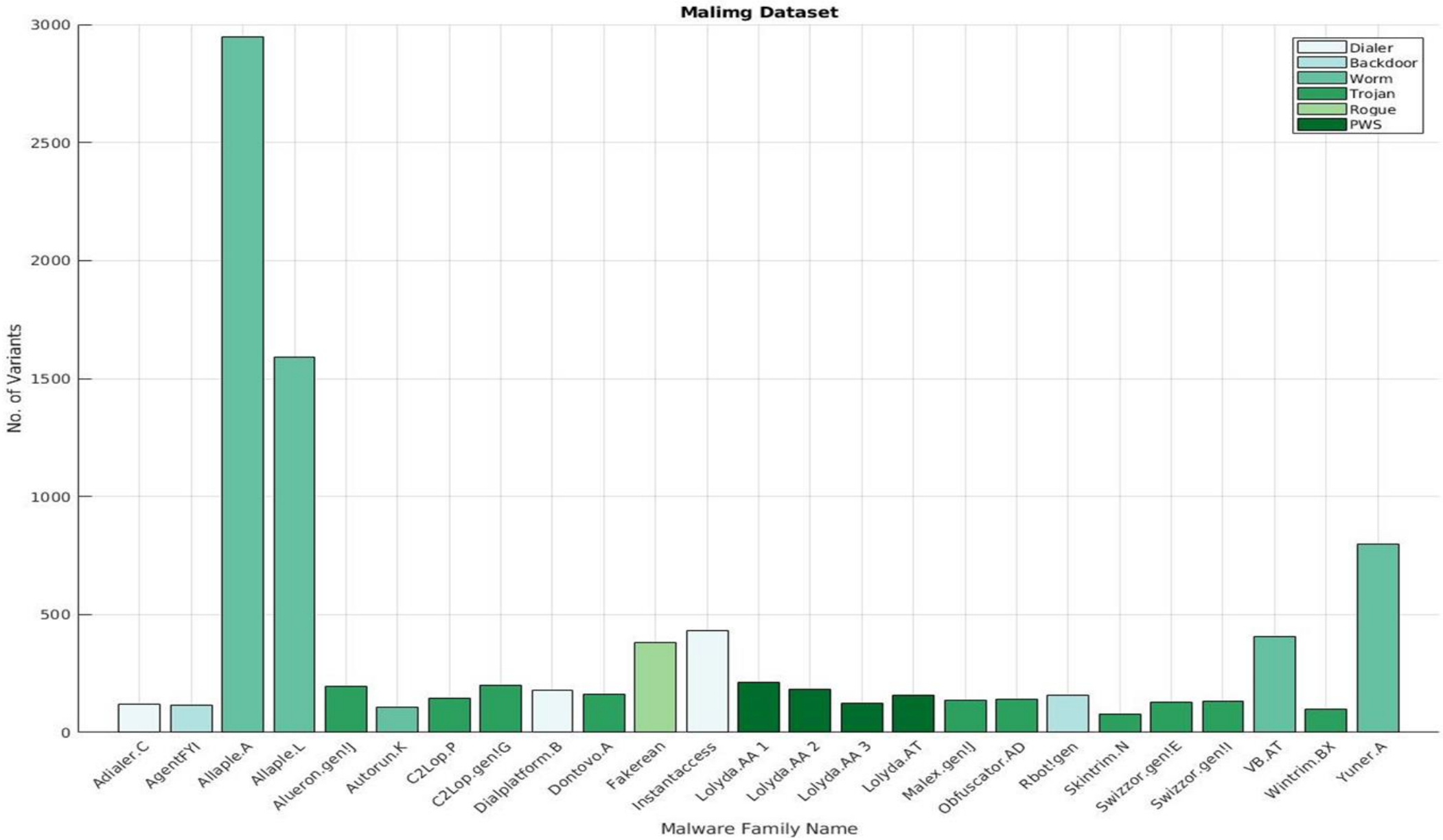


Il codice binario del malware viene trasformato in un'immagine 32x32 in scala di grigi

Data la difficoltà di ottenere codice malware originale e funzionante, si è usato il dataset *Malimg* scaricato dal sito web *Kaggle*, già predisposto per questo tipo di analisi.

Abbiamo quindi analizzato 25 diverse famiglie di virus, appartenenti a 6 diverse tipologie.





Istogramma riportante la composizione del dataset analizzato

RETI NEURALI CONVOLUZIONALI

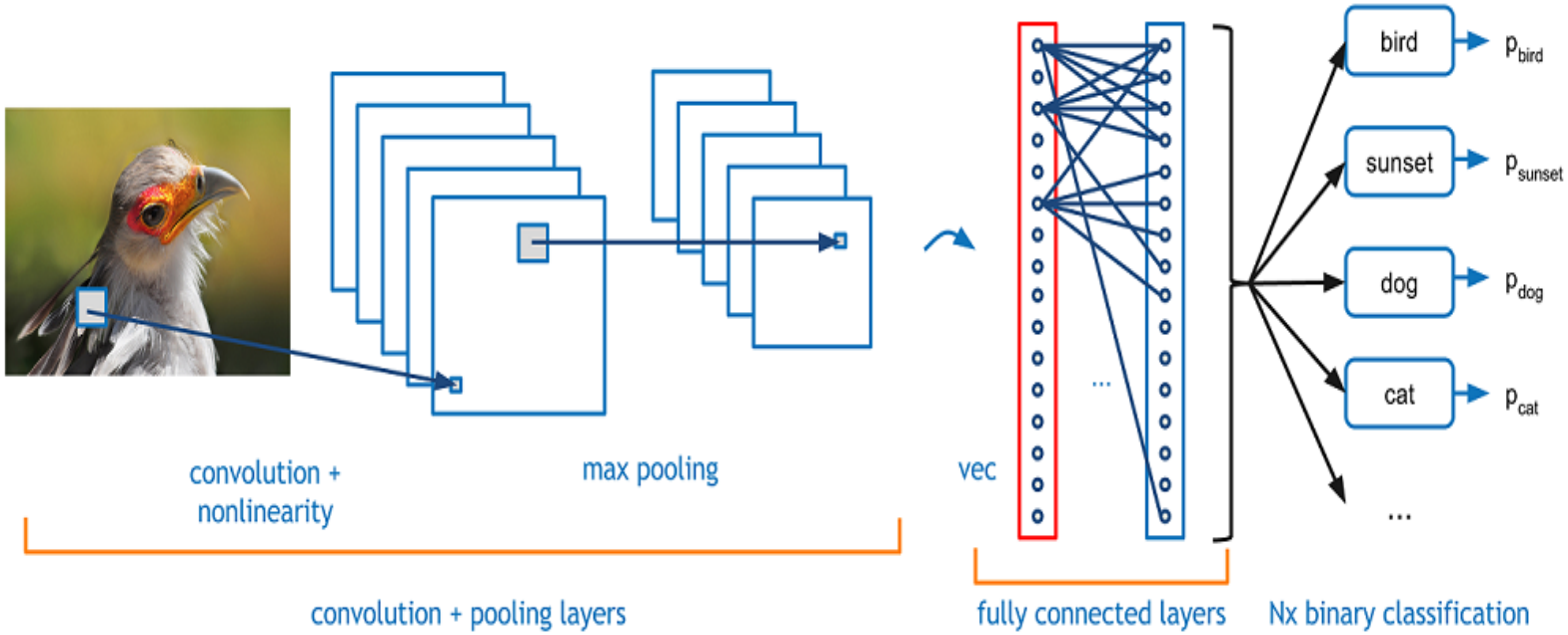
La classificazione di immagini è il processo che prendendo la stessa in input restituisce in uscita una classe di appartenenza (cane, gatto, ecc..) con una eventuale probabilità. Per noi umani, questo processo avviene in modo molto naturale già dai primi anni di vita.

Ad un computer, invece, le immagini appaiano come una matrice riportante i valori dei singoli pixels. A seconda della grandezza e della risoluzione dell'immagine, vedrà ad esempio una matrice di 32x32x3 di numeri interi tra 0 e 255 (il 3 si riferisce ai valori RGB).

What We See

What Computers See

Le reti neurali convoluzionali (CNN) sono quindi il tentativo di replicare il riconoscimento di pattern nell' immagine che avviene nell'uomo nella corteccia visiva, a livello matriciale utilizzando operazioni di *convoluzione* e di *pooling*, e i classici *fully connected layer* e *output layer* già visti nelle reti neurali classiche.



CONVOLUZIONE

Il meccanismo di riconoscimento di pattern nell'immagine viene quindi effettuato dal computer attraverso una serie di convoluzioni con filtri che vengono addestrati per riconoscere caratteristiche specifiche caratteristiche nell'immagine.

Maggiore il risultato dell'operazione di convoluzione, più quella determinata caratteristiche è presente.

Visualization of the filter on the image

0	0	0	0	0	0	0
0	40	0	0	0	0	0
40	0	40	0	0	0	0
40	20	0	0	0	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
25	25	0	50	0	0	0

Pixel representation of receptive field

*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = 0

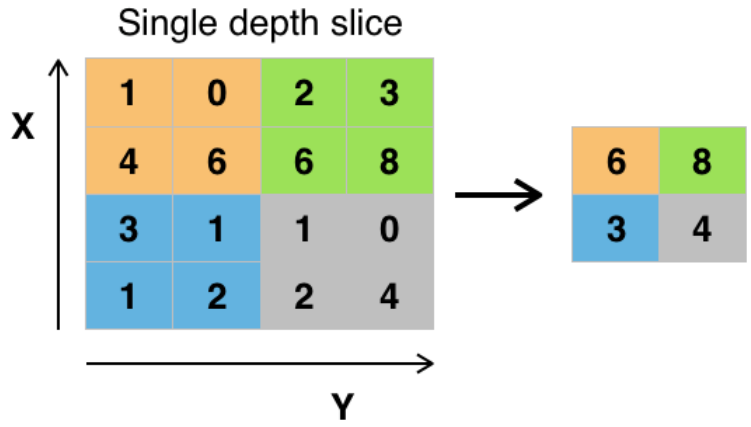
L'operazione viene quindi ripetuta andando a considerare un'altra porzione d'immagine (spostandosi di un valore detto stride) e riportando i risultati in una seconda matrice, che servirà da input per gli strati successivi, dopo aver applicato una funzione detta di attivazione, nel nostro caso la ReLU, che rende il sistema non lineare.

POOLING

Dopo l'operazione di convoluzione spesso segue l'operazione di pooling (anche conosciuta come subsampling).

E' un operazione molto semplice: dalla matrice ottenuta si vanno a considerare le varie regioni da cui è composta (generalmente 2x2) e si prende il valore massimo contenuto, andando a ridurre così la regione di spazio considerato.

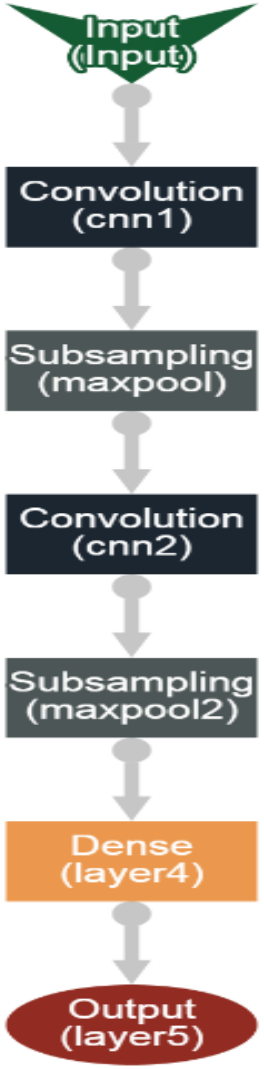
Il ragionamento intuitivo alla base di questo livello è che una volta che sappiamo che una caratteristica specifica è nel volume di input originale (ci sarà un alto valore di attivazione), la sua posizione esatta non è importante quanto la sua posizione relativa rispetto alle altre caratteristiche.



Example of Maxpool with a 2x2 filter and a stride of 2

I vantaggi sono molteplici: oltre a ridurre il numero di parametri o pesi successivamente considerati (riducendo lo sforzo computazionale), si generalizza l'immagine evitando che il modello si specializzi troppo su di quella e prevenendo così l'overfitting.

CNN UTILIZZATA



Il modello che ha fornito i risultati migliori è composto da:

- Immagini in input 32x32x1
- 50 filtri convoluzionali 5x5
- Pooling 2x2
- 100 filtri convoluzionali 5x5
- Pooling 2x2
- Dense layer 1024 percettroni
- Output layer 25 classi
- Tempo di addestramento circa 80 minuti.

PARAMETRI

Learning rate:	0.0094
Optimization algorithm:	stochastic_gradient_descent (momentum 0.9)
Activation Function:	ReLU Softmax (*solo output layer*)
Loss Function:	Negative Logarithmic Likelihood

VALUTAZIONE

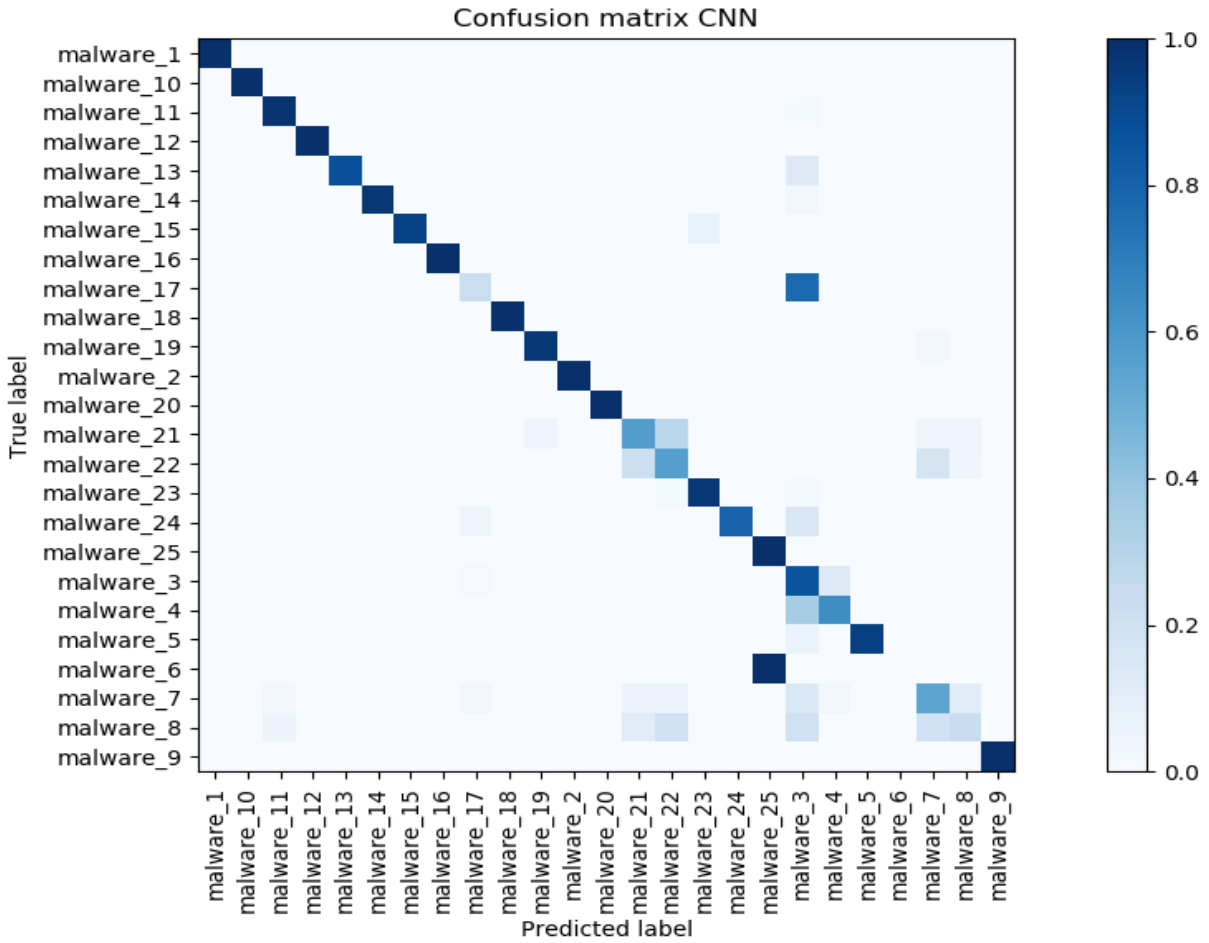
Qui sono riportati i risultati ottenuti dalla rete, il valore di accuracy potrebbe non essere molto indicato per la valutazione della rete avendo un dataset molto sbilanciato (Allaple.A ha una frequenza di occorrenza di quasi 1/3 rispetto a tutto il dataset).

```
==========================Scores=========================================
 # of classes:    25
 Accuracy:        0,8298
 Precision:       0,8638    (1 class excluded from average)
 Recall:          0,8032
 F1 Score:        0,8444    (1 class excluded from average)
Precision, recall & F1: macro-averaged (equally weighted avg. of 25 classes)
=========================================================================
```

MATRICE DI CONFUSIONE

Malware_3 e Malware_4 stessa famiglia due varianti diverse (Worm).

Il malware_6 non viene mai classificato correttamente, viene classificato come malware_3 ma anche qua stessa famiglia (Worm).



LAVORI FUTURI

- Eventuali lavori futuri che si possono sviluppare su questo problema potrebbero riguardare lo sviluppo di una rete con molti più livelli di convoluzione/pooling al suo interno il che potrebbe portare ad evidenziare feature che la nostra rete non è riuscita a trovare.