

# Anomaly Detection in Network Traffic Analysis

An application of Deep Neural Networks for intrusion detection systems by the analysis of the network traffic flow

Master Degree in DS  
Data Mining Project



DataSafe Team:  
Christian Riccio P37000002  
Giacomo Matrone P37000011

Date: 2020-07-29

“Instead of trying to produce a program to simulate the adult mind, why not rather try to produce one which simulates the child’s? If this were then subjected to an appropriate course of education one would obtain the adult brain.”

Alan Turing in Computing Machinery and Intelligence



# Speakers



Christian Riccio

Attending Master Degree in Data Science  
Bachelor in Physics



Giacomo Matrone

Attending Master Degree in Data Science  
Laurea Magistralis in Economics and Finances



# Introduzione

Viviamo in una società immersa nella nuova dimensione culturale pervasa dall' «iperconnettività» e da quello che filosofi come Floridi definiscono «On Life», ovvero una vera e propria vita parallela online.

Per il 2025 si stimano 463 exabytes di informazioni quotidiane. Tutte queste informazioni corrispondono a dati prodotti dalle più svariate sorgenti di informazioni.

Questo ammasso di dati impatta sulle politiche di cybersecurity. Infatti, abbiamo che vi sono molteplici fonti di dati in una rete aziendale quali, ad esempio:

- Computers (aziendali o personali a seconda delle policy BYOD)
- Smartphones (aziendali o personali a seconda delle policy BYOD)
- Sensori (IPS, IDS, etc.)
- Social media (inerente le policy su cosa condividere o meno)

Queste criticità rendono necessaria un'adeguata protezione dei dati, sempre più personali e sempre più inerenti l'intera sfera esistenziale individuale.



# Cosa si intende per Anomaly Detection?

«An outlier (or anomaly), is an observation which deviates so much from the other observations as to arouse suspicion that it was generated by different mechanism. 1»

- Dato un insieme di punti, è cruciale identificare quei punti che si discostano dal normale comportamento assunto da tutti gli altri
- Ci riferiamo quindi al problema di identificare patterns nei dati che non confermano il comportamento che ci si aspetta
- Le anomalie sono casi speciali di outliers che trasportano informazioni importanti che possono essere utili per l'analisi

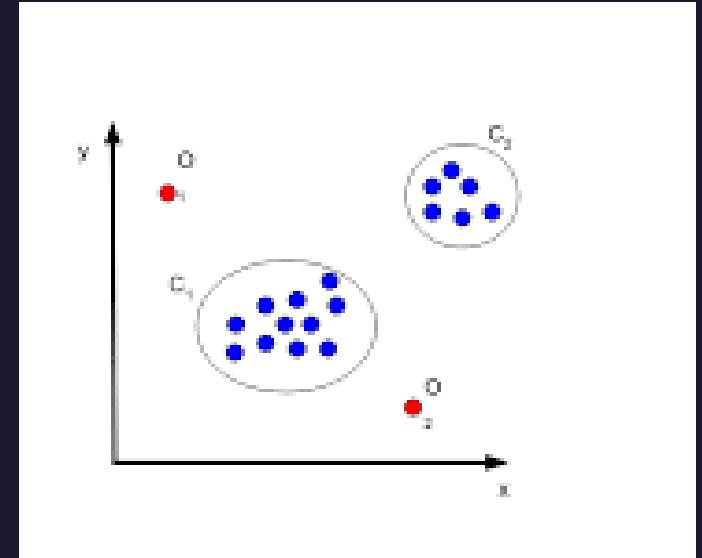


Fig.1. Rappresentazione di punti categorizzabili come anomali.

# Applicazioni dell' Anomaly Detection

- **Fraud Detection**

- Credit Card fraud
- Insurance claim fraud detection
- Insider trading detection



- **E-commerce**

- Manomissione prezzi
- Problemi di network

- **Network Intrusion Detection**

- Rilevamento attività malevole in sistemi informatici



- **Ambito medico**

- **Manifattura e sensoristica**



# Pericoli per l'industria 4,0 in Italia

Dal rapporto del Clusit 2 del 2020 si evince:

- 2019 anno peggiore di sempre in termini di evoluzione delle cyber minacce
- Gli attaccanti non sono più i semplici «hackers»
  - Vi sono decine di gruppi criminali organizzati e multinazionali fuori controllo che fatturano miliardi grazie al cyber crime
- Maggiore efficacia degli attacchi dovuta all'evoluzione degli attori e delle modalità che essi utilizzano
  - Malware
  - Phishing/Social engineering

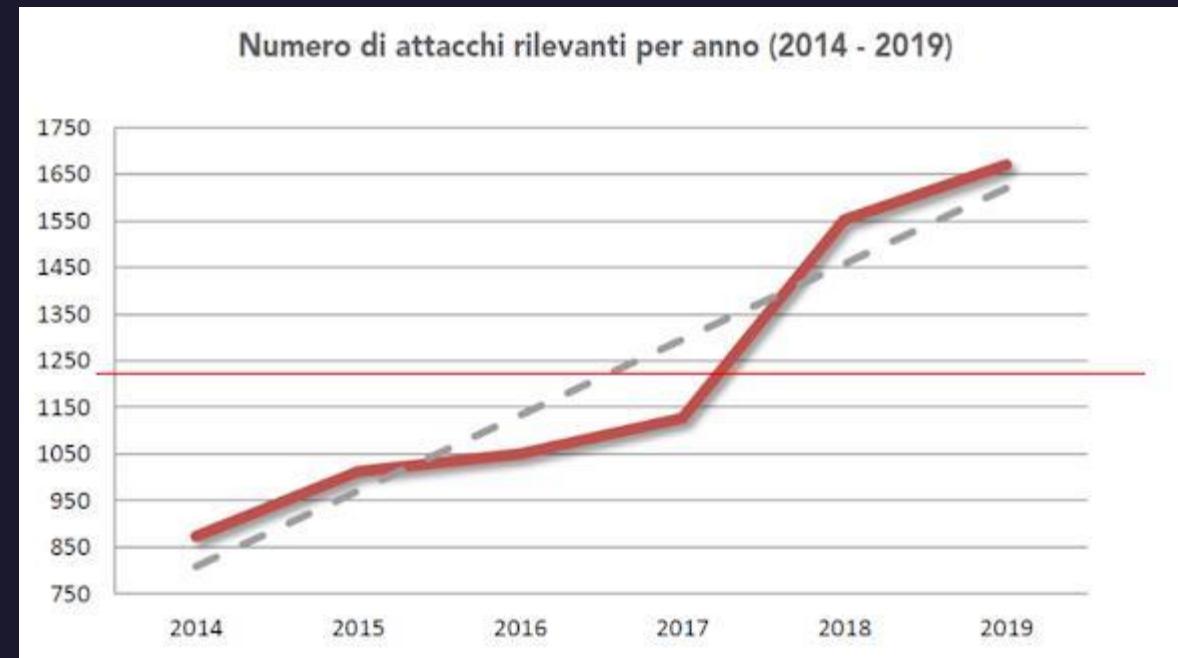


Fig.2. Dal rapporto sulla sicurezza informatica del Clusit si evince un incremento importante degli attacchi informatici nell'annualità 2014-2019.



# Cybersecurity - Overview

La cybersecurity si presenta come un gioco competitivo non cooperativo, dove la capacità di agire in modo tempestivo e veloce al verificarsi di un attacco è fondamentale.

Attualmente, l'approccio più seguito è quello signature based. Tuttavia, il diffondersi di un compute power a basso costo permette sempre di più, ad attaccanti e difensori, di ricorrere a tecniche di Machine Learning per rendere i propri attacchi o le proprie difese più sofisticate.



# Cybersecurity - Overview

In cybersecurity gli attori si identificano Red Team e Blue Team. I primi risultano essere gli attaccanti, mentre i secondi sono i difensori.

Ogni attacco inizia con una riconoscione, per poi terminare con la cancellazione di ogni traccia dell'attività malevola.

Ci sono moltissimi attacchi che spesso sono anche contemporanei.

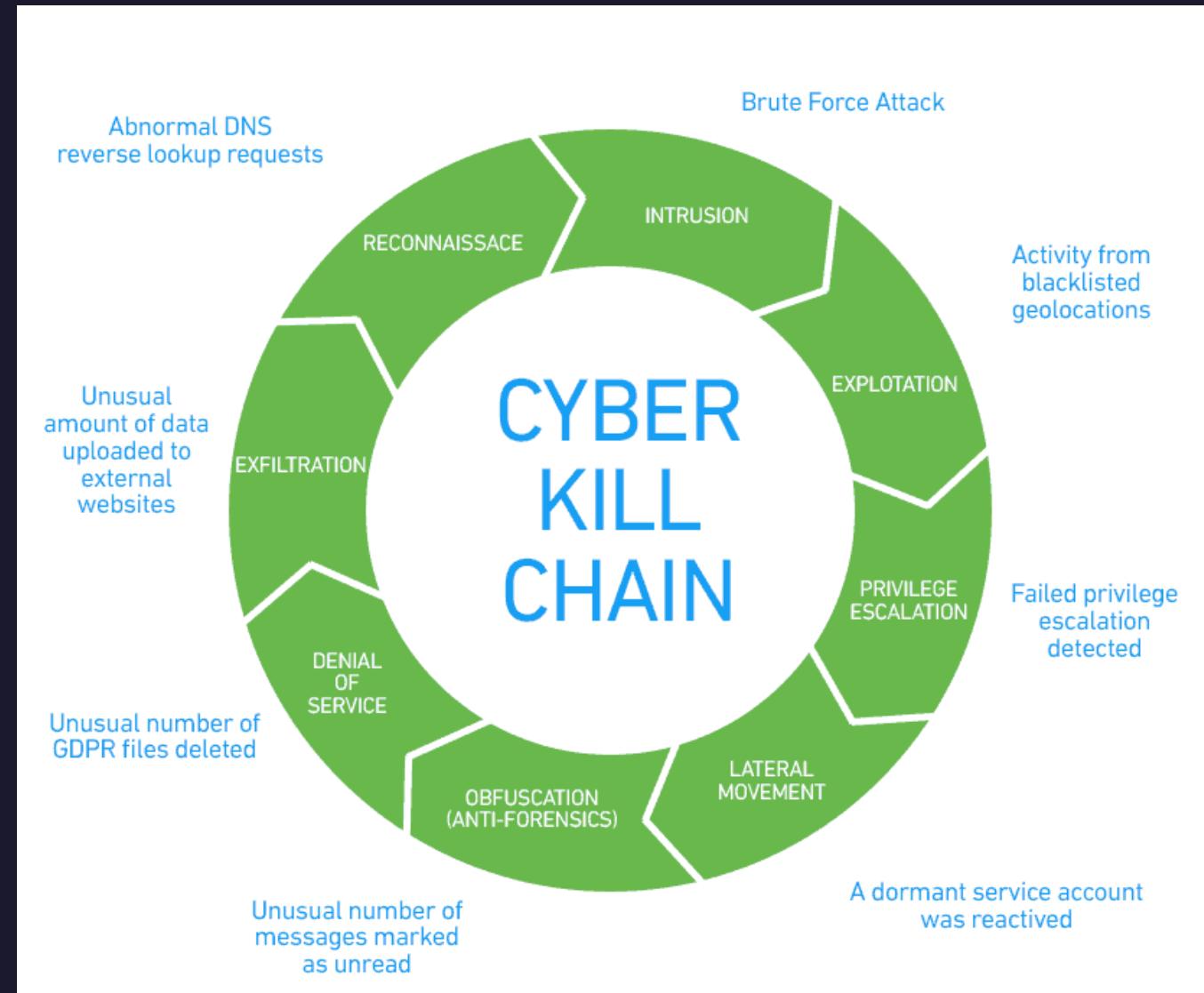


Fig.3. Ciclo delle fasi di un attacco informatico.

# Cybersecurity – COVID19

Con l'avanzare della pandemia, c'è stato un utilizzo sempre maggiore della rete internet a seguito della remotizzazione del lavoro. Conseguentemente, anche gli attacchi sono aumentati, sia tramite Social Engineering che tramite malware.

In particolare, c'è stata un'esplosione di ransomware, com'è possibile vedere dall'immagine.

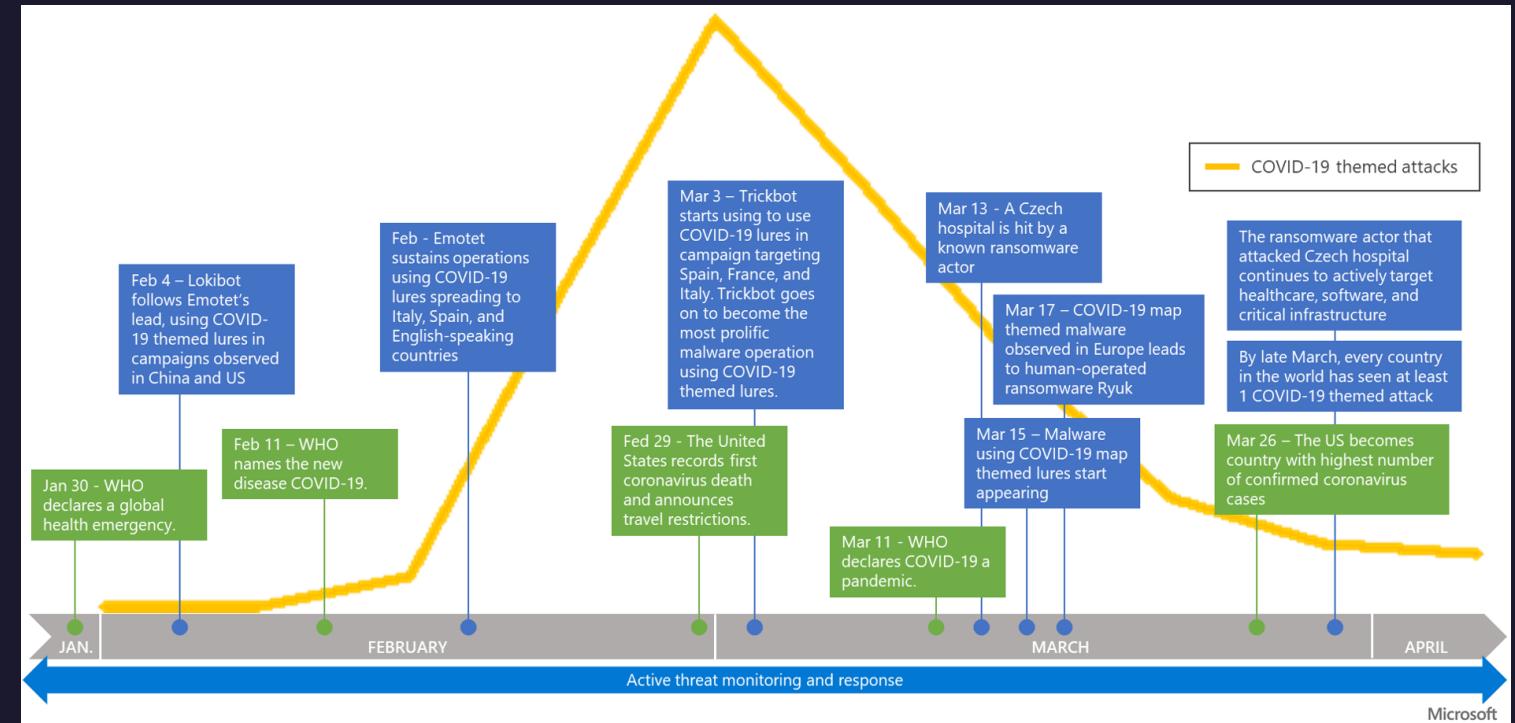


Fig.4. Aumento di attacchi informatici nei primi mesi di coronavirus

# Cybersecurity – Approccio per Signature

L'approccio ad oggi usato maggiormente è un approccio per signature, ovvero una «firma digitale» dell'attacco.

Le signature riescono a filtrare solo gli attacchi già noti, per cui spesso si ricorre ad un secondo elemento che è l'approccio per livelli fissi.

Questi approcci si dimostrano essere problematici, però, in quanto:

- Si generano molti falsi positivi
- I nuovi attacchi difficilmente vengono bloccati

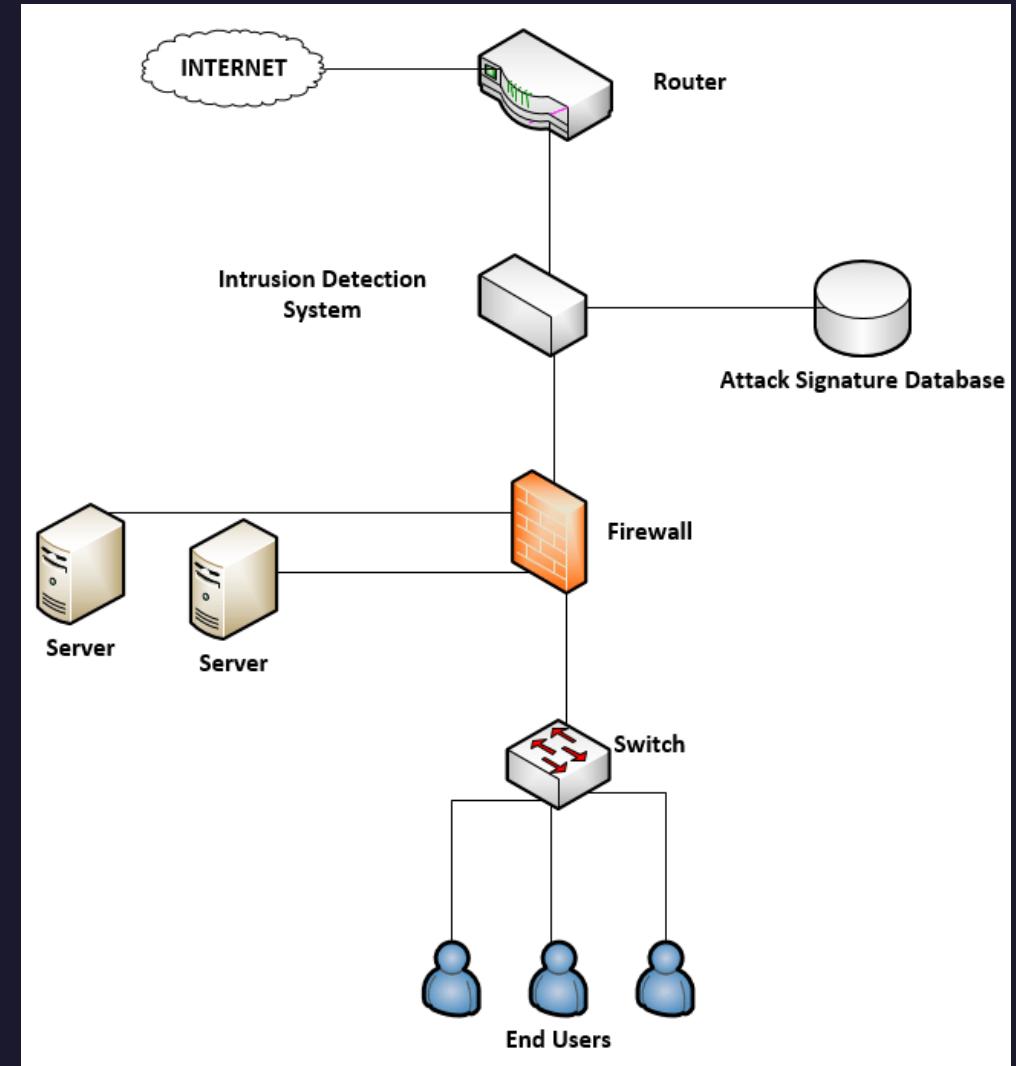


Fig.5. Schema architetturale di una rete aziendale basata su IDS con approccio per signatures.

# Cybersecurity – Intelligent NIDS Overview

L'approccio per signatures prevede la presenza di un DB che contenga al suo interno patterns malevoli già noti ed identificati nel traffico di rete in modo da poter successivamente identificare un'azione malevola.

In un'internet che cambia velocemente, è difficile avere un riscontro veloce e quindi identificare nuove signatures caratterizzanti un traffico di rete malevolo.



Un nuovo approccio si trova nell'adozione di *Intelligent NIDS* che provano ad identificare comportamenti malevoli a livello di network utilizzando tecniche di Data Mining, categorizzando la tipologia analisi di traffico di rete in:

- **Misuse:** vengono usati patterns già conosciuti con cui incrociare il traffico in entrata, e.g. le signatures
- **Anomaly detection systems:** si basano sul fatto che le attività di network normali e malevoli sono diverse, in particolare parti del traffico che deviano molto dal traffico considerato normale sono identificabili come anomalia

# Cybersecurity – Intelligent NIDS Overview 2

Sono stati sviluppati diversi tools che utilizzano tecniche di Data Mining su flussi di dati real-time.

Lo scopo è quello di identificare tra i patterns di traffico di una rete privata e/o aziendale incidenti importanti categorizzabili come anomalie. In particolare, per una data finestra temporale, vengono raccolti tutti i flussi di dati al suo interno.

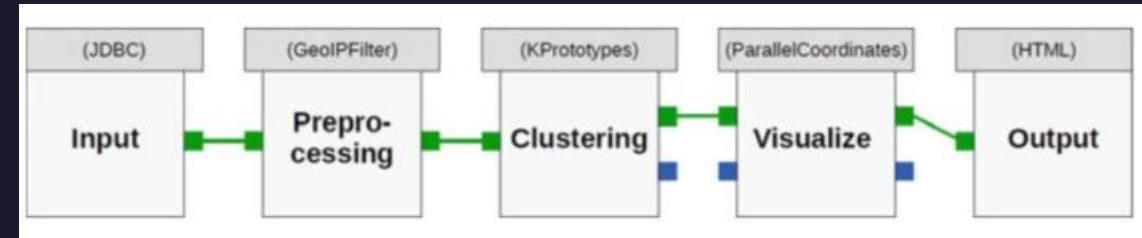


Fig.6. Architettura di un Intelligent NIDS basato su analisi real-time del traffico di rete. La struttura in figura generalmente è anche indicata come *pipes-and-filters*. \*

Nell'immagine troviamo:

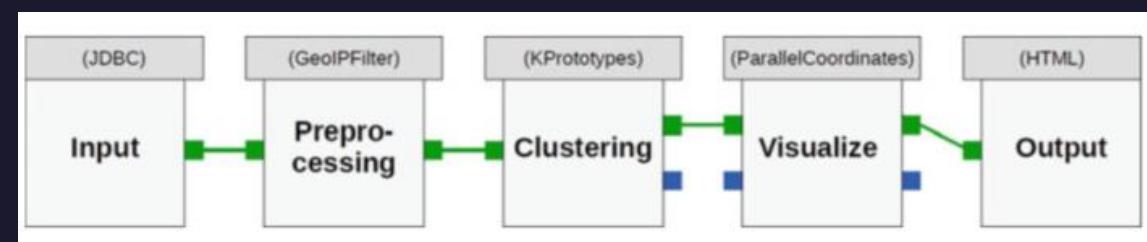
- I blocchi che rappresentano i ‘filtri’ di tutto il processo
- Le pipes, ovvero i collegamenti tra i vari blocchi



# Cybersecurity – Intelligent NIDS

## Overview 3

- I filtri di input e output sono responsabili della scrittura e lettura dei dati
- Il filtro di pre-processing pulisce i dati, trasforma le variabili e aggiunge ulteriori informazioni ai dati
  - I dati raccolti sono molto eterogenei, filtri appositi vengono utilizzati per renderli uniformi e poi digeribili dagli steps successivi
- Nel filtro di clustering si dividono i data points in clusters aventi patterns simili
  - A valle, viene poi posto un filtro di classificazione, su cui gira un algoritmo supervisionato (ad esempio Neural Networks, Decision Trees, ecc.) per assegnare ogni punto alla sua classe di appartenenza
- Il filtro visualize dispone di tools che permettono sia di esplorare graficamente i dati sia di valutare le performance degli algoritmi: Accuracy, Precisione, F1-score e etc.



# Network Traffic - La struttura dei Dati di Rete

Il traffico di rete può essere considerato come un flusso multidimensionale di records. Tali flussi di rete contengono meta-informationi circa la connessione tra due entità della rete (hosts).

La struttura base di un tipico flusso di rete è rappresentata dalla seguente tupla:

*(Source IP Address, Source Port Number, Destination IP Address, Destination Port Number, Transport Protocol)*

Ulteriori grandezze vengono altresì considerate, quali ad esempio:

- Durata della connessione
- Quantità di traffico al secondo (*rate/s*) scambiato tra le due entità
- Stato della connessione
- Quantità di byte scambiati tra sorgente e destinazione e viceversa

# DataSafe - La struttura dei Dati di Rete 2

- **Indirizzo IP:** variabile nominale, presentato in notazione dotted-decimals
- **Port Number:** variabile discreta e che assume valori tra 0-65535 che per motivi computazionali (limiti di RAM della macchina su cui abbiamo addestrato), abbiamo dovuto trattare per il momento come continua
- **Protocollo:** variabile nominale. Rappresenta una precisa regola di scambio di informazione tra due entità comunicanti
  - Es. due pc sono messi in comunicazione tra di loro in rete attraverso il protocollo IP
- **Stato della connessione:** variabile di tipo nominale. Rappresenta lo stato della connessione associato al rispettivo protocollo



# DataSafe - La struttura dei Dati di Rete 3



*Come vengono aggiunte ulteriori variabili che concorrono alla formazione del dataset finale?*

Due approcci in Network Anomaly Detection:

- **Packet-based analysis** che rappresenta l'approccio meno usato

- Aspetti concernenti la privacy
- Il contenuto del pacchetto (payload) potrebbe essere criptato
- I pacchetti di dati sono tutti di dimensioni diverse

- **Flow-based analysis**

- I dati possono essere ridotti
- Svincolo da problemi di privacy
- Facilmente ottenibili dalle entità centrali della rete (routers, switch e etc.)



Controllo più facile delle interfacce di rete per dati così strutturati

Vengono utilizzati:

- Tools di network scanning:
  - Argus
  - Bro-IDS
  - Nmap
  - Wireshark
- Algoritmi di parsing dei dati

# DataSafe - La struttura dei Dati di Rete 4

- **IP Address Info Filter:** riceve il flusso di dati dalla rete ed integra conoscenze di domain knowledge
- **Service Detection Filter:** riceve i dati dal filtro precedente ed identifica il servizio di ogni flusso (SSH, DNS, HTTP e etc)
- **Collecting Filter:** entità centrale del processo.
  - Raccoglie separatamente ogni flusso per ogni utente della rete
  - Controlla il window scanning parameter
  - Crea il network data-point associato ad ogni utente
  - Si ottiene la combinazione *user data-point – network data-point*

Dal collecting filter sono ottenute tutte le informazioni costituenti il dataset finale

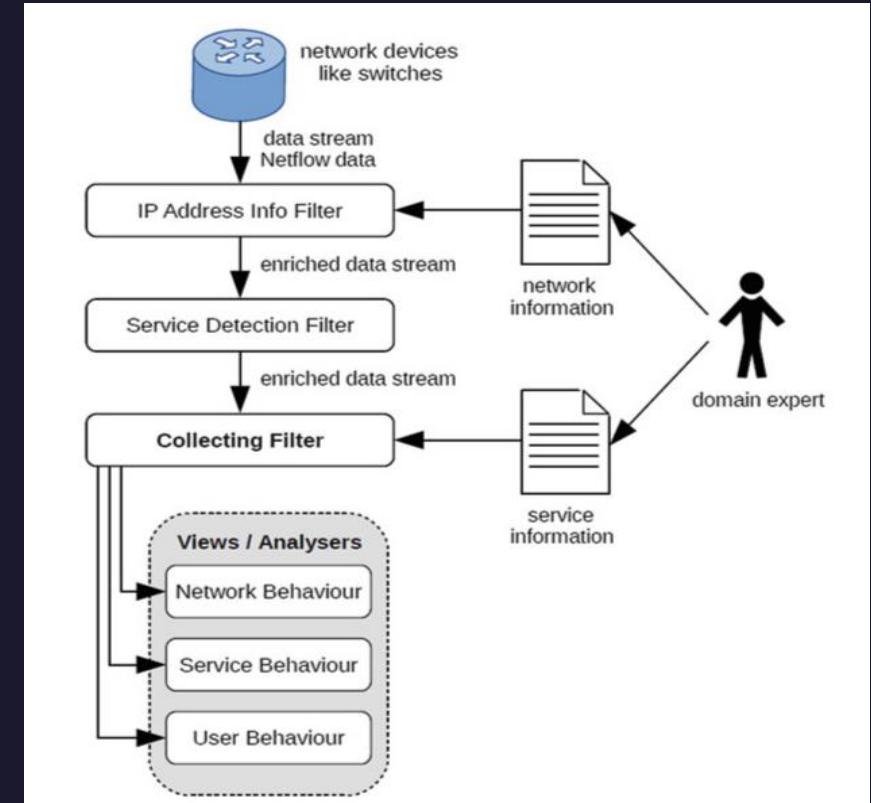


Fig.7. Workflow di processing per la costruzione del dataset, sul quale attuare tecniche di Data Mining e KDD.

Data Safe – Our Approach,  
your security

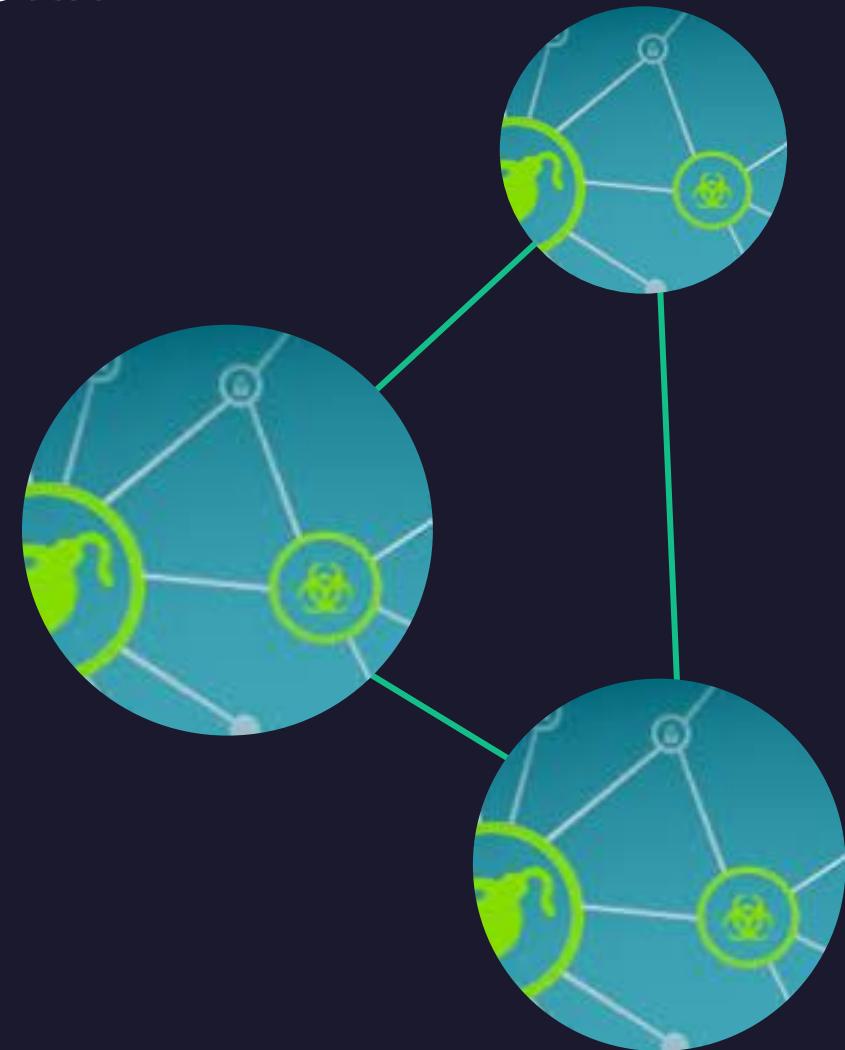
# DataSafe – Research Question

**Obiettivo: Real-Time Analysis su traffico di rete con l'obiettivo di discriminare il traffico normale e malevolo.**

- Questa tecnica può risultare molto utile nel tuning delle regole o nell'aiutare gli addetti al monitoring nella fase di threat hunting.

Proponiamo una rete neurale convolutiva con architettura ResNet50. Essa risulta essere composta da:

- 120 Layers differenti
- 20 milioni di parametri che può apprendere e grazie ai quali impara le features
  - Il training è reso più efficiente grazie all'impiego del **Residual Block**



# DataSafe – Panoramica degli attacchi

- **Backdoor:** codice che si autoinstalla in un computer per permettere l'accesso ad un'altra persona, l'obiettivo è superare le difese di un sistema firewall, accedendo da remoto ad un sistema attraverso falle nel software o hardware compromesso.
- **Fuzzers:** si inviano dati malformati ad un sistema con l'intento dichiarato di mandarlo in crash, rivelando così problematiche di affidabilità.
- **Shellcode:** scritto in linguaggio assembly, esegue una shell sul computer infetto garantendone il possesso all'attaccante.
- **Exploits:** sottoinsieme di malware, contenenti codici eseguibili in grado di sfruttare le vulnerabilità di software presente sul computer. Vengono progettati per colpire versioni specifiche del software che contengono vulnerabilità.
- **DoS (Denial of Service):** in tale attacco si fanno esaurire deliberatamente le risorse di un sistema informatico che fornisce un servizio ai client, ad esempio un sito web su un web server, fino a renderlo non più in grado di erogare il servizio ai client richiedenti.
- **Worms:** categoria di malware in grado di autoreplicarsi. Modifica il computer infettato in modo da essere eseguito ogni volta che si accende la macchina. Sono responsabili di un sovraccarico delle risorse di sistema.

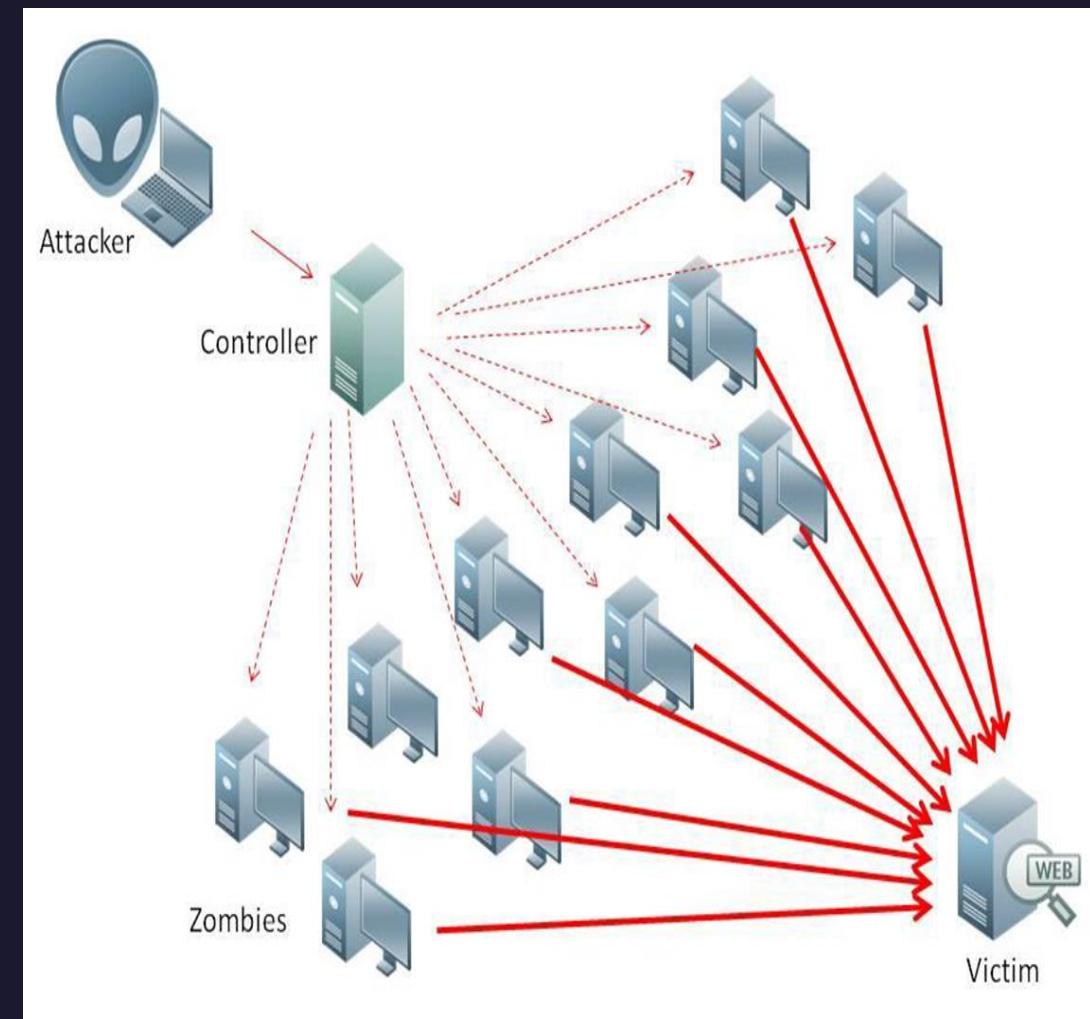


Fig.8. Schema del più comune degli attacchi informatici, il DoS. L'attaccante (Command&Control) assume il controllo di molte istanze che vengono usate per sovraccaricare il sistema vittima.

# DataSafe – Presentazione dei Dati

Il dataset a disposizione è stato creato dall'IXIA PerfectStorm tool del Cyber Range Lab del Centro Australiano per la Cybersecurity.

Esso consiste di:

- 2.058.768 osservazioni
- 49 features
- Circa 300 dopo operazioni di One Hot Encoding



# DataSafe – Presentazione dei Dati 2

- Alcune delle variabili **categoriche**, in quanto non presentanti un **ordine**, ad esempio «proto» sono state trattate con **One Hot Encoding**

- **Le variabili:**

- **Source IP**
- **Destination IP**

non sono state ritenute fondamentali, ai fini dell'analisi, motivo per cui si è deciso di escluderle

- **Le variabili:**

- **Source port**
- **Destination port**

per motivi **computazionali**, sono state considerate come **variabili continue** (ricordiamo il range di estensione 0-65535 di valori unici)



# DataSafe – Presentazione dei Dati 3

Da una valutazione delle classi è possibile notare che c'è un forte problema di Class Imbalance.

Infatti:

- Più del 50% dei dati è inerente a traffico normale.
- Le varie classi di attacco sono presenti in modo disomogeneo, incrementando l'**imbalance issue**.

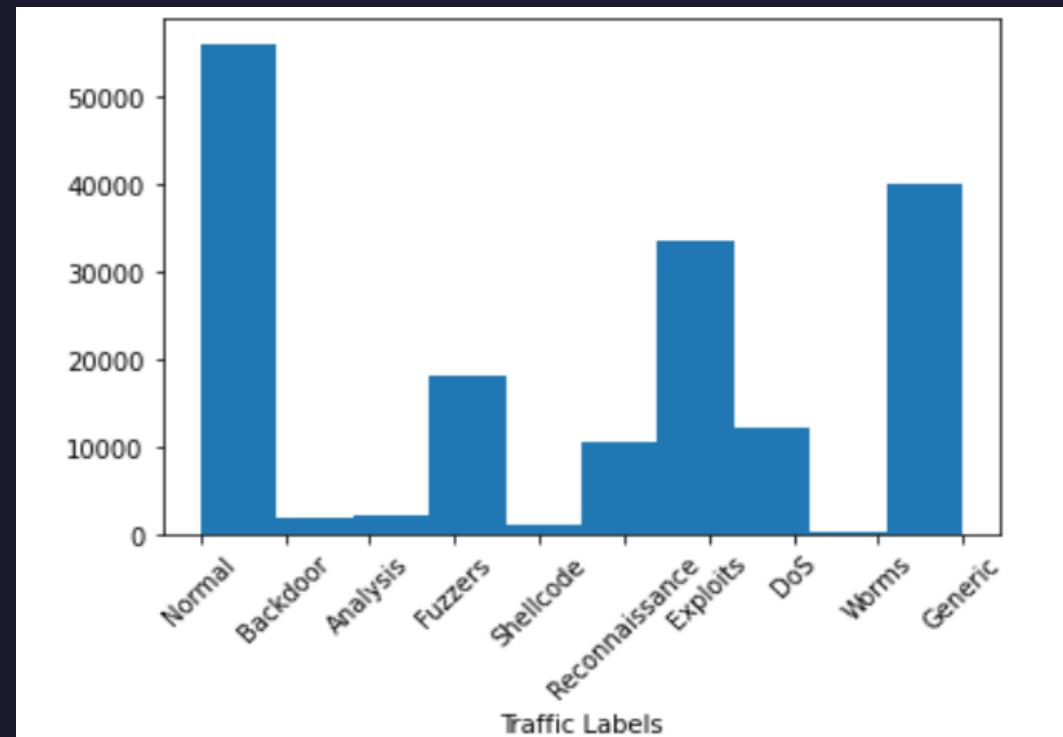


Fig.9. Istogramma raffigurante le labels del traffico di rete (dati di training). 1) *Normal*: indica traffico normale; 2) tutte le altre labels descrivono la tipologia di attacco

# DataSafe – Presentazione dei Dati 4

- La durata di traffico di rete registrato per singolo processo varia in circostanze di traffico normale e traffico associato ad un comportamento malevolo.
- Da notare un valore alto della durata di traffico di rete registrato in attacchi di tipo DoS, in cui come visto si cerca di sovraccaricare il sistema bersaglio

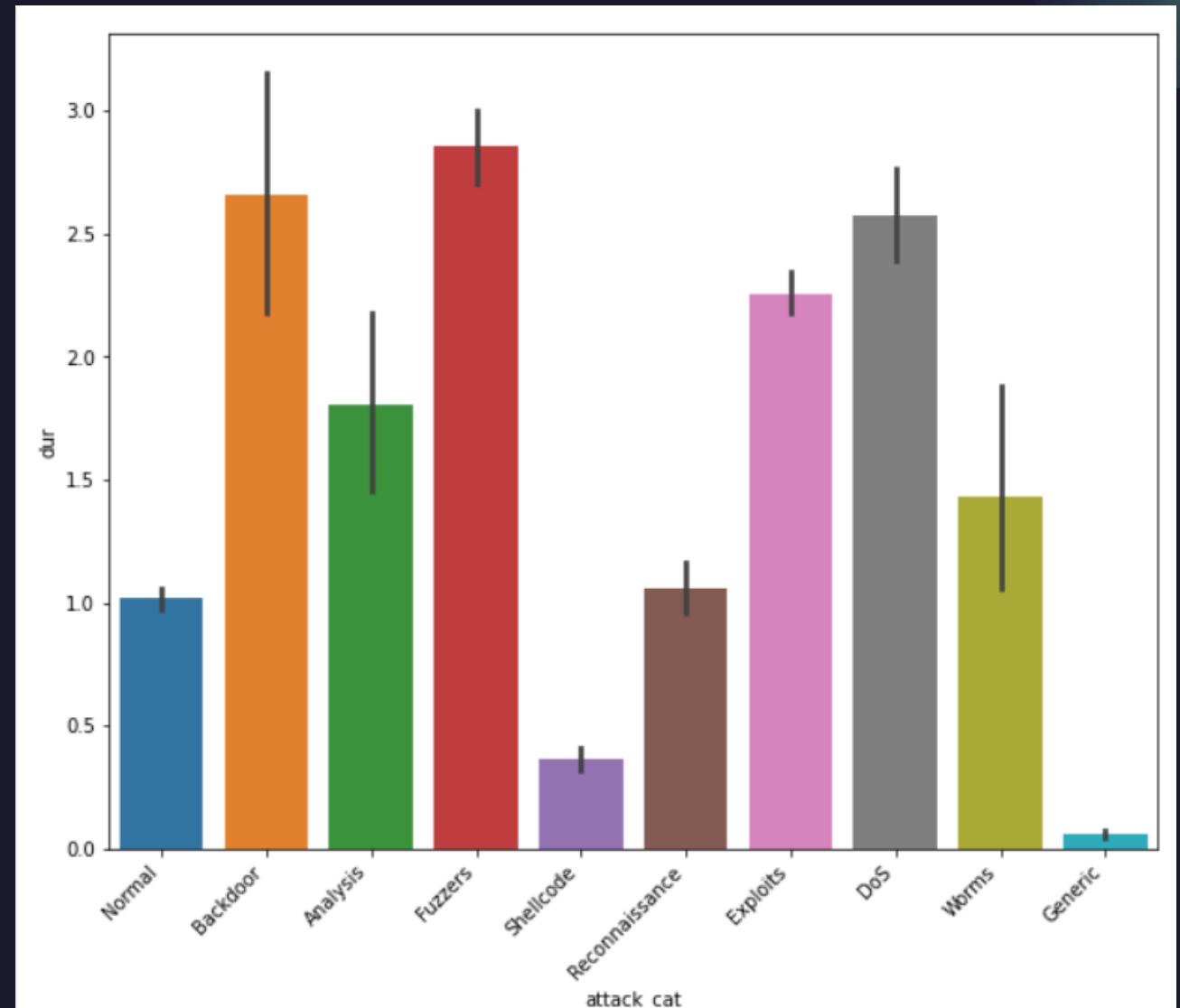


Fig.10. Grafico della durata di traffico di rete registrata nel caso di traffico normale e nel caso di traffico associato ad un comportamento malevolo.

# DataSafe – Presentazione dei Dati 5

- La quantità di traffico di rete si misura in Mbit/s, ad ogni processo si associa un *rate* di traffico di rete, che rappresenta la quantità di bit trasmessi.
- Tuttavia, in un problema di classificazione binaria risulta difficile distinguere tra traffico normale e traffico associato ad attacchi informatici.
- Il problema si sposta da una classificazione binaria ad un **many-class classification problem**, in cui ci si aspetta una *variable bit/rate* diverso per ogni attacco (VRB)

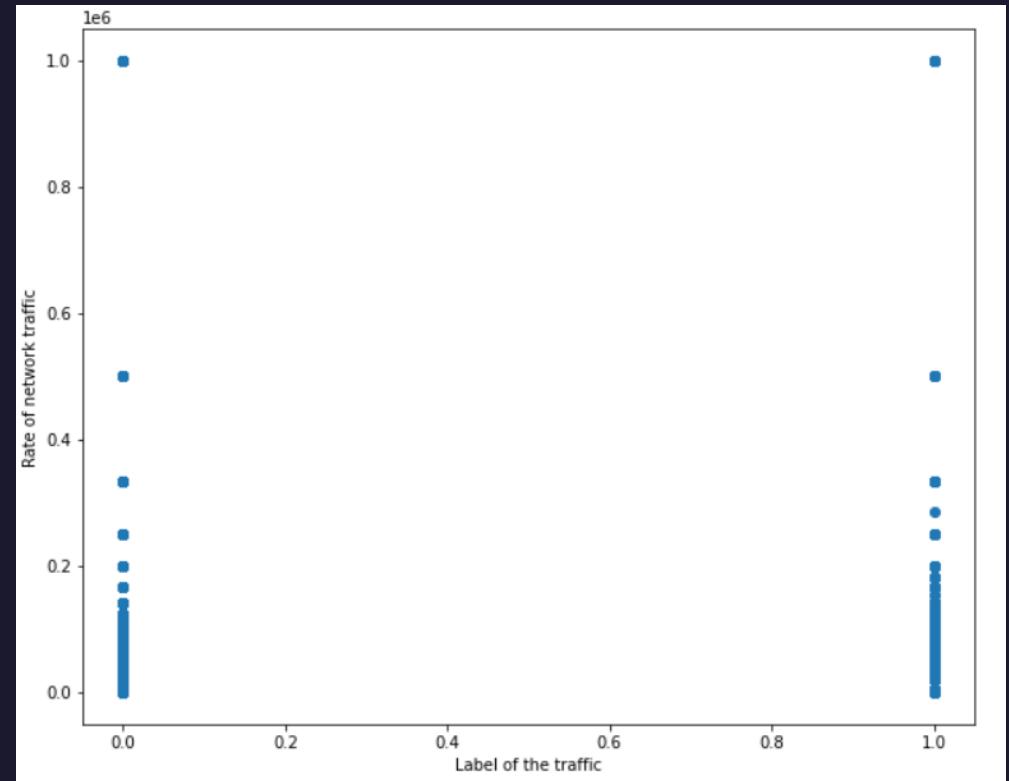


Fig. 11. Scatterplot del bit rate, rispettivamente traffico normale con label '0' e traffico malevolo con label '1'.

# DataSafe – Presentazione dei Dati 6

- Il traffico normale ha un bit/rate con distribuzione log-normale
- Gli attacchi informatici sono contraddistinti da:
  - comportamenti differenti in termini di bit/rate
  - presenza di outliers
  - Distribuzione multimodale

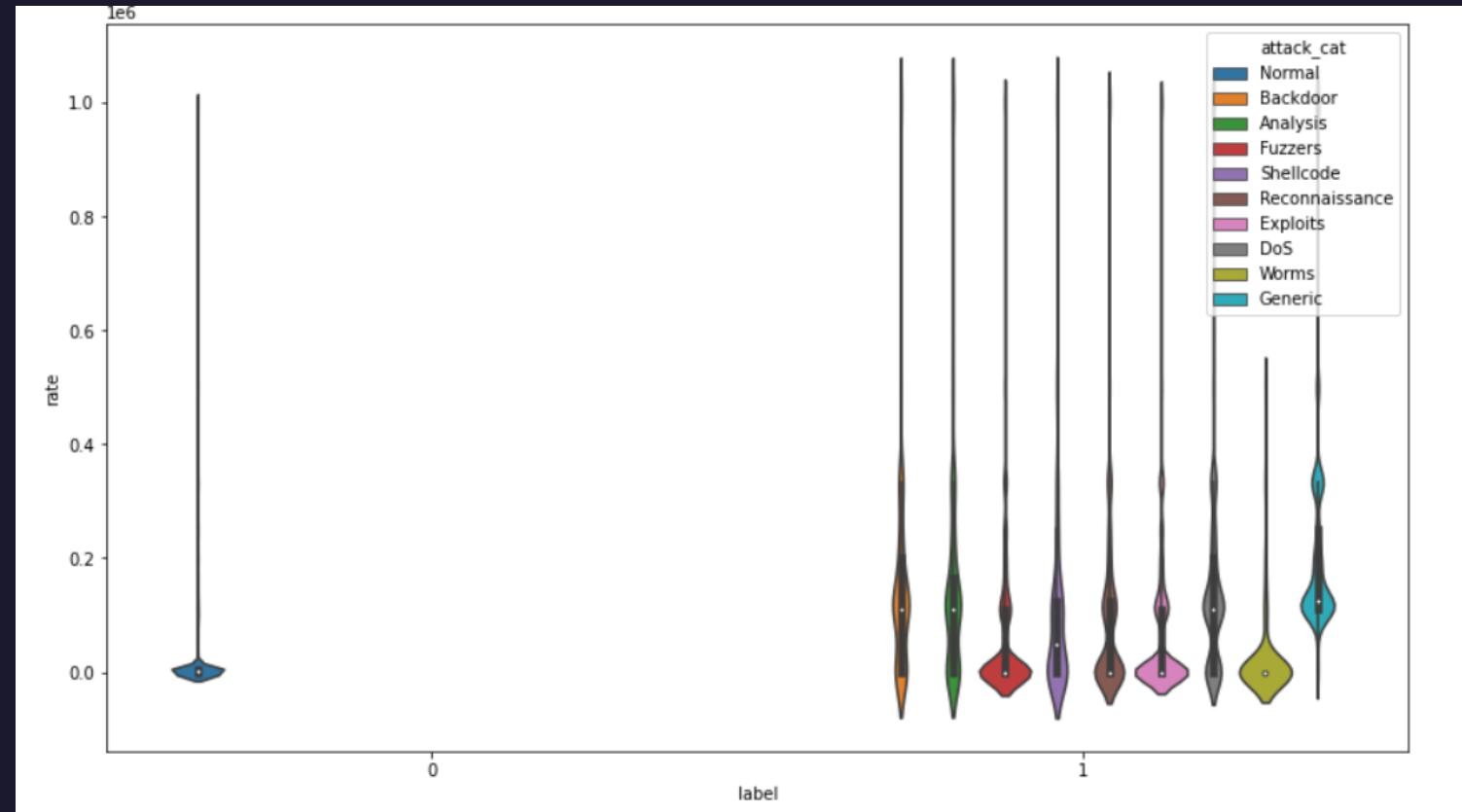


Fig.12. Violinplot del bit rate del traffico di rete per traffico normale e per categorica di attacco.

# DataSafe – Presentazione dei Dati 7

Un focus sui protocolli:

- Il traffico normale utilizza solo alcuni dei molteplici protocolli a disposizione
- Il traffico associato ad attacchi informatici fa largo uso dei tanti protocolli a disposizione. In particolare quasi tutti gli attacchi utilizzano gli stessi protocolli
- Difficoltà nell'identificare l'attacco da pattern tra le variabili

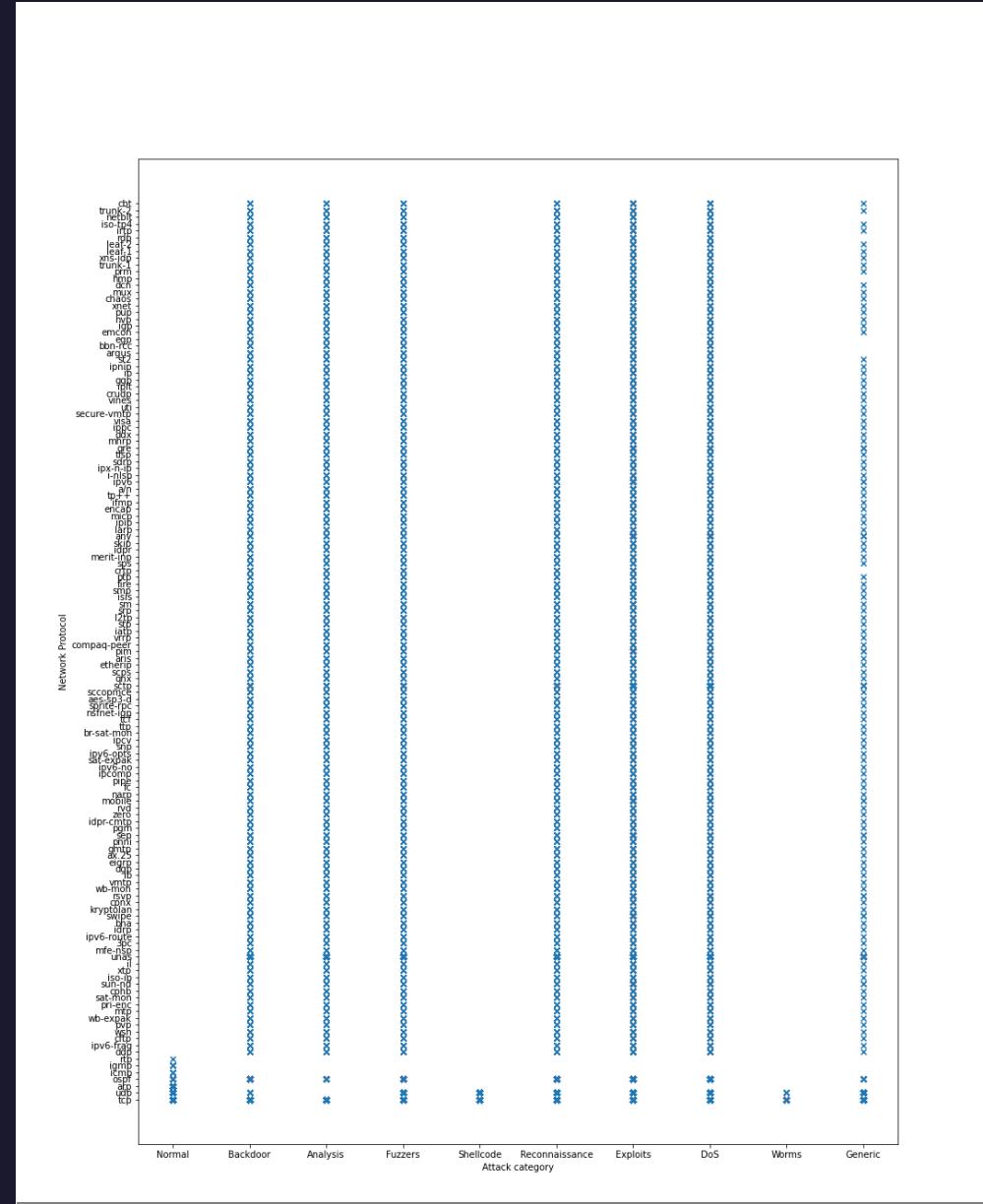


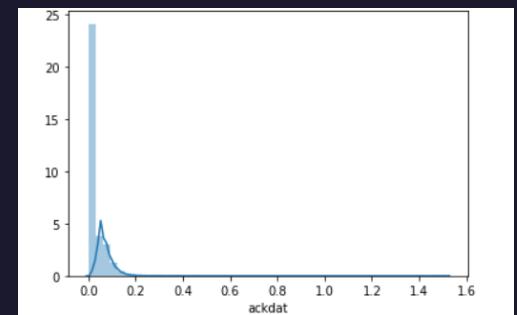
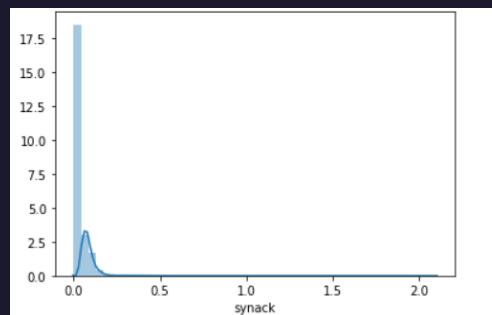
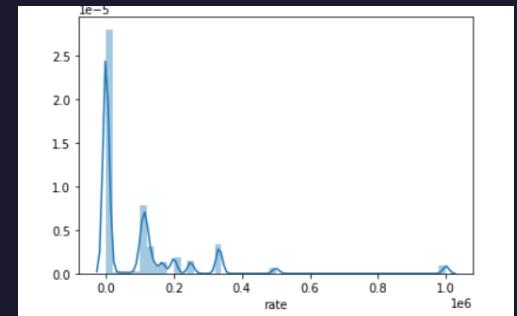
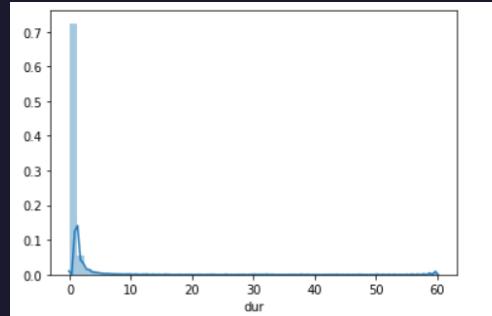
Fig.13. Grafico raffigurante i protocolli utilizzati da tutte le categorie di traffico presenti nel dataset.

# DataSafe – Analisi statistica

Il fulcro della nostra ricerca riguarda la possibilità non solo di andare a separare il traffico normale da quello anomalo, ma in quest'ultimo caso andare ad identificare l'attacco corretto.

Com'è possibile vedere, le variabili numeriche hanno differenti distribuzioni. Infatti, abbiamo notato che:

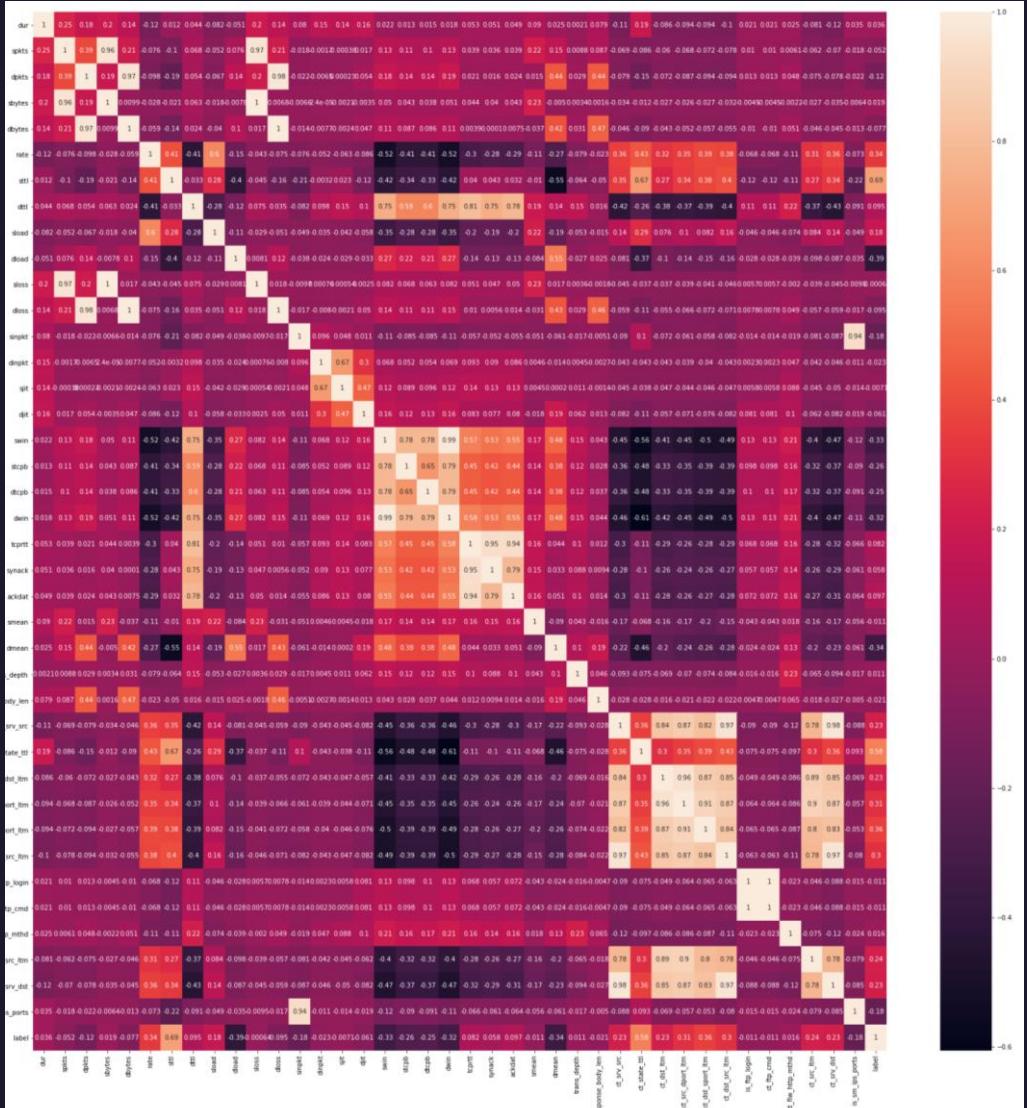
- La variabile di durata (dur) ha un'andamento log-normale,
- La variabile relativa al tasso al secondo di bit trasmessi tra le due fonti è multimodale



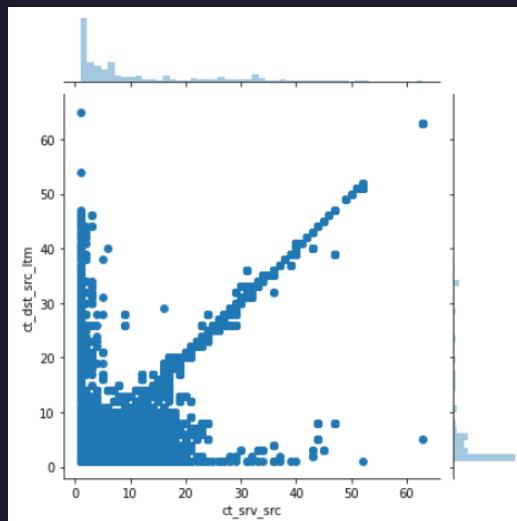
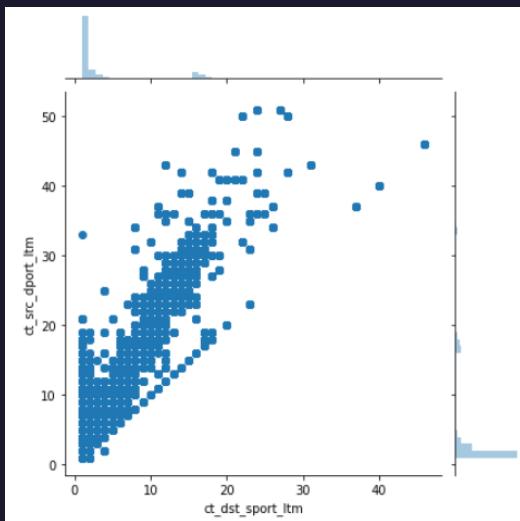
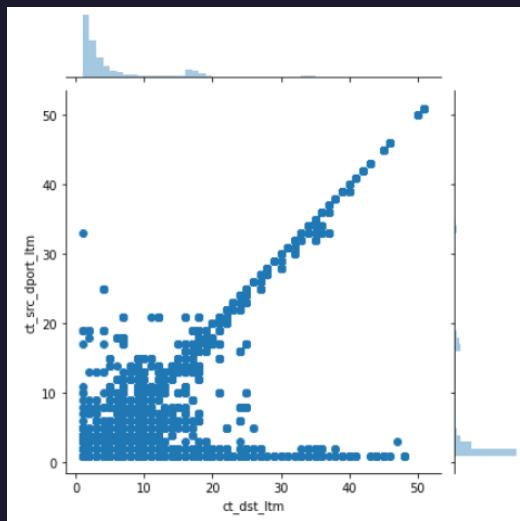
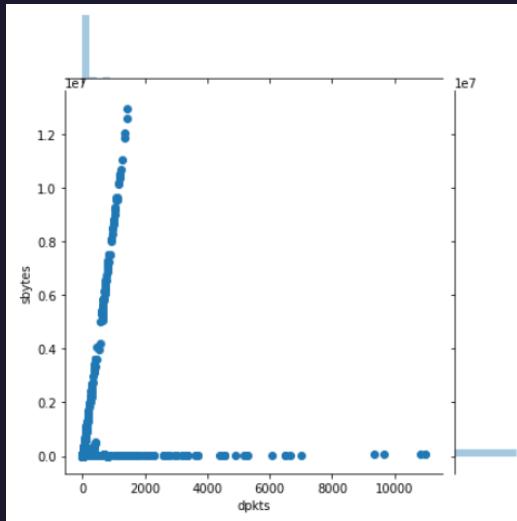
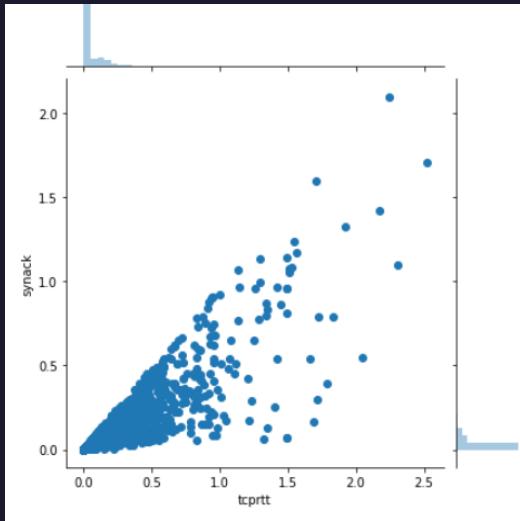
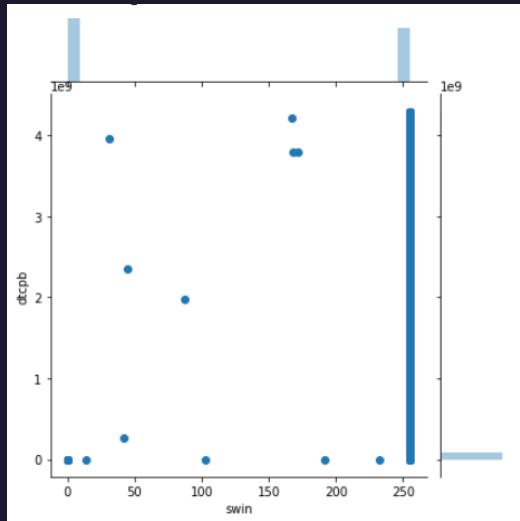
# DataSafe – Analisi della correlazione

La nostra analisi ha messo in luce quanto segue (bianco = correlazione perfetta):

- Le variabili sono scarsamente correlate con la nostra label di attacco.
  - Alcune variabili, come dpkts (I pacchetti a destinazione) e spkts (quelli sorgenti) o tcprtt (il round trip time del tcp) e synack (lo stato di connessione) o, ancora, swin (Source TCP window advertisement value) e dtcpb (Destination TCP base sequence number). Infine, una correlazione molto interessante risulta essere quella tra ct\_dst\_ltm (il numero di connessioni verso lo stesso indirizzo rispetto all'ultimo tempo, per 100 connessioni) e ct\_src\_dport\_ltm (il numero di connessioni alla sorgente per stessa porta)
  - In generale, i pattern presentati (prossima slide) risultano non essere lineari



# DataSafe – Analisi della correlazione



# DataSafe – Data Preprocessing

Per anomalie legate alla trascrizione di alcuni dati determinati valori di port sono stati inseriti in base esadecimale

222	192.168.241.243	49320	192.168.241.243	0xc0a8	icmp
2768	192.168.241.243	49320	192.168.241.243	0xc0a8	icmp
4205	192.168.241.243	49320	192.168.241.243	0xc0a8	icmp

Si è provveduto a convertire tali valori in base 10 ed escludendo valori superiori a 65535

```
b16 = lambda x: int(x,16)
hex_dsports.dsport = hex_dsports.loc[:, "dsport"].apply(b16)
hex_sports.sport = hex_sports.loc[:, "sport"].apply(b16)
```

# DataSafe – Data Preprocessing 2

La fase di preparazione dei dati è un aspetto fondamentale di tutto il processo di analisi dei dati. Il risultato finale è sicuramente influenzato dalle decisioni e mosse compiute in questa fase.

Per le seguenti variabili si è osservato:

- `ct_flw_http_mthd` (1): 53,2% missing values
- `is_ftp_login` (2): 56,3% missing values
- `ct_ftp_cmd` (3): 56,3% missing values

Circa il significato delle features, si rileva che, rispettivamente:

- 1) `ct_flw_http_mthd` indica il numero di flussi che usando il servizio http presentando i metodi Get/Post
- 2) `is_ftp_login` indica se la sessione ftp è stata acceduta tramite nome utente è password si assegna 1 altrimenti 0
- 3) `ct_ftp_cmd` indica il numero di flussi che hanno un comando di sessione ftp

# DataSafe – Data Preprocessing e NaN Hunting

Lo studio dei missing values ha rivelato che questi sono sistematici e nascondono un'informazione:

- I dati sono numerici continui
- Il missing value rappresenta un non utilizzo del protocollo di comunicazione relativo

Procediamo, quindi, con l'indagine dei singoli missing per servizio chiamato.

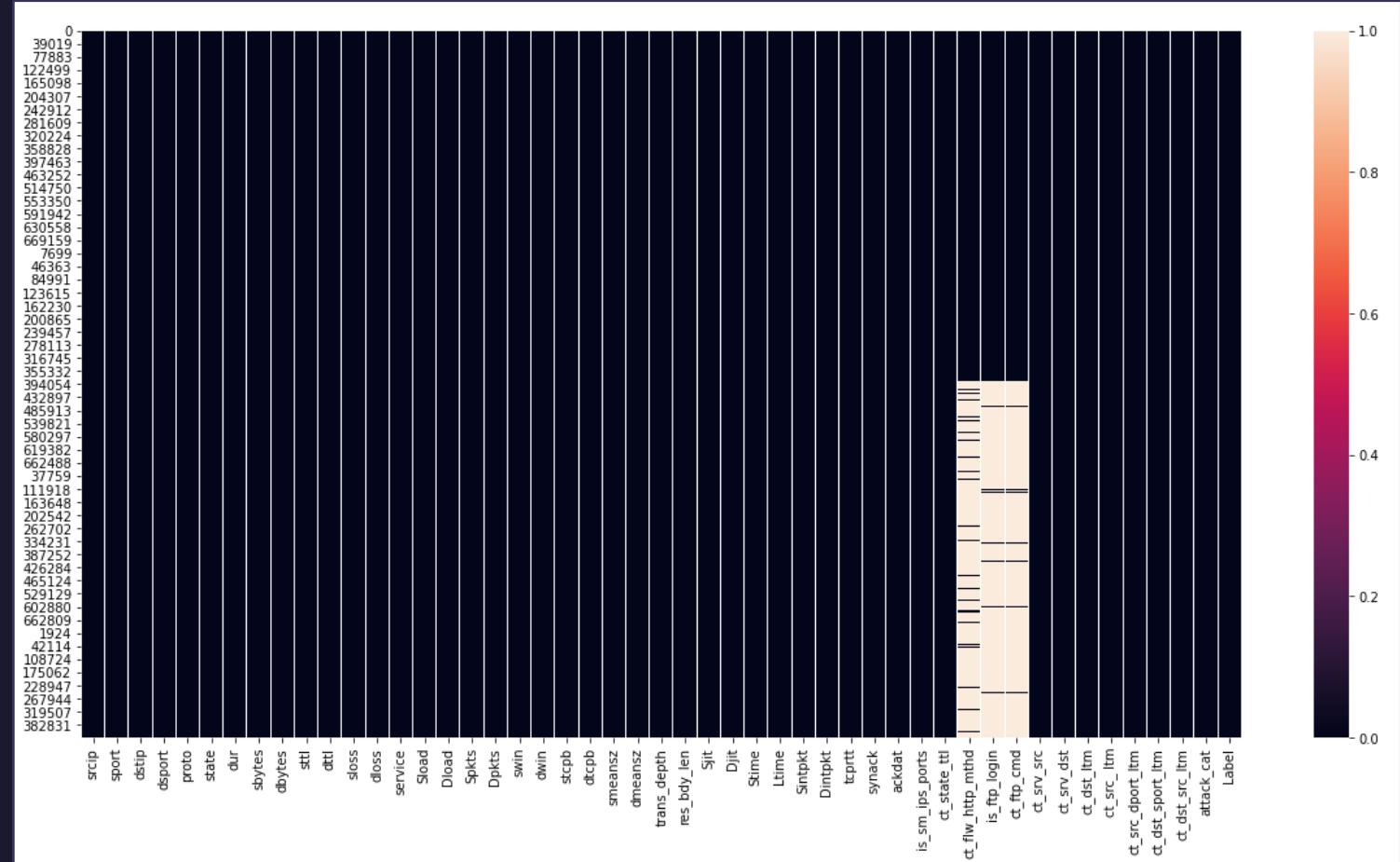


Fig.14. Matrice di rappresentazione degli NaN presenti del dataset.

# DataSafe – Data Preprocessing e NaN Hunting

Per studiare l'andamento degli NA, abbiamo dapprima fatto il subset dei dati per tipo di servizio, per poi visualizzare con delle heatmap i dati assenti per singolo elemento.

Nel caso del flusso http, abbiamo che `ct_flw_http_mthd` è presente a meno di poche osservazioni in cui ci sono dei packet drop.

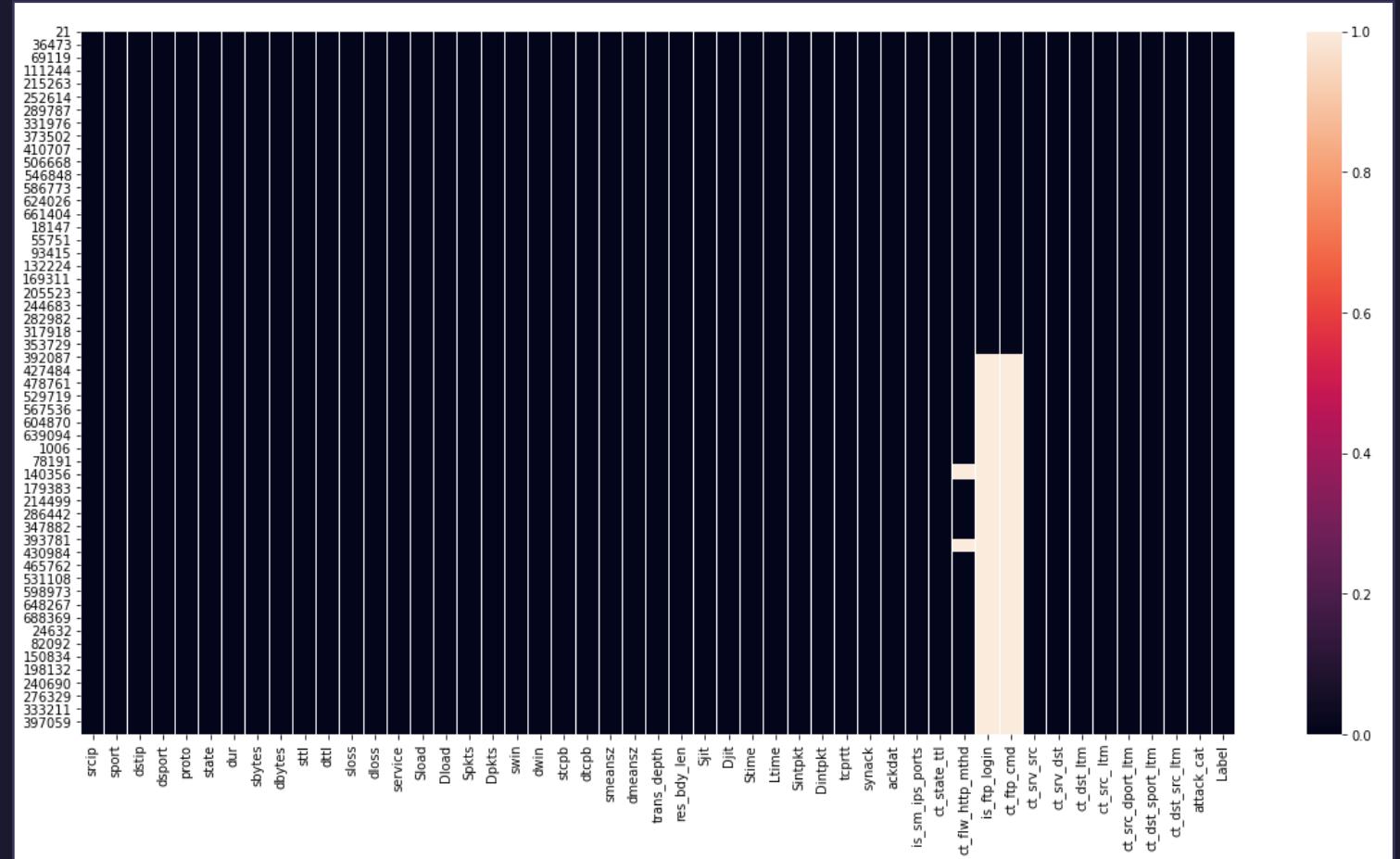


Fig.15. Matrice di rappresentazione degli NaN presenti del dataset.

# DataSafe – Data Preprocessing e NaN Hunting

Com'è possibile notare, il controllo flussi `ct_ftp_cmd` è presente, salvo sporadici casi inerenti il packet loss, nel momento in cui il servizio utilizzato è inerente a comandi ftp.

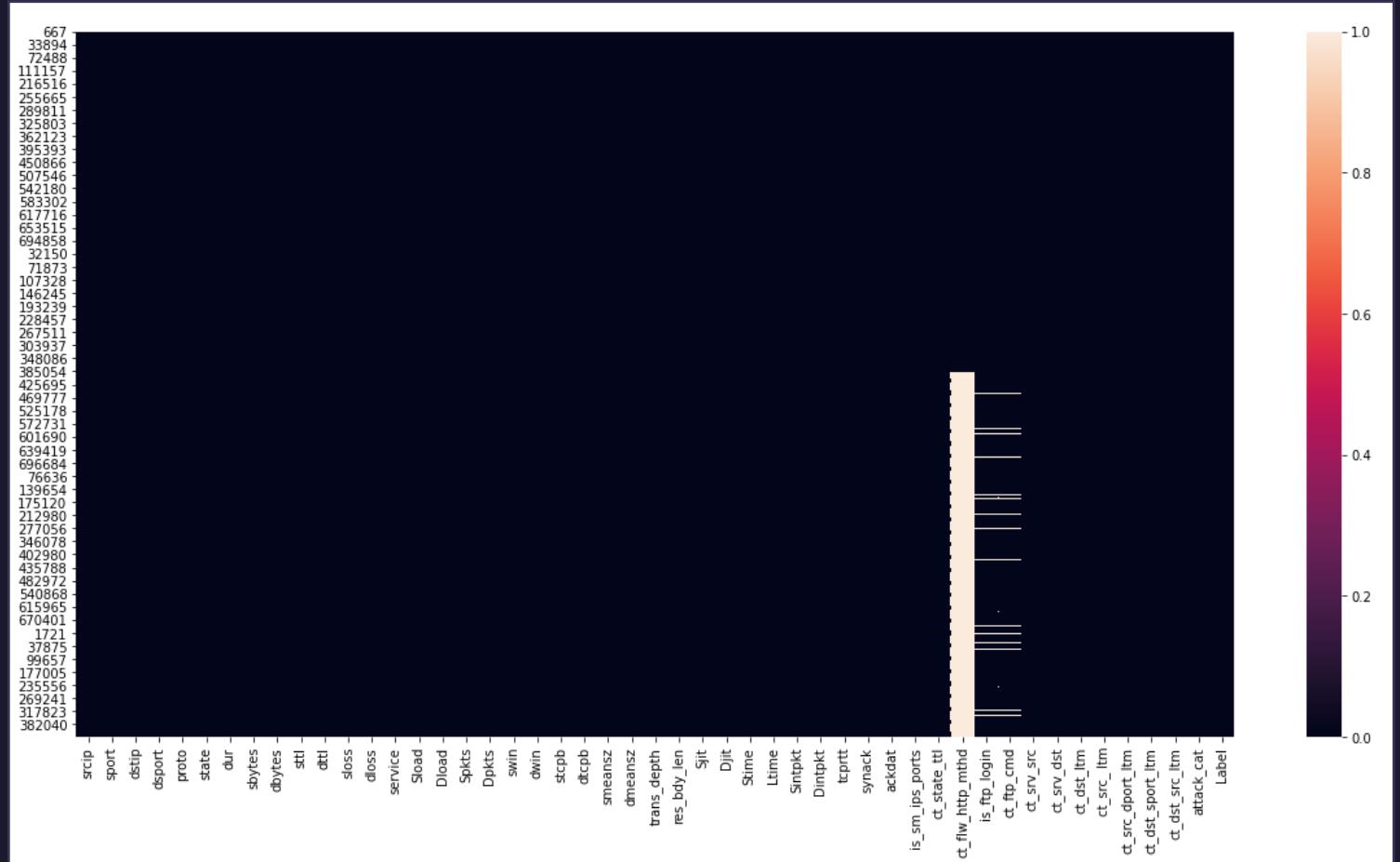


Fig.16. Matrice di rappresentazione degli NaN presenti del dataset.

# DataSafe – Data Preprocessing e NaN Hunting

Infine, `is_ftp_login` è in larga parte presente nel caso di login ftp, mentre, correttamente, non risulta essere ingaggiato da attività di scambio dati.

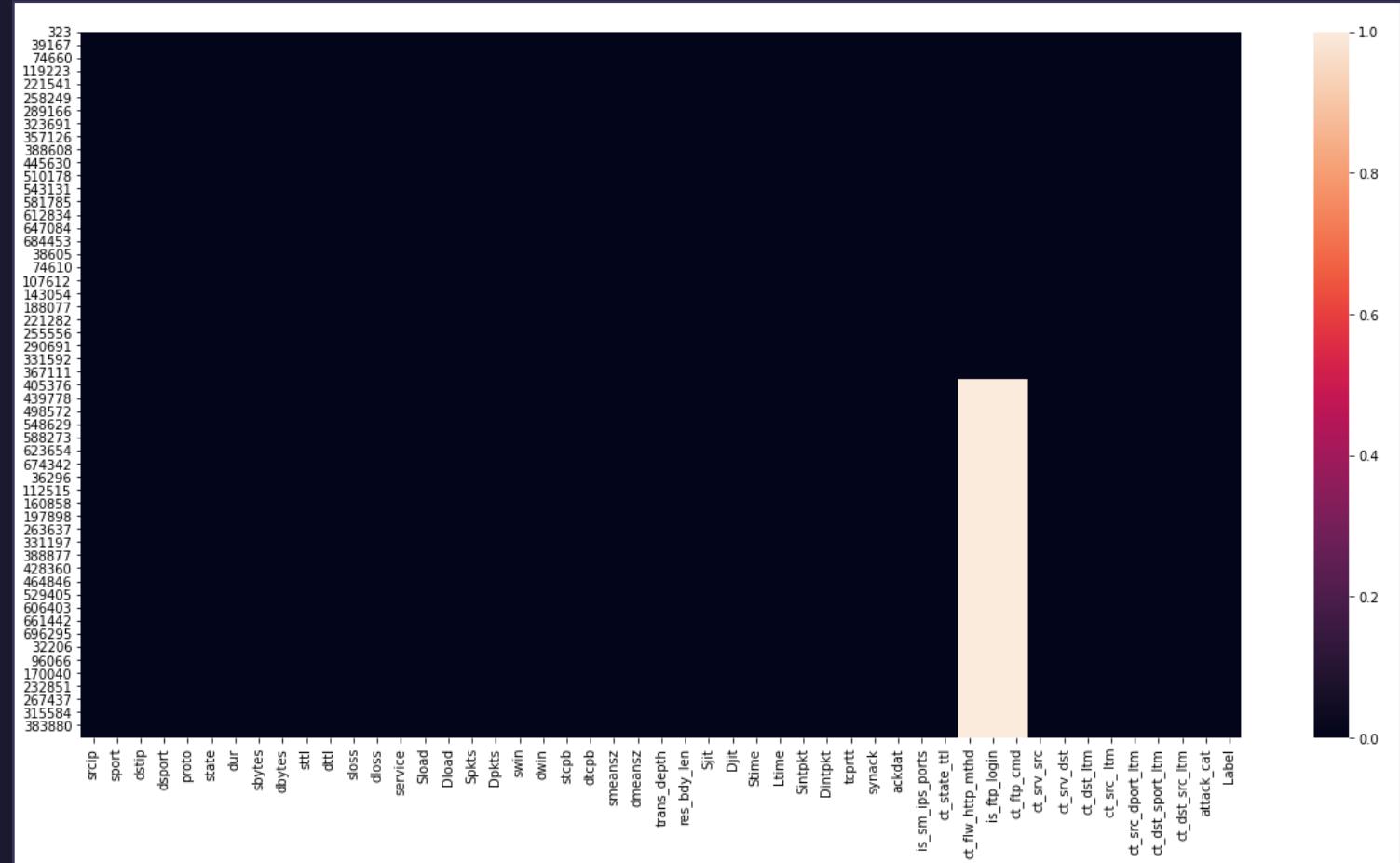


Fig.17. Matrice di rappresentazione degli NaN presenti del dataset.

# DataSafe – Data Preprocessing e NaN Imputation

Tanto premesso, di seguito la nostra strategia

- La strategia seguita è stata quella di andare a sostituire con ‘-1’ i valori mancanti, al fine di poter segnalare alla rete l’assenza del protocollo suddetto mediante pattern non lineare.
- Inoltre, questa entry è tipicamente out of range, andando a segnalare alla rete neurale il non utilizzo del protocollo, senza impattare in modo significativo su location e shift.

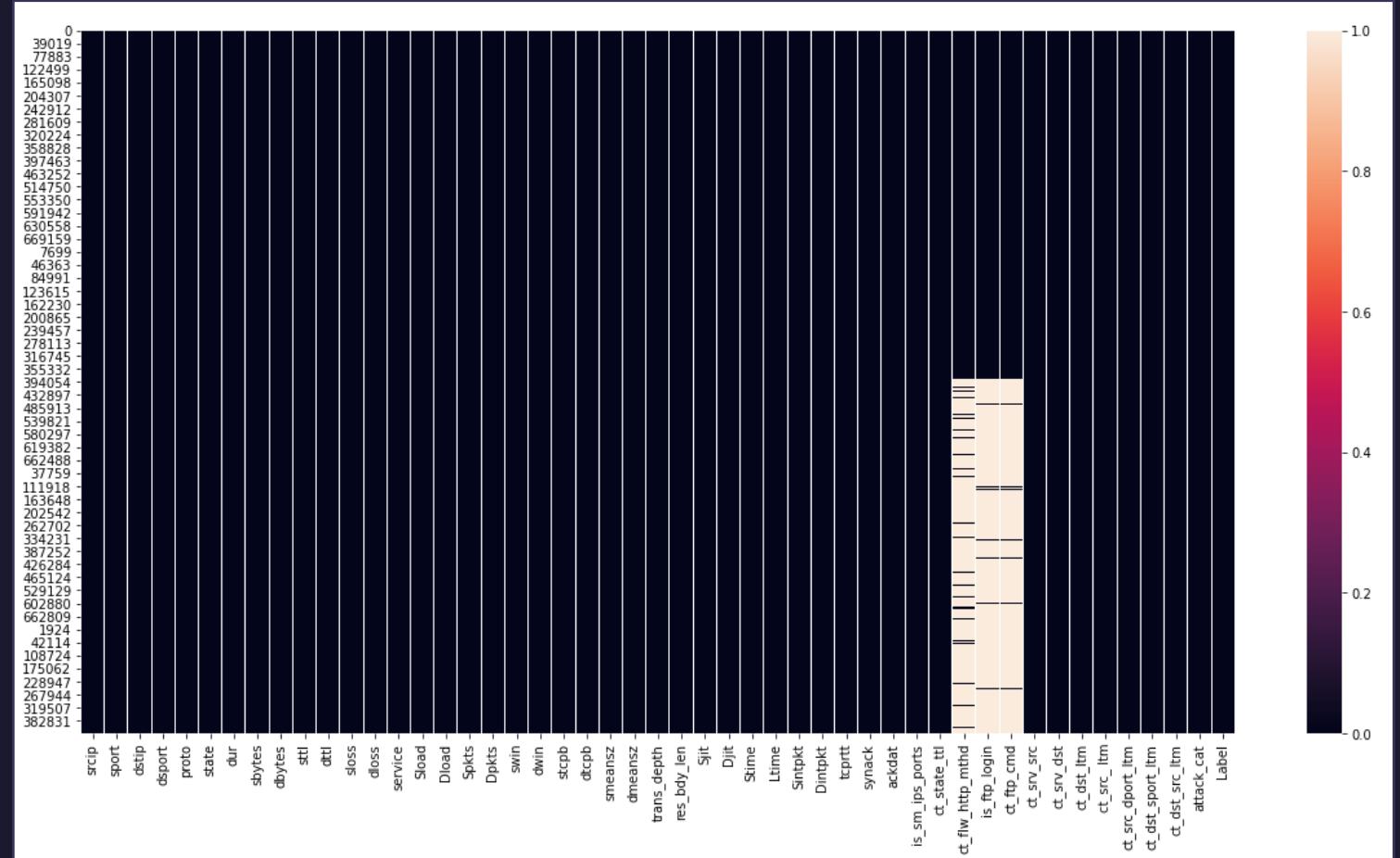


Fig.18. Matrice di rappresentazione degli NaN presenti del dataset.

# DataSafe – Modello Scelto

**Perché non lasciar scegliere alla rete quanti e quali layers usare per imparare ogni variabile?**

Il modello adottato è una ResNet 50. Tale modello è stato scelto perché:

- La sua struttura convolutiva permette di avere shared weights tra le varie features, andando ad alleggerire il training process
- Grazie agli skip elements, è possibile prevenire sia il gradient explosion che il gradient vanishing, andando a rendere il processo di addestramento più stabile
- La sua struttura a repeated bottleneck va a creare sia un feature engineering implicito che un processo di segmentazione dei dati, andando a focalizzarsi di più sulle anomalie

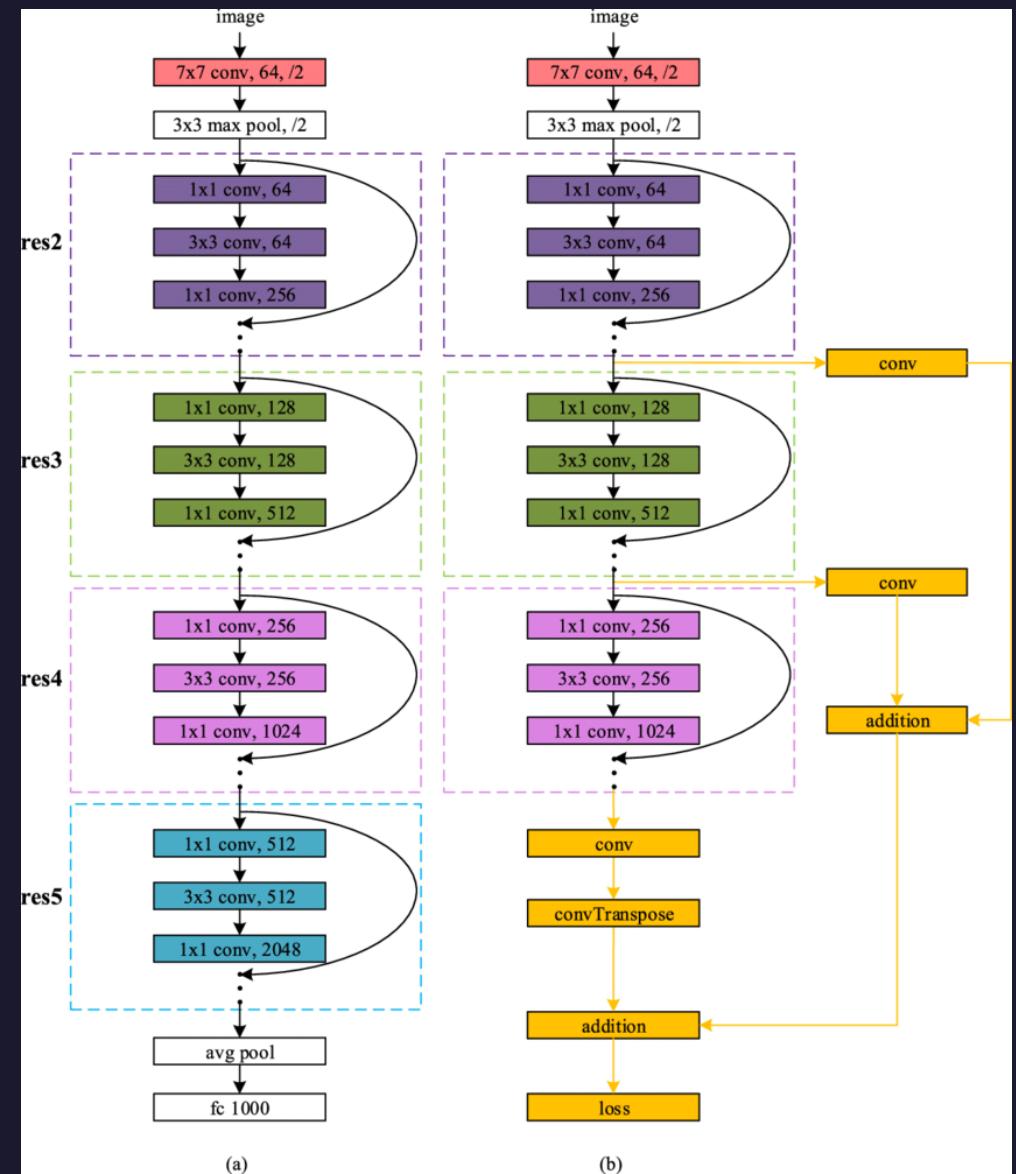
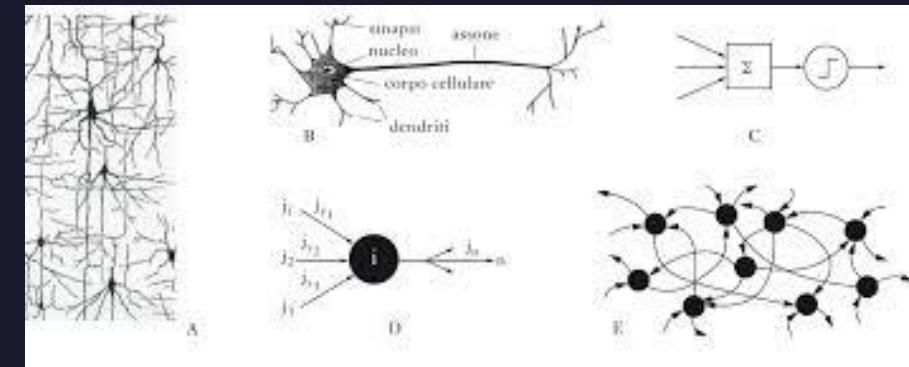


Fig.19. Architettura di una ResNet.

# DataSafe – Il Nostro Modello



Il nostro modello consiste nell'adattamento ai dati monodimensionali dell'architettura ResNet.

In particolare, abbiamo usato:

- Un **identity block**, costituito da strati Conv1D la cui finestra ha dimensione 1
- Un **convolutional block**, costituito da strati Conv1D la cui finestra ha dimensione 3
- Entrambi gli strati hanno uno **skip element**, che funge da scorciatoia
- L'identity block crea un **bottleneck** che permette di velocizzare il processo di addestramento, andando a creare una struttura ad autoencoding

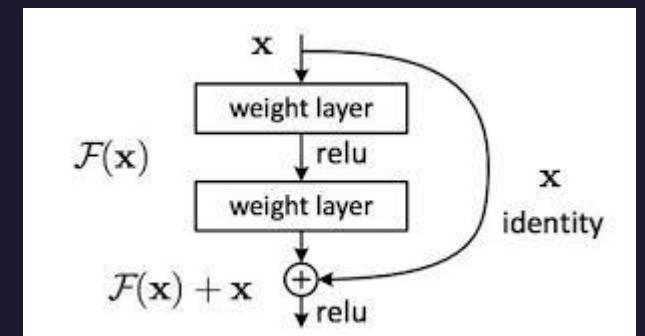


Fig.20. «Skip connection» scheme in una rete ResNet.

# DataSafe – Il Nostro Modello - Activation

La funzione logistica può essere utilizzata solo in ambiente di classificazione binario.

È possibile definire una generalizzazione della funzione logistica, garantendoci così di poter lavorare con many-class output: **softmax function**.

Tale funzione prende come input un vettore **Z (C-dim)** e restituisce come output un vettore **y (C-dim)** di valori reali tra 0-1.

$$y_c = S(z)_c = \frac{e^{zc}}{\sum_{d=1}^C e^{zd}}$$

- Il denominatore dell'espressione ha il compito di regolarizzazione per far sì che si abbiano  $\sum_{c=1}^C y_c = 1$

La probabilità che la classe  $t = c$  for  $c = 1 \dots C$ , dato  $z$ , è data da:

$$\begin{bmatrix} P(t = 1|z) \\ \vdots \\ P(t = C|z) \end{bmatrix} = \begin{bmatrix} S(z)_1 \\ \vdots \\ S(z)_C \end{bmatrix} = \frac{1}{\sum_{d=1}^C e^{zd}} \begin{bmatrix} e^{z1} \\ \vdots \\ e^{zC} \end{bmatrix}$$



# DataSafe – Il Nostro Modello – Ottimizzazione

L'uso della softmax è accompagnato dal calcolo della sua derivata in fase di ottimizzazione del modello, in cui ricava la **Loss**.

Per il nostro modello è stata utilizzata una **categorical cross-entropy**

La **likelihood** per M classi è così definita:

$$L = \prod_{c=1}^M p_c^{N_{y_c}}$$

Tale funzione si vuole sia massima per le stime ottenute per ciascuna classe a cui un dato campione appartiene:

$$\log L = \sum_{c=1}^M N_y \log(p_c) = -CE$$

Avendo definito con **CE** la cross-entropy.

Inoltre, è richiesta la massimizzazione della funzione di verosomiglianza:  $\max \log L_{pc}$

Per ogni osservazione la Loss è:  $CE = - \sum_{c=1}^M y_0 \log(p_{o,c})$    $\max - \sum_{c=1}^M y_0 \log(p_{o,c})$

# DataSafe – Benchmarks

Il nostro modello garantisce:

- un'accuracy molto elevata su grandi quantità di dati
  - Tale accuracy non è certamente raggiungibile mediante gli approcci tradizionali di ML (Figura 21)
- Un'accuracy maggiore rispetto a quanto presentato e raggiunto in *N. Shone, T. N. Ngoc, V. D. Phai and Q. Shi, "A Deep Learning Approach to Network Intrusion Detection \**
- L'algoritmo illustrato schematicamente in figura mostra avere un'accuratezza del 94.58% su una versione molto ridotta del dataset NSL - KDD

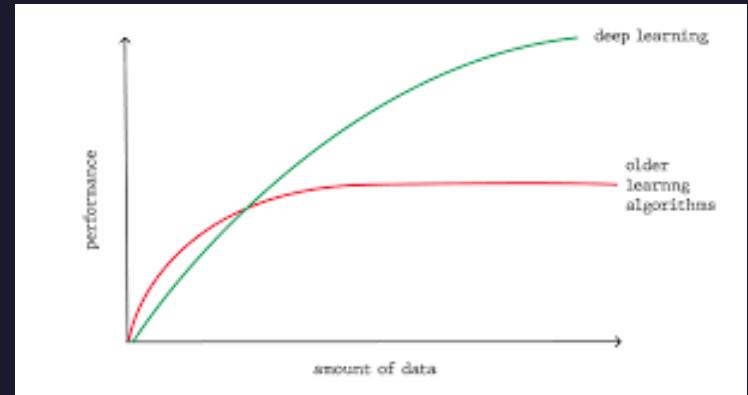


Fig.21. Confronto tra machine learning e deep learning



Fig.22. Rappresentazione architettura rete neurale in «A Deep Learning Approach to Network Intrusion Detection»

# DataSafe – Benchmarks

Attack Class	No. Training	No. Attacks	Accuracy (%)		Precision (%)		Recall (%)		F-Score (%)		False Alarm (%)	
			DBN	S-NDAE	DBN	S-NDAE	DBN	S-NDAE	DBN	S-NDAE	DBN	S-NDAE
Normal	97278	60593	99.49	99.49	94.51	100.00	99.49	99.49	96.94	99.75	5.49	8.92
DoS	391458	223298	99.65	99.79	98.74	100.00	99.65	99.79	99.19	99.89	1.26	0.04
Probe	4107	2377	14.19	98.74	86.66	100.00	14.19	98.74	24.38	99.36	13.34	10.83
R2L	1126	5993	89.25	9.31	100.00	100.00	89.25	9.31	94.32	17.04	0.00	0.71
U2R	52	39	7.14	0.00	38.46	0.00	7.14	0.00	12.05	0.00	61.54	100.00
Total	494021	292300	97.90	97.85	97.81	99.99	97.91	97.85	97.47	98.15	2.10	2.15

Tab.1- Performance sul dataset KDD CUP '99



Attack Class	No. Training	No. Attacks	Accuracy (%)		Precision (%)		Recall (%)		F-Score (%)		False Alarm (%)	
			DBN	S-NDAE	DBN	S-NDAE	DBN	S-NDAE	DBN	S-NDAE	DBN	S-NDAE
DoS	45927	5741	87.96	94.58	100.00	100.00	87.96	94.58	93.60	97.22	8.80	1.07
Normal	67343	9711	95.64	97.73	100.00	100.00	95.64	97.73	97.77	98.85	24.29	20.62
Probe	11656	1106	72.97	94.67	100.00	100.00	72.97	94.67	84.37	97.26	18.40	16.84
R2L	995	2199	0.00	3.82	0.00	100.00	0.00	3.82	0.00	7.36	0.00	3.45
U2R	52	37	0.00	2.70	0.00	100.00	0.00	2.70	0.00	5.26	0.00	50.00
Total	125973	18794	80.58	85.42	88.10	100.00	80.58	85.42	84.08	87.37	19.42	14.58

Tab.2- Performance sul dataset NSL – KDD

# DataSafe- Le metriche utilizzate

- $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$  misura la proporzione del numero totale di classificazioni corrette
- $Precision = \frac{TP}{TP+FP}$  misura la proporzione del numero corretto di classificazioni penalizzate dal numero di classificazioni incorrette
- $Recall = \frac{TP}{TP+FN}$  misura la proporzione del numero corretto di classificazioni penalizzate dal numero di classificazioni mancate
- $F1 - Score = \frac{precision*recall}{precision+recall}$  media armonica tra precision e recall. Fornisce una misura diretta delle performance dell'algoritmo



# DataSafe – Risultati

- Il processo di training è stato effettuato con batch di 2048 osservazioni. Tale decisione è dovuta alla volontà di rendere più consistenti le nostre stime.
- Con un addestramento per 10 epoch si ottiene un'accuracy del 99% sia per il validation che per il training set.
- Non vi è presenza di Overfitting considerando che anche l'accuracy sul test set risulta essere comunque del 99%
  - in fase di training è stata anche condotta una cross-validation avendo così la certezza della mancanza di overfitting sulle stime dei parametri
- Tuttavia, essendo il dataset fortemente afflitto da Class Imbalance, abbiamo effettuato anche le stime per:
  - Precision: 0.9998
  - Recall: 0.9992
  - F1-Score: 0.9995

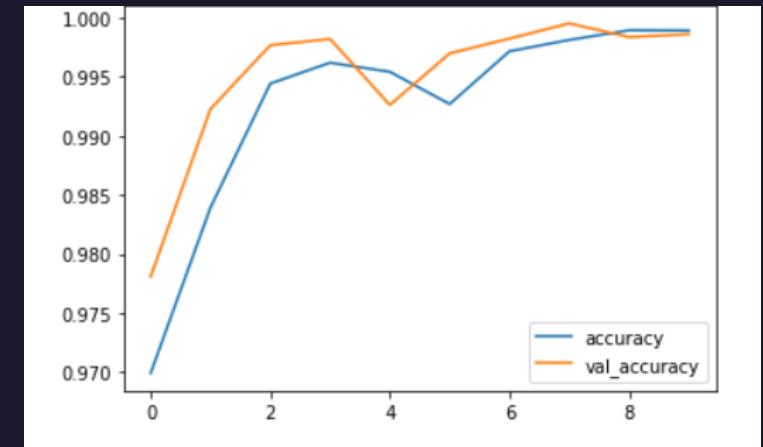
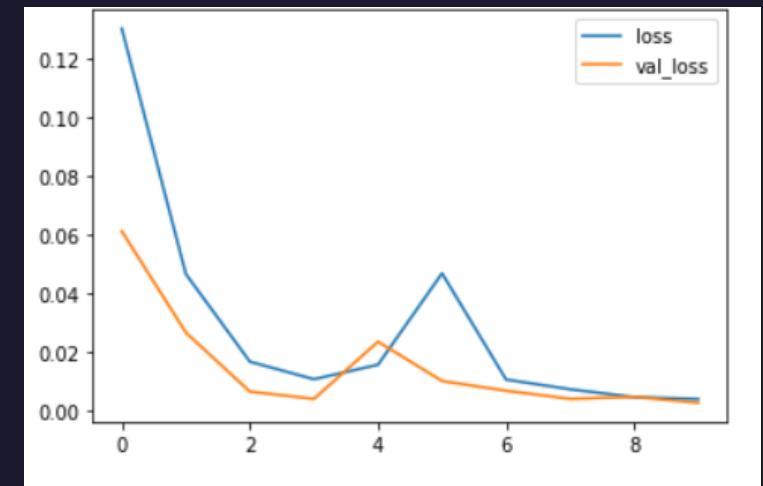


Fig.23. History Loss and Accuracy nella fase di training della rete. Ordinate = Numero di Epoche

# DataSafe – Risultati

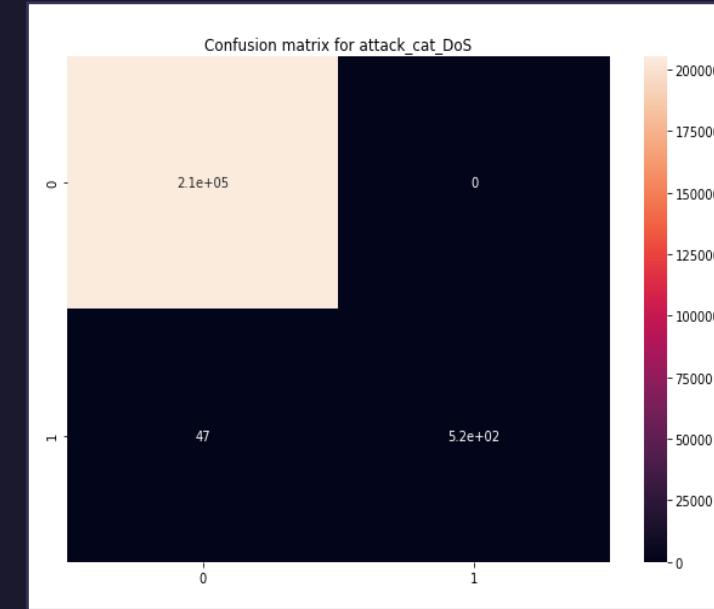
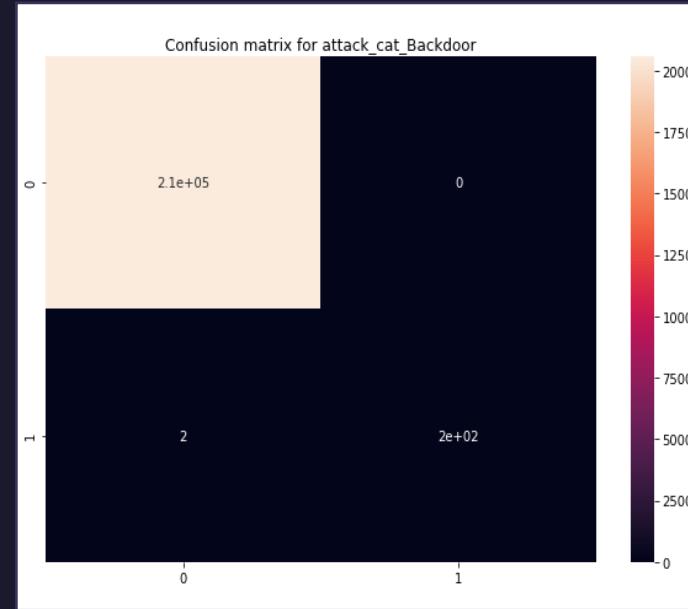
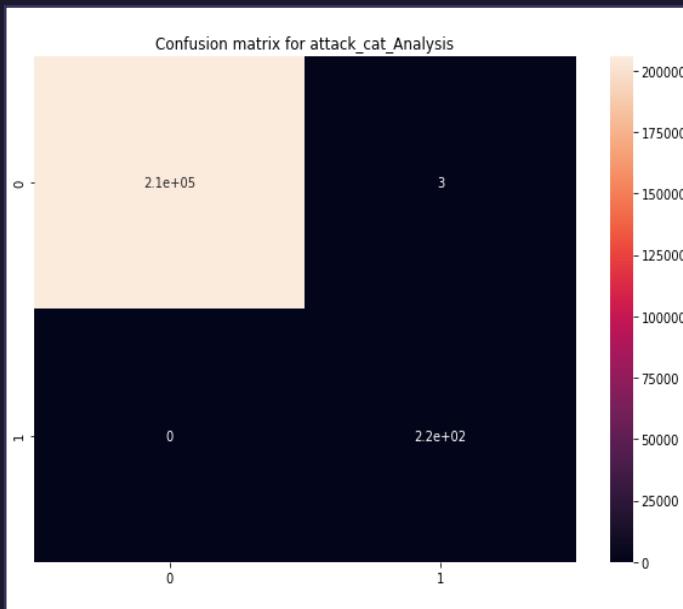


Fig.24-25-26. Confusion matrix per il classification report circa le classi di traffico

# DataSafe – Risultati

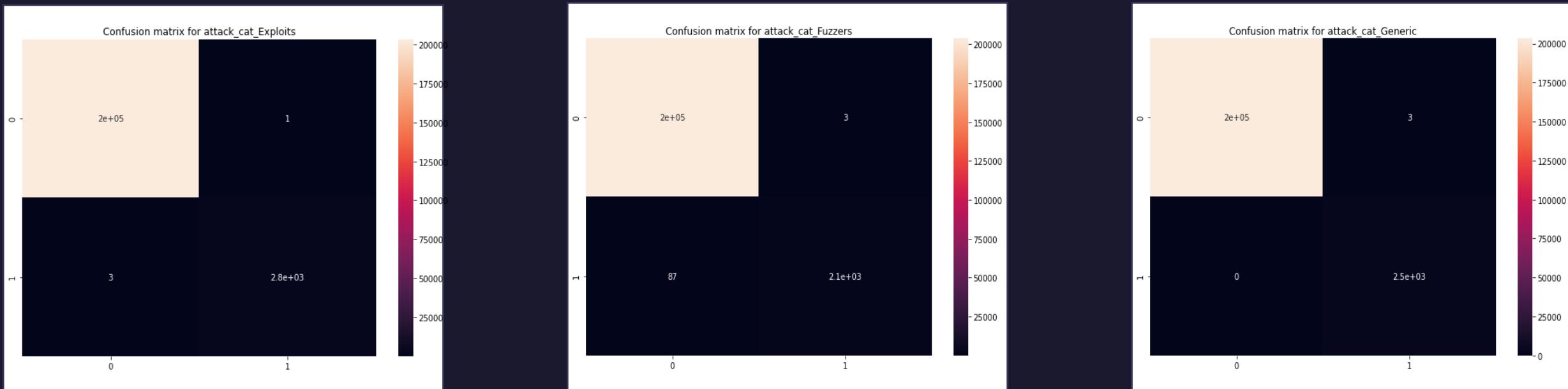


Fig.27-28-29. Confusion matrix per il classification report circa le classi di traffico

# DataSafe – Risultati

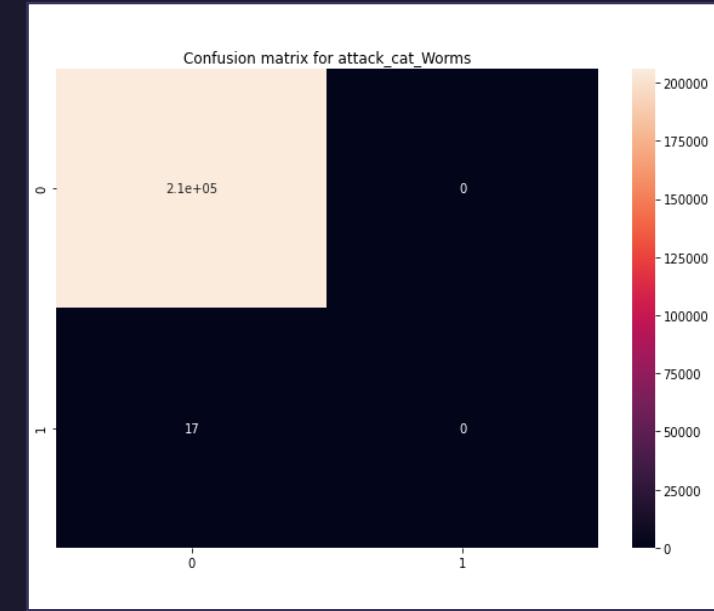
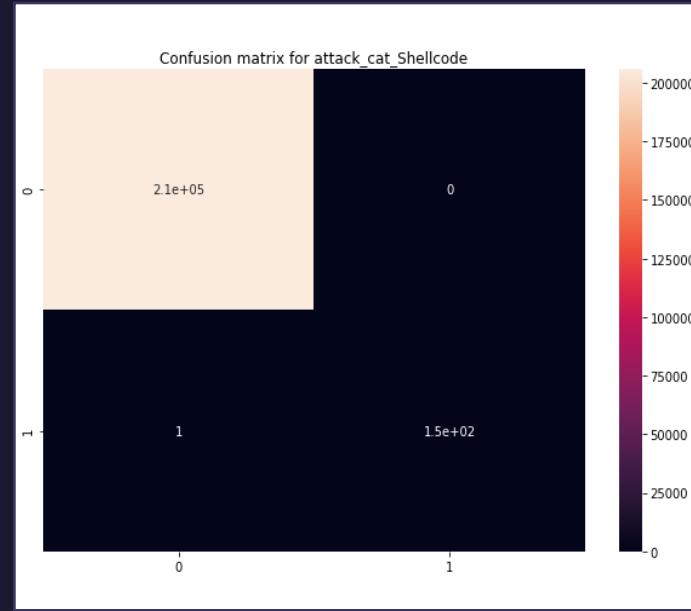
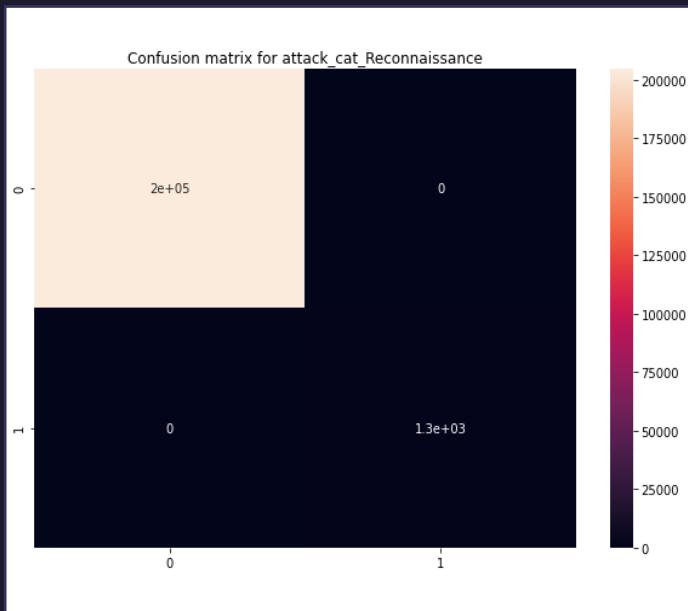


Fig.30-31-32. Confusion matrix per il classification report circa le classi di traffico

Grazie per l'attenzione

