



UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Laurea Triennale in Informatica

TESI DI LAUREA

Studio di modelli di IA per l'analisi del traffico di rete

RELATORE

Prof. Christiancarmine Esposito

TUTOR AZIENDALE

Dott. Filippo Bogetti

CANDIDATO

Daniele Liguori

Matricola: 0512113209

Anno Accademico 2023-2024

Questa tesi è stata realizzata con la collaborazione di



Abstract

La crescente complessità delle minacce informatiche ha accentuato la necessità di Intrusion Detection Systems (IDS) sempre più sofisticati. Tuttavia, molti dei modelli attuali presentano limiti significativi, come scarsa capacità di adattamento a nuove tipologie di attacchi e performance inadeguate con dati altamente sbilanciati. Questa tesi esplora l'applicazione di tecniche di machine learning per migliorare l'efficacia degli IDS, focalizzandosi su cinque modelli principali: Naive Bayes, Regressione Logistica, Analisi Discriminante Lineare, Reti Neurali e Random Forest. Utilizzando i dataset NSL-KDD e CIC-IDS2017, il lavoro analizza le prestazioni dei modelli attraverso il pre-processing dei dati, il bilanciamento, la feature selection e l'ottimizzazione dei parametri tramite grid search. I risultati evidenziano che i modelli più complessi, come Random Forest e le Reti Neurali, offrono prestazioni superiori, specialmente nella classificazione multiclasse. In particolare, Random Forest ha dimostrato una robustezza e una precisione superiori sia nel dataset NSL-KDD che in quello CIC-IDS2017. Sebbene questo lavoro non abbia superato completamente i limiti dello stato dell'arte, ha fornito approfondimenti utili e un altro punto di vista sui modelli più promettenti e sulle loro capacità di affrontare le sfide nella rilevazione degli attacchi.

Indice

Elenco delle Figure	iv
Elenco delle Tabelle	vi
1 Introduzione	1
1.1 L'importanza della sicurezza informatica in azienda	1
1.2 Breve panoramica sui rischi e le minacce attuali	4
1.3 Principali Cyber-Gang	6
2 Analisi del traffico di rete	8
2.1 Intrusion Detection Systems (IDS)	8
2.1.1 Definizione di IDS e Scopo	8
2.1.2 Tipologie di IDS	9
2.2 Intrusion Prevention Systems (IPS)	10
2.2.1 Definizione di IPS e differenze rispetto agli IDS	10
2.2.2 Ruolo degli IPS nella prevenzione degli attacchi	10
2.2.3 Architettura e funzionamento degli IPS	11
2.2.4 Suricata - Applicazioni reali	12
2.3 Next Generation Firewall (NGFW)	15
2.3.1 Definizione di NGFW e differenze rispetto ai firewall tradizionali	15
2.3.2 Funzionalità avanzate dei NGFW	15

2.3.3	Altri tipi di firewall	16
2.4	Software Defined Networks (SDN)	22
2.4.1	Introduzione alle SDN	22
2.4.2	Il ruolo delle SDN nell'analisi del traffico di rete	22
2.4.3	SDN e il traffico generato: sfide e opportunità per gli IDS . . .	23
3	Intelligenza Artificiale applicata alla sicurezza informatica	25
3.1	Concetto di IA e sua evoluzione nell'ambito della sicurezza informatica	25
3.2	Ruolo dell'IA nella rilevazione e prevenzione delle intrusioni	27
3.3	Tecniche di IA utilizzate nella rilevazione e prevenzione delle intrusioni	28
3.4	Vantaggi e sfide nell'integrazione di IDS, IPS, NGFW e IA	34
4	Sviluppo di un IDS con tecniche di ML	36
4.1	Introduzione	36
4.2	Dataset Selection	37
4.2.1	NSL-KDD	37
4.2.2	CIC-IDS2017	38
4.2.3	Confronto tra NSL-KDD e CIC-IDS2017	38
4.3	Analisi della Letteratura	38
4.3.1	Performance del dataset NSL-KDD	39
4.3.2	Performance del dataset CIC-IDS2017	40
4.4	Data Pre-Processing	42
4.4.1	Data Cleaning	42
4.4.2	One-Hot Encoding delle Variabili Categoriche	42
4.4.3	Standardizzazione delle Feature	43
4.5	Data Balancing	43
4.5.1	Data Balancing nella Classificazione Binaria	43
4.5.2	Data Balancing nella Classificazione Multiclasse	45
4.6	Suddivisione del Dataset	45
4.7	Training	46
4.7.1	Scelta dei Modelli	46
4.7.2	Confronto tra Classificazione Binaria e Multiclasse	47
4.7.3	NSL-KDD	48

4.7.4	CIC-IDS2017	57
4.7.5	Confronto delle Performance in Letteratura	68
4.7.6	NSL-KDD	68
4.7.7	CIC-IDS2017	69
4.8	Feature Selection con XGBoost	70
4.8.1	Differenze nelle Performance	71
4.9	GridSearch	73
4.9.1	Parametri e Modalità di Testing	73
4.9.2	Confronto delle Prestazioni	74
4.9.3	Scelta della Configurazione	76
5	Architettura dell'IDS	78
5.1	Introduzione all'architettura	78
5.2	Implementazione della SDN con Ryu	78
5.3	Modulo di Machine Learning	80
5.3.1	Scelta dei Modelli di Machine Learning	80
5.3.2	Pipeline di addestramento e predizione	80
5.3.3	Reazione dinamica agli attacchi basata su ML	81
5.4	Funzionamento del Flusso di Dati	82
5.5	Generazione del traffico	83
5.5.1	Strumenti per la generazione di traffico	83
5.5.2	Simulazione di attacchi e comportamento anomalo	84
5.5.3	Vantaggi e sfide nella generazione di traffico	85
5.6	Vantaggi e Limitazioni dell'architettura	86
6	Conclusioni	88
Bibliografia		91

Elenco delle figure

1.1	Attacchi per anno 2019-2023	2
1.2	Confronto crescita % Italia Vs Global	3
1.3	Severity % 2019-2023	3
1.4	Tecniche di attacco nel 2023	5
2.1	Eve, la principale dashboard di logging di Suricata	14
2.2	Integrazione con Splunk Enterprise	14
2.3	Applicazione di una regola statica in iptables per bloccare il traffico in arrivo sull'interfaccia wlan0	20
2.4	Uno sguardo nella dashboard di Cloudflare sviluppata in collaborazione con Graylog	21
3.1	Esempio di funzionamento di Semgrep Assistant	26
3.2	Architettura del modello descritto nel paper "An Unsupervised Deep Learning Model for Early Network Traffic Anomaly Detection", incluso il preprocessing (a sinistra) e il modulo di training e auto-learning (a destra)	29
3.3	Architettura del modello descritto nel paper "Intrusion Detection Systems using Linear Discriminant Analysis and Logistic Regression"	30
3.4	Architettura di LeNet-5	31

3.5	Architettura del modello descritto nel paper "Network Traffic Anomaly Detection Using Recurrent Neural Networks"	32
3.6	Architettura del modello descritto nel paper "Network Traffic Classifier With Convolutional and Recurrent Neural Networks for Internet of Things"	33
3.7	Rilevamento delle anomalie descritto nel paper "Traffic Anomaly Detection Using K-Means Clustering"	34
4.1	Risultati delle prestazioni sul dataset NSL-KDD	40
4.2	Distribuzione classi in NSL-KDD	44
4.3	Distribuzione classi in NSL-KDD dopo undersampling	44
4.4	Distribuzione classi in CIC-IDS2017	44
4.5	Distribuzione classi in CIC-IDS2017 dopo undersampling	44
4.6	Matrice di confusione - classificazione binaria con Naive Bayes (NSL-KDD)	49
4.7	Matrice di confusione - classificazione binaria con Regressione Logistica (NSL-KDD)	51
4.8	Matrice di confusione - classificazione binaria con LDA (NSL-KDD) .	53
4.9	Matrice di confusione - classificazione binaria con Random Forest (NSL-KDD)	55
4.10	Matrice di confusione - classificazione binaria con MLP (NSL-KDD) .	56
4.11	Matrice di confusione - classificazione binaria con Naive Bayes (CIC-IDS2017)	59
4.12	Matrice di confusione - classificazione binaria con Regressione Logistica (CIC-IDS2017)	61
4.13	Matrice di confusione - classificazione binaria con LDA (CIC-IDS2017)	63
4.14	Matrice di confusione - classificazione binaria con Random Forest (CIC-IDS2017)	65
4.15	Matrice di confusione - classificazione binaria con MLP (CIC-IDS2017)	67
5.1	Esempio di codice python che implementa uno switch L2 con Ryu controller in un ambiente SDN	79
5.2	Esempio di schema di architettura IDS/SDN con classificatore [1]. .	83

Elenco delle tabelle

4.1	Risultati delle prestazioni sul dataset CIC-IDS2017	41
4.2	Report di classificazione binaria con Naive Bayes (NSL-KDD)	50
4.3	Report di classificazione binaria con cross-validation e Naive Bayes (NSL-KDD)	50
4.4	Report di classificazione multiclasse con cross-validation e Naive Bayes (NSL-KDD)	50
4.5	Report di classificazione binaria con Regressione Logistica (NSL-KDD)	51
4.6	Report di classificazione binaria con cross-validation e Regressione Logistica (NSL-KDD)	52
4.7	Report di classificazione multiclasse con cross-validation e Regressione Logistica (NSL-KDD)	52
4.8	Report di classificazione binaria con LDA (NSL-KDD)	53
4.9	Report di classificazione binaria con cross-validation e LDA (NSL-KDD)	53
4.10	Report di classificazione multiclasse con cross-validation e LDA (NSL-KDD)	54
4.11	Report di classificazione binaria con Random Forest (NSL-KDD) . . .	54
4.12	Report di classificazione binaria con cross-validation e Random Forest (NSL-KDD)	55

4.13 Report di classificazione multiclasse con cross-validation e Random Forest (NSL-KDD)	55
4.14 Report di classificazione binaria con MLP (NSL-KDD)	56
4.15 Report di classificazione binaria con cross-validation e MLP (NSL-KDD)	57
4.16 Report di classificazione multiclasse con cross-validation e MLP (NSL-KDD)	57
4.17 Report di classificazione binaria con Naive Bayes (CIC-IDS2017)	59
4.18 Report di classificazione binaria con cross-validation e Naive Bayes (CIC-IDS2017)	59
4.19 Report di classificazione multiclasse con cross-validation e Naive Bayes (CIC-IDS2017)	60
4.20 Report di classificazione binaria con Regressione Logistica (CIC-IDS2017)	61
4.21 Report di classificazione binaria con cross-validation e Regressione Logistica (CIC-IDS2017)	61
4.22 Report di classificazione multiclasse con cross-validation e Regressione Logistica (CIC-IDS2017)	62
4.23 Report di classificazione binaria con LDA (CIC-IDS2017)	63
4.24 Report di classificazione binaria con cross-validation e LDA (CIC-IDS2017)	63
4.25 Report di classificazione multiclasse con cross-validation e LDA (CIC-IDS2017)	64
4.26 Report di classificazione binaria con Random Forest (CIC-IDS2017)	65
4.27 Report di classificazione binaria con cross-validation e Random Forest (CIC-IDS2017)	65
4.28 Report di classificazione multiclasse con cross-validation e Random Forest (CIC-IDS2017)	66
4.29 Report di classificazione binaria con MLP (CIC-IDS2017)	67
4.30 Report di classificazione binaria con cross-validation e MLP (CIC-IDS2017)	67
4.31 Report di classificazione multiclasse con cross-validation e MLP (CIC-IDS2017)	68
4.32 Prime 5 feature per importanza (NSL-KDD)	71

4.33 Prime 5 feature per importanza (CIC-IDS2017)	71
4.34 Confronto delle accuratezze prima e dopo l'uso di XGBoost per NSL-KDD	72
4.35 Confronto delle accuratezze prima e dopo l'uso di XGBoost per CIC-IDS2017	72
4.36 Risultati della Grid Search per NSL-KDD (in grassetto i parametri con le prestazioni migliori)	75
4.37 Risultati della Grid Search per CIC-IDS2017 (in grassetto i parametri con le prestazioni migliori)	76

CAPITOLO 1

Introduzione

1.1 L'importanza della sicurezza informatica in azienda

In un panorama digitale sempre più complesso, la sicurezza informatica riveste un'importanza cruciale per le aziende. Quest'ultime trattano una quantità sempre maggiore di dati sensibili come: dati personali dei clienti, informazioni finanziarie e proprietà intellettuale. La perdita o il furto di questi dati possono avere conseguenze devastanti tra cui il furto di identità e frodi finanziarie. Inoltre, le aziende dipendono sempre più dai sistemi informatici per le loro operazioni quotidiane. Le violazioni di questi sistemi possono avere costi finanziari enormi, compresi i costi di riparazione dei danni, le perdite di reddito dovute a interruzioni delle attività e le spese legali associate alle azioni correttive. Tuttavia, gli impatti non si limitano solo ai costi finanziari: le violazioni della sicurezza possono anche danneggiare irrimediabilmente la reputazione aziendale, minando la fiducia dei clienti e degli investitori e compromettendo la posizione competitiva dell'azienda sul mercato.

Come evidenziato dal "*Rapporto Clusit 2024 sulla sicurezza ICT in Italia*"¹ tra il 2019 e il 2023, si è verificato un significativo aumento degli attacchi informatici. Questi

¹Clusit – Associazione Italiana per la Sicurezza Informatica: <https://clusit.it>

dati indicano un aumento numerico del 60%, passando da 1.667 attacchi nel 2019 a 2.779 nel 2023 (**Fig. 1.1**).

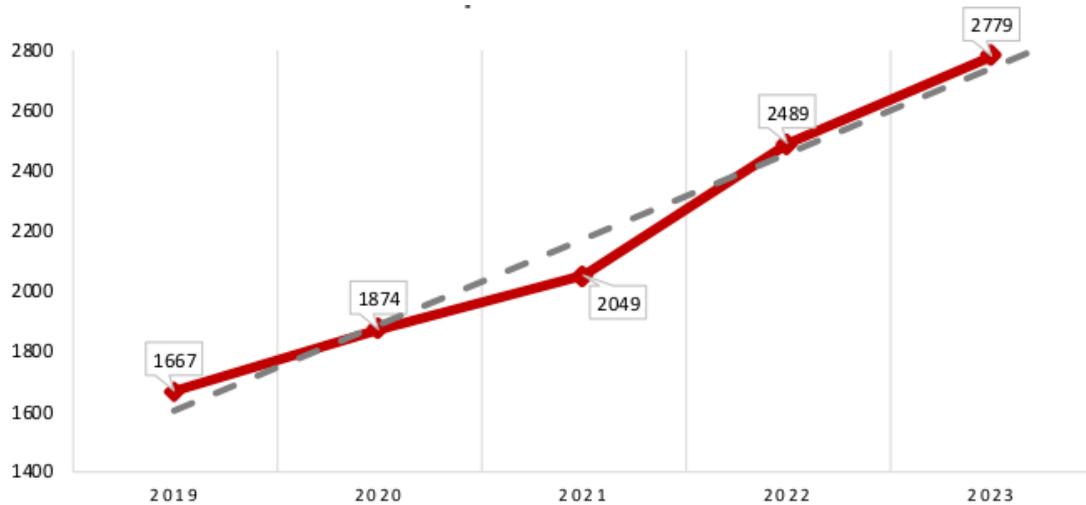


Figura 1.1: Attacchi per anno 2019-2023

L’aumento di questi attacchi potrebbe essere attribuito a diversi fattori. La crescente dipendenza dalla tecnologia ha incrementato la superficie di attacco dei sistemi informatici, mentre gli aggressori informatici hanno adottato tattiche sempre più sofisticate. Le organizzazioni potrebbero non essere adeguatamente preparate a difendersi, a causa di vulnerabilità nel software e di una mancanza di risorse o competenze. Inoltre, una maggiore consapevolezza dei rischi informatici potrebbe aver portato a una più ampia segnalazione degli incidenti. Infine, la crescente diffusione di minacce informatiche è probabilmente accentuata anche dallo scacchiere geopolitico attuale che è marcato dai conflitti tra Russia-Ucraina e Israele-Hamas. Tali minacce sfruttano opportunisticamente le instabilità politiche per colpire infrastrutture critiche, come sistemi finanziari, reti elettriche e settore sanitario.

Un altro dato interessante è il fatto che, in un anno, gli attacchi sono aumentati dell’11% al livello globale, mentre solo in Italia si è registrato un incremento del 65% (**Fig. 1.2**).

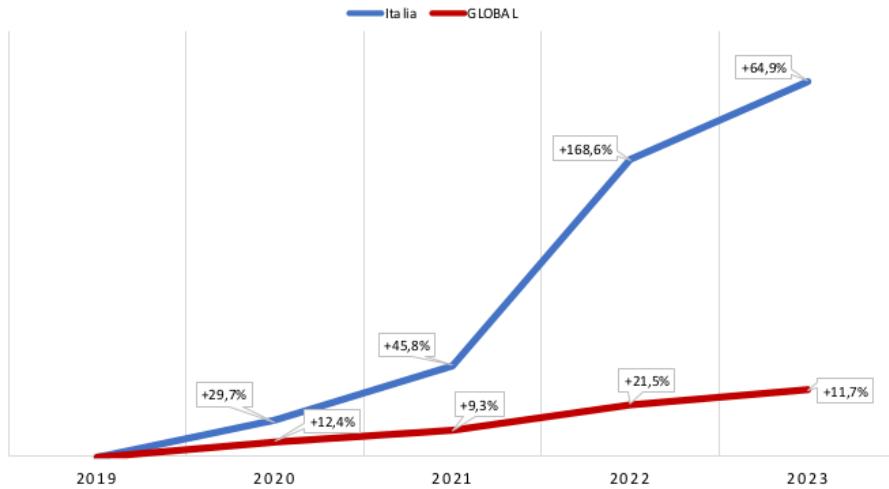


Figura 1.2: Confronto crescita % Italia Vs Global

Dunque, in Italia si osserva un aumento degli attacchi a un tasso notevolmente superiore, il che potrebbe indicare sia una tendenza dei cybercriminali a prendere di mira le vittime italiane, sia, più verosimilmente, una carenza nella capacità di difesa delle stesse vittime.

Non solo la frequenza degli attacchi è aumentata, ma anche la gravità degli stessi è peggiorata nel corso degli anni. Nel 2023, gli attacchi classificati come "critici" o "gravi" rappresentano l'80% del totale, rispetto al 47% registrato nel 2019 (Fig. 1.3).

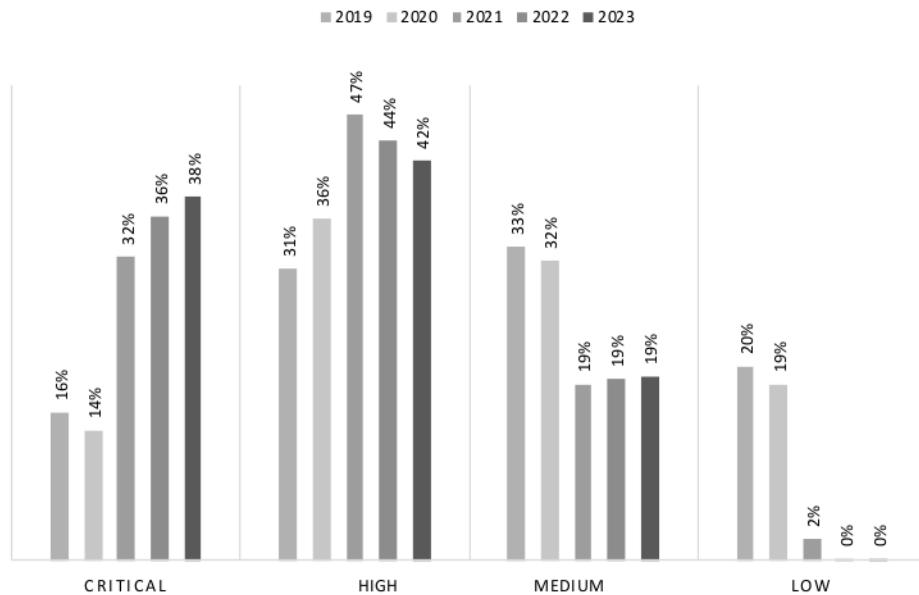


Figura 1.3: Severity % 2019-2023

Questi dati mettono in luce una realtà che non possiamo più ignorare: il mondo digitale è diventato sempre più vulnerabile agli attacchi informatici. Tuttavia, dietro a queste cifre allarmanti c'è anche un'opportunità. Cioè quella di rafforzare le nostre difese e proteggere le nostre risorse digitali in modo più efficace.

1.2 Breve panoramica sui rischi e le minacce attuali

Le minacce alla sicurezza informatica sono in costante evoluzione, spinte dall'avanzamento tecnologico e dalla creatività dei criminali informatici.

Secondo i dati del *Rapport Clusit*², nel 2023 il **malware** è rimasto la scelta preferita, colpendo il 36% delle volte (**Fig. 1.4**). Questo termine si riferisce a software dannoso che può assumere varie forme, ma il ransomware è particolarmente diffuso e remunerativo per i suoi autori. Questo tipo di malware cifra i dati dell'utente o dell'azienda e richiede un pagamento per il ripristino, garantendo così un profitto rapido e significativo. In secondo luogo, c'è lo sfruttamento delle **vulnerabilità**, coinvolgendo il 18% degli attacchi. Questo approccio può riguardare sia vulnerabilità già note che vulnerabilità sconosciute, come i cosiddetti zero-day, appena state scoperte e non hanno soluzioni di sicurezza disponibili. Le **tecniche sconosciute** rappresentano un'altra minaccia significativa, rappresentando circa il 20% degli attacchi analizzati. Queste tecniche, di cui non si conoscono pubblicamente i dettagli specifici, possono rendere difficile per le aziende rilevare e rispondere prontamente agli attacchi. Altre strategie diffuse includono: il **phishing**, che sfrutta l'ingegneria sociale per ingannare le persone; gli attacchi **DDoS** (Distributed Denial of Service), che sovraccaricano i server di destinazione con un eccesso di richieste; gli **attacchi Web**; l'**identity theft**, il processo attraverso il quale si ottengono illegalmente informazioni personali di un'altra persona al fine di assumere la sua identità e commettere frodi e accessi non autorizzati; l'**account cracking**, quando si ottiene accesso non autorizzato ad un account principalmente utilizzando tecniche di forza bruta per ottenere l'accesso.

²Clusit – Associazione Italiana per la Sicurezza Informatica: <https://clusit.it>

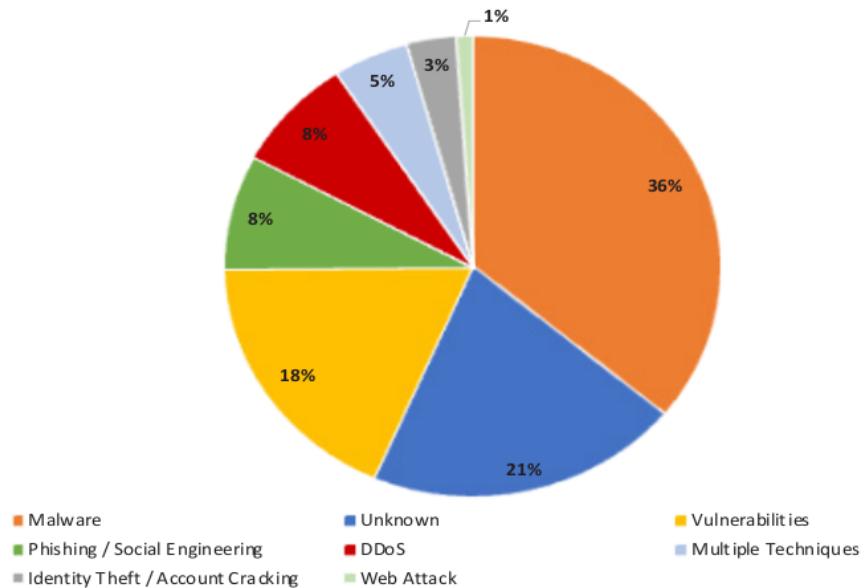


Figura 1.4: Tecniche di attacco nel 2023

Oltre alle tecniche già citate, è interessante notare l'adozione di metodi sempre più sofisticati. Evidenziati anche dal *Global Threat Report 2024* di CrowdStrike³. Tra questi metodi, l'**IA generativa** è sempre più utilizzata per sfruttare le debolezze umane attraverso l'ingegneria sociale. Gli attaccanti possono creare messaggi e contenuti falsi altamente persuasivi per indurre le vittime a compiere azioni dannose, come cliccare su link malevoli o rilasciare informazioni sensibili. L'IA viene impiegata anche per creare malware avanzati e individuare automaticamente vulnerabilità nei sistemi informatici. Un altro metodo in crescita è il **Ransomware as a Service (RaaS)**, che consente agli aspiranti criminali informatici di noleggiare o acquistare ransomware preconfezionati e strumenti di attacco da sviluppatori esperti, abbassando così la barriera all'ingresso nel mondo del ransomware e permettendo a più individui di lanciare attacchi senza necessariamente avere competenze tecniche avanzate. Gli **access brokers**, invece, sono individui o gruppi che si specializzano nel commercio di accesso non autorizzato a reti informatiche, account utente e sistemi. Questo accesso può essere ottenuto tramite violazioni dei dati o altre forme di compromissione. Inoltre, gli **attacchi ai fornitori di terze parti** stanno diventando sempre più frequenti. Gli attaccanti mirano ai fornitori di servizi di terze parti che hanno accesso a una vasta rete di clienti: compromettendo questi fornitori, gli attaccanti possono ottenere acces-

³Fonte: <https://www.crowdstrike.com/global-threat-report/>

so a numerosi sistemi e dati sensibili, amplificando così l'impatto dei loro attacchi. In concomitanza con la diffusione sempre più capillare del **cloud**, gli attaccanti sono sempre più consapevoli dell'importanza di questi servizi e dell'enorme quantità di dati sensibili accessibili attraverso di essi. Questa consapevolezza li spinge a mirare a piattaforme come Microsoft 365, SharePoint e repository di codice per raccogliere informazioni proprietarie, credenziali utente e altro ancora, aumentando il potenziale danno.

L'attuale panorama delle minacce informatiche è, dunque, caratterizzato da una vasta gamma di tecniche utilizzate dai criminali informatici per compromettere la sicurezza delle reti aziendali (e non solo) e trarne vantaggio finanziario. È fondamentale che le aziende comprendano queste minacce per implementare strategie di difesa efficaci e proteggere i propri dati e la propria reputazione.

1.3 Principali Cyber-Gang

La crescita esponenziale della criminalità informatica negli ultimi anni è stata caratterizzata dall'attività di gruppi sempre più sofisticati che sfruttano eventi globali per raggiungere i propri obiettivi. Il picco attuale ha avuto inizio durante la pandemia da COVID-19, quando si è verificato un massiccio spostamento dei dipendenti al lavoro da casa, esponendo molti di loro a dispositivi personali privi di adeguate protezioni informatiche contro malware e phishing. Più recentemente, l'invasione russa dell'Ucraina ha scatenato una guerra informatica parallela, con attacchi mirati da parte di gruppi russi contro infrastrutture governative e commerciali ucraine. In questo contesto, si presenta un elenco dei cinque gruppi più pericolosi attualmente attivi secondo *jamcyber.com*⁴:

- **DarkSide:** con base in Russia, è noto per offrire "ransomware-as-a-service": un modello di business illegale che consiste nel vendere il proprio codice ad altri hacker con il quale a loro volta mettono in atto attacchi ransomware. Si concentra principalmente su grandi aziende, poiché sono più propense a

⁴Fonte: <https://jamcyber.com/blog/cyber-insights/cyber-crimes-gangs/>

pagare grandi riscatti. Infatti, hanno acquisito notorietà per il loro attacco di ransomware alla Colonial Pipeline negli Stati Uniti nel maggio 2021.

- **Evil Corp:** è un gruppo criminale ben consolidato attivo dal 2009, con sede in Russia. Si stima che i leader del gruppo vivano uno stile di vita da milionari grazie all'estorsione di oltre 100 milioni di dollari da vittime corporate. Utilizzano una vasta gamma di tecniche, tra cui virus macro, email di phishing con foto malevole allegate e attacchi di tipo trojan.
- **LockBit 3.0:** altamente attiva e efficiente, è famosa per il furto di 78GB di dati dall'Agenzia delle Entrate italiana e per aver richiesto un riscatto per la restituzione sicura dei dati. Utilizzano Ransomware-as-a-Service come principale strategia di attacco, ma si distinguono anche per l'offerta di ricompense monetarie a chiunque trovi vulnerabilità nei sistemi o informazioni sui loro obiettivi.
- **Lapsus\$:** un gruppo relativamente nuovo nel mercato della criminalità informatica che si concentra principalmente sul furto di dati e sull'estorsione. Nonostante la relativa mancanza di organizzazione rispetto ad altri gruppi più esperti, hanno guadagnato notorietà per aver compromesso le reti di Microsoft, Samsung e Nvidia. Anche se alcuni leader sono stati arrestati, il gruppo ha continuato le sue attività.
- **FIN7:** operativo dal 2012, è noto per essere estremamente sofisticato e operare come un'impresa. Si concentrano principalmente sul furto di dati di carte di credito dai siti di e-commerce. Si stima che abbiano accumulato miliardi di dollari attraverso le loro attività criminali. Hanno ottenuto notorietà per campagne di spear phishing ai danni della SEC degli Stati Uniti.

Ciascuna di queste gang ha le proprie caratteristiche distintive e strategie di attacco, e rappresentano una seria minaccia per le aziende e le istituzioni di tutto il mondo.

CAPITOLO 2

Analisi del traffico di rete

2.1 Intrusion Detection Systems (IDS)

2.1.1 Definizione di IDS e Scopo

Un sistema di rilevamento delle intrusioni (IDS) è un componente hardware e/o software progettato per monitorare e analizzare il traffico di rete o le attività del sistema al fine di identificare e rispondere a potenziali intrusioni o violazioni della sicurezza.

In particolare, un IDS analizza i pacchetti di dati in arrivo e in uscita e i log di sistema e, utilizzando regole predefinite o modelli di comportamento basati sull'apprendimento automatico, identifica pattern di traffico insoliti che potrebbero indicare una potenziale violazione della sicurezza. Questo può includere tentativi di accesso non autorizzati, attacchi di tipo malware o exploit di vulnerabilità. Una volta rilevata un'attività sospetta, l'IDS genera un allarme per avvisare gli amministratori di sistema o i team di sicurezza della potenziale minaccia. Questo avviso può includere dettagli come l'indirizzo IP dell'origine dell'attacco, il tipo di attacco e altre informazioni rilevanti. Gli amministratori possono quindi esaminare i dettagli dell'evento segnalato per determinare la natura della minaccia e avviare delle contro-

misure necessarie per mitigare l’attacco in corso o prevenire future intrusioni. Questo potrebbe includere l’isolamento di dispositivi compromessi, l’applicazione di patch di sicurezza, la modifica delle politiche di accesso o altre azioni correttive.

2.1.2 Tipologie di IDS

Esistono diverse tipologie di IDS perché affrontano minacce e vulnerabilità in modi diversi, e ognuna ha i suoi punti di forza e di debolezza.

IDS basati sulla rete (NIDS) sono efficaci nel rilevare attacchi che transitano attraverso la rete, come scansioni di porte, tentativi di intrusioni e attacchi DDoS. È particolarmente utile per monitorare il traffico in tempo reale su reti di grandi dimensioni.

IDS basati sull’host (HIDS) forniscono una visibilità più dettagliata sull’attività dei singoli host e possono rilevare minacce che sfuggono alla rilevazione a livello di rete, come attacchi locali o malware che agiscono direttamente sul sistema. Tuttavia, possono essere limitati dalla necessità di essere installati e gestiti su ogni singolo host.

IDS ibridi utilizzando entrambe le prospettive, a livello di rete e a livello di host, possono rilevare una gamma più ampia di attacchi e fornire una maggiore contextualizzazione delle minacce. Tuttavia, possono richiedere una maggiore complessità nell’implementazione e nella gestione.

Non c’è una soluzione “migliore” in assoluto, poiché dipende dalle esigenze specifiche di sicurezza di un’organizzazione, dalle sue risorse e dall’ambiente IT. Spesso, la migliore strategia è utilizzare una combinazione di diverse tipologie di IDS, integrandole con altre misure di sicurezza, come firewall, sistemi di prevenzione delle intrusioni e monitoraggio dei log, per ottenere una difesa più completa contro le minacce informatiche.

2.2 Intrusion Prevention Systems (IPS)

2.2.1 Definizione di IPS e differenze rispetto agli IDS

Un sistema di prevenzione delle intrusioni (IPS) è un componente software progettato per identificare e bloccare minacce informatiche in tempo reale. Funziona monitorando il traffico di rete in cerca di comportamenti sospetti o anomalie che potrebbero indicare un attacco in corso. Una volta rilevato un potenziale attacco, l'IPS può intraprendere diverse azioni per proteggere il sistema, ad esempio bloccare il traffico proveniente dall'indirizzo IP sospetto, bloccare l'accesso a determinati servizi o inviare avvisi agli amministratori di sistema.

Rispetto ad un IDS, un IPS va oltre la semplice rilevazione: funziona in modo proattivo per impedire che le minacce raggiungano il loro obiettivo. Invece di generare solo avvisi, un IPS è in grado di bloccare automaticamente il traffico dannoso o sospetto, adottando azioni come la modifica delle regole del firewall o il blocco di pacchetti dannosi.

2.2.2 Ruolo degli IPS nella prevenzione degli attacchi

Gli IPS svolgono un ruolo cruciale nella prevenzione degli attacchi informatici agendo come uno scudo proattivo per le reti.

L'efficacia degli IPS deriva dalla loro capacità di rispondere rapidamente agli attacchi, spesso in tempo reale, e dalla loro capacità di adattarsi e apprendere nuovi modelli di minaccia. Questi sistemi sono spesso supportati da database di firme e algoritmi di intelligenza artificiale che consentono loro di riconoscere e neutralizzare le minacce in evoluzione.

I database di firme, una componente fondamentale degli IPS (nonché degli IDS), contengono pattern predefiniti che identificano specifiche minacce o tipologie di attacchi noti. Quando il traffico di rete viene analizzato, gli IPS confrontano i pacchetti di dati con queste firme per rilevare e bloccare attacchi già documentati e conosciuti.

Tuttavia, poiché le minacce informatiche sono in costante evoluzione e possono assumere forme nuove e sofisticate, gli IPS si affidano anche agli algoritmi di intelligenza artificiale per rilevare modelli anomali e comportamenti potenzialmente

dannosi che potrebbero non essere stati precedentemente identificati e catalogati nei database di firme.

Questa capacità di adattamento e apprendimento continuo permette agli IPS di rimanere efficaci nel proteggere le reti da una vasta gamma di attacchi, contribuendo così a mantenere la sicurezza dei sistemi informatici.

Questa flessibilità ha, però, un costo. Gli IPS basati sull'anomalia possono generare un numero significativo di falsi positivi, ossia avvisi che segnalano attività apparentemente sospette che in realtà sono legittime. Quest'ultimi possono sovraccaricare gli amministratori di sistema, causando la cosiddetta "alert fatigue" e potenzialmente portando a una minore reattività a veri incidenti di sicurezza. In altre parole, potrebbero diventare meno attenti agli avvisi se devono costantemente distinguere tra falsi allarmi e reali minacce.

Nonostante questo, è generalmente preferibile avere falsi positivi in un sistema di sicurezza. Un sistema che genera falsi positivi è vigile e attento a ogni possibile anomalia, garantendo un livello più alto di protezione. Gli amministratori, con il tempo, possono affinare le configurazioni e gli algoritmi degli IPS per ridurre il numero di falsi positivi, migliorando l'efficacia del sistema senza compromettere la sicurezza.

2.2.3 Architettura e funzionamento degli IPS

L'architettura di un IPS può variare leggermente a seconda del fornitore e delle specifiche esigenze dell'ambiente di rete, ma in generale includono dei componenti principali.

Un componente essenziale sono i **sensori di rete**, i quali sono distribuiti strategicamente lungo la rete e monitorano costantemente il traffico di rete in ingresso e in uscita. Possono essere implementati come dispositivi hardware dedicati o come software che viene eseguito su server all'interno della rete. Il **motore di rilevamento delle intrusioni**, invece, è responsabile dell'analisi del traffico di rete per identificare potenziali intrusioni o comportamenti sospetti, utilizzando una varietà di tecniche come firme di attacco, euristica e analisi comportamentale per rilevare le minacce. Quest'ultimo utilizza una vasta libreria di firme di attacco note, contenuta nel **database**.

se delle firme, per confrontare il traffico di rete in tempo reale e identificare eventuali corrispondenze. Le **policy di sicurezza** definiscono le regole e le politiche che determinano come il sistema di prevenzione delle intrusioni risponde alle minacce rilevate. Queste politiche possono essere personalizzate per adattarsi alle esigenze specifiche dell’organizzazione e possono includere azioni come il blocco del traffico sospetto, la registrazione degli eventi e la notifica agli amministratori di sistema. Un altro componente importante è l’**interfaccia di amministrazione** che fornisce agli amministratori un’interfaccia utente per configurare e gestire il sistema. Attraverso di essa, si possono visualizzare i report di sicurezza, aggiornare le firme di attacco, modificare le politiche di sicurezza e monitorare lo stato del sistema. Inoltre, è fondamentale che l’IPS registri tutte le attività rilevanti, incluse le minacce individuate e le azioni intraprese, attraverso il **logging**. Questi log possono essere utilizzati per l’analisi forense, la conformità normativa e il miglioramento della sicurezza complessiva. Infine, un IPS può essere integrato con altri componenti del sistema di sicurezza, come i firewall e i sistemi di rilevamento delle intrusioni (IDS), per fornire una difesa multi-livello contro le minacce informatiche.

2.2.4 Suricata - Applicazioni reali

Sebbene nei capitoli precedenti abbiamo parlato di IDS e IPS come se fossero due entità a sé stanti, nella pratica, molti sistemi di sicurezza integrano sia funzionalità di rilevamento che di prevenzione, rendendo la loro distinzione meno significativa. Gli strumenti moderni sono progettati per fornire una protezione completa e reattiva, combinando il monitoraggio continuo con la capacità di risposta automatizzata. Questa integrazione è cruciale per affrontare le minacce avanzate che richiedono tempi di risposta minimo.

Suricata è uno degli esempi più avanzati e flessibili disponibili oggi. Sviluppato dall’Open Information Security Foundation (OISF), è un motore IDS/IPS open source che offre capacità di rilevamento delle intrusioni, prevenzione e analisi approfondita del traffico di rete. Grazie alla sua architettura modulare e alla capacità di gestire grandi volumi di dati, Suricata si è affermato come una scelta popolare sia per piccole reti che per grandi infrastrutture aziendali.

Una delle caratteristiche distintive è la sua capacità di analizzare in modo dettagliato il traffico di rete, utilizzando un motore di rilevamento basato su regole simile a quello di Snort, ma con significative migliorie in termini di performance e scalabilità. Suricata supporta il multi-threading, il che gli consente di sfruttare appieno l'hardware per processare il traffico di rete a velocità molto elevate. Questo è particolarmente importante in ambienti ad alto traffico, dove la capacità di analizzare e rispondere rapidamente agli eventi di sicurezza è cruciale.

Suricata non si limita a rilevare minacce: può essere configurato per prevenire attacchi attuando misure di difesa automatica. Ad esempio, può bloccare il traffico proveniente da indirizzi IP malevoli identificati o interrompere connessioni che presentano comportamenti anomali.

Un'altra funzionalità importante è il suo supporto per numerosi formati di log e protocolli, che consente di integrarlo facilmente con altri strumenti di sicurezza e piattaforme di gestione dei log. Può esportare dati in formati come JSON, YAML e CSV, facilitando l'analisi e la correlazione degli eventi di sicurezza con altri dati raccolti nella rete. Infatti, è compatibile con varie piattaforme SIEM (Security Information and Event Management), che permettono una gestione centralizzata degli eventi di sicurezza e una risposta coordinata agli incidenti (vedi l'integrazione con Splunk Enterprise, Figura 2.2).

L'aspetto open source, inoltre, ne favorisce l'adozione e l'adattamento in una vasta gamma di scenari. La community di sviluppatori contribuisce costantemente al miglioramento del codice e alla creazione di nuove regole di rilevamento, rendendolo un progetto in continua evoluzione.

In conclusione, Suricata rappresenta un esempio emblematico di come gli strumenti IDS e IPS possano fondersi in un'unica soluzione robusta e versatile. La sua capacità di rilevare e prevenire attacchi, unita alla scalabilità e alla flessibilità di integrazione, lo rende uno strumento prezioso.

2.2 – Intrusion Prevention Systems (IPS)

Figura 2.1: Eve, la principale dashboard di logging di Suricata

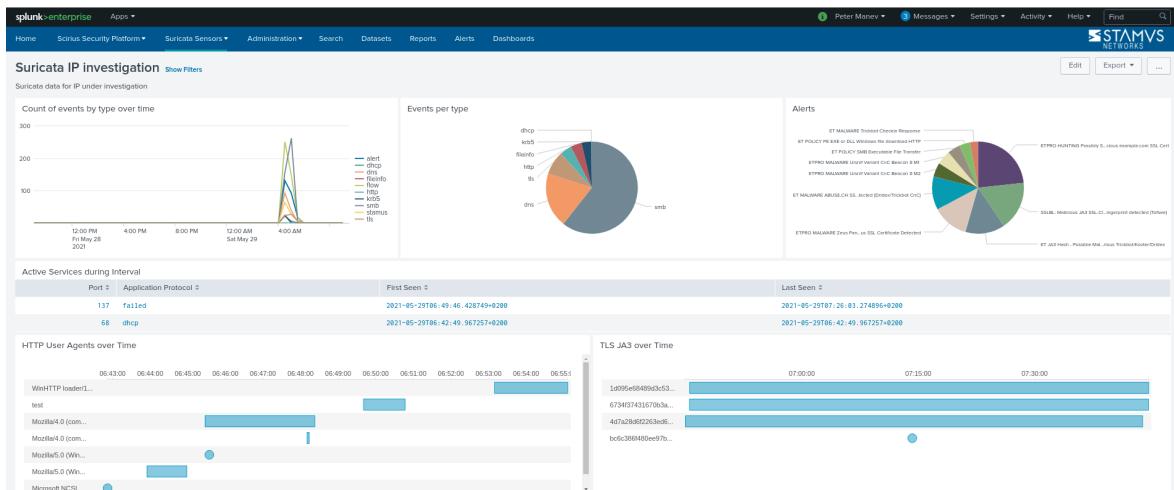


Figura 2.2: Integrazione con Splunk Enterprise

2.3 Next Generation Firewall (NGFW)

2.3.1 Definizione di NGFW e differenze rispetto ai firewall tradizionali

Un Next Generation Firewall (NGFW) è una evoluzione dei firewall tradizionali progettata per affrontare le sfide di sicurezza più complesse che emergono nel contesto delle reti moderne. A differenza dei firewall tradizionali, che si limitano principalmente a filtrare il traffico di rete basandosi su regole di indirizzo IP e porte, i NGFW incorporano funzionalità avanzate di sicurezza e analisi del traffico. Infatti, integrano tecniche di analisi dei contenuti e del comportamento delle applicazioni per identificare e bloccare minacce informatiche più sofisticate, come malware, attacchi di phishing e intrusioni avanzate. Utilizzano spesso metodi di rilevamento e prevenzione delle intrusioni (IDS/IPS), sandboxing per isolare e analizzare file sospetti, filtraggio dei contenuti web, e controlli più avanzati sull'accesso alle applicazioni e sui dati. Inoltre, spesso supportano funzionalità di gestione centralizzata e di reporting avanzato, consentendo agli amministratori di monitorare e gestire in modo più efficace la sicurezza della rete.

Dunque, la differenza principale rispetto ai firewall tradizionali sta nella capacità dei NGFW di adottare un approccio più intelligente e completo alla protezione della rete, tenendo conto non solo di informazioni di base, ma anche del contenuto e del comportamento delle applicazioni.

2.3.2 Funzionalità avanzate dei NGFW

I Next-Generation Firewall offrono una serie di funzionalità avanzate che vanno oltre le capacità dei firewall tradizionali. Una caratteristica importante, nota come **application-awareness**, è la capacità di riconoscere il traffico di rete in base alle applicazioni specifiche che lo generano. Questo va oltre la semplice identificazione dei protocolli di rete, consentendo di applicare politiche di sicurezza più granulari basate sulle applicazioni stesse. I NGFW includono anche il **content filtering**, che permette di bloccare o consentire l'accesso a determinati tipi di contenuti in base a criteri specifici. Ad esempio, è possibile filtrare l'accesso a siti web in base alla categoria

(adulti, social media, giochi, ecc.) oppure bloccare specifici tipi di file (come i file eseguibili) per mitigare il rischio di malware. Un'altra funzione cruciale è l'**intrusion prevention**, attraverso la quale sono in grado di rilevare e prevenire intrusioni nella rete. Monitorano il traffico in tempo reale e lo confrontano con firme e pattern noti di attacchi informatici, utilizzando tecniche come la rilevazione di anomalie per identificare comportamenti sospetti e bloccare il traffico dannoso prima che raggiunga la rete interna. La **deep packet inspection** (DPI), invece, è una capacità avanzata, che consente ai NGFW di analizzare il contenuto dei pacchetti di dati in modo dettagliato, incluso il loro payload. Il **policy-based routing** è un'altra funzionalità che consente di instradare il traffico di rete in base a politiche specifiche. Ad esempio, si può inviare il traffico web attraverso un servizio di proxy per il filtraggio dei contenuti o instradare il traffico verso un sistema di rilevamento delle minacce per l'ispezione approfondita. È degno di nota che il fattore che amplia ulteriormente le capacità di protezione e rilevamento delle minacce della rete è l'**integrazione** con servizi di sicurezza esterni, come sistemi di gestione delle minacce, sandboxing per l'analisi avanzata del malware e servizi di intelligence. Infine, la **threat intelligence** è il processo mediante il quale si ottengono e si utilizzano informazioni aggiornate sulle minacce informatiche da varie fonti. Questo approccio dinamico consente di apportare regolari aggiornamenti alle politiche di sicurezza in tempo reale, in risposta alle minacce emergenti.

Tutte queste funzionalità avanzate consentono ai NGFW di fornire una protezione più efficace contro una vasta gamma di minacce informatiche, garantendo nel contempo una maggiore visibilità e controllo sul traffico di rete.

2.3.3 Altri tipi di firewall

Host Firewall

I firewall basati su host controllano il traffico entrante e uscente su singoli dispositivi.

Un vantaggio significativo di questa tipologia è la loro capacità di essere configurati in modo specifico per le esigenze di ciascuna macchina, consentendo una gestione granulare delle regole di sicurezza. Tuttavia, questa granularità comporta an-

che una gestione complessa, poiché richiede che ogni dispositivo venga configurato individualmente.

Inoltre, i firewall basati su host possono consumare risorse significative del sistema su cui operano, riducendo le prestazioni del dispositivo.

Network Firewall

I network firewall operano a livello di rete, monitorando e controllando il traffico tra diverse reti o tra una rete interna e il mondo esterno.

Uno dei principali vantaggi è la loro capacità di proteggere un'intera rete con una singola implementazione. Questo approccio centralizzato semplifica la gestione delle regole di sicurezza, poiché le modifiche devono essere apportate solo in un unico punto invece che su ciascun dispositivo della rete. Inoltre, sono generalmente più potenti in termini di capacità di elaborazione e gestione del traffico rispetto ai firewall basati su host, il che li rende più adatti a reti di grandi dimensioni con elevati volumi di traffico.

Tuttavia, i network firewall presentano anche alcuni svantaggi. La centralizzazione della sicurezza può creare un punto di fallimento unico, rendendo la rete vulnerabile in caso di malfunzionamento. Inoltre, l'acquisizione e la manutenzione possono comportare costi elevati, sia in termini di hardware che di competenze necessarie per la loro configurazione e gestione.

Application Firewall

Gli application firewall operano ad un livello più profondo, esaminando i dati specifici delle applicazioni. Sono in grado di riconoscere protocolli applicativi specifici e bloccare o permettere il traffico in base alle regole definite.

Un esempio calzante sono i Web Application Firewall (WAF) che operano a livello applicativo e sono progettati per monitorare e filtrare il traffico HTTP(S). Essi offrono una protezione più granulare e dettagliata, esaminando il contenuto delle richieste e delle risposte tra il client e l'applicazione web. Questa capacità consente di identificare e bloccare attacchi specifici come gli attacchi di tipo SQL injection, XSS e SSRF.

Tra i WAF più noti c'è ModSecurity, un modulo open-source per server web come Apache, che fornisce funzionalità di rilevamento delle intrusioni e prevenzione, permettendo di scrivere regole personalizzate per proteggere le applicazioni.

Tuttavia, i firewall applicativi presentano anche alcuni svantaggi. La configurazione e la gestione possono essere complesse e richiedono competenze specialistiche. Se non configurati correttamente, infatti, possono generare facilmente falsi positivi, bloccando traffico legittimo e influenzando negativamente l'esperienza dell'utente. Inoltre, poiché analizzano il contenuto delle comunicazioni a livello applicativo, possono introdurre latenze aggiuntive, rallentando le prestazioni del sistema.

Circuit-level Firewall

Un circuit-level firewall opera al livello di sessione del modello OSI, diversamente dai firewall classici che operano principalmente ai livelli di rete o di trasporto.

A differenza dei firewall tradizionali, non esaminano il contenuto dei pacchetti individuali. Invece, essi permettono o bloccano il traffico basandosi sulla connessione stabilita tra il client e il server. Quando una connessione viene stabilita, il firewall crea un circuito virtuale attraverso il quale il traffico può fluire senza ulteriori controlli dettagliati sui pacchetti. Questo approccio ha il vantaggio di essere meno intensivo dal punto di vista computazionale rendendoli più efficienti in termini di prestazioni quando si tratta di gestire un alto volume di traffico.

Infatti, uno dei principali vantaggi è proprio la loro capacità di fornire un livello di sicurezza contro le connessioni non autorizzate senza necessitare di risorse eccessive per l'ispezione del contenuto. Ad esempio, possono essere configurati per bloccare connessioni provenienti da indirizzi IP non affidabili o da sessioni che non seguono un pattern di comportamento atteso.

Ciononostante, la loro mancanza di ispezione dettagliata del contenuto dei pacchetti significa che non possono rilevare e bloccare attacchi nascosti all'interno del traffico legittimo. Per esempio, un attacco basato su payload maligni che viaggia all'interno di una connessione apparentemente legittima potrebbe passare inosservato, poiché il firewall non analizza i dati trasportati una volta stabilita la connessione.

Stateless/Stateful Firewall

Un firewall stateless si limita ad analizzare i pacchetti di dati individualmente, basandosi su regole predefinite per decidere se consentire o bloccare ciascun pacchetto.

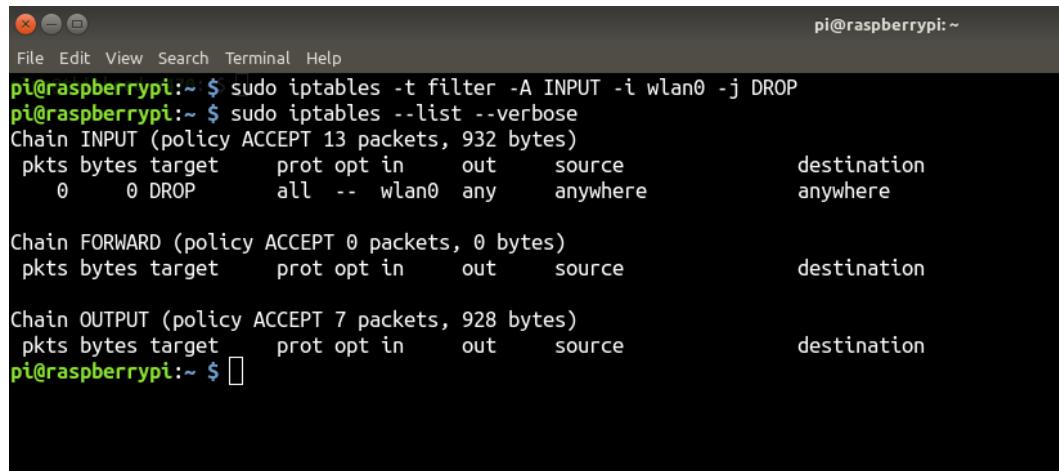
Questa caratteristica lo rende particolarmente semplice e veloce, poiché non deve mantenere alcuna informazione sullo stato delle connessioni in corso. Essendo meno complessi, questi firewall possono processare pacchetti a una velocità maggiore e con un minor utilizzo di memoria e potenza di calcolo. Questo li rende ideali per applicazioni in cui le risorse sono limitate o dove la velocità di elaborazione è critica.

Un esempio pratico di utilizzo di un firewall stateless è in ambienti con dispositivi a risorse limitate, come i dispositivi IoT. In tali contesti, l'efficienza e la rapidità di elaborazione dei pacchetti sono essenziali, e la loro semplicità può garantire una protezione di base senza compromettere le prestazioni del dispositivo.

D'altra parte, l'approccio stateless presenta anche significativi svantaggi. La mancanza di consapevolezza dello stato delle connessioni implica che questi firewall non possono rilevare attacchi più sofisticati che sfruttano le caratteristiche delle connessioni di rete, come gli attacchi a iniezione di pacchetti o quelli di session hijacking.

Un esempio di implementazione reale di questo tipo di firewall è iptables in Linux (quando configurato in modalità stateless). Infatti, iptables può essere configurato per applicare regole di filtraggio in base alle caratteristiche di ciascun pacchetto, come l'indirizzo IP di origine e di destinazione, la porta di origine e di destinazione, il protocollo utilizzato, e così via, senza mantenere alcuna informazione di stato (Figura 2.3).

Contrariamente, i firewall stateful monitorano e mantengono informazioni sullo stato di ogni connessione di rete, analizzando il contesto del singolo pacchetto. Questo permette loro di rilevare e prevenire tentativi di intrusione più avanzati, offrendo un livello di sicurezza superiore. Questa maggiore protezione però comporta un costo in termini di complessità e consumo di risorse.



```
pi@raspberrypi:~ $ sudo iptables -t filter -A INPUT -i wlan0 -j DROP
pi@raspberrypi:~ $ sudo iptables --list --verbose
Chain INPUT (policy ACCEPT 13 packets, 932 bytes)
 pkts bytes target     prot opt in     out     source               destination
      0   0 DROP        all    --  wlan0  any     anywhere            anywhere
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
Chain OUTPUT (policy ACCEPT 7 packets, 928 bytes)
 pkts bytes target     prot opt in     out     source               destination
pi@raspberrypi:~ $
```

Figura 2.3: Applicazione di una regola statica in iptables per bloccare il traffico in arrivo sull’interfaccia wlan0

Cloud Firewall

Questo tipo di firewall è una soluzione basata su software che sfrutta la potenza e la flessibilità del cloud computing per offrire protezione alle risorse digitali.

Uno dei principali vantaggi dei cloud firewall è la scalabilità. Essi possono essere facilmente adattati alle esigenze crescenti di un’azienda senza richiedere interventi hardware costosi. La gestione centralizzata attraverso piattaforme cloud permette un aggiornamento continuo e automatico delle definizioni di sicurezza, assicurando che la protezione sia sempre aggiornata contro le minacce più recenti. Inoltre, la configurazione e la gestione remota riducono significativamente i costi operativi e semplificano l’implementazione in ambienti distribuiti.

Tuttavia, i cloud firewall presentano anche alcuni svantaggi. La dipendenza dalla connettività internet può rappresentare un punto debole, poiché un’interruzione della connessione potrebbe compromettere la capacità di proteggere le risorse. Inoltre, poiché i dati attraversano l’infrastruttura cloud del provider del servizio, vi è una maggiore preoccupazione riguardo alla privacy e alla sicurezza dei dati sensibili. È essenziale che i fornitori di cloud firewall offrano garanzie robuste in termini di crittografia e conformità alle normative sulla protezione dei dati.

Un esempio concreto di cloud firewall è Cloudflare, una piattaforma molto conosciuta per le sue funzionalità avanzate relative alle protezione di siti, applicazioni e reti da varie minacce. Il firewall di Cloudflare include una protezione efficace contro

attacchi DDoS che utilizza il rate limiting e una rete globale distribuita in grado di assorbire e mitigare grandi volumi di traffico malevolo, regole personalizzabili per filtrare il traffico, un WAF che protegge da vulnerabilità web comuni e strumenti per la gestione e la difesa delle API che sono spesso bersaglio di attacchi sofisticati. Tutto ciò utilizzando l'intelligenza artificiale e l'apprendimento automatico in modo tale da rilevare e bloccare minacce in tempo reale.

Altri esempi sono AWS Firewall Manager per gli applicativi sviluppati su Amazon Web Services (AWS) e Google Cloud Armor, che protegge le applicazioni e le infrastrutture ospitate su Google Cloud Platform contro attacchi DDoS e altre minacce, offrendo funzionalità di gestione basate su criteri predefiniti.

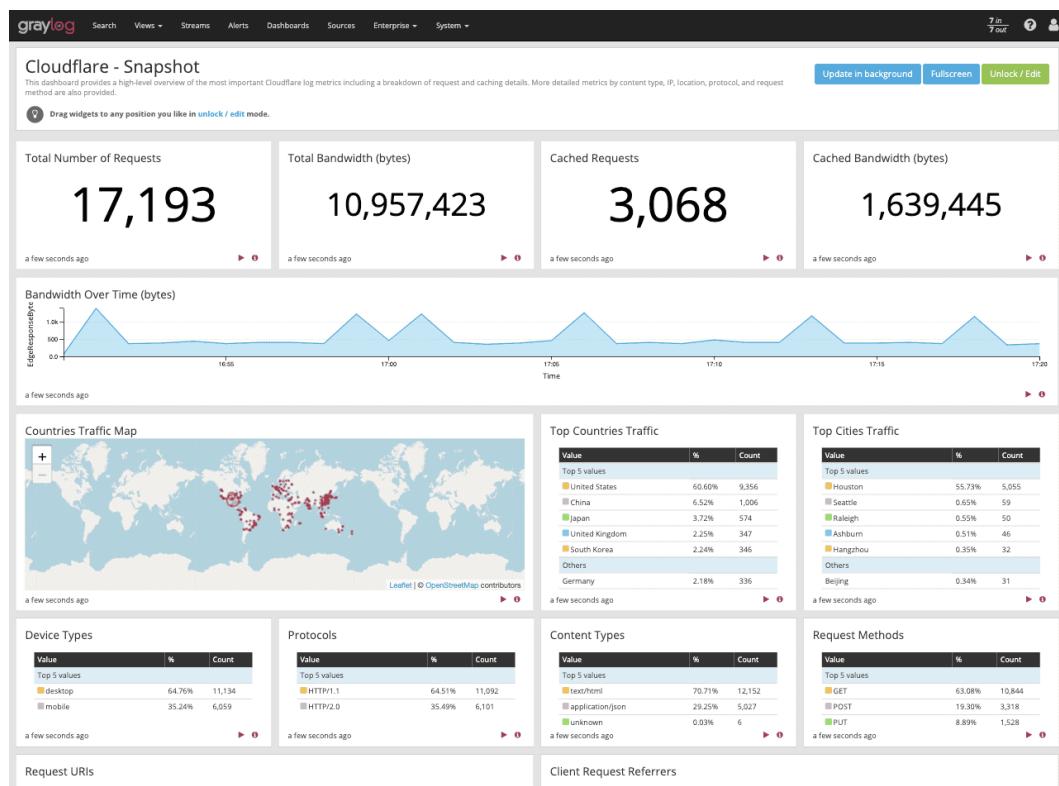


Figura 2.4: Uno sguardo nella dashboard di Cloudflare sviluppata in collaborazione con Graylog

2.4 Software Defined Networks (SDN)

2.4.1 Introduzione alle SDN

Le *Software Defined Networks* (SDN) rappresentano una delle innovazioni più significative nel campo della gestione del traffico di rete. L'approccio SDN separa il piano di controllo (*control plane*) dal piano di inoltro (*data plane*), permettendo un controllo centralizzato e programmabile della rete. Questa separazione consente agli amministratori di rete di configurare dinamicamente il comportamento dei dispositivi di rete (come switch e router) attraverso un controller centrale, migliorando così la flessibilità, la scalabilità e la gestione delle risorse della rete stessa.

Il controller SDN, che agisce come "cervello" della rete, comunica con i dispositivi di rete sottostanti utilizzando protocolli come *OpenFlow*. Questa architettura consente la programmazione delle regole di gestione del traffico in modo centralizzato, piuttosto che distribuire le decisioni di *routing* e *forwarding* tra più dispositivi. Ne risulta una maggiore facilità di gestione, un'ottimizzazione dinamica del traffico e una migliore reattività alle esigenze della rete.

Le SDN sono particolarmente utili negli ambienti di rete complessi e dinamici, dove i flussi di traffico possono variare rapidamente in base alle applicazioni in esecuzione, ai requisiti di servizio o ai cambiamenti della topologia della rete. Inoltre, l'integrazione delle SDN con strumenti avanzati di analisi del traffico e sicurezza, come gli *Intrusion Detection Systems* (IDS), offre nuove opportunità per il monitoraggio e la protezione proattiva delle infrastrutture di rete.

2.4.2 Il ruolo delle SDN nell'analisi del traffico di rete

Le SDN giocano un ruolo cruciale nell'analisi del traffico di rete, offrendo un controllo centralizzato e programmabile che facilita la raccolta e la gestione dei dati relativi ai flussi di rete. Grazie alla loro architettura flessibile, le SDN permettono di monitorare il traffico in modo più dettagliato e di applicare dinamicamente regole che consentono l'analisi approfondita dei dati di rete in tempo reale.

Un vantaggio chiave delle SDN nell'analisi del traffico è la capacità del controller di estrarre informazioni dettagliate sui flussi di dati, come la larghezza di banda

utilizzata, le latenze, le destinazioni, i protocolli utilizzati e altre metriche rilevanti. Questa visibilità approfondita consente agli amministratori di individuare anomalie nel traffico di rete e di identificare comportamenti potenzialmente pericolosi in modo rapido ed efficace. In aggiunta, il piano di controllo delle SDN può essere programmato per reagire immediatamente a determinati eventi o soglie, come la deviazione o il blocco del traffico sospetto per una più accurata ispezione da parte di un IDS.

Inoltre, le SDN offrono la possibilità di integrare funzionalità di analisi avanzata basate su *machine learning*. Utilizzando questi strumenti, il traffico di rete può essere classificato automaticamente, e il traffico anomalo può essere rilevato con maggiore accuratezza rispetto ai metodi tradizionali basati su regole. La flessibilità delle SDN, combinata con la capacità di analizzare dati di rete in tempo reale, rende questo approccio particolarmente adatto per la gestione delle reti moderne, dove la quantità e la complessità del traffico sono in costante aumento.

2.4.3 SDN e il traffico generato: sfide e opportunità per gli IDS

Le SDN non solo consentono una gestione dinamica del traffico di rete, ma possono anche essere utilizzate per generare traffico simulato per testare e ottimizzare le prestazioni degli *Intrusion Detection Systems* (IDS). Tuttavia, la capacità delle SDN di generare e manipolare il traffico introduce una serie di sfide e opportunità per i sistemi di rilevamento delle intrusioni.

Uno dei principali vantaggi delle SDN è la possibilità di generare scenari di traffico altamente controllati e personalizzabili. Gli amministratori di rete possono utilizzare questa funzionalità per simulare attacchi reali, come attacchi DDoS (*Distributed Denial of Service*) o tentativi di compromissione, e osservare come l'IDS reagisce a queste situazioni. Questo approccio permette non solo di testare l'efficacia del sistema, ma anche di migliorare il suo addestramento, affinando le capacità di rilevamento attraverso modelli di *machine learning* che imparano da questi scenari.

Tuttavia, il traffico generato dalle SDN può anche introdurre sfide complesse per gli IDS. Poiché le SDN sono altamente programmabili, un attaccante che riesca a comprometterne il controller potrebbe generare traffico malevolo altamente sofisti-

cato e distribuito, difficile da rilevare con metodi tradizionali. Inoltre, la dinamicità delle SDN può rendere il traffico di rete molto variabile, complicando la creazione di modelli statici per il rilevamento delle anomalie. Gli IDS devono quindi adattarsi a questa variabilità, utilizzando tecniche di apprendimento continuo per riconoscere nuovi *pattern* di traffico anomalo.

D'altro canto, l'integrazione delle SDN con gli IDS offre opportunità uniche. Ad esempio, un controller SDN può essere programmato per fornire visibilità completa sul traffico di rete e indirizzare flussi sospetti direttamente all'IDS per un'ispezione più approfondita. Questo approccio riduce il carico di lavoro dell'IDS, migliorando al contempo la precisione nel rilevamento degli attacchi.

CAPITOLO 3

Intelligenza Artificiale applicata alla sicurezza informatica

3.1 Concetto di IA e sua evoluzione nell'ambito della sicurezza informatica

L'intelligenza artificiale (IA) è un campo dell'informatica che si occupa della creazione di sistemi capaci di eseguire compiti che normalmente richiederebbero l'intelligenza umana. Tra questi compiti vi sono il riconoscimento vocale, la traduzione di lingue, il riconoscimento delle immagini e la presa di decisioni. Negli ultimi decenni, l'IA ha fatto passi da gigante, evolvendosi da semplici sistemi di regole e logica a complessi algoritmi di apprendimento automatico. In parallelo a questa evoluzione, ha trovato applicazione in numerosi settori, tra cui quello della sicurezza informatica.

Un esempio attuale di come l'IA stia influenzando la sicurezza informatica è la sua integrazione con strumenti come Semgrep¹. Semgrep è uno strumento avanzato di ricerca del codice basato su regole predefinite o personalizzate, utilizzato

¹"We put GPT-4 in Semgrep to point out false positives & fix code": <https://semgrep.dev/blog/2023/gpt4-and-semgrep-detailed>

ampiamente per le scansioni di sicurezza (SAST - Static Application Security Testing). Recentemente, ha integrato GPT-4 nel proprio servizio cloud per migliorare la gestione dei falsi positivi, ovvero segnalazioni erronee di vulnerabilità che non rappresentano effettivamente un rischio per la sicurezza. Questo problema è comune negli strumenti SAST e può aumentare il carico di lavoro dei team di sviluppo, che devono dedicare tempo a esaminare e correggere segnalazioni non pertinenti. Il modello IA, infatti, analizza il contesto delle segnalazioni di Semgrep, inclusi dettagli come il codice sorgente coinvolto e le regole di scansione applicate, ed è in grado di determinare se rappresentano effettivamente un problema di sicurezza o se possono essere ignorata come falso positivo. Inoltre, GPT-4 supporta anche la generazione automatica di correzioni per i problemi identificati, sebbene con tassi di accettazione attuali intorno al 40%.

```

76      76          .catch((_) => {
77      77              // if there's a 404, survey has not been filled out
78 -      -                  history.push("/survey");
78 +      +                  window.location.href = "https://forms.google.com/u/0/fill";

```

semgrep-app bot 5 days ago

Direct modification of the window location object causes React to lose state. Instead, use the `history` object from `react-router-dom`.

Semgrep Assistant thinks this is a **false positive** that should be ignored. The code is navigating to an external website, so using the `history` object from `react-router-dom` is not applicable in this case.

Reply with `/semgrep ignore <reason>` to ignore the finding created by `direct-window-location-use`.

status open

Reply...

Resolve conversation

Figura 3.1: Esempio di funzionamento di Semgrep Assistant

Tra gli ambiti più rilevanti di applicazione dell’IA nella sicurezza informatica vi è anche il rilevamento delle minacce: gli algoritmi di machine learning e deep learning sono impiegati per analizzare i flussi di dati di rete e individuare pattern anomali che possono indicare un attacco in corso. L’IA è anche utilizzata per l’analisi dei malware. Gli algoritmi di apprendimento automatico possono essere addestrati per riconoscere

caratteristiche comuni nei malware, permettendo di identificare nuove varianti basate su quelle esistenti. Tecniche di apprendimento supervisionato e non supervisionato vengono utilizzate per classificare i file come benigni o maligni, riducendo il tempo necessario per la rilevazione e la risposta. Un'altra applicazione cruciale è la risposta automatica agli incidenti. Infatti, gli algoritmi di IA possono analizzare rapidamente l'origine e l'impatto di un attacco, suggerendo o implementando automaticamente misure di contenimento e mitigazione. Questo è particolarmente utile in scenari in cui il tempo di reazione è critico. Nel settore finanziario, l'IA è ampiamente utilizzata per la prevenzione delle frodi. Con essa si possono analizzare le transazioni finanziarie in tempo reale, identificando attività sospette che possono indicare tentativi di frode. Tecniche come il clustering e la rilevazione delle anomalie sono essenziali per distinguere tra comportamenti legittimi e malevoli.

Sebbene l'IA offra numerosi vantaggi nella sicurezza informatica, presenta anche delle sfide. Ad esempio, gli algoritmi di IA possono essere suscettibili ad attacchi di adversarial machine learning, dove gli aggressori manipolano i dati di input per ingannare i modelli di apprendimento automatico. Inoltre, l'uso solleva questioni etiche riguardanti la privacy e l'accuratezza delle decisioni automatizzate.

3.2 Ruolo dell'IA nella rilevazione e prevenzione delle intrusioni

L'IA gioca un ruolo cruciale nella rilevazione e prevenzione delle intrusioni, rivoluzionando il modo in cui le minacce alla sicurezza informatica vengono affrontate. Tradizionalmente, come già citato nei capitoli precedenti, i sistemi di rilevamento delle intrusioni (IDS) e i sistemi di prevenzione delle intrusioni (IPS) si basavano su firme statiche e regole predefinite per identificare attività malevole. Tuttavia, questi approcci erano limitati nella loro capacità di rilevare minacce nuove e sofisticate.

Con l'introduzione dell'intelligenza artificiale, è possibile analizzare grandi quantità di dati di rete in tempo reale, identificando pattern anomali e comportamenti sospetti che potrebbero indicare un'intrusione. Ad esempio, tecniche come l'analisi delle serie temporali e il clustering possono rilevare anomalie nei flussi di dati,

permettendo di individuare attacchi non ancora noti e zero-day. Inoltre, l'IA può apprendere continuamente dalle nuove minacce, migliorando costantemente la sua efficacia nel rilevare intrusioni.

Un altro aspetto significativo è la sua capacità di automatizzare la risposta alle minacce. Quando un'anomalia viene rilevata, sistemi basati su IA possono eseguire azioni immediate per contenere la minaccia, come isolare il segmento di rete compromesso, bloccare indirizzi IP sospetti, o applicare patch di sicurezza. Questo non solo riduce il tempo di risposta, ma minimizza anche l'impatto di eventuali attacchi, proteggendo le risorse critiche dell'organizzazione.

Infine, l'IA può essere integrata con altre tecnologie di sicurezza, come la threat intelligence e la gestione degli eventi di sicurezza (SIEM), per fornire una visione olistica delle potenziali minacce. Questa integrazione consente di correlare dati provenienti da diverse fonti, migliorando la capacità di rilevamento e la precisione delle previsioni.

3.3 Tecniche di IA utilizzate nella rilevazione e prevenzione delle intrusioni

Nella ricerca di soluzioni per la rilevazione e la prevenzione delle intrusioni, diversi approcci basati sull'intelligenza artificiale sono stati esplorati.

Uno di questi approcci è stato presentato nel paper "**An Unsupervised Deep Learning Model for Early Network Traffic Anomaly Detection**" [2] che propone un modello di deep learning non supervisionato per rilevare prematuramente anomalie nel traffico di rete, mirando a prevenire intrusioni. Il modello è composto da due blocchi: il primo, una rete neurale convoluzionale (CNN), apprende automaticamente le feature dei dati senza necessità di ingegnerizzazione manuale; il secondo blocco è un autoencoder che classifica il traffico come legittimo o illegittimo. I dati vengono preprocessati selezionando i primi n pacchetti di un flow e estraendo i primi l byte di ciascun pacchetto, trasformandoli poi in un array monodimensionale per l'input della CNN. Questo metodo permette un'analisi in tempo reale applicando l'analisi solo ai primi byte dei pacchetti iniziali. Il modello è progettato per essere veloce e agnostico

rispetto all'IP, rendendolo adatto per l'uso in contesti di sicurezza IoT. I risultati sperimentali mostrano un'accuratezza quasi del 100%, evidenziando l'efficacia del metodo proposto.

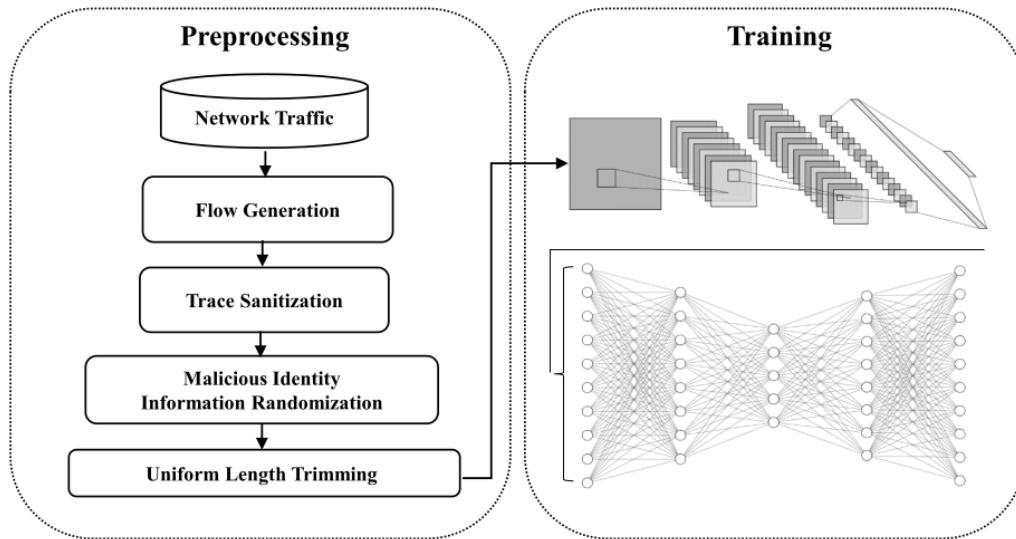


Figura 3.2: Architettura del modello descritto nel paper "An Unsupervised Deep Learning Model for Early Network Traffic Anomaly Detection", incluso il preprocessing (a sinistra) e il modulo di training e auto-learning (a destra)

Un altro approccio presentato nel paper "**Intrusion Detection Systems using Linear Discriminant Analysis and Logistic Regression**" [3] presenta un approccio supervisionato e statistico per la rilevazione delle intrusioni, utilizzando LDA (Linear Discriminant Analysis) e LR (Logistic Regression). Queste tecniche, pur non essendo di apprendimento automatico, offrono prestazioni comparabili con un minore peso computazionale, facilitando l'implementazione di sistemi in tempo reale. Le feature, 23 in totale, sono selezionate manualmente utilizzando un algoritmo di Chi Quadro e sono sottoposte a un leggero preprocessing (discretizzazione delle feature categoriche e scalatura). Sebbene la classificazione multi-classe richiederebbe la creazione di diverse funzioni di discriminazione per LDA e iperpiani per LR, nel contesto della classificazione binaria (traffico legittimo/non legittimo) la complessità si riduce notevolmente. L'architettura dei modelli è semplice, focalizzandosi principalmente sul preprocessing delle feature e sul testing. I metodi proposti sono promettenti per la loro facilità di implementazione e flessibilità, con un numero di feature personalizzabile e la possibilità di applicarli a dati diversi dal traffico di rete.

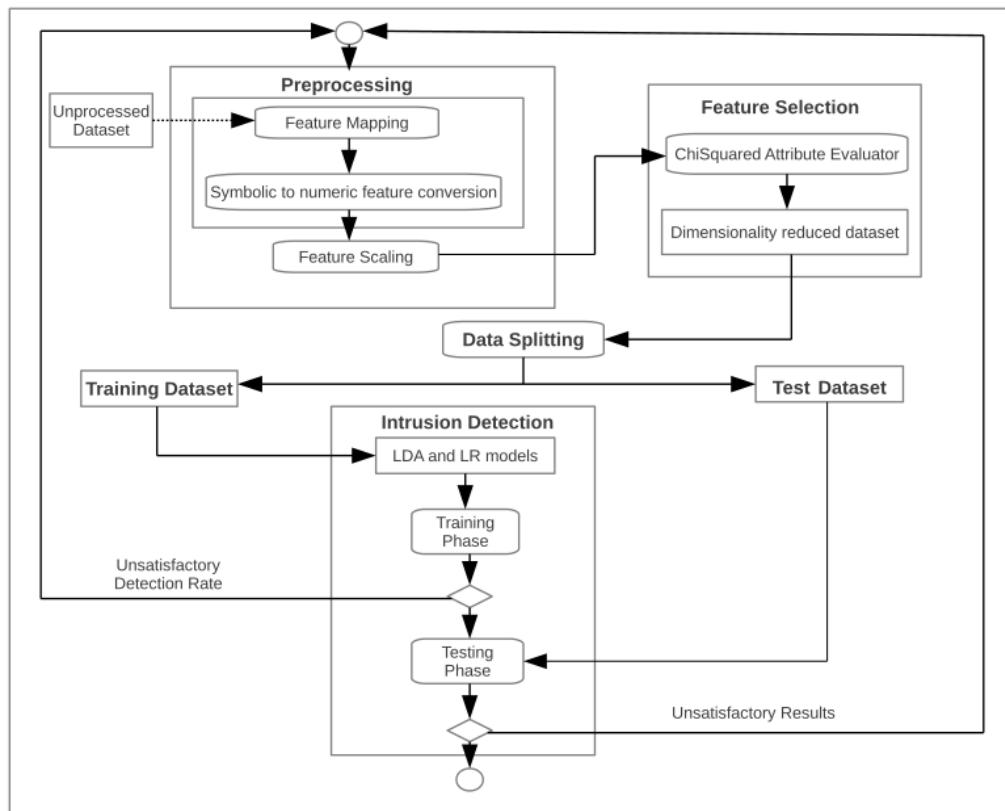


Figura 3.3: Architettura del modello descritto nel paper "Intrusion Detection Systems using Linear Discriminant Analysis and Logistic Regression"

Un terzo approccio, descritto nel paper "**Malware Traffic Classification Using Convolutional Neural Network for Representation Learning**" [4] propone un modello di CNN per la classificazione del traffico di rete, utilizzando il representation learning per analizzare e classificare vari tipi di traffico senza dipendere da specifici elementi come porte o protocolli. Il modello distingue inizialmente tra traffico legittimo e illegittimo, per poi classificare ulteriormente ogni tipo di traffico. L'architettura è molto simile a quella della LeNet-5 (una popolare rete convoluzionare) ma con un maggior numero di canali. Il modello ha mostrato un'accuratezza media del 99,41%, con migliori prestazioni nel classificatore binario rispetto a quello multi-classe. Tuttavia, la complessità della CNN e la necessità di un dataset adeguato per l'addestramento rappresentano delle sfide per l'implementazione pratica.

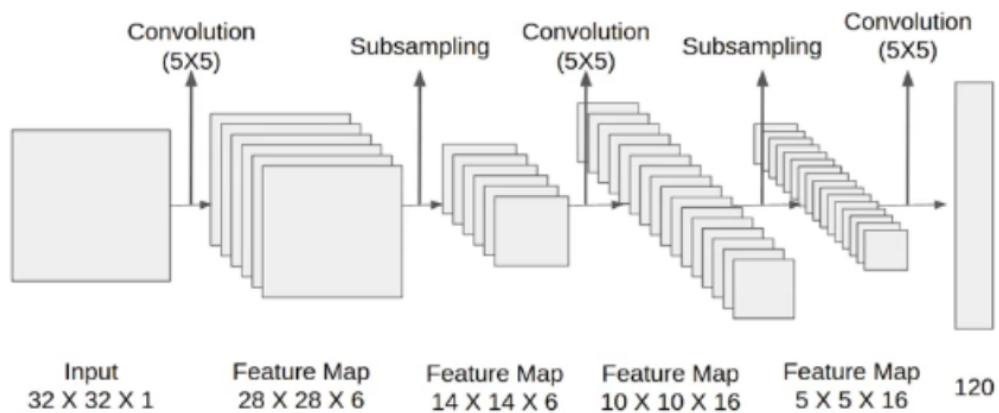


Figura 3.4: Architettura di LeNet-5

Il paper **"Network Traffic Anomaly Detection Using Recurrent Neural Networks"** [5], invece, propone un approccio innovativo utilizzando reti neurali ricorrenti (RNN) per l'analisi del traffico di rete, applicando tecniche di Natural Language Processing (NLP) alle comunicazioni. Il modello di apprendimento non supervisionato è addestrato su una grande quantità di traffico legittimo, imparando a generalizzare e predire l'attività della rete. La rilevazione delle anomalie avviene quando c'è una discrepanza significativa tra la predizione del modello e il traffico effettivo. I dati utilizzati sono registrazioni di traffico di rete, preprocessati in record di flow, che rappresentano comunicazioni unidirezionali caratterizzate da coppie di indirizzi IP e porte. Questi flow vengono raggruppati in "diadi", rappresentando sequenze ordinate di flow tra due IP, considerate come "conversazioni". Le RNN predicono il prossimo flow basandosi sui precedenti, segnalando un'anomalia quando la predizione differisce dal flow osservato. L'architettura del modello include due reti LSTM (Long-Short Term Memory) per l'analisi delle sequenze, un layer fully connected e un layer di softmax per generare una distribuzione di probabilità. Per prevenire l'overfitting, sono inseriti dropout layer dopo ciascun altro layer della rete. Il modello è in grado di analizzare intervalli minimi di un'ora, predicendo principalmente traffico già visto, il che potrebbe limitarne la flessibilità e l'adattabilità. Di conseguenza, questo approccio potrebbe non essere adatto per scenari che richiedono analisi in tempo reale o che coinvolgono traffico molto variabile.

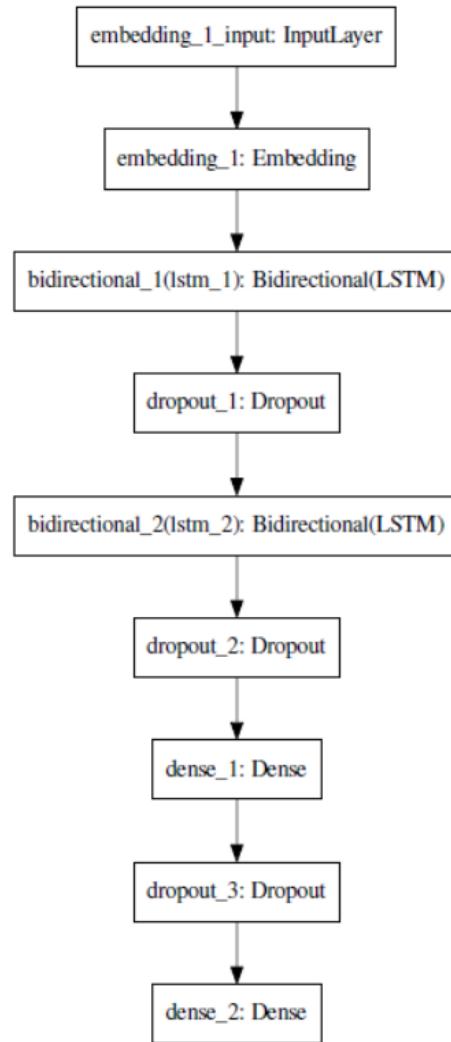


Figura 3.5: Architettura del modello descritto nel paper "Network Traffic Anomaly Detection Using Recurrent Neural Networks"

Il paper **"Network Traffic Classifier With Convolutional and Recurrent Neural Networks for Internet of Things"** [6] esplora un'architettura che combina CNN e RNN per classificare il traffico di rete nei vari servizi. I dati sono raggruppati in flow, ossia comunicazioni unidirezionali caratterizzate da protocollo, indirizzi IP e porte di origine e destinazione. L'analisi si concentra solo sugli header dei datagram, ignorando il payload, il che migliora la velocità e la confidenzialità dell'analisi. I dati dei flow sono arricchiti con statistiche come la quantità di byte trasmessi e la finestra temporale TCP, e sono rappresentati come una matrice di n vettori, ciascuno con 6 feature. L'accuratezza della rete varia in base al valore di n, con valori più alti che migliorano l'accuratezza ma rallentano la rete.

Sebbene l'architettura sia promettente, il modello è pensato principalmente per la classificazione del traffico, non per la rilevazione delle anomalie. Tuttavia, potrebbe essere adattato a questo scopo con ulteriori studi.

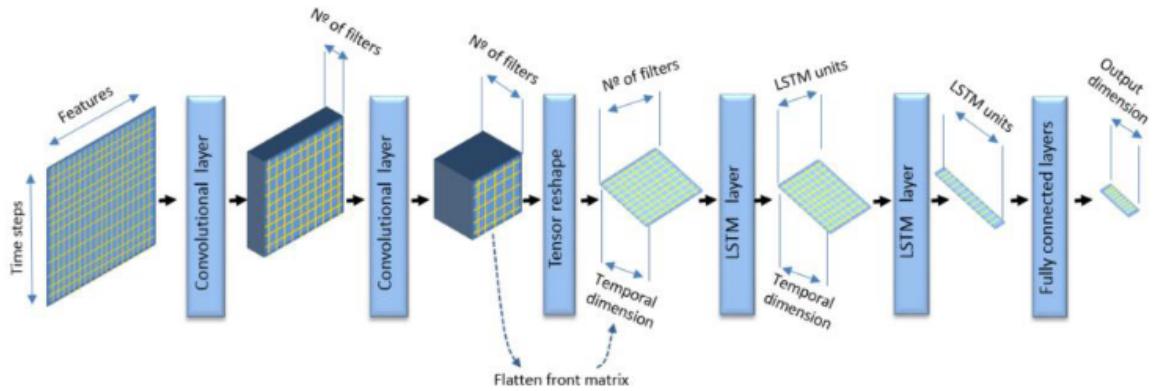


Figura 3.6: Architettura del modello descritto nel paper "Network Traffic Classifier With Convolutional and Recurrent Neural Networks for Internet of Things"

Infine, il paper **"Traffic Anomaly Detection Using K-Means Clustering"** [7] propone l'uso di tecniche di data mining non supervisionate per classificare i flow di traffico di rete. Il metodo principale impiegato è il K-Means clustering, che suddivide i dati di training in cluster definiti dai centroidi. I dati che risultano distanti dai centroidi di tutti i cluster sono considerati outliers, e quindi classificati come traffico illegittimo. I dati analizzati sono flussi unidirezionali di pacchetti IP, identificati da quintupla (protocollo, IP sorgente, IP destinazione, porto sorgente, porto destinazione) e statistiche sul flow come numero di pacchetti e byte scambiati nell'intervallo di tempo. La prima scrematura dei dati avviene raggruppandoli per tipo di protocollo (HTTP, FTP, ecc.). Se questa classificazione non è efficace, si usano categorie più specifiche come TCP, UDP e ICMP. L'analisi dei dati avviene su intervalli temporali, il che significa che si identificano periodi con anomalie senza specificare il pacchetto, flow o dispositivo responsabile. L'architettura del sistema è relativamente semplice, basandosi esclusivamente sull'algoritmo K-Means con due cluster principali: traffico legittimo e anomalie. La distanza euclidea pesata è utilizzata come funzione di distanza, poiché alcune feature sono meno rilevanti per la rilevazione di anomalie. Ogni macro categoria di protocollo (HTTP, FTP, TCP, UDP, ICMP, ecc.) ha il proprio algoritmo di clustering, richiedendo un'architettura complessa che prima inferisca il

protocollo e poi lo confronti con i cluster appropriati. Nonostante l'approccio sembra efficace e semplice, è considerato piuttosto datato in termini di tecnologie utilizzate, dalla fase di estrazione delle feature fino all'algoritmo di classificazione scelto.

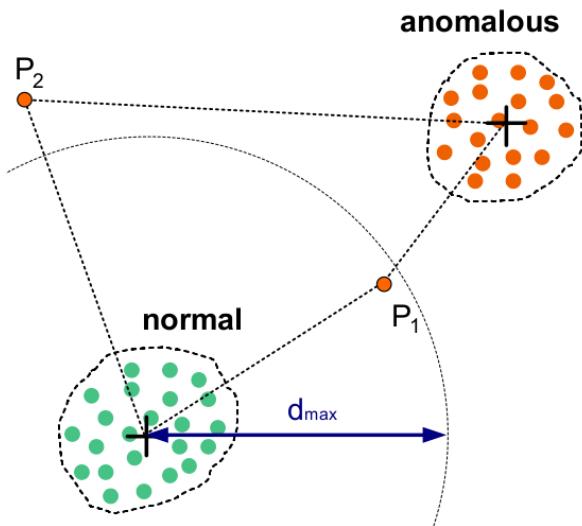


Figura 3.7: Rilevamento delle anomalie descritto nel paper "Traffic Anomaly Detection Using K-Means Clustering"

3.4 Vantaggi e sfide nell'integrazione di IDS, IPS, NG-FW e IA

L'integrazione di IDS, IPS, NGFW e IA crea un sistema di sicurezza più robusto e completo. Gli IDS e IPS monitorano il traffico di rete per rilevare e prevenire intrusioni, mentre i NGFW aggiungono funzionalità avanzate come il controllo delle applicazioni e l'ispezione profonda dei pacchetti. L'IA, con le sue capacità di apprendimento automatico e analisi predittiva, migliora ulteriormente queste tecnologie, permettendo di identificare modelli di comportamento anomali e nuove minacce in tempo reale. Uno dei principali vantaggi dell'integrazione è la maggiore accuratezza nel rilevamento delle minacce. Gli IDS e gli IPS tradizionali possono generare un elevato numero di falsi positivi, che richiedono tempo e risorse per essere analizzati. L'IA riduce drasticamente questo problema, analizzando grandi quantità di dati e distinguendo con maggiore precisione tra attività legittime e potenziali attacchi. Inoltre, può apprendere e adattarsi continuamente, migliorando la sua efficacia nel

tempo. L'integrazione facilita anche la gestione centralizzata della sicurezza. Gli amministratori di rete possono controllare e monitorare tutti gli aspetti di sicurezza da una singola piattaforma, semplificando le operazioni e riducendo il rischio di errori umani. Questa gestione unificata consente una visione più chiara delle minacce e una risposta più rapida e coordinata agli incidenti di sicurezza.

Nonostante i numerosi vantaggi, questa integrazione presenta diverse sfide. Una delle principali è la complessità tecnica: implementare e gestire un sistema integrato richiede competenze avanzate in vari campi, inclusi la sicurezza informatica, l'amministrazione di rete e l'intelligenza artificiale. La formazione del personale e l'acquisizione delle competenze necessarie possono rappresentare un investimento significativo in termini di tempo e risorse. Un'altra sfida rilevante è la gestione dei dati. L'IA richiede l'accesso a grandi quantità di dati per essere efficace, il che solleva preoccupazioni riguardo alla privacy e alla protezione delle informazioni sensibili. Inoltre, la raccolta e l'archiviazione di grandi volumi di dati possono comportare problemi di capacità e prestazioni. La compatibilità tra i vari componenti è un'altra sfida significativa. Gli IDS, IPS e NGFW spesso provengono da fornitori diversi e possono non essere progettati per funzionare insieme in modo armonioso. L'integrazione richiede un'accurata configurazione e talvolta l'adozione di soluzioni di interoperabilità per garantire che tutti i componenti lavorino in sinergia. Questo può aumentare la complessità dell'implementazione e la necessità di supporto tecnico.

In conclusione, solo attraverso un approccio olistico e ben pianificato è possibile realizzare il pieno potenziale di queste tecnologie avanzate, creando un ambiente resiliente e reattivo alle minacce emergenti.

CAPITOLO 4

Sviluppo di un IDS con tecniche di ML

4.1 Introduzione

L’obiettivo principale di questa tesi è la progettazione e valutazione di un Intrusion Detection System (IDS) basato su tecniche di Machine Learning, utilizzando i dataset NSL-KDD e CIC-IDS2017, per la rilevazione di attacchi informatici. L’intento è quello di studiare l’efficacia di diversi modelli di apprendimento automatico nella classificazione degli attacchi, confrontando le loro performance in scenari di classificazione binaria e multiclasse.

Il progetto si focalizza sui seguenti modelli di Machine Learning: Naive Bayes, Regressione Logistica, Analisi Discriminante Lineare, Random Forest e Neural Network, valutando le loro prestazioni su entrambi i dataset per diverse tipologie di attacchi. Per migliorare ulteriormente la precisione e l’efficienza del sistema, è stata implementata una procedura di feauture selection utilizzando XGBoost. Questo passaggio permette di ridurre la complessità del modello, concentrandosi solo sulle feature più rilevanti per la classificazione degli attacchi.

Inoltre, uno degli obiettivi chiave è l’ottimizzazione delle prestazioni delle reti neurali tramite l’applicazione della tecnica di Grid Search, che consente di trovare

la combinazione ottimale di iperparametri per migliorare la capacità predittiva del modello.

Infine, il progetto mira a fornire un’analisi dettagliata delle differenze tra la classificazione binaria e quella multiclasse, evidenziando i pro e i contro di ciascun approccio in termini di accuratezza, tempi di esecuzione e complessità computazionale. Attraverso questa analisi, si intende offrire una visione completa delle potenzialità dei modelli di Machine Learning applicati agli IDS e delle loro possibili implementazioni in un contesto reale.

4.2 Dataset Selection

Nella progettazione di un IDS basato su Machine Learning, la scelta del dataset è fondamentale per garantire la rappresentatività degli scenari di attacco e la validità delle valutazioni dei modelli. In questo progetto, sono stati utilizzati due dataset ampiamente conosciuti e studiati nel campo della sicurezza informatica: NSL-KDD e CIC-IDS2017. Questi dataset sono stati scelti per la loro capacità di rappresentare attacchi reali e per la varietà di tipologie di attacchi che includono.

4.2.1 NSL-KDD

Il dataset NSL-KDD è una versione migliorata del famoso KDD’99, che è stato utilizzato come benchmark per la competizione KDD Cup 1999, focalizzata sul rilevamento degli attacchi attraverso l’analisi della rete. Il KDD’99 ha subito critiche per la presenza di dati ridondanti e sbilanciati, che potevano influenzare negativamente le prestazioni dei modelli di Machine Learning. NSL-KDD cerca di risolvere questi problemi riducendo la quantità di dati ridondanti e fornendo un insieme di dati più bilanciato.

Il dataset contiene sia traffico di rete normale che diversi tipi di attacchi. Gli attacchi sono classificati in quattro principali categorie: Denial of Service (DoS), Probe, Remote to Local (R2L), e User to Root (U2R). Ogni record del dataset è rappresentato da 41 features che descrivono le proprietà del traffico di rete, come durata della connessione, protocollo utilizzato, numero di pacchetti, e altro.

4.2.2 CIC-IDS2017

Il dataset CIC-IDS2017 è stato creato dal Canadian Institute for Cybersecurity ed è considerato uno dei dataset più completi e realistici per il rilevamento degli attacchi. A differenza del NSL-KDD, il CIC-IDS2017 è stato costruito simulando scenari reali di attacco in un ambiente di rete controllato.

Il dataset contiene attacchi di Brute Force, Denial of Service (DoS), Distributed Denial of Service (DDoS), Infiltrazione, Phishing, Port Scanning, e altre minacce. Ogni record è rappresentato da 80 features che descrivono le proprietà del traffico di rete, tra cui informazioni temporali, flussi di rete e statistiche derivate.

4.2.3 Confronto tra NSL-KDD e CIC-IDS2017

Mentre il NSL-KDD rappresenta un set limitato di attacchi comuni fino al 1999, il CIC-IDS2017 offre una gamma molto più ampia di attacchi moderni, riflettendo scenari di sicurezza informatica più attuali e complessi. Il NSL-KDD contiene 41 features, mentre il CIC-IDS2017 ne contiene 80, fornendo un livello di dettaglio maggiore sulle connessioni di rete. Inoltre, il CIC-IDS2017 è considerato più realistico rispetto al NSL-KDD, in quanto è stato generato in un ambiente di test che simula scenari di attacchi del mondo reale, mentre il NSL-KDD è un dataset derivato da attacchi storici.

In conclusione, entrambi i dataset forniscono una solida base in questo contesto, offrendo una combinazione di dati storici consolidati (NSL-KDD) e scenari più recenti e realistici (CIC-IDS2017). Questo permette di confrontare i modelli in situazioni differenti e di esplorare le prestazioni in contesti sia tradizionali che moderni di attacchi informatici.

4.3 Analisi della Letteratura

Come analizzato nel capitolo 3.3, diverse ricerche hanno proposto approcci innovativi e promettenti per la rilevazione degli attacchi. I modelli di deep learning non supervisionato, come quello descritto nel paper "*An Unsupervised Deep Learning Model for Early Network Traffic Anomaly Detection*" [2], offrono un'accuratezza quasi perfetta

nella classificazione delle anomalie di rete, combinando reti neurali convoluzionali con autoencoder per un’analisi in tempo reale. Approcci più semplici, come quelli basati su tecniche statistiche supervisionate (LDA e regressione logistica), offrono soluzioni pratiche con un carico computazionale ridotto, mantenendo buone prestazioni, come dimostrato nel paper "*Intrusion Detection Systems using Linear Discriminant Analysis and Logistic Regression*" [3].

Modelli più complessi come le reti convoluzionali (CNN) e le reti ricorrenti (RNN), presentati rispettivamente nei paper "*Malware Traffic Classification Using Convolutional Neural Network for Representation Learning*" [4] e "*Network Traffic Anomaly Detection Using Recurrent Neural Networks*" [5], dimostrano grande potenzialità nella classificazione del traffico e nella predizione delle anomalie, ma richiedono dataset adeguati e presentano sfide di implementazione.

La combinazione di CNN e RNN, come proposta per l’IoT, rappresenta un compromesso tra accuratezza e complessità, mentre tecniche di clustering come il K-Means offrono soluzioni semplici ma ormai datate.

A questo punto, approfondiamo la letteratura relativa ai dataset selezionati che sono stati utilizzati per addestrare e testare i modelli.

4.3.1 Performance del dataset NSL-KDD

Come già accennato in precedenza, il dataset NSL-KDD è una versione migliorata del famoso dataset KDD CUP 99, il quale è stato ampiamente utilizzato per l’addestramento degli IDS nonostante i vari problemi . In risposta alle diverse critiche, il dataset NSL-KDD è stato proposto nel paper "*A Detailed Analysis of the KDD CUP 99 Data Set*" [8] come una versione perfezionata che rimuove le ridondanze e riequilibra il numero di istanze, permettendo una valutazione più equa delle capacità di classificazione degli algoritmi.

Per valutare le prestazioni del dataset, sono stati utilizzati diversi algoritmi di machine learning. L’algoritmo J48, che è una variante migliorata dell’albero decisionale C4.5, ha raggiunto l’accuracy più alta con un valore del 93.82%, dimostrando un’elevata capacità di classificazione delle diverse tipologie di traffico di rete. Anche l’algoritmo NB Tree, che combina la semplicità di Naive Bayes con la potenza predit-

tiva degli alberi decisionali, ha ottenuto prestazioni notevoli con una precisione del 93.51%.

L'algoritmo **Random Forest**, un ensemble di alberi decisionali, ha mostrato una precisione del 92.79%, offrendo un buon compromesso tra accuratezza e robustezza. In modo simile, il **Random Tree** ha ottenuto una precisione del 92.53%, confermando l'efficacia dei metodi basati su alberi per questo tipo di dataset.

Il **Multi-layer Perceptron (MLP)**, una rete neurale a strati, ha ottenuto una precisione del 92.26%, dimostrando che le reti neurali possono essere efficaci per la classificazione degli attacchi, anche se leggermente meno accurate rispetto agli alberi decisionali. Questo risultato riflette la capacità delle reti neurali di catturare relazioni complesse tra le caratteristiche, a fronte di un maggiore costo computazionale.

L'algoritmo **Naive Bayes** ha ottenuto una precisione inferiore, pari all'81.66%, sottolineando i limiti delle assunzioni di indipendenza delle caratteristiche su un dataset complesso come NSL-KDD. Infine, l'**SVM** ha raggiunto una precisione del 65.01%, indicando che questo algoritmo potrebbe richiedere una maggiore ottimizzazione o potrebbe non essere il più adatto per la natura del dataset.

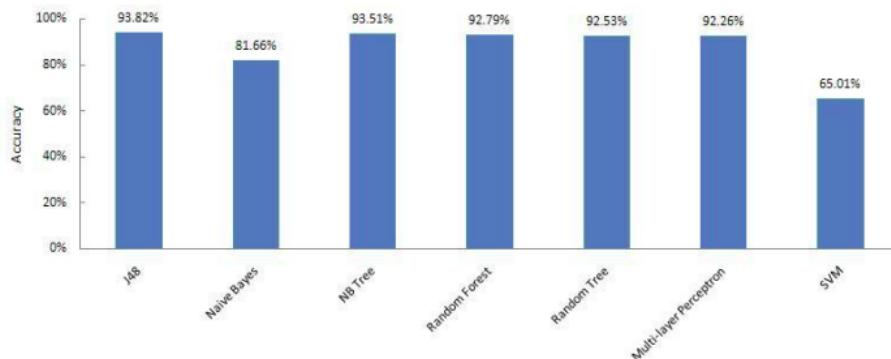


Figura 4.1: Risultati delle prestazioni sul dataset NSL-KDD

4.3.2 Performance del dataset CIC-IDS2017

Nel paper *"Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization"* [9], gli autori hanno condotto un'analisi delle prestazioni del dataset **CIC-IDS2017** seguendo un approccio strutturato in quattro fasi.

Inizialmente, sono state estratte 80 feature dai flussi di traffico di rete utilizzando il tool **CICflowMeter**. Esse includono parametri come la durata del flusso, il numero di byte e pacchetti inviati e ricevuti, e l'intervallo di tempo tra i pacchetti (Inter Arrival Time, IAT). Le feature così estratte sono state utilizzate per rappresentare i flussi di rete e individuare i pattern correlati agli attacchi.

Successivamente, per identificare il miglior sottoinsieme di feature per rilevare ciascun attacco, è stata impiegata la classe **RandomForestRegressor** di Scikit-learn. Questo metodo ha permesso di calcolare l'importanza di ognuna di esse su tutto il dataset. Il risultato finale è stato ottenuto moltiplicando il valore medio standarizzato di ciascuna feature per il suo valore di importanza relativa. Ad esempio, la durata del flusso e le feature legate all'IAT sono risultate particolarmente efficaci per il rilevamento degli attacchi DoS.

Le prestazioni del dataset sono state quindi testate utilizzando sette algoritmi di machine learning comuni: **K-Nearest Neighbors (KNN)**, **Random Forest (RF)**, **ID3**, **Adaboost**, **Multilayer Perceptron (MLP)**, **Naive Bayes (NB)** e **Quadratic Discriminant Analysis (QDA)**. Le metriche di valutazione impiegate sono state la **Precisione (Pr)**, il **Recall (Rc)** e il **F1-score**. I risultati di questo test sono riportati nella seguente tabella:

Algoritmo	Precisione	Recall	F1-score	Tempo di Esecuzione (Sec)
KNN	0.96	0.96	0.96	1908.23
Random Forest	0.98	0.97	0.97	74.39
ID3	0.98	0.98	0.98	235.02
Adaboost	0.77	0.84	0.77	1126.24
MLP	0.77	0.83	0.76	575.73
Naive Bayes	0.88	0.04	0.04	14.77
QDA	0.97	0.88	0.92	18.79

Tabella 4.1: Risultati delle prestazioni sul dataset CIC-IDS2017

Dai risultati ottenuti emerge che gli algoritmi **KNN**, **Random Forest** e **ID3** hanno mostrato le migliori prestazioni complessive in termini di precisione, recall e F1-score.

Tuttavia, il **Random Forest** si è distinto come l'algoritmo più efficiente, completando il processo di addestramento e testing in soli 74.39 secondi, risultando quindi molto più veloce rispetto al **KNN**, che ha impiegato 1908.23 secondi, rivelandosi l'algoritmo più lento.

4.4 Data Pre-Processing

4.4.1 Data Cleaning

Il processo di data cleaning è un passaggio cruciale per garantire la qualità dei dati e migliorare le prestazioni dei modelli di Machine Learning. Nel progetto sono state effettuate diverse operazioni di pulizia sui dataset utilizzati, a partire dalla gestione dei valori nulli. Sono state eliminate tutte le righe contenenti valori mancanti, poiché la presenza di dati incompleti può compromettere l'addestramento dei modelli. Inoltre, si è intervenuti sui valori infiniti, sia positivi che negativi, presenti nelle colonne numeriche: i record contenenti questi valori sono stati rimossi dal dataset, in quanto potevano causare anomalie nei modelli. Infine, sono state individuate e rimosse le righe duplicate, al fine di evitare ridondanze che avrebbero potuto distorcere le analisi e le previsioni dei modelli.

Dunque, è stato adottato il row pruning per gestire i valori nulli, infiniti e duplicati nel dataset. Questa tecnica è stata preferita per la sua semplicità e per la sua capacità di garantire l'integrità del dataset senza introdurre distorsioni. Poiché i record problematici erano pochi e non influenzavano significativamente la dimensione complessiva del dataset, la rimozione diretta delle righe è risultata una soluzione pratica ed efficace. Questo approccio ha permesso di mantenere un dataset pulito e di alta qualità senza complicare la pre-elaborazione con metodi più complessi, assicurando così che i modelli potessero operare su dati affidabili e rappresentativi.

4.4.2 One-Hot Encoding delle Variabili Categoriche

Dopo la pulizia dei dati, è stata applicata la tecnica del One-Hot Encoding per trasformare specifiche variabili categoriche in una rappresentazione numerica, consentendo ai modelli di elaborare correttamente tali informazioni.

In particolare, le colonne del dataset NSL-KDD *protocol_type*, *service* e *flag*, che contengono valori categorici, sono state convertite in vettori binari. Queste colonne rappresentano attributi importanti per la classificazione degli attacchi di rete. Ad esempio, *protocol_type* identifica il protocollo di rete utilizzato (come TCP, UDP, o ICMP), *service* specifica il tipo di servizio di rete (come HTTP, FTP, o SSH), mentre *flag* indica lo stato della connessione (come SF per "successful connection").

4.4.3 Standardizzazione delle Feature

Infine, le feature numeriche del dataset sono state standardizzate per garantire che abbiano una distribuzione con media 0 e deviazione standard 1. La standardizzazione è un passaggio cruciale per migliorare la performance dei modelli, specialmente quelli che utilizzano distanze euclidee o gradienti nella fase di ottimizzazione.

4.5 Data Balancing

Lo sbilanciamento delle classi è un problema critico nei dataset utilizzati nel contesto di rilevamento degli attacchi. La presenza di un numero significativamente maggiore di istanze appartenenti alla classe "benigna" rispetto a quelle "malevoli" può portare i modelli di Machine Learning a favorire la classe maggioritaria, riducendo la capacità del modello di identificare correttamente gli attacchi.

4.5.1 Data Balancing nella Classificazione Binaria

Per mitigare questo problema, nella classificazione binaria è stato applicato l'undersampling, che consiste nel ridurre il numero di istanze della classe maggioritaria per rendere il dataset più bilanciato. In particolare, è stata utilizzata la tecnica di random undersampling, che prevede la selezione casuale di un sottoinsieme delle istanze della classe più rappresentata (traffico benigno) per far sì che il numero di istanze tra le due classi sia più simile.

Per il dataset NSL-KDD, la distribuzione originale delle classi mostrava che il 53.46% delle istanze apparteneva alla classe "BENIGN" e il 46.54% alla classe "Malicious" (Figura 4.2). Dopo aver applicato l'undersampling, la distribuzione è

stata equamente bilanciata, con il 50% delle istanze in ciascuna classe (Figura 4.3). In termini numerici, il numero di istanze della classe "BENIGN" è stato ridotto a 58.630, mentre il numero di istanze della classe "Malicious" è rimasto invariato, risultando in un totale di 117.260 istanze, equamente suddiviso tra le due classi.

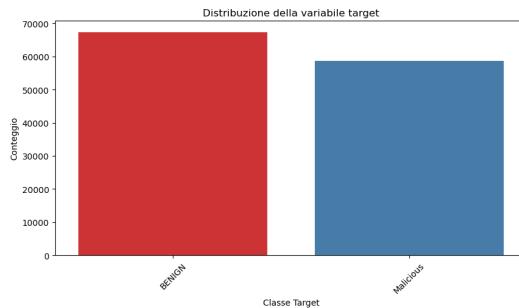


Figura 4.2: Distribuzione classi in NSL-KDD

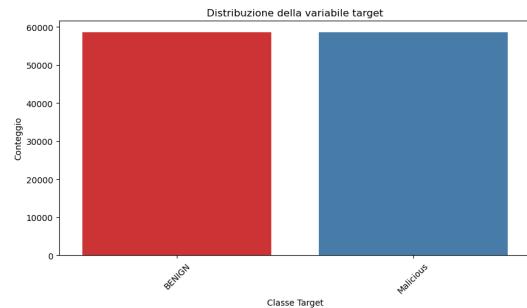


Figura 4.3: Distribuzione classi in NSL-KDD dopo undersampling

Per quanto riguarda il dataset CIC-IDS2017, la distribuzione originale era molto sbilanciata: la classe "BENIGN" costituiva l'83.11% del dataset, mentre la classe "Malicious" rappresentava solo il 16.89% (Figura 4.4). Dopo l'applicazione dell'undersampling, entrambe le classi hanno raggiunto una distribuzione del 50% (Figura 4.5). Specificamente, il numero di istanze per ciascuna classe è stato uniformato a 425.741, portando il totale a 851.482 istanze.

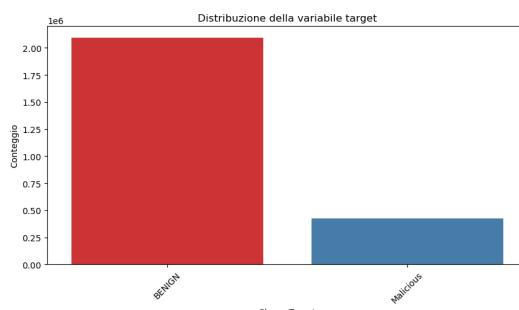


Figura 4.4: Distribuzione classi in CIC-IDS2017

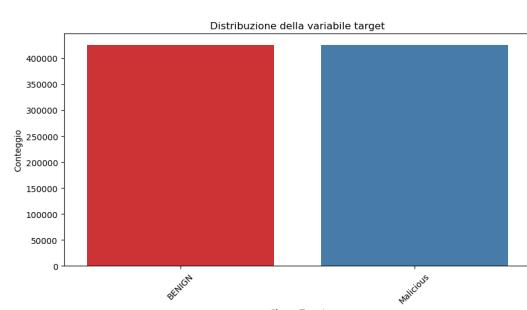


Figura 4.5: Distribuzione classi in CIC-IDS2017 dopo undersampling

Da una parte l'undersampling ha permesso di migliorare l'attenzione del modello verso le classi minoritarie, riducendo il rischio di sbilanciamento delle predizioni a favore della classe maggioritaria. D'altra parte, un potenziale svantaggio dell'undersampling è la perdita di informazioni derivante dalla riduzione delle istanze

della classe maggioritaria, che potrebbe influire negativamente sulla performance del modello.

4.5.2 Data Balancing nella Classificazione Multiclasse

Nel contesto della classificazione multiclasse, l'undersampling non è stato utilizzato. Questo perché alcune delle classi presenti nel dataset, soprattutto per gli attacchi più rari, contenevano pochissime istanze. Ridurre ulteriormente il numero di istanze per bilanciare le classi avrebbe potuto comportare una perdita significativa di dati, rendendo ancora più difficile l'addestramento del modello per riconoscere quelle classi.

Invece di applicare l'undersampling, per trattare lo sbilanciamento nella classificazione multiclasse è stata utilizzata la tecnica di Stratified K-Fold Cross-Validation. Questa tecnica garantisce che ogni fold abbia la stessa proporzione di istanze di ciascuna classe, mantenendo così un bilanciamento relativo tra le classi durante la fase di addestramento e validazione. In questo modo, si è riusciti a trattare il problema dello sbilanciamento senza sacrificare preziose informazioni del dataset, migliorando la capacità del modello di generalizzare anche su classi meno rappresentate.

4.6 Suddivisione del Dataset

La suddivisione del dataset in set di addestramento e test è un passaggio essenziale nella costruzione e valutazione dei modelli di Machine Learning. Questo processo consente di addestrare il modello su una porzione dei dati e di verificarne le prestazioni su dati separati, che non sono stati utilizzati durante l'addestramento.

In questo progetto, è stato riservato il 33% del dataset per il test set. Questa proporzione è stata selezionata per garantire un equilibrio adeguato tra la quantità di dati disponibili per l'addestramento e la dimensione del test set, assicurando che quest'ultimo sia abbastanza grande da fornire una valutazione accurata delle prestazioni del modello.

Inoltre, per evitare bias derivanti da ordinamenti preesistenti nei dati, è stata effettuata una mescolatura casuale dei dati prima della suddivisione. Tale approccio

assicura che la distribuzione dei dati rifletta fedelmente la distribuzione complessiva del dataset.

4.7 Training

4.7.1 Scelta dei Modelli

Durante la fase di training, sono stati addestrati e valutati diversi modelli di Machine Learning per confrontare le loro prestazioni. La scelta dei modelli è stata guidata dall'esigenza di bilanciare robustezza, efficienza computazionale e adattabilità ai dati specifici.

Il modello **Naive Bayes** è stato selezionato per la sua semplicità e per l'efficienza nell'elaborazione di grandi volumi di dati, oltre che per la capacità di gestire in modo efficace feature indipendenti, caratteristica comune nei dati di rete.

La **Regressione Logistica** e l'**Analisi Discriminante Lineare (LDA)** sono stati selezionati per la loro interpretabilità e capacità di gestire problemi di classificazione. Entrambi i modelli, come evidenziato nel paper "*Intrusion Detection Systems using Linear Discriminant Analysis and Logistic Regression*" [3], forniscono buone prestazioni quando le classi sono chiaramente separabili, il che è tipico nei dataset in questo contesto dove si distingue tra traffico legittimo e illegittimo. Questi modelli si distinguono inoltre per il loro basso costo computazionale, rendendoli adatti per applicazioni in tempo reale.

Il **Multi-Layer Perceptron (MLP)**, un tipo specifico di rete neurale feedforward, è stato incluso per la sua capacità di apprendere rappresentazioni complesse e non lineari dai dati mantenendo comunque una struttura relativamente semplice. Diversi paper analizzati in precedenza ("*An Unsupervised Deep Learning Model for Early Network Traffic Anomaly Detection*" [2], "*Malware Traffic Classification Using Convolutional Neural Network for Representation Learning*" [4] e "*Network Traffic Anomaly Detection Using Recurrent Neural Networks*" [5]) hanno dimostrato l'efficacia delle reti neurali nel rilevare anomalie nel traffico di rete con un'accuratezza quasi perfetta, grazie alla loro abilità di esaminare pattern nascosti nei dati. Sebbene esse richiedano un elevato

costo computazionale e siano più difficili da interpretare rispetto ai modelli lineari, il loro potenziale le rende cruciali per questo tipo di task.

Infine, il modello **Random Forest** è stato utilizzato per la sua robustezza contro l’overfitting e la capacità di gestire dataset con molte feature, come sottolineato nella letteratura del dataset NSL-KDD e quella del dataset CIC-IDS2017. Random Forest ha mostrato un’ottima combinazione tra accuratezza e tempi di esecuzione, rendendolo particolarmente adatto per scenari che richiedono efficienza senza compromettere le prestazioni.

4.7.2 Confronto tra Classificazione Binaria e Multiclasse

Nel contesto dell’analisi del traffico di rete è fondamentale scegliere l’approccio di classificazione più adatto per ottenere risultati accurati e affidabili. Esistono due approcci principali per affrontare il problema: la classificazione binaria e quella multiclasse.

La **classificazione multiclasse** mira a distinguere tra diverse categorie di attacchi, oltre alla classe *BENIGN*, offrendo vantaggi come una maggiore granularità nella rilevazione degli attacchi. Questa granularità è utile per rispondere in modo mirato, ad esempio distinguendo tra un attacco *DDoS* e un *Web Attack*, e permette di rilevare nuove varianti o tendenze specifiche, come l’aumento degli attacchi basati su *PortScan*. Tuttavia, presenta anche svantaggi significativi: il principale è lo sbilanciamento delle classi, che rende difficile la rilevazione delle classi meno rappresentate come *Infiltration* e *Bot*, causando basse prestazioni in termini di precisione e recall. Inoltre, la maggiore complessità richiede modelli più sofisticati, il che rende difficile bilanciare le prestazioni tra tutte le classi e può ridurre l’interpretabilità dei risultati, aumentando il rischio di overfitting, specialmente con modelli complessi come le reti neurali.

La **classificazione binaria**, d’altro canto, semplifica il problema riducendolo alla distinzione tra *BENIGN* e *ATTACK*. Questo approccio presenta alcuni vantaggi chiave, come la semplicità dei modelli, che consente di raggiungere elevate prestazioni nel rilevamento di attacchi. I modelli binari gestiscono meglio lo sbilanciamento delle classi, essendo più robusti con dataset dove prevale la classe *BENIGN*. Inoltre, l’interpretabilità dei risultati è migliorata, poiché con meno classi è più facile comprendere

le prestazioni del modello e impostare soglie decisionali chiare per la rilevazione. Tuttavia, la classificazione binaria ha lo svantaggio di perdere informazioni specifiche sull'attacco. In contesti reali, sapere se un attacco è di tipo *DDoS* o *Brute Force* può essere cruciale per una risposta tempestiva e appropriata.

In sintesi, la scelta tra classificazione multiclasse e binaria dipende dall'obiettivo finale dell'analisi. Se lo scopo è semplicemente rilevare la presenza di un attacco, la classificazione binaria offre un approccio più semplice, robusto e interpretabile. Tuttavia, quando è necessario ottenere maggiori dettagli sugli attacchi e differenziare le varie tipologie, la classificazione multiclasse diventa indispensabile, nonostante la sua complessità e le sfide legate allo sbilanciamento delle classi.

Inoltre, per la classificazione binaria, il training è stato effettuato utilizzando sia un approccio classico di suddivisione del dataset in set di addestramento e di test, sia tramite una procedura di validazione più robusta con stratified k-fold cross-validation. Per la classificazione multiclasse, invece, il training è stato effettuato esclusivamente con stratified k-fold cross-validation e senza l'uso di undersampling, come spiegato nei capitoli precedenti. Questo tipo di convalida incrociata, infatti, garantisce che ogni fold rifletta la distribuzione complessiva delle classi nel dataset, migliorando così l'affidabilità delle stime delle performance del modello, soprattutto quando si lavora con dataset complessi e sbilanciati.

4.7.3 NSL-KDD

Nella classificazione multiclasse del dataset NSL-KDD, gli attacchi sono stati raggruppati in quattro categorie principali: *DoS*, *Probe*, *R2L*, e *U2R*. Questa categorizzazione ha permesso di semplificare l'analisi e migliorare la comprensione delle performance dei modelli. Di seguito è riportata la mappatura utilizzata per raggruppare le diverse classi di attacchi:

- **DoS:** neptune, teardrop, smurf, pod, back, land
- **Probe:** ipsweep, portsweep, nmap, satan
- **R2L:** warezclient, guess_passwd, ftp_write, multihop, imap, phf, spy, warezmaster

- **U2R**: rootkit, buffer_overflow, loadmodule, perl

Gaussian Naive Bayes

Nel caso della **classificazione binaria**, il modello Gaussian Naive Bayes ha ottenuto un'accuratezza dell'84%. Sebbene il recall per la classe *BENIGN* sia stato vicino al 100%, il recall per la classe *Malicious* è stato solo del 68%, indicando uno squilibrio nelle performance. Durante la validazione incrociata stratificata, l'accuratezza è scesa al 50.06%, suggerendo una scarsa generalizzazione del modello nonostante l'undersampling.

Per la **classificazione multiclasse**, il modello ha raggiunto un'accuratezza del 38.83%. Le performance erano migliori per la classe *DoS* con un recall del 97.93% e un F1-score di 0.5682, mentre le altre classi, come *BENIGN* e *U2R*, hanno mostrato precisione e recall significativamente più bassi. Questo evidenzia le difficoltà del modello nel gestire la complessità dei dati multiclasse.

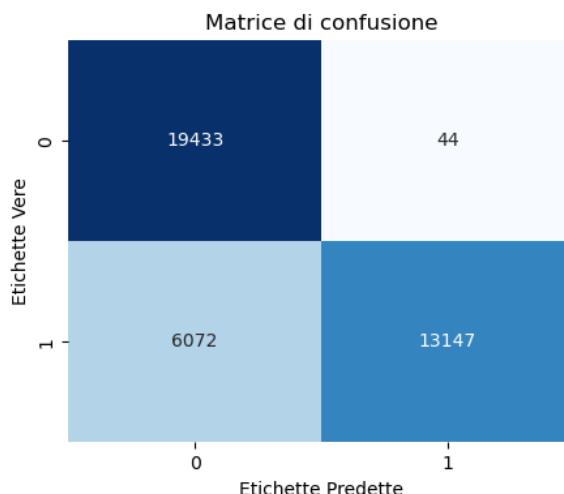


Figura 4.6: Matrice di confusione - classificazione binaria con Naive Bayes (NSL-KDD)

Classe	Precision (%)	Recall (%)	F1-Score
BENIGN	76%	100%	0.86
Malicious	100%	68%	0.81
Accuracy			84%

Tabella 4.2: Report di classificazione binaria con Naive Bayes (NSL-KDD)

Classe	Precision (%)	Recall (%)	F1-Score
BENIGN	50.03%	98.43%	0.6634
Malicious	51.91%	1.70%	0.0329
Accuracy: 50.06%			

Tabella 4.3: Report di classificazione binaria con cross-validation e Naive Bayes (NSL-KDD)

Classe	Precision (%)	Recall (%)	F1-Score
BENIGN	72.46%	4.50%	0.0845
DoS	40.02%	97.93%	0.5682
Probe	56.35%	7.64%	0.1344
R2L	1.19%	0.50%	0.0070
U2R	0.17%	24.91%	0.0034
Accuracy: 38.83%			

Tabella 4.4: Report di classificazione multiclasse con cross-validation e Naive Bayes (NSL-KDD)

Regressione Logistica

Nel contesto della **classificazione binaria**, la Regressione Logistica ha mostrato prestazioni superiori, raggiungendo un'accuratezza del 99%. Il modello ha ottenuto elevati valori di precisione e recall per entrambe le classi, con un F1-score di 0.99 per ciascuna. Anche con la cross-validation stratificata, ha mantenuto buone performance

con un'accuratezza dell'87.54%, dimostrando di essere una scelta robusta per la classificazione binaria, grazie anche al bilanciamento del dataset.

Per la **classificazione multiclasse**, la Regressione Logistica ha significativamente migliorato le prestazioni rispetto al Gaussian Naive Bayes, ottenendo un'accuratezza complessiva dell'84.18%. Ha mostrato eccellenti risultati nella classificazione delle classi *BENIGN* e *DoS*, con precisioni superiori all'84%. Tuttavia, le classi meno rappresentate come *Probe*, *R2L*, e *U2R* hanno mostrato risultati inferiori, suggerendo difficoltà del modello nel trattare classi meno frequenti e più complesse.

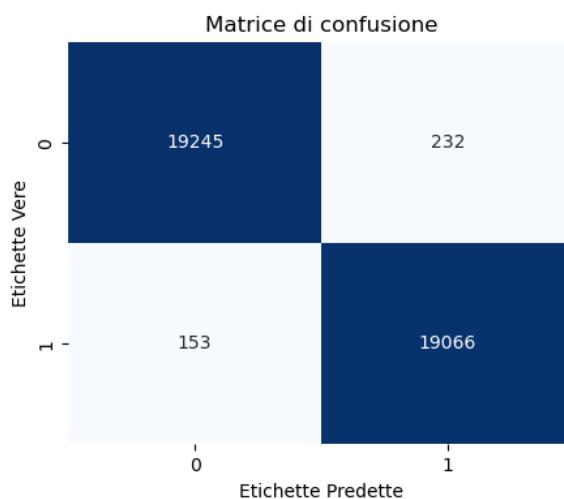


Figura 4.7: Matrice di confusione - classificazione binaria con Regressione Logistica (NSL-KDD)

Classe	Precision	Recall	F1-Score
BENIGN	99%	99%	0.99
Malicious	99%	99%	0.99
Accuracy	99%		

Tabella 4.5: Report di classificazione binaria con Regressione Logistica (NSL-KDD)

Classe	Precision	Recall	F1-Score
BENIGN	87.03%	88.23%	0.8763
Malicious	88.07%	86.85%	0.8745
Accuracy	87.54%		

Tabella 4.6: Report di classificazione binaria con cross-validation e Regressione Logistica (NSL-KDD)

Classe	Precision	Recall	F1-Score
BENIGN	86.56%	93.68%	0.8998
DoS	84.49%	91.27%	0.8775
Probe	30.06%	8.88%	0.1369
R2L	3.64%	0.20%	0.0038
U2R	0.00%	0.00%	0.0000
Accuracy	84.18%		

Tabella 4.7: Report di classificazione multiclasse con cross-validation e Regressione Logistica (NSL-KDD)

Analisi Discriminante Lineare (LDA)

Nel contesto della **classificazione binaria**, l’Analisi Discriminante Lineare (LDA) ha raggiunto un’accuratezza del 98%, mostrando prestazioni eccellenti nel discriminare tra le classi *BENIGN* e *Malicious*. I risultati sono stati coerenti anche dopo la cross-validation stratificata, confermando la robustezza del modello in un dataset bilanciato.

Per la **classificazione multiclasse**, LDA ha ottenuto un’accuratezza del 97.39%. Ha classificato con alta precisione le classi *BENIGN*, *DoS*, e *Probe*, con precisioni superiori al 98%. Anche la classe *R2L*, che è difficile da distinguere, ha mostrato un buon F1-score di 0.6216, dimostrando che LDA è capace di gestire efficacemente dati complessi.

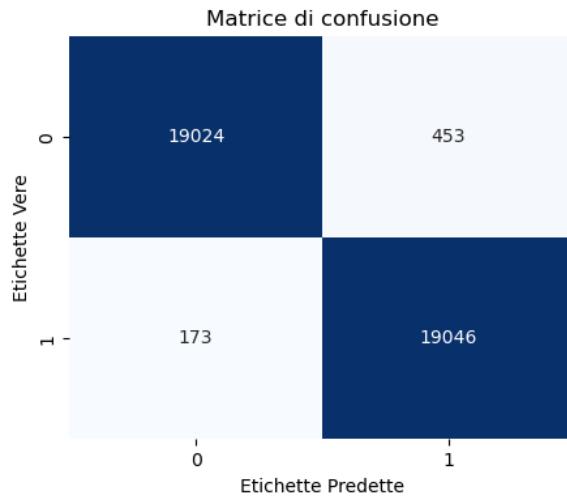


Figura 4.8: Matrice di confusione - classificazione binaria con LDA (NSL-KDD)

Classe	Precision	Recall	F1-Score
BENIGN	99%	98%	0.98
Malicious	98%	99%	0.98
Accuracy	98%		

Tabella 4.8: Report di classificazione binaria con LDA (NSL-KDD)

Classe	Precision	Recall	F1-Score
BENIGN	99.04%	97.74%	0.9838
Malicious	97.77%	99.05%	0.9840
Accuracy	98.39%		

Tabella 4.9: Report di classificazione binaria con cross-validation e LDA (NSL-KDD)

Classe	Precision	Recall	F1-Score
BENIGN	98.89%	97.53%	0.9821
DoS	98.73%	97.75%	0.9824
Probe	93.91%	95.34%	0.9462
R2L	45.69%	97.39%	0.6216
U2R	21.49%	50.00%	0.2993
Accuracy	97.39%		

Tabella 4.10: Report di classificazione multiclasse con cross-validation e LDA (NSL-KDD)

Random Forest

Per la **classificazione binaria**, il modello Random Forest ha ottenuto un'accuratezza eccezionale vicina al 100% (i valori della figura 4.11 sono arrotondati, per maggior dettagli osservare la matrice di confusione 4.9). La matrice di confusione indica un'accurata classificazione delle istanze, con precisione e recall prossimi al 100% per entrambe le classi. La cross-validation stratificata ha confermato queste prestazioni, con un'accuratezza complessiva del 99.96%.

Nella **classificazione multiclasse**, il modello Random Forest ha raggiunto un'accuratezza del 99.95%, mostrando eccellenti capacità predittive per tutte le classi. Le classi *BENIGN*, *DoS*, *Probe* e *R2L* hanno ottenuto F1-score molto elevati, mentre la classe *U2R* ha raggiunto un F1-score di 0.7890. Questo conferma l'eccezionale efficacia del modello nella gestione della classificazione multiclasse su un dataset complesso come NSL-KDD.

Classe	Precision	Recall	F1-Score
BENIGN	100%	100%	1.00
Malicious	100%	100%	1.00
Accuracy	100%		

Tabella 4.11: Report di classificazione binaria con Random Forest (NSL-KDD)

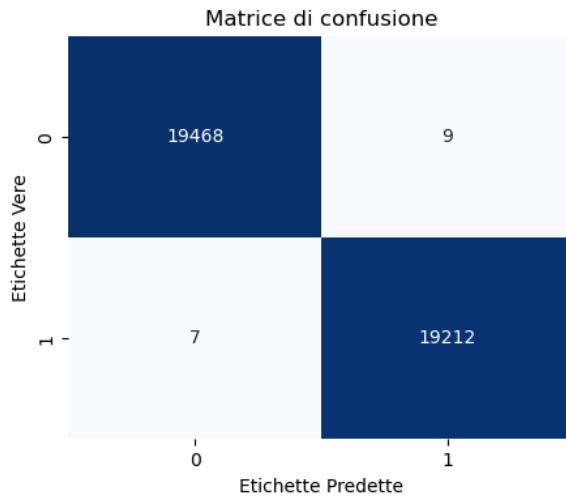


Figura 4.9: Matrice di confusione - classificazione binaria con Random Forest (NSL-KDD)

Classe	Precision	Recall	F1-Score
BENIGN	99.97%	99.96%	0.9996
Malicious	99.96%	99.97%	0.9996
Accuracy	99.96%		

Tabella 4.12: Report di classificazione binaria con cross-validation e Random Forest (NSL-KDD)

Classe	Precision	Recall	F1-Score
BENIGN	99.94%	99.97%	0.9996
DoS	99.99%	99.99%	0.9999
Probe	99.91%	99.94%	0.9993
R2L	99.28%	97.69%	0.9848
U2R	89.97%	71.45%	0.7890
Accuracy	99.95%		

Tabella 4.13: Report di classificazione multiclasse con cross-validation e Random Forest (NSL-KDD)

Multi-Layer Perceptron (MLP)

Nel contesto della **classificazione binaria**, anche il Multi-Layer Perceptron (MLP) ha raggiunto un'accuratezza vicina al 100%, dimostrando una classificazione quasi perfetta per entrambe le classi. Anche dopo la cross-validation stratificata, l'accuratezza si è mantenuta alta al 97.48%.

Per la **classificazione multiclasse**, l'MLP ha ottenuto un'accuratezza complessiva del 97.18%. Ha mostrato ottime performance nelle classi principali, come *BENIGN*, *DoS* e *Probe*, ma ha avuto difficoltà con le classi meno rappresentate, come *R2L* e *U2R*. Nonostante queste difficoltà, il MLP ha dimostrato una buona capacità di generalizzazione, mantenendo elevate precisione e recall per le classi predominanti.

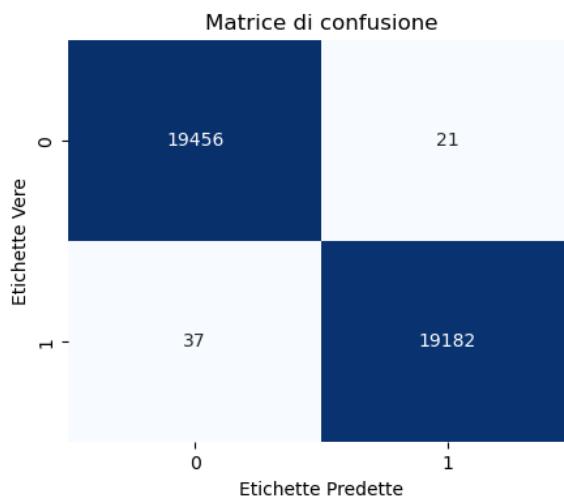


Figura 4.10: Matrice di confusione - classificazione binaria con MLP (NSL-KDD)

Classe	Precision	Recall	F1-Score
BENIGN	100%	100%	1.00
Malicious	100%	100%	1.00
Accuracy	100%		

Tabella 4.14: Report di classificazione binaria con MLP (NSL-KDD)

Classe	Precision	Recall	F1-Score
BENIGN	97.87%	97.09%	0.9747
Malicious	97.11%	97.88%	0.9749
Accuracy	97.48%		

Tabella 4.15: Report di classificazione binaria con cross-validation e MLP (NSL-KDD)

Classe	Precision	Recall	F1-Score
BENIGN	97.02%	98.14%	0.9757
DoS	98.94%	96.83%	0.9787
Probe	94.03%	97.20%	0.9555
R2L	65.92%	52.96%	0.5678
U2R	20.00%	2.00%	0.0364
Accuracy	97.18%		

Tabella 4.16: Report di classificazione multiclasse con cross-validation e MLP (NSL-KDD)

Conclusioni

Dai risultati emerge che *LDA* e i modelli più complessi come *Random Forest* e *MLP* hanno ottenuto le migliori performance nella classificazione, specialmente quella multiclasse. I modelli più semplici, come *Gaussian Naive Bayes* e *Regressione Logistica*, nonostante abbiano avuto buone performance nella classificazione binaria, hanno mostrato difficoltà nel distinguere correttamente le classi meno rappresentate, come *R2L* e *U2R*. Complessivamente, *Random Forest* si è rivelato il modello più robusto, con la capacità di mantenere elevati livelli di accuratezza anche nelle classi più difficili da classificare.

4.7.4 CIC-IDS2017

Nel contesto dell’analisi del dataset CIC-IDS2017, gli attacchi sono stati raggruppati in macro-classi secondo lo schema seguente:

- **Brute Force:** FTP-Patator, SSH-Patator
- **DoS:** DoS slowloris, DoS Slowhttptest, DoS Hulk, DoS GoldenEye, Heartbleed
- **Web Attack:** Web Attack – Brute Force, Web Attack – XSS, Web Attack – Sql Injection
- **Infiltration:** Infiltration
- **DDoS:** DDoS
- **PortScan:** PortScan
- **Bot:** Bot

Questo raggruppamento è stato fatto per semplificare l'analisi multiclassa, accorpando vari tipi di attacchi simili sotto categorie più ampie, per un totale di 7 classi.

Gaussian Naive Bayes

Per la **classificazione binaria**, il modello *Gaussian Naive Bayes* ha ottenuto un'accuracy del 60%. La matrice di confusione mostra una forte asimmetria: la classe *Malicious* ha un recall del 99% ma una precisione del 56%, mentre la classe *BENIGN* ha una precisione del 97% ma un recall molto basso del 22%, risultando in un F1-score di 0.35 per *BENIGN*. Durante la validazione incrociata, l'accuracy è scesa al 54.80%, con un recall per *Malicious* del 94.51% e un recall per *BENIGN* del 15.09%.

Per la **classificazione multiclassa**, il modello *Gaussian Naive Bayes* ha mostrato un'accuracy complessiva molto bassa del 14%. Le performance sono risultate deludenti, con precisioni e recall molto bassi per la maggior parte delle classi, inclusa la classe *BENIGN* e *PortScan*. Tuttavia, la classe *Web Attack* ha mostrato un recall elevato del 92% ma una precisione molto bassa del 2%, indicando un comportamento sbilanciato e inaffidabile del modello nella classificazione multiclassa.

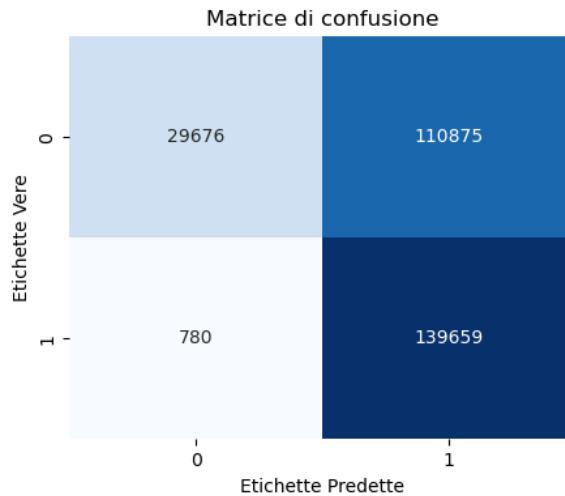


Figura 4.11: Matrice di confusione - classificazione binaria con Naive Bayes (CIC-IDS2017)

Classe	Precision	Recall	F1-Score
BENIGN	97%	22%	0.35
Malicious	56%	99%	0.72
Accuracy	60%		

Tabella 4.17: Report di classificazione binaria con Naive Bayes (CIC-IDS2017)

Classe	Precision	Recall	F1-Score
BENIGN	80.97%	15.09%	0.2464
Malicious	52.66%	94.51%	0.6758
Accuracy	54.80%		

Tabella 4.18: Report di classificazione binaria con cross-validation e Naive Bayes (CIC-IDS2017)

Classe	Precision	Recall	F1-Score
BENIGN	96.87%	7.85%	0.1435
Bot	0.23%	52.16%	0.0046
Brute Force	0.78%	68.45%	0.0155
DDoS	31.52%	82.38%	0.4545
DoS	29.75%	47.57%	0.3649
Infiltration	0.06%	77.78%	0.0011
PortScan	0.29%	0.98%	0.0043
Web Attack	2.53%	92.35%	0.0492
Accuracy	14.76%		

Tabella 4.19: Report di classificazione multiclasse con cross-validation e Naive Bayes (CIC-IDS2017)

Regressione Logistica

Per la **classificazione binaria**, il modello di *Regressione Logistica* ha mostrato un'accuratezza iniziale molto alta del 95%, con precisione e recall ben bilanciati. Tuttavia, durante la validazione incrociata, l'accuratezza è scesa al 76.08% e la precisione per la classe *Malicious* è calata al 67.39%, mentre il recall è sceso al 66.80%, indicando una possibile sensibilità ai cambiamenti nei dati di training.

Per la **classificazione multiclasse**, il modello ha ottenuto un'accuratezza complessiva del 88.83%. Sebbene la classe *BENIGN* abbia ottenuto un buon F1-score di 0.9395, il modello ha mostrato scarse prestazioni nel riconoscere le classi di attacco come *Bot*, *Brute Force*, *PortScan* e *Web Attack*, con un F1-score pari a zero per queste classi. Questo suggerisce una tendenza del modello a favorire la classe *BENIGN*, trascurando le classi di attacco.

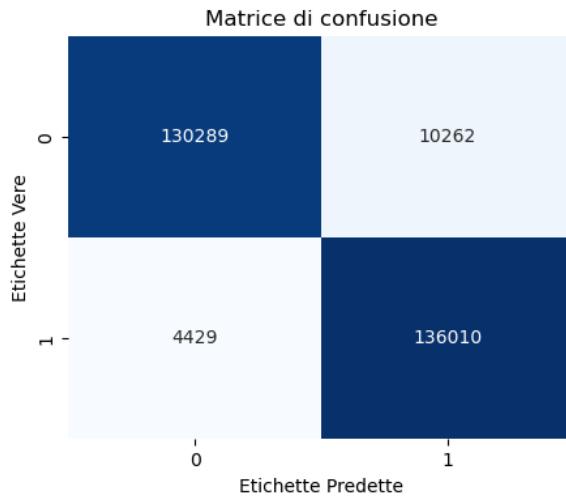


Figura 4.12: Matrice di confusione - classificazione binaria con Regressione Logistica (CIC-IDS2017)

Classe	Precision	Recall	F1-Score
BENIGN	97%	93%	0.95
Malicious	93%	97%	0.95
Accuracy	95%		

Tabella 4.20: Report di classificazione binaria con Regressione Logistica (CIC-IDS2017)

Classe	Precision	Recall	F1-Score
BENIGN	77.55%	85.35%	0.7981
Malicious	67.39%	66.80%	0.6658
Accuracy	76.08%		

Tabella 4.21: Report di classificazione binaria con cross-validation e Regressione Logistica (CIC-IDS2017)

Classe	Precision	Recall	F1-Score
BENIGN	91.69%	96.34%	0.9395
Bot	0.00%	0.00%	0.0000
Brute Force	0.00%	0.00%	0.0000
DDoS	66.99%	68.72%	0.6687
DoS	76.21%	68.56%	0.7155
Infiltration	0.00%	0.00%	0.0000
PortScan	0.00%	0.00%	0.0000
Web Attack	0.00%	0.00%	0.0000
Accuracy	88.83%		

Tabella 4.22: Report di classificazione multiclasse con cross-validation e Regressione Logistica (CIC-IDS2017)

Analisi Discriminante Lineare (LDA)

Nel caso della **classificazione binaria**, LDA ha ottenuto un'accuratezza del 91%. La classe *BENIGN* ha mostrato una precisione del 96% e un recall dell'85%, mentre la classe *Malicious* ha avuto una precisione dell'87% e un recall del 96%. Queste prestazioni indicano una buona capacità di generalizzazione del modello, confermata anche dalla validazione incrociata con un'accuratezza dell'81.42%, suggerendo robustezza.

Per la **classificazione multiclasse**, LDA ha raggiunto un'accuratezza complessiva del 88.95%. Il modello ha mostrato buone prestazioni nella rilevazione della classe *BENIGN* e risultati accettabili per classi come *DoS* (F1-Score 0.7873) e *PortScan* (F1-Score 0.7688). Tuttavia, classi come *Bot* e *Web Attack* hanno presentato difficoltà, con F1-Score molto bassi. Nonostante LDA sia migliore rispetto ad altri modelli nel bilanciare le classi, persistono problemi con le classi meno rappresentate.

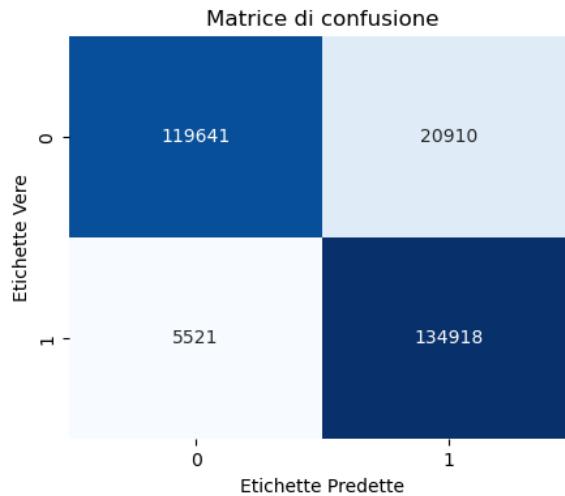


Figura 4.13: Matrice di confusione - classificazione binaria con LDA (CIC-IDS2017)

Classe	Precision	Recall	F1-Score
BENIGN	96%	85%	0.90
Malicious	87%	96%	0.91
Accuracy	91%		

Tabella 4.23: Report di classificazione binaria con LDA (CIC-IDS2017)

Classe	Precision	Recall	F1-Score
BENIGN	85.53%	86.78%	0.8443
Malicious	69.77%	76.06%	0.7244
Accuracy	81.42%		

Tabella 4.24: Report di classificazione binaria con cross-validation e LDA (CIC-IDS2017)

Classe	Precision	Recall	F1-Score
BENIGN	95.87%	92.44%	0.9404
Bot	6.28%	1.18%	0.0064
Brute Force	12.26%	36.79%	0.1832
DDoS	78.46%	56.64%	0.6514
DoS	90.12%	71.46%	0.7873
Infiltration	6.13%	27.78%	0.0667
PortScan	67.01%	98.66%	0.7688
Web Attack	4.16%	84.83%	0.0794
Accuracy	88.95%		

Tabella 4.25: Report di classificazione multiclasse con cross-validation e LDA (CIC-IDS2017)

Random Forest

Nel caso della **classificazione binaria**, il modello *Random Forest* ha raggiunto un'accuratezza molto vicina al 100% (ancora una volta, la tabella 4.26 presenta valori arrotondati). Entrambe le metriche di precisione e recall sono prossime al 100%, dimostrando una straordinaria capacità di distinguere tra campioni benigni e maligni. La matrice di confusione ha evidenziato pochissimi errori di classificazione e la validazione incrociata ha confermato un'accuratezza del 96.88%.

Per la **classificazione multiclasse**, *Random Forest* ha ottenuto un'accuratezza complessiva del 98.66%, mostrando ottime performance su molte classi. Le classi *BENIGN* e *DDoS* hanno registrato F1-Scores molto elevati, rispettivamente 0.9922 e 0.9618, mentre *DoS* e *PortScan* hanno ottenuto punteggi eccellenti del 0.9456 e 0.9859. Anche le classi meno frequenti, come *Infiltration*, hanno mostrato miglioramenti, sebbene il recall per queste classi rimanga relativamente basso al 47.22%.

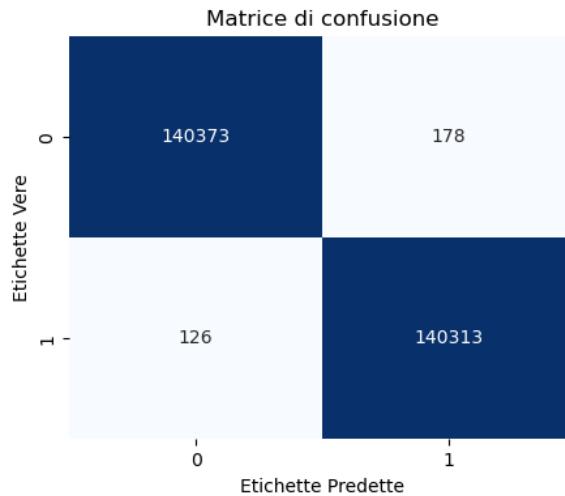


Figura 4.14: Matrice di confusione - classificazione binaria con Random Forest (CIC-IDS2017)

Classe	Precision	Recall	F1-Score
BENIGN	100%	100%	1.00
Malicious	100%	100%	1.00
Accuracy	100%		

Tabella 4.26: Report di classificazione binaria con Random Forest (CIC-IDS2017)

Classe	Precision	Recall	F1-Score
BENIGN	94.47%	99.89%	0.9704
Malicious	99.88%	93.87%	0.9670
Accuracy	96.88%		

Tabella 4.27: Report di classificazione binaria con cross-validation e Random Forest (CIC-IDS2017)

Classe	Precision	Recall	F1-Score
BENIGN	99.07%	99.37%	0.9922
Bot	78.84%	63.03%	0.6177
Brute Force	99.95%	98.19%	0.9905
DDoS	93.20%	99.91%	0.9618
DoS	99.61%	90.23%	0.9456
Infiltration	100.00%	47.22%	0.6151
PortScan	97.80%	99.41%	0.9859
Web Attack	98.83%	95.62%	0.9715
Accuracy	98.66%		

Tabella 4.28: Report di classificazione multiclasse con cross-validation e Random Forest (CIC-IDS2017)

Multi-Layer Perceptron (MLP)

Nel caso della **classificazione binaria**, il modello *Multi-Layer Perceptron (MLP)* ha raggiunto un'accuratezza molto alta del 99%, con performance eccellenti per entrambe le classi e precisione e recall vicini al 99%. Tuttavia, durante la validazione incrociata, l'accuratezza è scesa al 68.99%, con una significativa riduzione del recall per la classe *BENIGN* (46.95%), indicando possibili difficoltà nel generalizzare in contesti diversi.

Per la **classificazione multiclasse**, il *MLPClassifier* ha ottenuto un'accuratezza complessiva del 83.16%, mostrando buone prestazioni nella classificazione della classe *BENIGN* (F1-Score 0.9080). Tuttavia, ha avuto seri problemi nel riconoscere la maggior parte delle classi di attacco, come *Bot*, *Brute Force* e *DDoS*, con un F1-Score pari a zero per queste classi. L'unica eccezione è *PortScan*, con un F1-Score molto basso del 0.0276. Questi risultati indicano che il modello non è riuscito a generalizzare adeguatamente per le classi di attacco.

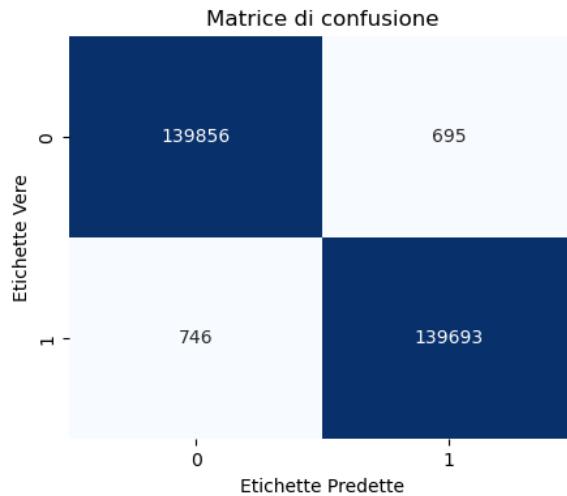


Figura 4.15: Matrice di confusione - classificazione binaria con MLP (CIC-IDS2017)

Classe	Precision	Recall	F1-Score
BENIGN	100%	99%	0.99
Malicious	99%	100%	0.99
Accuracy	99%		

Tabella 4.29: Report di classificazione binaria con MLP (CIC-IDS2017)

Classe	Precision	Recall	F1-Score
BENIGN	91.55%	46.95%	0.5850
Malicious	63.74%	91.03%	0.7409
Accuracy	68.99%		

Tabella 4.30: Report di classificazione binaria con cross-validation e MLP (CIC-IDS2017)

Classe	Precision	Recall	F1-Score
BENIGN	83.15%	99.99%	0.9080
Bot	0.00%	0.00%	0.0000
Brute Force	0.00%	0.00%	0.0000
DDoS	0.00%	0.00%	0.0000
DoS	0.00%	0.00%	0.0000
Infiltration	0.00%	0.00%	0.0000
PortScan	29.26%	1.45%	0.0276
Web Attack	0.00%	0.00%	0.0000
Accuracy	83.16%		

Tabella 4.31: Report di classificazione multiclasse con cross-validation e MLP (CIC- IDS2017)

Conclusioni

Dai risultati ottenuti, emerge chiaramente, ancora una volta, che il modello *RandomForestClassifier* è il più performante nella classificazione binaria e multiclasse sul dataset CIC-IDS2017, riuscendo a mantenere un buon equilibrio tra le varie classi, incluse quelle di attacco. Al contrario, modelli come *GaussianNB* e *MLPClassifier* hanno mostrato performance inadeguate per questo compito, con difficoltà significative nel rilevamento delle classi di attacco.

4.7.5 Confronto delle Performance in Letteratura

4.7.6 NSL-KDD

Il confronto tra gli algoritmi comuni sviluppati nella tesi e quelli riportati in letteratura evidenzia delle differenze in termini di performance, soprattutto per quanto riguarda la capacità di classificazione e l'accuratezza generale.

Per il **Random Forest**, l'accuratezza raggiunta è stata del 99.95%, mentre in letteratura è riportata un'accuratezza del 92.79%. Questo indica un miglioramento nel modello sviluppato, probabilmente dovuto a una migliore ottimizzazione del

modello e dei suoi iperparametri, oltre a un eventuale affinamento nella gestione del dataset e nella selezione delle feature rilevanti.

Nel caso del **Multi-Layer Perceptron (MLP)**, l'accuratezza ottenuta nella tesi è stata del 97.18%, superando la performance del 92.26% riportata in letteratura. Questo miglioramento può essere attribuito a una migliore configurazione della rete neurale, probabilmente con più livelli o un diverso schema di allenamento, che ha permesso al modello di catturare meglio le caratteristiche complesse del dataset.

Per quanto riguarda **Linear Discriminant Analysis (LDA)** e **Regessione Logistica**, i risultati ottenuti nella tesi sono coerenti con quanto riportato nel paper "*Intrusion Detection Systems using Linear Discriminant Analysis and Logistic Regression*" [3]. In particolare, LDA ha raggiunto un'accuratezza del 97.39%, mentre la Regressione Logistica ha ottenuto un'accuratezza dell'84.18%. Questi risultati riflettono quanto osservato nel paper, che prevede un'accuracy molto elevata in questo contesto.

Infine, per l'algoritmo **Naive Bayes**, la versione utilizzata nella tesi (Gaussian Naive Bayes) ha ottenuto un'accuratezza del 38.83%, significativamente inferiore rispetto all'81.66% riportato in letteratura. Questo evidenzia una forte limitazione del modello sviluppato nel trattare un problema multiclass complesso come quello presente nel dataset NSL-KDD, dimostrando che non è ideale per questo tipo di dati quando non viene adeguatamente adattato.

In conclusione, va sottolineato che un confronto accurato tra i modelli sviluppati nella tesi e quelli riportati nella letteratura non può essere effettuato in modo completo, a causa delle difficoltà riscontrate nella classificazione delle classi più rare, come *R2L* e *U2R*. Questi risultati sono stati particolarmente scarsi per la maggior parte degli algoritmi, ad eccezione di Random Forest, che ha mostrato prestazioni elevate su tutte le classi. Questo squilibrio nelle prestazioni rende difficile un confronto diretto e significativo con i risultati della letteratura, dove generalmente vengono riportate performance migliori su tutte le classi del dataset NSL-KDD.

4.7.7 CIC-IDS2017

Anche nel confronto tra i risultati ottenuti nella letteratura per il dataset CIC-IDS2017 e quelli sviluppati nella tesi, emergono alcune differenze nelle prestazioni

degli algoritmi.

Random Forest, ad esempio, ha mostrato una precisione del 98.66% nel modello sviluppato nella tesi, che è paragonabile ai risultati riportati nel paper di Sharafaldin [9], dove ha raggiunto una precisione del 98%. Tuttavia, nella tesi, il modello ha evidenziato maggiori difficoltà nel gestire le classi meno rappresentate come *Infiltration*, dove il recall è sceso al 47.22%.

Anche il **Multi-Layer Perceptron (MLP)** ha avuto un comportamento simile: in letteratura, il modello ha ottenuto un F1-score di 0.76, mentre nella tesi ha raggiunto un'accuratezza complessiva dell'83.16%. Tuttavia, in quest'ultimo si è riscontrata una difficoltà maggiore nel classificare correttamente le classi di attacco, con F1-scores pari a zero per classi come Bot, Brute Force, e DDoS, segnalando problemi di generalizzazione che non sono stati evidenziati nella letteratura.

Infine, anche in questo caso, sia la **Regressione Logistica** che l'**Analisi Discriminante Lineare (LDA)** hanno mostrato risultati coerenti con quanto riportato nel paper *"Intrusion Detection Systems using Linear Discriminant Analysis and Logistic Regression"* [3], dove si prevedeva un'accuratezza molto elevata. Entrambi i modelli hanno raggiunto un'accuratezza superiore all'88%, ma hanno presentato difficoltà con le classi di attacco più rare, come Bot e Web Attack, confermando che il loro limite principale risiede nella gestione delle classi meno rappresentate.

Queste differenze, soprattutto per le classi meno comuni, suggeriscono che i modelli sviluppati non hanno gestito interamente la complessità del dataset, e le loro performance non sono paragonabili a quelle riportate nella letteratura, dove le classi sono state riconosciute con maggiore equilibrio e accuratezza.

4.8 Feature Selection con XGBoost

Nel processo di sviluppo dei modelli, la selezione delle feature è una fase cruciale che influisce direttamente sulla performance e sull'efficienza del modello. Una selezione accurata delle feature consente di migliorare la qualità delle predizioni e ridurre il tempo di addestramento, evitando l'overfitting e migliorando l'interpretabilità del modello. In questo progetto, la tecnica di Feature Selection è stata realizzata utilizzando il modello XGBoost.

XGBoost (Extreme Gradient Boosting) è un potente algoritmo di boosting basato su alberi decisionali che viene utilizzato non solo per l’addestramento del modello, ma anche per valutare l’importanza delle feature nel dataset. Questo modello fornisce una misura di importanza per ciascuna feature, che riflette quanto una di esse contribuisce al processo di previsione del modello.

Per determinare quali feature selezionare, è stato stabilito un soglia di importanza dell’1% per garantire che solo le più rilevanti siano incluse.

Nel dataset NSL-KDD, il processo di feature selection con XGBoost ha portato alla selezione di 18 feature significative su un totale di 123 (Figura 4.32). Invece, nel dataset CIC-IDS2017, ne sono state selezionate 21 su un totale di 78 (Figura 4.33).

Feature	Importanza
src_bytes	26.24%
dst_bytes	13.31%
service_ftp_data	9.17%
service_ecr_i	7.04%
service_http	7.00%

Tabella 4.32: Prime 5 feature per importanza (NSL-KDD)

Feature	Importanza
Bwd Packet Length Std	15.75%
Average Packet Size	10.13%
Avg Bwd Segment Size	9.37%
Bwd Header Length	8.26%
Fwd Packet Length Max	7.95%

Tabella 4.33: Prime 5 feature per importanza (CIC-IDS2017)

4.8.1 Differenze nelle Performance

Le tabelle qui presenti mostrano un confronto delle accuratezze dei modelli di classificazione binaria sui dataset NSL-KDD e CIC-IDS2017 prima e dopo l’uso della feature selection con XGBoost. Le performance sono state valutate utilizzando la tecnica stratified K-fold cross validation, assicurando così che i risultati riflettano un’accurata stima delle capacità di generalizzazione dei modelli.

La feature selection ha portato a una riduzione della dimensionalità dei dati, il che ha comportato un vantaggio significativo in termini di tempo di training. I modelli hanno infatti richiesto meno tempo per essere addestrati sui dataset ridotti, senza

compromettere significativamente le loro performance, sebbene in alcuni casi si siano registrati leggeri decrementi di accuratezza.

Nel dataset NSL-KDD, per esempio, il modello *RandomForestClassifier* ha mostrato solo una leggera riduzione dell'accuratezza (dal 99.96% al 99.94%). Anche altri modelli, come *GaussianNB* e *LogisticRegression* hanno subito variazioni di accuratezza trascurabili rispetto ai benefici offerti.

Modello	Accuratezza Prima (%)	Accuratezza Dopo (%)
GaussianNB	50.06%	49.84%
LogisticRegression	87.54%	87.89%
LDA	98.39%	95.56%
RandomForest	99.96%	99.94%
MLP	97.48%	94.96%

Tabella 4.34: Confronto delle accuratezze prima e dopo l'uso di XGBoost per NSL-KDD

Anche nel caso del dataset CIC-IDS2017, l'uso di XGBoost ha comportato una riduzione del tempo di training, ma in alcuni modelli, come *LogisticRegression* e *MLPClassifier*, si è osservato un calo di accuratezza più significativo rispetto al dataset NSL-KDD. Tuttavia, *RandomForestClassifier* ha mantenuto un'accuratezza elevata, con una leggera diminuzione dal 96.88% al 96.39%, confermando la robustezza del modello.

Modello	Accuratezza Prima (%)	Accuratezza Dopo (%)
GaussianNB	54.80%	53.38%
LogisticRegression	76.08%	67.95%
LDA	81.42%	74.58%
RandomForest	96.88%	96.39%
MLP	68.99%	64.77%

Tabella 4.35: Confronto delle accuratezze prima e dopo l'uso di XGBoost per CIC-IDS2017

4.9 GridSearch

La ricerca dei migliori iperparametri per un modello di Machine Learning rappresenta un ulteriore passo nel processo di ottimizzazione delle performance. In questo contesto, la tecnica di Grid Search si distingue come uno degli approcci più utilizzati e sistematici per identificare la combinazione ottimale di iperparametri.

In particolare, Grid Search esplora esaustivamente tutte le possibili combinazioni di valori definiti all'interno di un intervallo specificato per ciascun iperparametro del modello. Sebbene questo approccio possa risultare computazionalmente costoso per spazi di ricerca di grandi dimensioni, offre il vantaggio di garantire che tutte le possibili configurazioni siano prese in considerazione, evitando il rischio di ignorare potenziali combinazioni ottimali.

4.9.1 Parametri e Modalità di Testing

I parametri testati con la tecnica di Grid Search sono:

- **Architettura della rete (hidden_layer_sizes):** sono state provate diverse configurazioni relative alla dimensione e al numero di hidden layers:
 - (20,): Una singola rete con 20 neuroni.
 - (50,): Una singola rete con 50 neuroni.
 - (20, 20): Due hidden layers con 20 neuroni ciascuno.
 - (50, 20): Due hidden layers con 50 e 20 neuroni rispettivamente.
 - (50, 50): Due hidden layers con 50 neuroni ciascuno.
- **Funzione di attivazione (activation):** sono state valutate due funzioni di attivazione:
 - *ReLU* (Rectified Linear Unit), comunemente usata nelle reti neurali "profonde" per la sua capacità di accelerare il processo di addestramento.
 - *Tanh*, un'alternativa alla ReLU che mappa gli input in un intervallo compreso tra -1 e 1.

- **Ottimizzatore (solver)**: sono stati testati due algoritmi per l’aggiornamento dei pesi:
 - *Adam* (Adaptive Moment Estimation), noto per la sua efficienza e velocità di convergenza.
 - *SGD* (Stochastic Gradient Descent), che effettua aggiornamenti dei pesi più graduali, richiedendo generalmente un numero maggiore di iterazioni per la convergenza.

Combinando queste variabili, la Grid Search ha esplorato un totale di 20 configurazioni differenti per ogni dataset. Ogni combinazione è stata valutata sul task di classificazione binaria del modello MLP utilizzando la validazione incrociata a 3 fold (K-fold cross-validation), al fine di garantire la generalizzabilità dei risultati.

4.9.2 Confronto delle Prestazioni

Per il **dataset NSL-KDD**, la miglior configurazione ottenuta tramite Grid Search è stata ‘activation’: ‘relu’, ‘hidden_layer_sizes’: (50, 50), ‘solver’: ‘adam’, che ha raggiunto un punteggio medio di accuracy pari al 99.71%, con un tempo di addestramento relativamente lungo di 33 secondi. In generale, l’attivazione relu ha prodotto migliori risultati rispetto a tanh, soprattutto con architetture più complesse come (50, 50).

Nel caso del **dataset CIC-IDS2017**, la configurazione ottimale risultante è stata ‘activation’: ‘tanh’, ‘hidden_layer_sizes’: (50, 50), ‘solver’: ‘adam’, che ha ottenuto un punteggio medio di accuracy pari al 99.31%, ma con un tempo di addestramento più elevato pari a 299 secondi.

In sintesi, i risultati evidenziano come l’uso di configurazioni più complesse (ad esempio, reti con più hidden layers) e l’ottimizzatore adam tendano a produrre le migliori performance in termini di accuratezza, a scapito però di un aumento considerevole del tempo di addestramento. Di seguito sono riportate le tabelle riassuntive delle performance delle varie combinazioni di parametri per entrambi i dataset.

Activation	Hidden Layer Sizes	Solver	Mean Test Score (%)	Fit Time (s)
relu	(20,)	adam	99.56%	6.08
relu	(20,)	sgd	99.14%	5.94
relu	(50,)	adam	99.61%	24.92
relu	(50,)	sgd	99.12%	23.49
relu	(20, 20)	adam	99.66%	12.38
relu	(20, 20)	sgd	99.21%	6.68
relu	(50, 20)	adam	99.68%	12.03
relu	(50, 20)	sgd	99.34%	9.25
relu	(50, 50)	adam	99.71%	33.71
relu	(50, 50)	sgd	99.35%	33.00
tanh	(20,)	adam	99.61%	6.31
tanh	(20,)	sgd	98.99%	5.47
tanh	(50,)	adam	99.62%	19.77
tanh	(50,)	sgd	99.12%	30.44
tanh	(20, 20)	adam	99.66%	9.17
tanh	(20, 20)	sgd	99.27%	7.93
tanh	(50, 20)	adam	99.65%	10.43
tanh	(50, 20)	sgd	99.31%	11.28
tanh	(50, 50)	adam	99.66%	50.00
tanh	(50, 50)	sgd	99.29%	43.48

Tabella 4.36: Risultati della Grid Search per NSL-KDD (in grassetto i parametri con le prestazioni migliori)

Activation	Hidden Layer Sizes	Solver	Mean Test Score (%)	Fit Time (s)
relu	(20,)	adam	98.50%	95.44
relu	(20,)	sgd	96.14%	75.09
relu	(50,)	adam	98.56%	326.99
relu	(50,)	sgd	96.25%	300.91
relu	(20, 20)	adam	99.10%	111.11
relu	(20, 20)	sgd	97.76%	109.46
relu	(50, 20)	adam	99.15%	92.82
relu	(50, 20)	sgd	97.98%	136.56
relu	(50, 50)	adam	99.24%	394.39
relu	(50, 50)	sgd	98.10%	328.03
tanh	(20,)	adam	98.44%	60.74
tanh	(20,)	sgd	96.10%	53.74
tanh	(50,)	adam	99.11%	197.17
tanh	(50,)	sgd	95.96%	177.55
tanh	(20, 20)	adam	99.00%	75.73
tanh	(20, 20)	sgd	97.30%	75.18
tanh	(50, 20)	adam	99.23%	65.53
tanh	(50, 20)	sgd	97.43%	93.87
tanh	(50, 50)	adam	99.31%	299.62
tanh	(50, 50)	sgd	97.59%	355.33

Tabella 4.37: Risultati della Grid Search per CIC-IDS2017 (in grassetto i parametri con le prestazioni migliori)

4.9.3 Scelta della Configurazione

Si è scelto di utilizzare la combinazione di parametri `hidden_layer_sizes=(20, 20)`, `activation="relu"`, e `solver="adam"` per il training finale, dopo un'attenta analisi dei

risultati ottenuti durante la Grid Search.

I risultati hanno evidenziato che la rete neurale con questi parametri ha raggiunto punteggi di accuratezza molto elevati su entrambi i dataset utilizzati: NSL-KDD e CIC-IDS2017, rispettivamente del 99.66% e 99.10%.

L'uso della funzione di attivazione ReLU ha contribuito significativamente a questo successo, grazie alla sua capacità di introdurre non-linearietà senza incorrere nei problemi di saturazione che altre funzioni di attivazione potrebbero presentare. Inoltre, il solver Adam ha dimostrato di essere particolarmente efficace nella gestione dei grandi dataset, garantendo una convergenza rapida e tempi di addestramento più brevi rispetto ad altri metodi. La scelta di una configurazione con *hidden_layer_sizes*=(20, 20) si è rivelata un compromesso ideale. Pur offrendo risultati di alta qualità, ha evitato il costo computazionale eccessivo associato a reti neurali con più unità nascoste, come quelle con dimensioni (50, 50).

In definitiva, questa configurazione ha dimostrato di offrire un eccellente equilibrio tra prestazioni di classificazione e tempi di addestramento, soddisfacendo così le esigenze del progetto in modo ottimale.

CAPITOLO 5

Architettura dell'IDS

5.1 Introduzione all'architettura

L'architettura proposta in vista di uno sviluppo futuro integra un Intrusion Detection System (IDS) con un Software Defined Network (SDN), sfruttando la capacità di programmare dinamicamente il comportamento della rete attraverso un controller centrale. Il controller SDN, in questo contesto, funge da strumento per la raccolta, l'analisi e la gestione dei flussi di traffico, che vengono poi esaminati dal modulo IDS.

Inoltre, l'integrazione del machine learning nell'architettura permette di analizzare il traffico in modo intelligente, sfruttando modelli di apprendimento automatico per rilevare anomalie e attacchi non ancora conosciuti.

5.2 Implementazione della SDN con Ryu

Ryu è un framework open-source per la gestione delle reti SDN che offre una piattaforma altamente programmabile e flessibile per il controllo centralizzato del traffico. Nell'architettura menzionata, svolge un ruolo cruciale nel coordinare il

flusso dei dati tra i diversi dispositivi di rete, monitorando costantemente i pacchetti e applicando regole di controllo definite dal sistema di sicurezza.

Uno dei vantaggi principali del *Ryu Controller* è la sua capacità di interfacciarsi facilmente con il modulo IDS, fornendo una visione globale e in tempo reale della rete. Questo consente all'IDS di ricevere informazioni dettagliate sui flussi di traffico, come la frequenza, la latenza e le anomalie, permettendo una rilevazione più accurata delle intrusioni.

Attraverso il *Ryu*, è possibile implementare politiche di sicurezza dinamiche, che reagiscono in tempo reale alle minacce identificate, bloccando o modificando i flussi di traffico sospetti direttamente sul piano dati. Questo garantisce un alto livello di flessibilità e adattabilità nella difesa contro attacchi sofisticati.

```
from ryu.base import app_manager
from ryu.controller import ofp_event
from ryu.controller.handler import MAIN_DISPATCHER
from ryu.controller.handler import set_ev_cls
from ryu.ofproto import ofproto_v1_0

class L2Switch(app_manager.RyuApp):
    OFP_VERSIONS = [ofproto_v1_0.OFP_VERSION]

    def __init__(self, *args, **kwargs):
        super(L2Switch, self).__init__(*args, **kwargs)

    @set_ev_cls(ofp_event.EventOFPPacketIn, MAIN_DISPATCHER)
    def packet_in_handler(self, ev):
        msg = ev.msg
        dp = msg.datapath
        ofp = dp.ofproto
        ofp_parser = dp.ofproto_parser

        actions = [ofp_parser.OFPActionOutput(ofp.OFPP_FLOOD)]

        data = None
        if msg.buffer_id == ofp.OFP_NO_BUFFER:
            data = msg.data

        out = ofp_parser.OFPPacketOut(
            datapath=dp, buffer_id=msg.buffer_id, in_port=msg.in_port,
            actions=actions, data = data)
        dp.send_msg(out)
```

Figura 5.1: Esempio di codice python che implementa uno switch L2 con Ryu controller in un ambiente SDN

5.3 Modulo di Machine Learning

Il modulo di Machine Learning nell’architettura rappresenta il cuore del sistema di rilevamento degli attacchi, sfruttando algoritmi avanzati per l’analisi del traffico di rete e la classificazione delle anomalie. In questo contesto, il modello si occupa di identificare pattern sospetti nel traffico, distinguendo tra attività legittime e potenziali minacce. L’integrazione del Machine Learning in un’architettura SDN permette di beneficiare della flessibilità e programmabilità delle SDN per raccogliere dati di traffico in tempo reale e analizzarli in modo efficiente.

5.3.1 Scelta dei Modelli di Machine Learning

Per il rilevamento degli attacchi, due modelli principali si sono dimostrati particolarmente efficaci: **random forest** e **reti neurali**.

Il modello Random Forest, un algoritmo di apprendimento supervisionato, è altamente performante nei compiti di classificazione grazie alla sua struttura basata su una combinazione di più alberi decisionali. Questa architettura lo rende robusto contro l’overfitting e ben adatto a gestire dataset complessi come quelli relativi al traffico di rete. Un altro punto di forza è la sua capacità di interpretare i feature più rilevanti, un aspetto cruciale in ambito sicurezza.

Le reti neurali, in particolare le *Deep Neural Networks* (DNN), offrono una capacità superiore di modellare relazioni non lineari e catturare pattern intricati nel traffico di rete. Grazie alla loro flessibilità e capacità di apprendimento, si sono dimostrate estremamente efficaci in questo contesto. Tuttavia, il loro utilizzo richiede più risorse computazionali rispetto a modelli più semplici.

Entrambi i modelli forniscono vantaggi significativi: le reti neurali eccellono nel riconoscere pattern complessi, mentre le Random Forest offrono efficienza e interpretabilità.

5.3.2 Pipeline di addestramento e predizione

La pipeline di Machine Learning nell’architettura segue diversi passaggi cruciali per l’addestramento e la predizione. Il processo inizia con la raccolta del traffico di

rete, monitorato tramite il controller SDN come Ryu, dove i pacchetti sono etichettati e organizzati per l’addestramento. Successivamente, i dati vengono pre-processati, includendo operazioni come la rimozione di valori mancanti, la normalizzazione e il *one-hot encoding* delle feature categoriche. In seguito, avviene l’estrazione e selezione delle feature rilevanti del traffico di rete, utilizzando tecniche come XGBoost, per ridurre la complessità e migliorare le prestazioni del modello.

I modelli di Machine Learning, come random forest e reti neurali, vengono poi addestrati per imparare a distinguere tra traffico legittimo e malevolo. Una volta addestrati, i modelli vengono valutati su un set di test per misurare accuratezza, precisione, richiamo e F1-score, scegliendo il modello migliore in base alle prestazioni. Infine, il modello selezionato viene implementato per analizzare il traffico in tempo reale. Dato che il traffico evolve, è previsto un aggiornamento periodico del modello attraverso nuovi dati, garantendo l’efficacia del sistema nel tempo.

5.3.3 Reazione dinamica agli attacchi basata su ML

Uno dei principali vantaggi dell’integrazione tra IDS, SDN e Machine Learning è la capacità di reagire dinamicamente agli attacchi. Questa architettura consente un ciclo continuo di rilevamento, classificazione e risposta alle minacce quasi in tempo reale.

Grazie ai modelli di Machine Learning, che sono stati addestrati su grandi quantità di dati, l’analisi del traffico di rete avviene in pochi millisecondi, permettendo di individuare comportamenti anomali o potenzialmente dannosi con alta precisione. Quando viene rilevato un attacco, l’architettura consente di implementare azioni dinamiche per contrastarlo. Il controller SDN può, ad esempio, modificare le regole di inoltro degli switch per bloccare il traffico sospetto o limitare le connessioni a determinate risorse di rete. Inoltre, con l’afflusso di nuovi dati, i modelli vengono costantemente aggiornati, migliorando la loro capacità di rilevare e rispondere a nuove minacce.

In conclusione, l’integrazione del Machine Learning non solo consente una difesa proattiva e adattabile alle reti moderne, ma ottimizza anche la velocità e l’efficacia

della risposta, garantendo una protezione avanzata contro le minacce informatiche in evoluzione.

5.4 Funzionamento del Flusso di Dati

Il flusso di dati tra l’SDN e l’IDS segue un processo ben definito:

1. **Monitoraggio del traffico di rete:** Gli switch SDN, configurati attraverso il controller SDN (ad esempio Ryu), monitorano il traffico di rete e inviano regolarmente i dati al modulo IDS. Questo traffico viene raccolto a livello di flusso, permettendo una granularità dettagliata nella visibilità e analisi del traffico.
2. **Inoltro dei dati all’IDS:** Il controller SDN riceve le informazioni sui flussi di traffico dagli switch e li trasmette all’IDS per un’analisi approfondita. In questa fase, vengono selezionati i flussi più rilevanti per ridurre il carico computazionale sull’IDS.
3. **Analisi tramite Machine Learning:** Il modulo di Machine Learning dell’IDS analizza i flussi di dati ricevuti, cercando pattern che possano indicare attività malevoli. A seconda del modello utilizzato (ad esempio, Random Forest o Reti Neurali), i flussi vengono classificati come legittimi o potenzialmente malevoli.
4. **Decisione e risposta:** Se un flusso di traffico viene identificato come sospetto, l’IDS notifica il controller SDN, il quale applica dinamicamente politiche di rete per mitigare la minaccia. Questo può includere il blocco del traffico, il reindirizzamento a una sandbox per un’analisi approfondita o la limitazione della larghezza di banda.
5. **Feedback continuo:** Il sistema è in grado di apprendere continuamente attraverso il feedback che riceve dal traffico analizzato. I modelli di Machine Learning possono essere aggiornati con nuovi dati per migliorare l’accuratezza nel rilevamento delle minacce.

Questa struttura offre un’architettura scalabile e reattiva, in grado di adattarsi rapidamente ai cambiamenti nel traffico di rete e di rispondere in tempo reale a possibili intrusioni

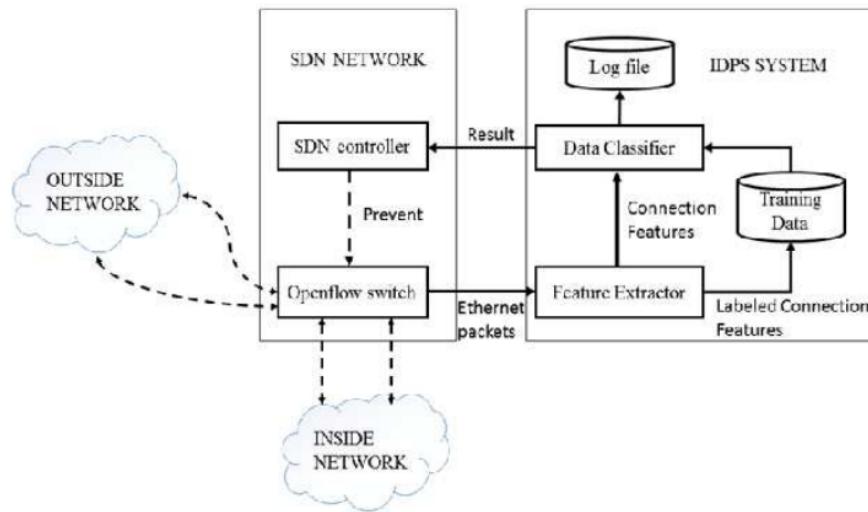


Figura 5.2: Esempio di schema di architettura IDS/SDN con classificatore [1].

5.5 Generazione del traffico

La generazione di traffico di rete è una componente essenziale per testare e valutare le prestazioni di un IDS in ambienti SDN. Tecniche avanzate di simulazione del traffico permettono di riprodurre scenari realistici in cui l'IDS può essere valutato sia in termini di efficienza nel rilevamento delle minacce sia in termini di reattività a diverse tipologie di attacchi.

5.5.1 Strumenti per la generazione di traffico

Nel contesto delle SDN, diversi strumenti possono essere utilizzati per generare traffico di rete, ognuno con le sue peculiarità. Tra gli strumenti più utilizzati vi sono Mininet e D-ITG, che forniscono capacità avanzate per la simulazione e l'emulazione del traffico di rete.

Mininet

Mininet è uno strumento ampiamente utilizzato per emulare reti SDN, fornendo un ambiente leggero e scalabile che consente di simulare topologie di rete complesse. Permette di creare reti virtuali su un singolo computer, simulando in modo realistico

host, switch, controller e link. È particolarmente utile per simulare ambienti SDN, gestendo dinamicamente il traffico tramite controller come Ryu.

D-ITG (Distributed Internet Traffic Generator)

D-ITG è uno strumento avanzato per la generazione di traffico, capace di simulare vari tipi di applicazioni come HTTP, VoIP e FTP. Grazie alla sua capacità di generare traffico in modo distribuito, è utile per simulare reti su larga scala e replicare traffico realistico. Integrato con SDN, D-ITG permette di testare le capacità di rilevamento di anomalie e attacchi da parte del sistema, generando traffico complesso e altamente configurabile che riflette sia condizioni normali che situazioni di attacco.

5.5.2 Simulazione di attacchi e comportamento anomalo

La simulazione di attacchi e comportamenti anomali è un componente essenziale per il testing e il miglioramento degli IDS. In un ambiente di rete simulato, è possibile replicare una vasta gamma di scenari di attacco, come attacchi di tipo DDoS, scansioni di porte, iniezioni di codice, tentativi di accesso non autorizzato e attacchi basati sul traffico anomalo. Utilizzando un'infrastruttura SDN con il supporto di strumenti come Mininet e D-ITG, è possibile generare in modo controllato sia traffico legittimo che malevolo, simulando una rete reale.

Un approccio comunemente utilizzato è quello di generare traffico normale su cui vengono iniettati flussi malevoli o anomali. Ad esempio, è possibile simulare un attacco DDoS inviando un numero elevato di richieste a un singolo host, osservando come l'IDS rileva il comportamento anomalo e interviene per mitigare l'impatto. La simulazione può anche includere varianti di attacchi più sofisticati, come attacchi mirati che imitano flussi legittimi per sfuggire al rilevamento.

La possibilità di controllare ogni aspetto del traffico generato permette di testare non solo le capacità di rilevamento dell'IDS, ma anche la sua resilienza e la capacità di gestire attacchi complessi o su larga scala. Inoltre, attraverso l'uso di tecniche di machine learning integrate nell'IDS, è possibile monitorare come il sistema reagisce a nuovi tipi di attacchi non precedentemente conosciuti, valutando l'efficacia del modello nell'adattarsi a comportamenti imprevisti.

5.5.3 Vantaggi e sfide nella generazione di traffico

La generazione di traffico in un ambiente simulato offre numerosi vantaggi per il testing di un IDS. In primo luogo, consente di creare scenari di traffico altamente personalizzati, in cui è possibile controllare ogni aspetto del flusso, dalla topologia di rete al comportamento dei nodi. Questo permette di eseguire test mirati e riproducibili, facilitando il processo di validazione e miglioramento delle prestazioni dell'IDS.

Un altro vantaggio importante è la possibilità di simulare condizioni di rete reali senza influenzare reti di produzione. Questo aspetto è cruciale per eseguire test su larga scala o in presenza di attacchi distruttivi, come attacchi DDoS o infezioni malware, senza rischiare di interrompere i servizi aziendali o compromettere la sicurezza delle informazioni.

Tuttavia, la generazione di traffico per il testing di IDS presenta anche diverse sfide. Una delle principali difficoltà è la creazione di dataset realistici e sufficientemente diversificati. Se il traffico simulato non rappresenta adeguatamente la complessità del traffico di rete reale, i risultati dei test potrebbero non riflettere accuratamente le prestazioni dell'IDS in ambienti di produzione. Inoltre, la generazione di traffico malevolo particolarmente sofisticato o di attacchi zero-day richiede un'elevata competenza tecnica e una profonda conoscenza dei vettori di attacco più recenti.

Un'altra sfida è la necessità di bilanciare il traffico legittimo con quello anomalo, in modo da evitare un bias nei modelli di machine learning. In un contesto di test, l'eccessiva concentrazione di traffico malevolo potrebbe portare a risultati distorti, sovrastimando la capacità dell'IDS di rilevare minacce. È fondamentale quindi creare un ambiente di test equilibrato, in cui l'IDS possa essere valutato sia nella sua capacità di rilevare attacchi che nel ridurre i falsi positivi durante l'analisi del traffico legittimo.

Dunque, la generazione di traffico per il testing degli IDS rappresenta uno strumento potente, ma richiede un'attenta progettazione per garantire la validità dei test e la riproducibilità dei risultati, soprattutto in presenza di attacchi sofisticati.

5.6 Vantaggi e Limitazioni dell’architettura

L’architettura proposta, che integra tecniche di ML con l’infrastruttura di una rete a controllo centralizzato, offre numerosi vantaggi, ma presenta anche alcune limitazioni che devono essere considerate.

Uno dei principali vantaggi di questa architettura è la flessibilità e centralizzazione del controllo della rete, garantita dall’uso di SDN. Grazie al Ryu Controller, è possibile monitorare e gestire in modo dinamico il traffico di rete, applicando politiche di sicurezza in tempo reale e adattando il comportamento della rete in risposta a minacce emergenti. Questo riduce il tempo di reazione e aumenta la capacità di prevenire intrusioni su larga scala.

L’integrazione con tecniche di machine learning consente di migliorare notevolmente la capacità di rilevamento degli attacchi. Algoritmi come le reti neurali e le random forest offrono ottime performance nell’identificazione di pattern anomali nel traffico, riducendo i falsi positivi rispetto ai sistemi tradizionali basati su regole statiche. Inoltre, l’uso del machine learning permette di adattare l’IDS a nuove minacce, migliorando continuamente la precisione del rilevamento grazie al retraining periodico sui nuovi dati raccolti.

La scalabilità dell’architettura è un altro vantaggio significativo. Con l’uso di SDN, è possibile gestire reti di dimensioni variabili senza compromettere la capacità di monitorare e analizzare il traffico. La modularità del sistema, inoltre, permette l’integrazione di nuovi componenti senza dover ricostruire l’intera infrastruttura.

D’altro canto, una delle principali limitazioni è la complessità nella configurazione e manutenzione. L’integrazione tra SDN e ML richiede una profonda conoscenza sia delle reti che delle tecniche di intelligenza artificiale. Il training dei modelli di machine learning richiede inoltre dataset ben curati e bilanciati, che possono non essere sempre disponibili, soprattutto in ambienti reali.

Un altro limite è la dipendenza dalla qualità del dataset utilizzato per l’addestramento del modulo di machine learning. Se il dataset non è sufficientemente rappresentativo del traffico reale o non copre una gamma sufficientemente ampia di scenari di attacco, l’IDS può non essere in grado di rilevare efficacemente nuove minacce o attacchi complessi.

Inoltre, il tempo di latenza può rappresentare un problema, specialmente in reti su larga scala o con un elevato volume di traffico. L’analisi in tempo reale di grandi quantità di dati può richiedere risorse computazionali significative, e l’implementazione di misure di sicurezza basate su machine learning potrebbe introdurre ritardi nel flusso di rete.

Infine, un’altra limitazione riguarda la vulnerabilità della SDN stessa. Essendo un’infrastruttura centralizzata, il controller SDN diventa un punto critico e può essere un obiettivo di attacchi mirati. Un attacco diretto al controller potrebbe compromettere l’intero sistema di sicurezza, inclusa l’efficacia dell’IDS.

CAPITOLO 6

Conclusioni

Il presente lavoro di tesi ha approfondito l'uso delle tecniche di machine learning per lo sviluppo e miglioramento di un sistema di rilevamento delle intrusioni (IDS), con un focus su cinque modelli principali: Naive Bayes, Regressione Logistica, Analisi Discriminante Lineare, Reti Neurali e Random Forest. La valutazione dei modelli è stata effettuata utilizzando i dataset NSL-KDD e CIC-IDS2017, con particolare attenzione ai risultati del training e all'ottimizzazione delle performance.

Nel **capitolo 1**, si è discussa l'importanza della sicurezza informatica nelle aziende, esaminando i rischi e le minacce attuali e le principali cyber-gang. Questa base teorica ha fornito il contesto necessario per comprendere l'importanza di sviluppare sistemi IDS efficaci.

Il **capitolo 2** ha introdotto le principali tecnologie per la protezione della rete, inclusi gli Intrusion Detection Systems (IDS), gli Intrusion Prevention Systems (IPS) e i Next Generation Firewall (NGFW). Sono stati esplorati in dettaglio le definizioni, le differenze e le applicazioni reali di queste tecnologie, evidenziando come esse contribuiscono alla sicurezza della rete.

Nel **capitolo 3**, viene esaminata l'evoluzione dell'intelligenza artificiale (IA) nel campo della sicurezza informatica, le sue applicazioni nella rilevazione e prevenzione delle intrusioni, e le sfide legate alla sua integrazione con IDS, IPS e NGFW. Questo

capitolo ha posto le basi per il successivo sviluppo di un IDS basato su tecniche di machine learning.

Nel **capitolo 4** si discute dell’effettivo sviluppo. Analizza la selezione dei dataset, il pre-processing dei dati, il bilanciamento dei dati, e la suddivisione dei dataset. La sezione sul training ha incluso un confronto tra classificazione binaria e multclasse, fornendo dettagli specifici sui dataset NSL-KDD e CIC-IDS2017. In seguito, è stata trattata la feature selection con XGBoost per valutare l’importanza delle variabili e il loro impatto sulle performance dei modelli. Infine, viene mostrato il processo di Grid Search per l’ottimizzazione dei parametri dei modelli.

Il **capitolo 5** ha descritto una possibile architettura dell’IDS sviluppato integrato con una rete SDN, sfruttando il controllo centralizzato del traffico tramite un controller come Ryu. L’IDS utilizza modelli di machine learning, come random forest e reti neurali, per analizzare il traffico di rete e rilevare minacce in modo dinamico. Il flusso di dati tra SDN e IDS segue un processo di monitoraggio, analisi ML e risposta in tempo reale alle minacce. Inoltre, sono stati descritti strumenti come Mininet e D-ITG per la simulazione di traffico di rete, utili per testare l’IDS in scenari realistici.

Questa tesi si propone di contribuire alla conoscenza nel campo della sicurezza informatica applicata al machine learning, fornendo una valutazione dettagliata di vari modelli di apprendimento automatico per il rilevamento degli attacchi. Sebbene non sia stato possibile superare completamente i limiti dello stato dell’arte, il lavoro ha confermato l’efficacia di modelli complessi come Random Forest e Reti Neurali, offrendo un’analisi approfondita delle loro performance su dataset realistici. Inoltre, l’uso di tecniche avanzate di ottimizzazione, come la Grid Search e la feature selection, ha mostrato come sia possibile migliorare l’efficienza dei modelli esistenti, aprendo la strada a ulteriori ricerche in questo ambito.

Per ulteriori sviluppi futuri, si raccomanda di esplorare ulteriori tecniche di machine learning, come gli approcci basati su ensemble e i modelli deep learning più recenti, che potrebbero migliorare ulteriormente la capacità di rilevamento. Un altro campo promettente è la generazione di traffico realistico per testare in modo più efficace l’IDS, in particolare simulando scenari di attacco complessi e dinamici. Questo permetterebbe di valutare in modo più accurato la resilienza e l’efficacia del sistema di rilevamento in ambienti reali. Inoltre, sarebbe interessante sviluppare

l'intera architettura dell'IDS in un'applicazione reale, estendendo quanto realizzato in questa tesi, che si è concentrata principalmente sull'implementazione del modello di machine learning. Infine, sarebbe utile investigare l'integrazione di tecniche di machine learning con sistemi di protezione attiva, come gli Intrusion Prevention Systems (IPS), per ottenere un approccio più completo e proattivo.

Bibliografia

- [1] V. A. Le, P. Dinh, H. Le, and C. Tran, "Flexible network-based intrusion detection and prevention system on software-defined networks," 11 2015, pp. 106–111. (Citato alle pagine 5 e 83)
- [2] R.-H. Hwang, M.-C. Peng, C.-W. Huang, P.-C. Lin, and V.-L. Nguyen, "An unsupervised deep learning model for early network traffic anomaly detection," *IEEE Access*, vol. 8, pp. 30 387–30 399, 2020. (Citato alle pagine 28, 38 e 46)
- [3] B. Subba, S. Biswas, and S. Karmakar, "Intrusion detection systems using linear discriminant analysis and logistic regression," in *2015 Annual IEEE India Conference (INDICON)*, 2015, pp. 1–6. (Citato alle pagine 29, 39, 46, 69 e 70)
- [4] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *2017 International Conference on Information Networking (ICOIN)*, 2017, pp. 712–717. (Citato alle pagine 30, 39 e 46)
- [5] B. J. Radford, L. M. Apolonio, A. J. Trias, and J. A. Simpson, "Network traffic anomaly detection using recurrent neural networks," *CoRR*, vol. abs/1803.10769, 2018. [Online]. Available: <http://arxiv.org/abs/1803.10769> (Citato alle pagine 31, 39 e 46)

- [6] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Network traffic classifier with convolutional and recurrent neural networks for internet of things," *IEEE Access*, vol. 5, pp. 18 042–18 050, 2017. (Citato a pagina 32)
- [7] R. Kumari, Sheetanshu, M. K. Singh, R. Jha, and N. Singh, "Anomaly detection in network traffic using k-mean clustering," in *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, 2016, pp. 387–393. (Citato a pagina 33)
- [8] M. Tavallaei, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009, pp. 1–6. (Citato a pagina 39)
- [9] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *International Conference on Information Systems Security and Privacy*, 2018. [Online]. Available: <https://api.semanticscholar.org/CorpusID:4707749> (Citato alle pagine 40 e 70)

Ringraziamenti

Desidero esprimere la mia sincera gratitudine a tutte le persone che mi hanno supportato durante il percorso di tesi.

In primo luogo, un ringraziamento speciale a **Rino Gegnacorsi** e **Raffaele Martorelli**, per avermi accolto con fiducia in azienda e avermi offerto l'opportunità di crescere professionalmente in un ambiente stimolante.

Sono profondamente grato a **Filippo Bogetti**, il mio tutor aziendale, per la sua disponibilità e preziosa assistenza. Il suo supporto, in particolare nelle questioni legate alla cybersecurity, è stato essenziale per il successo di questo lavoro. La sua competenza e pazienza mi hanno permesso di approfondire aspetti cruciali della materia.

Un ringraziamento a **Mariano Caccavale**, per l'aiuto fornito nell'ambito dell'intelligenza artificiale. La sua esperienza e i suoi consigli sono stati una risorsa inestimabile durante lo sviluppo del progetto.

Esprimo la mia gratitudine anche al professore **Christiancarmine Esposito**, il mio relatore, per la sua guida costante che ha rappresentato un pilastro fondamentale per il completamento dell'intero percorso accademico.

A tutti voi, grazie di cuore per il vostro contributo e per aver reso possibile la realizzazione di questa tesi.