Università degli Studi di Padova
Dipartimento di Scienze Statistiche
Corso di Laurea Triennale in
Statistica per le Tecnologie e le Scienze

Individuazione di utenti malevoli in reti informatiche tramite modelli statistici

Relatore Prof. Antonio Canale
Dipartimento di Scienze Statistiche

Laureando: Andrea Longhin
Matricola N 1189525

Anno Accademico 2021/2022

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

1.1 Obiettivi dell'analisi

L'analisi svolta riguarda una rete di dispositivi interconnessi, in particolare si tratta di una rete di dispositivi IIoT, sintetica, quindi artificialmente progettata, descritta nel seguito. I dati pervenuti da tale rete, assieme ad altre misure ottenute dall'elaborazione della struttura della stessa, sono stati utilizzati per adattare un modello di regressione logistica, l'obiettivo del quale è quello di classificare i nodi appartenenti alla rete come utenti normali o come offensori, atti ad eseguire un attacco informatico all'interno della rete. Il ruolo di un'analisi di questo tipo è quello di sfruttare le informazioni derivanti dall'aver subito un attacco informatico, per poi adottare successivamente opportune misure di sicurezza che garantiscano l'esclusione della possibilità che analoghe situazioni possano ripetersi in eventi futuri, nonché l'eventuale possibilità di interrompere l'esecuzione di un attacco nel caso in cui si fosse in grado di individuare la presenza di utenti malevoli nella rete prima che essi abbiano completato il ciclo operativo dell'offesa. Lo scopo principale di questa analisi non è quindi quello di prevenire un attacco informatico prima che questo si presenti, ma di offrire uno strumento che consenta di acquisire informazioni successivamente all'avvenuta offesa per impedire che possa ripresentarsi. Inoltre, è stata privilegiata la semplicità interpretativa del modello, la quale conseguentemente permette di poter disporre di una contenuta collezione di informazioni che consentano l'apprendimento delle caratteristiche e la conseguente discriminazione di utenti malintenzionati.

1.2 Attacchi informatici

L'attacco informatico è qualsiasi tentativo di ottenere un accesso non autorizzato a un computer, sistema informatico o rete informatica con l'intento di causare danni. Gli attacchi informatici mirano a disattivare, interrompere, distruggere o controllare i sistemi informatici o ad alterare, bloccare, cancellare, manipolare o rubare i dati contenuti in questi sistemi. Servendosi di una o più strategie di offesa, un attacco informatico può essere eseguito da qualsiasi luogo e da qualsiasi individuo o gruppo, questi ultimi sono generalmente considerati criminali informatici. Spesso indicati come "bad actors", "threat actors" e "hackers", essi includono individui che agiscono da soli, attingendo alle loro competenze informatiche per progettare ed eseguire attacchi

dannosi, ma possono anche appartenere ad un'organizzazione criminale, lavorando con altri attori di minacce informatiche per trovare debolezze o vulnerabilità nei sistemi informatici, che possono essere sfruttate per il fine malevolo di tali individui. Inoltre, esistono anche gruppi di esperti informatici sponsorizzati dal governo, il quale incarico è quello di eseguire attacchi informatici. Tra i loro fini c'è quello di attaccare le infrastrutture informatiche di altri governi, così come entità non governative, come aziende, organizzazioni no-profit e servizi pubblici. Ad alimentare le preoccupazioni riguardo gli attacchi a sistemi informatici nell'attuale periodo, si aggiunge la loro rapida e crescente diffusione. Si noti che in Italia, secondo un report di Trend Micro Research, sono stati praticati 28 milioni di attacchi nel solo primo semestre del 2021, posizionando l'Italia al primo posto per numero di attacchi informatici in Europa e quarta nel mondo. Secondo CORCOM, inoltre, gli attacchi sul suolo italiano sono raddoppiati in un anno, e ciò sembra essere un sottoprodotto della pandemia, dal momento che essa tende ad indebolire le strutture istituzionali e sanitarie, rendendole un obiettivo più semplicemente sfruttabile. Si evince quindi che, per le possibili vittime, risulti sempre più importante adottare misure di contrasto adeguate al problema in esame. Tale obiettivo può essere perseguito adottando opportune pratiche, tra le quali:

- Implementazione di difese perimetrali, quali firewall, software incaricati di monitorare il traffico in entrata e in uscita in un sistema informatico, per bloccare l'accesso alle risorse da parte di entità notoriamente malevoli.
- Utilizzo di software per la protezione da malware, ovvero gli antivirus, che aggiungono uno strato di protezione a quanto citato precedentemente.
- Adozione di un programma per la gestione delle patch, ovvero un programma incaricato di mantenere aggiornati i software appartenenti al sistema, andando a correggere eventuali errori o vulnerabilità.
- Impostazione di opportune configurazioni di sicurezza tramite l'introduzione di politiche inerenti alle password e all'accesso degli utenti.
- Adozione di un programma per il monitoraggio e il rilevamento di attività sospette che sappia adeguatamente identificarle e segnalarle.
- Creazione di un piano per la risposta ad avvenuti insediamenti malevoli che possa guidare un'adeguata risposta a tali minacce.
- Training e informazione degli utenti individuali riguardo possibili scenari di attacco e come reagire in caso di avvenuta offesa.

Per essere in grado di rispondere adeguatamente ad un attacco informatico è, però, fondamentale mantenere i sistemi di sicurezza costantemente aggiornati, andando iterativamente a studiare le cause degli attacchi subiti per poi implementare sistemi di prevenzione adeguati. L'analisi successiva ad un attacco risulta quindi essere cruciale

per garantire la protezione futura da insediamenti malevoli analoghi a quelli precedentemente osservati.

1.3 Reti informatiche e grafi

Tra gli esistenti sistemi informatici si trova la rete, un insieme di apparati hardware e software interconnessi, in grado di scambiarsi dati e informazioni attraverso determinati canali e protocolli di comunicazione. Un dispositivo hardware connesso alla rete può essere un'attrezzatura di comunicazione dati (o Data Communication Equipment, DCE), come modem, hub o switch, oppure un'attrezzatura terminale (o Data Terminal Equipment, DTE), come computer, smartphone, stampanti o altri dispositivi smart, quindi dotati di una connessione ad Internet. Tali dispositivi sono generalmente identificati da un indirizzo IP (Internet Protocol), ovvero l'insieme di regole che governano il formato dei dati inviati via internet o nelle reti locali. In particolare, gli indirizzi IP sono l'identificatore che permette di inviare informazioni tra i dispositivi di una rete: contengono informazioni sulla posizione e rendono i dispositivi accessibili per la comunicazione. Due dispositivi si dicono collegati nella rete quando essi sono in grado di scambiarsi dati e informazioni attraverso canali trasmissivi per il corretto invio dei messaggi. La connessione tra due dispositivi, o link, è il mezzo di trasmissione nel quale vengono inviati dati e informazioni, questa può essere fisica, come cavi e fibra ottica, o wireless, usufruendo di protocolli quali Bluetooth e WiFi. In una rete informatica, i dispositivi comunicanti seguono un insieme di regole o protocolli che definiscono il corretto invio e ricezione dei dati elettronici attraverso i collegamenti. L'architettura di una rete informatica definisce, quindi, il design di queste componenti fisiche e logiche, fornendo le specifiche per l'organizzazione funzionale degli attori, i protocolli di comunicazione e le procedure operative. Tra i protocolli comunicativi tipicamente adottati dalle architetture di rete si trovano ICMP (Internet Control Message Protocol), UDP (User Datagram Protocol) e TCP (Transmission Control Protocol). Per quanto riguarda quest'ultimo, esso supporta l'utilizzo delle cosiddette TCP flag, codici binari contenuti all'interno dei pacchetti inviati nel corso di una comunicazione tra due dispositivi. Le flag sono usate per indicare un particolare stato della connessione o per fornire informazioni aggiuntive utili agli scopi di risoluzione dei problemi e di gestione dei controlli di una particolare connessione. Le flag più comunemente usate sono SYN, ACK, FIN e RST.

È possibile interpretare una rete informatica come un grafo, struttura matematica descrivibile come un insieme di nodi e un insieme di archi. Mentre un nodo è interpretabile come un soggetto del grafo, un arco è definibile come coppia di nodi, suoi estremi: se è importante l'ordine con cui essi sono indicati, l'arco risulterà orientato e rappresenterà una relazione non simmetrica tra i nodi suoi estremi, la

presenza di almeno uno di questi archi rende il grafo orientato. Su ciascun nodo può incidere un numero arbitrario di archi, tale numero ne rappresenta in qualche misura la complessità e prende il nome di grado del nodo. Nel caso di grafo orientato, si fa distinzione tra archi entranti e archi uscenti dal nodo andando così a definire, per ogni nodo, un grado entrante e un grado uscente. Un'altra misura di complessità di un nodo è la cosiddetta betweenness, ovvero il numero di volte in cui un nodo si trova nel percorso più breve tra due nodi, dove per percorso si intende la collezione di archi che mettono in comunicazione due nodi. Servendosi quindi dell'interpretazione di rete informatica come grafo, i dispositivi hardware interconnessi nella rete sono concepibili come nodi, mentre le comunicazioni tra essi come archi. Tali archi saranno orientati, dal momento che le comunicazioni hanno necessariamente un mittente e un destinatario.

1.4 Industrial Internet of Things e sue vulnerabilità

L'Internet of Things, o IoT, si riferisce ai miliardi di dispositivi fisici sparsi in tutto il mondo che si trovano in questo istante collegati a Internet, tutti impegnati a raccogliere e condividere dati. Grazie all'arrivo di chip informatici super economici e all'ubiquità delle reti wireless, è possibile trasformare qualsiasi oggetto, da qualcosa di piccolo come una pillola a qualcosa di grande come un aereo, in una parte dell'IoT. Collegare tutti questi oggetti diversi e aggiungere loro dei sensori aggiunge un livello di intelligenza digitale a dispositivi che altrimenti sarebbero muti, permettendo loro di comunicare dati in tempo reale senza il coinvolgimento di un essere umano. L'Industrial Internet of Things (IIoT) si riferisce all'applicazione della tecnologia IoT in contesti industriali, con l'obiettivo di utilizzare una combinazione di sensori, reti wireless, big data, intelligenza artificiale e sistemi analitici per misurare e ottimizzare processi industriali. Di recente, le industrie hanno utilizzato la comunicazione machine-to-machine (M2M) per ottenere l'automazione e il controllo wireless. Ma con l'emergere del cloud e di tecnologie quali analisi e apprendimento automatici, le industrie possono raggiungere un nuovo livello di automazione e con esso creare nuovi ricavi e modelli di business, si parla quindi di quarta ondata della rivoluzione industriale o industria 4.0. L'industria 4.0 enfatizza la tecnologia intelligente, i dati, l'automazione, l'interconnettività, l'intelligenza artificiale, e tutto ciò è reso possibile dall'introduzione di dispositivi IoT nel contesto industriale. Tuttavia, la convergenza tra Tecnologia dell'Informazione (IT), dovuta all'introduzione di sistemi intelligenti, e Tecnologia Operativa (OT), originariamente presente nel contesto industriale, ha ampliato il panorama delle minacce e aumentato il potenziale rischio di attacchi informatici condotti contro tali critici sistemi. Recenti pubblicazioni sulle minacce informatiche forniscono dettagli riguardo diverse campagne di hacking mirate a critici Sistemi di Controllo Industriale (ICS) e dispositivi IIoT. Ad esempio, gli attacchi

ransomware Ekans e LockerGoga sono stati appositamente progettati per compromettere più macchine Windows che operano in applicazioni industriali. Questi attacchi causavano la crittografia di file relativi a storici di dati e applicazioni web, portando all'interruzione dei sistemi informatici e, conseguentemente, della produzione. Più recentemente si sono registrati attacchi che sfruttano la precarietà dell'attuale pandemia da COVID-19, durante i quali vengono compromessi diversi sistemi sanitari dotati di tecnologie IIoT per violare la proprietà intellettuale dei vaccini. Pertanto, a causa del fatto che, in un sistema industriale interconnesso, qualsiasi vulnerabilità può immediatamente diventare una minaccia per la sicurezza pubblica, risulta di particolare importanza porre l'attenzione sulle misure di sicurezza da implementare nella struttura.

Capitolo 2

Il dataset

2.1 Generazione dei dati

Il dataset utilizzato per la ricerca è X-IIoTID, una raccolta di dati sviluppata da Muna Al-Hawawreh, Elena Sitnikova e Neda Aboutorab, e pubblicata su IEEE Internet of Things Journal. Il set di dati è il risultato di un framework, il quale fine è quello di generare dati sintetici che rappresentino accuratamente il panorama IoT in un contesto industriale, considerandone l'eterogeneità del tipo di dispositivi, del traffico, dei protocolli di connettività, dei pattern comunicativi e del genere di possibili intrusioni. Il dataset è stato generato implementando, nel laboratorio IoT dell'Università di New South Wales (UNSW) in Canberra, il testbed Brown-IIoTbed, dove per testbed si intende una piattaforma di sperimentazione controllata in cui le soluzioni possono essere implementate e testate in un ambiente che replica le condizioni del mondo reale. Il Brown-IIoTbed è costituito da una varietà di protocolli di connettività M2M (machine to machine), M2H (machine to human) e H2M (human to machine), sensori, attuatori, vari dispositivi mobili e informatici, API (Application Programming Interface, collezione di protocolli per lo sviluppo e l'integrazione di prodotti software), il tutto distribuito sui tre livelli di un sistema IIoT: edge tier, platform tier, enterprise tier. Per l'utilizzo del testbed citato, è stato implementato il modello IIRA, un modello architetturale standard a livello internazionale per applicazioni IIoT, pubblicato dall'Industrial Internet Consortium (IIC), per definire i principali requisiti dei sistemi IIoT reali.

2.2 Generazione degli attacchi informatici

Gli attacchi informatici praticati sulla rete sono stati simulati rispecchiando gli approcci più recenti alla modellazione delle minacce su reti IIoT, i quali danno la priorità a un livello perimetrale, in particolare un edge gateway, che è la parte principalmente presa di mira dagli attacchi informatici a reti IIoT più avanzati. L'edge gateway rappresenta un punto di connessione della rete per i dispositivi ad essa connessi. Di solito si tratta di router, switch, o altre apparecchiature elettroniche volte alla connessione dei dispositivi appartenenti alla rete. Ciò che rende l'edge gateway particolarmente propenso a trovarsi vittima di un attacco è la sua posizione nel sistema IIoT come ponte tra IT e OT, interfacciandosi a parti critiche dei circuiti di controllo e connettendosi

direttamente alle riserve di dati. Inoltre, gli attacchi sono stati simulati secondo una struttura pensata ad hoc per problemi inerenti a reti di dispositivi IIoT. Questa struttura prende spunto da altri framework di attacchi informatici, come CKC, MALC e ATT&CK, i quali però si riferiscono a generiche reti di calcolatori, e sono quindi stati utilizzati come base per poi essere adattati ad una nuova struttura di attacco pensata specificamente per reti IIoT. Tale struttura è interpretabile come il ciclo vitale di un attacco informatico diviso in più fasi, le quali scandiscono il livello di penetrazione dell'utente malevolo nella rete. Di seguito sono riportate le fasi degli attacchi informatici che sono stati simulati nella rete:

- *Reconnaissance*. Durante la fase di ricognizione, l'attaccante cerca un ipotetico bersaglio e identifica e seleziona l'opportuna metodologia di offesa. L'obiettivo principale è quello di raccogliere informazioni riguardanti il bersaglio utilizzando varie tecniche e procedure, alcune delle quali sono state applicate nella simulazione di questa fase e sono riportate di seguito:
    - Scansione generica: si riferisce al processo di definizione delle porte attive del dispositivo bersaglio, il suo sistema operativo e i servizi di cui dispone.
    - Scansione delle vulnerabilità: si riferisce al processo di determinazione delle vulnerabilità note e delle configurazioni errate nella macchina bersaglio.
    - *Fuzzing*: mira a trovare errori di sistema o di software inviando dati casuali o semi validi nel sistema o software bersaglio, per poi analizzare la risposta a tali messaggi per scoprire possibili falle.
    - Scoperta delle risorse: si riferisce al processo di ottenimento dell'elenco delle risorse CoAP (Constrained Application Protocol) disponibili al bersaglio. Queste risorse sono legate a processi fisici, ad esempio la lettura dei valori dei sensori o il controllo degli attuatori.
- *Weaponization*. Questa fase riguarda le attività svolte dall'attaccante per instaurare la sua posizione sulla rete, trasmettendo malware alla macchina obiettivo. Le tecniche utilizzate per la simulazione di questa fase sono riportate di seguito:
    - *Brute force attack*: consiste nel fare accesso nell'account di un utente ottenendo le credenziali di accesso (nome utente e password) tramite strategie trial-and-error, le quali consistono nel ripetuto tentativo di generare possibili credenziali di accesso fin quando non vengono trovate quelle corrette.
    - *Dictionary attack*: particolare tipo di attacco brute force, durante il quale i termini utilizzati per ottenere le credenziali di accesso vengono reperiti da un dizionario predisposto a tale fine.
    - *Malicious insider*: questo attacco viene eseguito da un fidato utente della rete che ha accesso legittimo alle risorse del sistema, pur essendo guidato

da intenzioni dannose. Durante questo attacco, l'utente fidato insedia un codice malevolo nella macchina bersaglio tramite comunicazioni legittime, le quali possono essere eseguite dall'utente vista la sua posizione apparentemente innocua nella rete.

- *Exploitation*. In questa fase viene sfruttata la posizione ormai instaurata da parte dell'attaccante nella rete. Gli scenari simulati sono i seguenti:
    - *Reverse shell*: l'attaccante instaura una connessione tra la sua macchina e l'ingresso perimetrale (edge gateway) della rete. Sfruttando un precedentemente insediato malware, viene così instaurata una cosiddetta backdoor, ovvero un metodo mediante il quale utenti autorizzati e non sono in grado di aggirare le normali misure di sicurezza e ottenere un accesso utente di alto livello (noto anche come accesso root) su un sistema informatico, una rete o un'applicazione software.
    - *MitM attack*: generalmente, consiste nell'insediamento dell'attaccante nella comunicazione tra due dispositivi. In questa implementazione, i due dispositivi comunicanti sono l'edge gateway e un router, l'attaccante inoltra quindi pacchetti con un indirizzo falso all'edge gateway impersonando un router.
- *Lateral movement*. L'obiettivo di questa fase è esplorare l'ambiente della vittima, penetrare nelle risorse sempre più in profondità e compromettere quanti più sistemi possibile, anche arrivando ad accedere al cloud. Gli scenari simulati sono i seguenti:
    - *MQTT cloud broker-subscription*: l'utente attaccante sfrutta i privilegi precedentemente acquisiti per connettersi al cloud broker, il sistema che si occupa di coordinare le comunicazioni tra dispositivi fisici e servizi cloud. Una volta stabilita tale connessione, l'attaccante ha accesso a tutte le informazioni raccolte dai dispositivi della rete che sono state caricate sul cloud.
    - *Modbus-register reading*: l'attaccante sfrutta dispositivi preposti all'uso di Modbus, un protocollo di comunicazione per il trasferimento di dati che non richiede autenticazione, con lo scopo di ottenere gli indirizzi di altri dispositivi connessi alla rete.
    - *TCP Relay attack*: l'attaccante apre un canale di comunicazione tra il suo dispositivo e il server mail della rete attraverso l'ormai compromesso edge gateway, accedendo così ai contenuti del server mail come utente appartenente alla rete, pur non essendone legittimamente parte.
- *Command and Control* (C&C). Questa fase richiede l'interazione manuale da parte dell'attaccante, non vengono quindi attuate procedure automatiche. Il canale C&C viene stabilito connettendo il server dell'attaccante con i dispositivi della rete precedentemente compromessi per poi effettuare diverse attività

perpetuanti l'insediamento dell'attacco, tra le quali l'invio di comandi e istruzioni al malware già installato.

- *Exfiltration*. Durante questa fase vengono divulgate pubblicamente informazioni private e sensibili relative i dispositivi IIoT presenti nella rete, come dati provenienti da sensori, credenziali di accesso e versioni dei software utilizzati dai dispositivi appartenenti alla rete. Vengono solitamente utilizzate diverse tecniche e procedure per il raggiungimento di tale obiettivo, come la compressione, l'offuscamento e l'utilizzo di un canale differente da quello C&C, con lo scopo di evitare il rintracciamento dell'offensore.
- *Tampering*. Questa fase è caratterizzata dalla distruzione, manipolazione o alterazione dei dati in transito o memorizzati nei dispositivi appartenenti alla rete. In questa implementazione, è stato effettuato il cosiddetto poisoning dei dati su cloud, ovvero l'iniezione di dati fallaci, e l'invio di notifiche false. In particolare, in una rete IIoT, i dati collezionati dai sensori vengono inviati al cloud con lo scopo di fare training dei modelli analitici, i quali si occupano di operare previsioni e decisioni riguardanti la manutenzione della rete. Dunque, l'attaccante può generare e introdurre dati di questo genere per inficiare l'accuratezza dell'analisi di tali modelli. Alternativamente, l'attaccante può inviare false notifiche di email o di emergenza agli operatori collegati alla rete.
- *Crypto-Ransomware*. In questo tipo di attacco, l'offensore introduce un malware che impedisce l'accesso ai dati di un sistema, forzando la vittima a pagare un riscatto in forma di criptovaluta. In questo contesto, l'obiettivo del ransomware è quello di crittografare file inerenti configurazione e impostazione dei dispositivi fisici collegati alla rete, minacciando di formattare tali dispositivi nel caso in cui il riscatto non venisse pagato entro determinati termini tempistici.
- *Ransom Denial of Service* (RDoS). In questo attacco, l'offensore minaccia di attuare un massivo attacco DDoS nel caso in cui la vittima non saldasse un riscatto entro uno specifico lasso di tempo. Per attacco DDoS (*Distributed Denial of Service*) si intende il tentativo di rendere un servizio online inoperabile tramite il sovraccarico di traffico destinato al dispositivo offrente tale servizio, dove il travolgente ammontare di dati in traffico proviene da una moltitudine di dispositivi, giustificando così la natura distribuita dell'attacco.

2.3 Reperimento dei dati

La rete è stata organizzata predisponendo 114 dispositivi connessi (nodi), le quali comunicazioni simulate (archi) sono state effettuate in un periodo di circa quattro mesi, dal 5 Dicembre 2019 al 27 Marzo 2020. Durante tale periodo sono state effettuate comunicazioni di tipo normale e malevolo: l'inclusione di entrambi i tipi di interazione è fondamentale per costruire apposite soluzioni di sicurezza applicabili in contesti reali.

Il traffico è stato registrato in modalità non continuativa, alternando periodi di passività osservativa a periodi di collezione dei dati inerenti al traffico informatico, estesi per diverse ore in modo da garantire l'accurata rappresentatività delle caratteristiche delle attività presenti all'interno di un sistema IIoT. Per il raccoglimento di tali dati sono stati adoperati diversi dispositivi hardware e software, tra i quali uno strumento dumpcap e il software SAR. Lo strumento dumpcap, in questo caso installato nell'edge gateway, è un software incaricato di svolgere attività di lettura dei pacchetti instradati nella rete per poi estrarne le informazioni e salvarle su appositi file. Il compito del SAR (System Activity Reporter) è, invece, quello di raccogliere informazioni circa le prestazioni del sistema, riportandole in appositi documenti testuali. I dati estratti nelle modalità sopracitate sono poi stati elaborati utilizzando una pletora di script sviluppati tramite Python e Batch, due linguaggi di programmazione leader nel settore. Il risultato delle varie elaborazioni consiste in una collezione di archi arricchita da numerose variabili che sono dettagliate nell'Appendice A. In particolare, la Tabella A.1 riporta la descrizione delle variabili riguardanti il traffico nella rete, mentre nella Tabella A.2 è riportata la descrizione delle variabili inerenti ai dispositivi dai quali sono state estratte.

Una volta collezionati e elaborati i dati, sono quindi state estratte casualmente 421417 osservazioni di tipo normale e 399417 osservazioni di tipo malevolo, le quali, assieme, consistono negli archi della rete che sarà usata nelle seguenti analisi. Tali archi sono descritti dalle 66 variabili riportate nelle due tabelle precedentemente citate, più tre livelli di etichette che descrivono il tipo di comunicazione: il terzo livello, *class3*, distingue semplicemente una comunicazione normale da un attacco, il secondo, *class2*, descrive anche il tipo di attacco secondo la fase del ciclo vitale dello stesso, mentre il primo livello, *class1*, descrive nel dettaglio lo scenario che è stato applicato tra quelli possibili a seconda della fase del ciclo vitale dell'attacco.

Capitolo 3

Analisi esplorative

3.1 Elaborazione dei dati

I dati presentati nel paragrafo precedente sono stati codificati secondo lo standard CSV (Comma-Separated Values), un formato di trasmissione dati che consente di organizzare le informazioni in una struttura tabellare. Una volta importati tali dati in un ambiente di sviluppo R, essi sono stati opportunamente elaborati per consentirne l'utilizzo. Tale elaborazione è stata operata secondo le seguenti fasi:

- Rimozione di tuple problematiche. Le osservazioni che sono state rimosse dal dataset presentavano uno dei seguenti problemi:
    - Il mittente della comunicazione è sconosciuto, e quindi denotato dal simbolo "?" al posto dell'associato indirizzo IP.
    - Il destinatario della comunicazione è sconosciuto, e quindi denotato dal simbolo "?" al posto dell'associato indirizzo IP.
    - Il mittente e il destinatario della comunicazione sono il medesimo dispositivo. Sebbene una situazione di questo tipo possa presentarsi all'interno della rete, il particolare caso del dispositivo identificato dall'indirizzo IP 127.0.0.1 era tale per cui tutte le comunicazioni effettuate da tale dispositivo erano indirizzate esclusivamente a sé stesso. Ciò rende il nodo in questione sconnesso dal resto della rete, e quindi irrilevante per le successive analisi condotte sulla stessa, ed è per tale ragione stato rimosso dal dataset.
- Codifica dei nomi delle variabili. I nomi delle variabili contenute nel dataset non sembravano rispondere ad uno specifico standard di notazione, sono perciò state codificate secondo le seguenti specifiche dello standard snake case:
    - Utilizzo esclusivo di caratteri in stampatello minuscolo.
    - Spaziatura tra termini appartenenti al medesimo nome definita tramite il simbolo underscore, ovvero "_".

  I nomi così ricodificati sono quelli presenti nelle tabelle A.1 e A.2.

- Correzione interpretativa del tipo di variabile. In questa fase ci si è accertati che le variabili venissero correttamente interpretate dal linguaggio di programmazione, andando ad esplicitarne la tipologia. I tipi di variabili presenti nel dataset sono:

- Testuali, presenti nella forma di variabili qualitative sconnesse, come *protocol*, e di variabili qualitative ordinali, come *conn_state*.
- Numeriche, quindi quantitative discrete e continue, tra le quali si trovano *src_bytes* e *duration*.
- Dicotomiche, quindi assumenti i soli valori 0 e 1, come *syn_ack* e *payload*.
- Data, l'unica istanza di questo tipo è la variabile *date*, che rappresenta la data nella quale la comunicazione è avvenuta, codificata secondo lo standard giorno/mese/anno.

I dati così elaborati sono quindi stati memorizzati in una variabile di tipo data frame, uno degli standard più diffusi per l'elaborazione di dati tabulari in un ambiente di programmazione R. Si noti che le classi di attacco 1 e 2, ovvero le due variabili che forniscono informazioni dettagliate sui tipi di attacchi praticati sulla rete, non sono state inizialmente prese in considerazione per lo svolgimento dell'analisi, dal momento che il progetto che si sta descrivendo mira all'individuazione di utenti malevoli indipendentemente dalle fasi in cui si trovano gli attacchi che stanno praticando. Dunque, l'unica variabile discriminante attacchi da comunicazione normali mantenuta all'interno del dataset è *class3*, la quale è stata denominata come *label*.

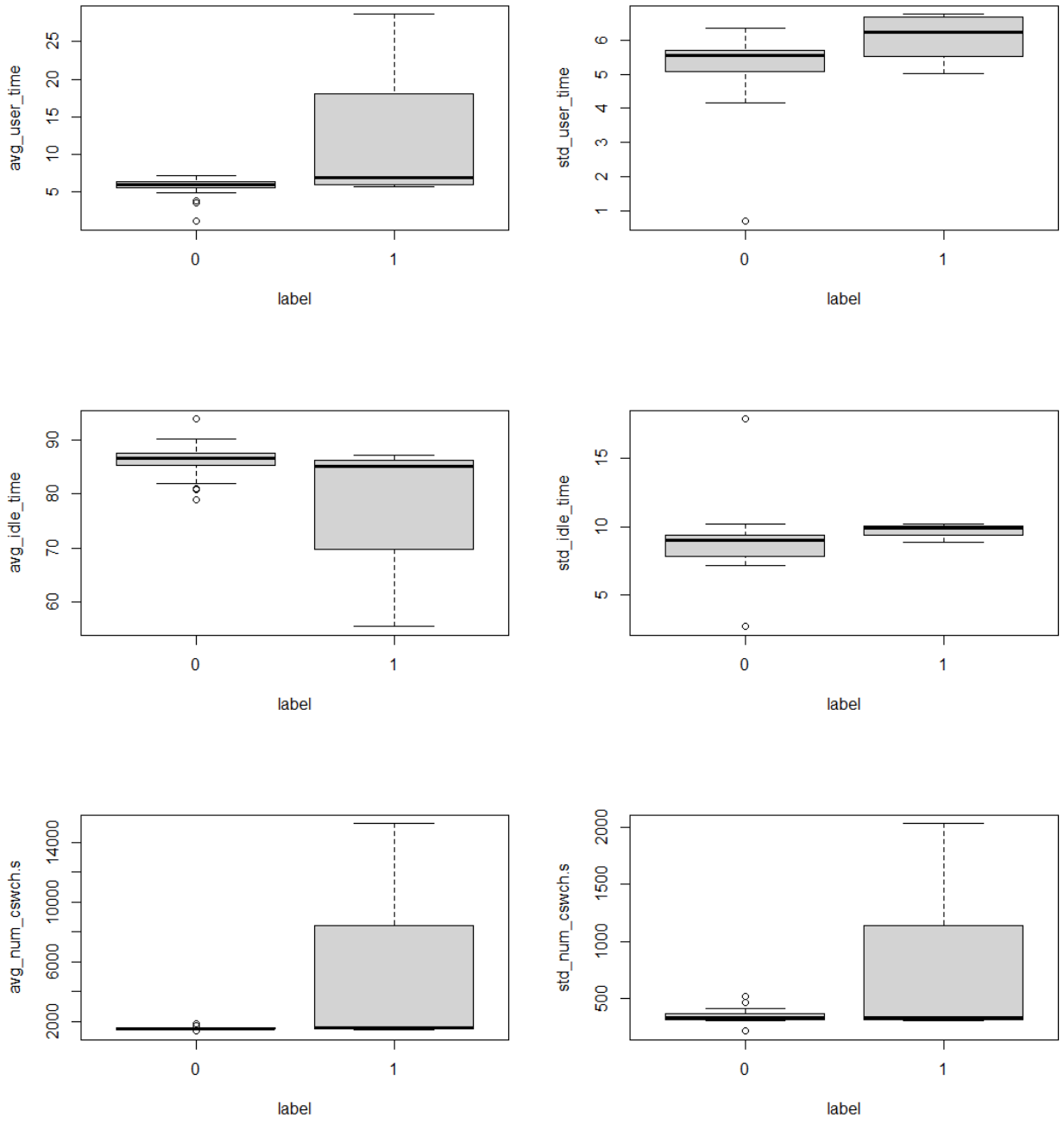
3.2 Studio dei valori mancanti

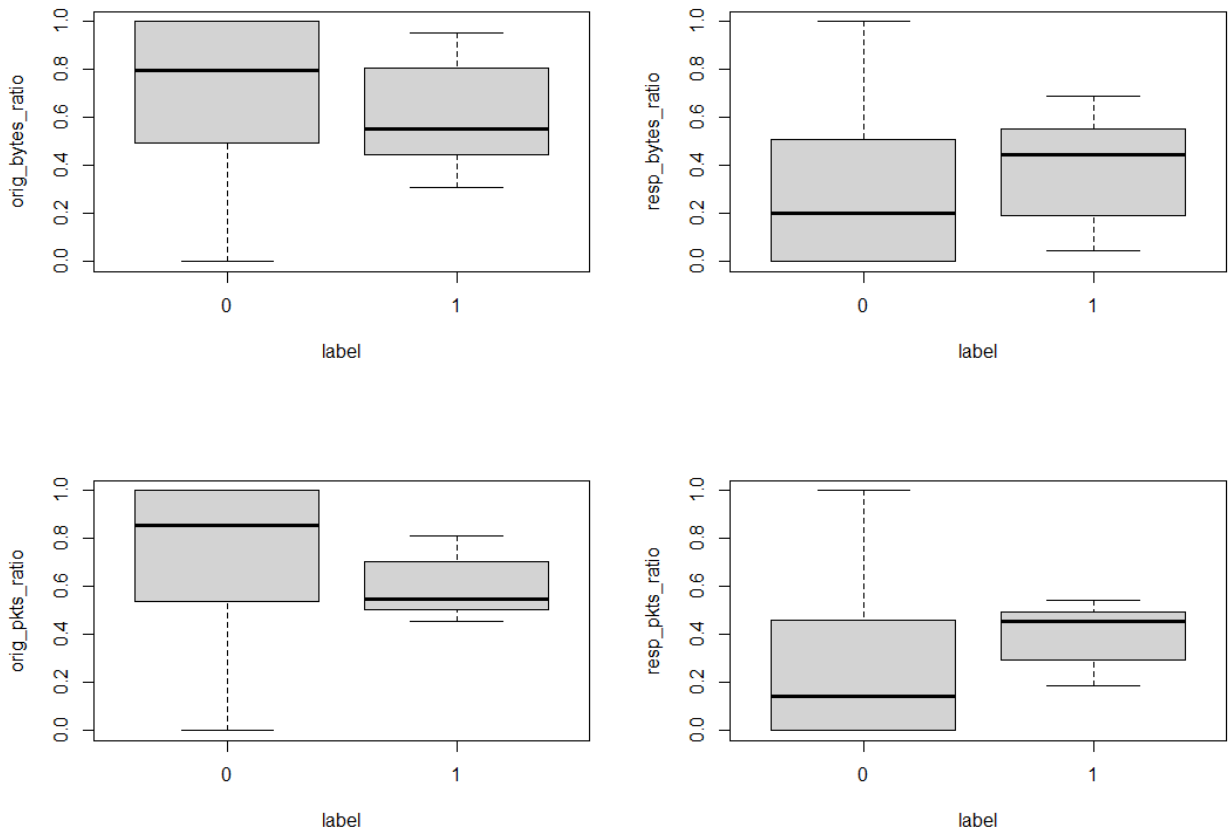
L'osservazione dei dati ha permesso di individuare la presenza di informazioni assenti. La plausibile motivazione per l'assenza di tali dati è attribuibile alla varietà dei dispositivi presenti nella rete, dal momento che ci si può aspettare che non tutti i terminali hardware disponessero dei medesimi sensori per la rilevazione di tali informazioni, per via della loro variegata natura. In particolare, si sono osservati un totale di 216870 archi presentanti valori assenti in corrispondenza di 40 variabili. È dunque risultato fondamentale studiare il corretto approccio per la gestione di tali informazioni mancanti, in particolare si è dovuto decidere quale tra le due componenti eliminare: le variabili o gli archi. Si noti che il numero di archi presentanti dati mancanti corrisponde al 27.8% del totale, mente le variabili problematiche equivalgono al 60.6%. Apparentemente, sembra quindi che la perdita minore di informazione si avrebbe rimuovendo le connessioni carenti di dati. Tuttavia, un'analisi più approfondita del problema ha rilevato che un totale di 8 nodi origine, ovvero nodi responsabili dell'invio della comunicazione, presentano osservazioni con dati mancanti. Quindi, considerando che il numero totale di nodi origine è 34, tale valore corrisponde al 23.5% del totale dei nodi origine presenti nella rete. Inoltre, uno dei nodi attaccanti presenta informazioni assenti in tutti gli archi corrispondenti ad attacchi da esso praticati, ciò si traduce nel fatto che, una volta eliminati gli archi, le uniche comunicazioni da lui effettuate sarebbero di tipo di normale, andando quindi ad

etichettare erroneamente tale utente come nodo comune anziché malevolo. Per evitare tale contraddizione, bisognerebbe quindi andare a rimuovere completamente tale nodo dalla rete, il che comporterebbe un'alterazione significativa della rete stessa.

Per andare, quindi, a completare l'analisi sui valori assenti, si è costruito un nuovo dataset comprendente le sole variabili problematiche e gli archi presentanti informazioni complete, con l'obiettivo di indagare l'effettiva validità di tali variabili nel compito di discriminare gli utenti offensori da quelli normali. A partire da tale collezione è stato, dunque, creato un nuovo data frame che osserva le stesse informazioni da una prospettiva differente: le righe, anziché rappresentare gli archi, riportano ora i nodi, mentre le colonne sono adesso una misura delle precedenti variabili, ottenuta dall'elaborazione delle stesse. In particolare, trattandosi di variabili quantitative continue, le nuove colonne corrispondono alle medie delle variabili originariamente osservate calcolate rispetto ai nodi. Si veda l'Appendice B per osservare il codice relativo a tale rielaborazione. Si noti, inoltre, che si è deciso di mantenere invariati i nomi delle variabili rielaborate rispetto a quelli delle variabili originali. La rielaborazione operata comporta una differente interpretazione delle variabili rilevate sulla rete: le colonne del nuovo dataset rappresentano ora il comportamento medio dei nodi origine rispetto a tutte le comunicazioni da loro effettuate all'intero della rete. Tale decisione è stata presa per il fatto che il fine del progetto è quello di individuare i nodi aggressori, non le singole comunicazioni malevole. Una volta ottenuto il dataset descritto, si è quindi operata un'analisi esplorativa dei dati, costruendo in primo luogo dei boxplot tra la variabile *label*, che etichetta il nodo come offensore nel caso esso avesse effettuato almeno una comunicazione malevola, e le altre variabili ottenute dall'elaborazione di quelle rilevate dalla rete. Tale analisi visiva ha permesso di osservare che 10 delle 40 variabili in questione presentano un comportamento diversificato a seconda del tipo di nodo. Risulta, tuttavia, rilevante notare che tali variabili si presentano come coppie. Infatti, considerando l'esempio di *avg_user_time* e *std_user_time*, tali variabili corrispondono ai valori medi del tempo medio di esecuzione di programmi o codici negli ultimi 10 secondi e della deviazione standard del tempo di esecuzione di programmi o codici negli ultimi 10 secondi. È, dunque, normale aspettarsi che tali coppie di variabili abbiano comportamenti simili all'interno della coppia rispetto alla variabile *label*. In Figura 3.1 sono riportati i soli boxplot relativi alle 10 variabili che hanno presentato un comportamento diversificato rispetto alla variabile *label*.

Figura 3.1: Boxplot relativi alle variabili problematiche apparentemente rilevanti, rispetto alla variabile label





Successivamente, nel tentativo di eseguire un'analisi della varianza a una via per valutare la presenza di comportamenti diversi delle variabili rispetto all'etichetta, si è studiata l'omogeneità della varianza nei gruppi dettati dalla variabile *label* rispetto alle dieci variabili in esame, dal momento che tra le ipotesi sottostanti l'analisi della varianza a una via vi è quella di omoschedasticità tra i gruppi. I risultati di tale analisi sono riportati nella Tabella 3.1. Come si può notare da tali esiti, la varianza all'interno dei gruppi non risulta essere omogenea, violando un prerequisito dell'analisi della varianza a una via. Per gli scopi precedentemente citati, si è, dunque, fatto ricorso al test Kruskal-Wallis, un approccio non parametrico equivalente all'ANOVA ad una via, il quale non prevede l'ipotesi di omoschedasticità. Tale test viene utilizzato per valutare la presenza di una differenza statisticamente significativa della mediana di gruppi di osservazioni indipendenti, il che andrebbe ad indicare se una variabile presenta comportamenti diversi, rispetto alla mediana, a seconda dei gruppi di nodi denotati dal valore della variabile *label*. In altre parole, si vuole studiare se gli utenti malevoli si comportano diversamente dagli utenti normali rispetto alle variabili in esame, il che indicherebbe che tali variabili risulterebbero utili a discriminare i due gruppi di utenti, nel caso tali comportamenti fossero sufficientemente differenti. Tale test ha quindi prodotto dei risultati che sembrano indicare che le sole variabili *avg_user_time*, *std_user_time* e *std_idle_time* presentino comportamenti differenti in mediana rispetto alla variabile *label*, andando ad indicarne la potenziale significatività nel caso in cui

tali variabili venissero utilizzate per discriminare gli utenti malevoli da quelli comuni. Tale conclusione può essere tratta confrontando i p-value del test in questione, i quali sono riportati nella Tabella 3.2, con una soglia nominale del 5%.

Tabella 3.1: Analisi dell'omogeneità della varianza nei gruppi dettati dalla variabile label

Variabile	Label	Deviazione standard
avg_user_time	0	1.48
	1	10.1
std_user_time	0	1.29
	1	0.659
avg_idle_time	0	3.55
	1	14.0
std_idle_time	0	2.77
	1	0.485
avg_num_cswch.s	0	107
	1	6282
std_num_cswch.s	0	67.3
	1	775
orig_bytes_ratio	0	0.315
	1	0.238
resp_bytes_ratio	0	0.315
	1	0.238
orig_pkts_ratio	0	0.292
	1	0.138
resp_pkts_ratio	0	0.292
	1	0.138

Tabella 3.2: Risultati del test Kruskal-Wallis condotto rispetto alle variabili presentati valori assenti

Variabile	Statistica H	Gradi di libertà	P-value
avg_user_time	4.4568	1	0.03476
std_user_time	4.4568	1	0.03476
avg_idle_time	2.7778	1	0.09558
std_idle_time	5.9753	1	0.01451
avg_num_cswch.s	0.5216	1	0.4702
std_num_cswch.s	0.049383	1	0.8241
orig_bytes_ratio	1.538	1	0.2149
resp_bytes_ratio	1.538	1	0.2149
orig_pkts_ratio	2.8599	1	0.09081
resp_pkts_ratio	2.8599	1	0.09081

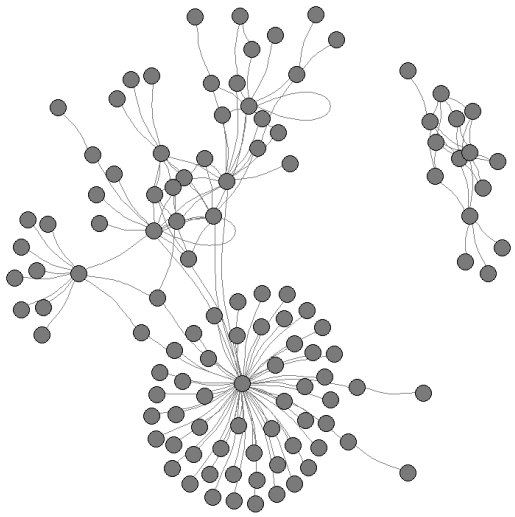
In conclusione, considerando che un'esigua porzione delle variabili presentanti valori assenti risulta potenzialmente utile per lo scopo di distinguere gli utenti offensori da quelli comuni, esse risultano infatti essere 3 su 40, il 7.5% di tali variabili, e una volta valutata la significativa perdita di informazioni che risulterebbe dal rimuovere gli archi che presentano dati incompleti, si decide di rimuovere le variabili problematiche mantenendo intatta la totalità delle comunicazioni presenti nel dataset iniziale.

3.3 Visualizzazione della rete

Tenendo presente la natura di una rete di dispositivi IIoT come quella in questione, l'aspetto visivo di una struttura informatica di questo tipo risulta distribuito su una superficie geografica indefinibilmente estesa, se non si conoscono informazioni riguardanti la localizzazione fisica dei dispositivi. Infatti, i canali trasmissivi utilizzati per lo scambio di dati tra dispositivi possono non essere esclusivamente o necessariamente tangibili, ma possono includere tecnologie in grado di sfruttare connessioni ad Internet o ad un cloud. In questo caso, trattandosi di dati generati sinteticamente, è noto che i dispositivi interconnessi si trovassero in unico laboratorio, tuttavia non è noto in che modo essi siano stati organizzati all'interno di tale ambiente. Tale considerazione non rappresenta, tuttavia, un particolare problema ai fini della ricerca, dal momento che la visualizzazione della rete in esame risulta comunque essere possibile da un punto di vista logico piuttosto che fisico, il che consente di osservare visivamente determinati comportamenti presenti nella rete. Dunque, per la visualizzazione della rete, sfruttando i dati sugli archi derivanti dal dataset, è stato utilizzato un algoritmo per la rappresentazione grafica di grafi chiamato *force-directed placement.* Tale algoritmo, noto anche come *spring-embedder* o *energy-based placement*, dispone i nodi sul piano in modo organico ed esteticamente apprezzabile, esponendo la struttura simmetrica e a grappolo di un grafo, mostrando una distribuzione bilanciata dei nodi e minimizzando gli incroci tra gli archi. L'algoritmo si basa su un modello fisico: i nodi sono rappresentati come punti in un piano, i quali sono elettricamente carichi e applicano forze repulsive l'uno contro l'altro, mentre gli archi collegano tali nodi simulando una forza a molla, attraendo i nodi adiacenti. Il modello determina iterativamente le forze risultanti che agiscono sui nodi e cerca di muovere i nodi nel tentativo di raggiungere un equilibrio dove tutte le forze si sommano a zero, consentendo la stabilità della posizione dei nodi. L'interpretazione dell'algoritmo *force-directed* utilizzata è quella proposta da Fructerman e Reingold, nella quale le forze si attenuano quando i nodi diventano rispettivamente più lontani o più vicini. È, inoltre, presente una temperatura globale che svolge il ruolo di ricottura simulata, una pratica che nella realtà viene effettuata su metalli e altri materiali per ridurre o annullare le tensioni interne all'oggetto. In questo contesto, tale concetto si traduce in una misura utilizzata per limitare lo spostamento massimo dei vertici ad ogni

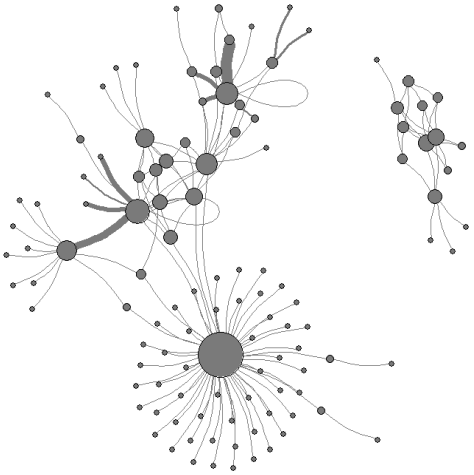
iterazione. Dunque, man mano che le forze tra ogni componente e la temperatura diminuiscono gradualmente, il layout si stabilizza. La rete rappresentata con l'utilizzo dell'algoritmo descritto è riportata in Figura 3.2.

Figura 3.2: Rappresentazione grafica della rete



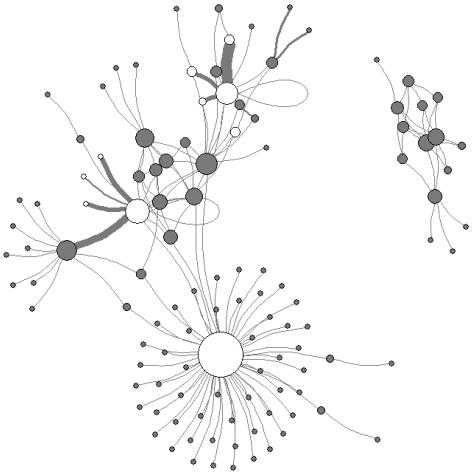
Per consentire un'appropriata lettura, si noti che non sono stati riportati gli indirizzi IP dei nodi e le direzioni degli archi. Sebbene questa prima rappresentazione consenta di notare alcune caratteristiche strutturali della rete, come la presenza di un cluster di nodi sconnesso dal resto della rete e l'influenza di un nodo che si trova circondato da decine di altri dispositivi, è possibile arricchire il grafico per maggiorare l'informazione ottenibile dallo stesso. Con questo fine è stato, dunque, prodotto il grafico in Figura 3.3, il quale diversifica la dimensione dei nodi rispetto al loro grado e lo spessore degli archi rispetto al numero di comunicazioni che essi rappresentano, infatti la rete presenta archi sovrapposti dal momento che due nodi possono avere effettuato più di una comunicazione tra loro. Per consentire un'opportuna visualizzazione della rete, si noti, inoltri, che i valori del grado e del numero di archi sovrapposti sono stati riscalati. In particolare, per quanto riguarda il grado, ad esso è stato applicato una radice quadrata e una successiva moltiplicazione per il valore 2.6, il quale coefficiente moltiplicativo è stato individuato a seguito di una ricerca iterativa di un'opportuna costante che garantisse una visualizzazione apprezzabile. Per quanto riguarda lo spessore degli archi, questo è stato determinato dividendo per 10000 il numero di comunicazioni che ciascun arco rappresenta, anche in questo caso il valore è stato individuato iterativamente.

Figura 3.3: Rappresentazione grafica della rete con pesi attribuiti a nodi e archi



Successivamente, risulta di interesse visualizzare la posizione dei nodi attaccanti all'interno della rete, il che potrebbe dare un'idea dell'influenza e del danno di cui tali nodi possono essere responsabili. Per tali ragioni, è stato prodotto il grafico rappresentato in Figura 3.4, nel quale i nodi attaccanti sono stati rappresentati dal colore bianco, mentre i rimanenti di colore grigio.

Figura 3.4: Rappresentazione grafica della rete con distinzione visiva tra nodi attaccanti e normali

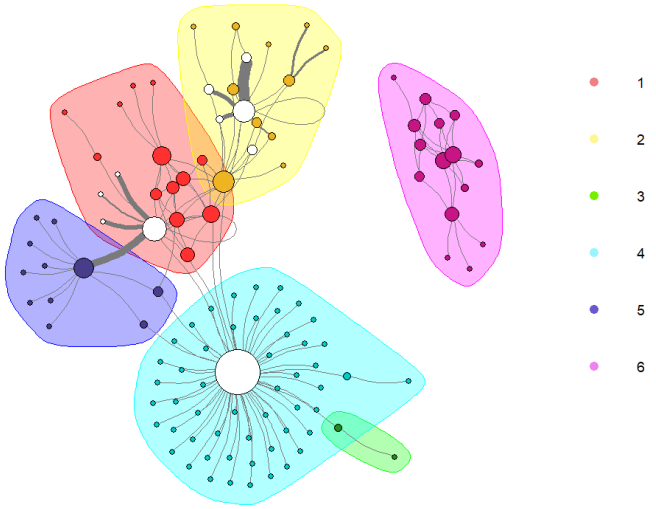


Da tale rappresentazione si può notare come il nodo particolarmente influente precedentemente citato sia un attaccante, permettendo dunque di quantificare preliminarmente il livello di insediamento da esso praticato. Si può, inoltre, notare

come spesso i nodi attaccanti siano comunicanti con altri nodi attraverso archi molto spessi, indicando un comportamento comunicativo identificato da ripetute connessioni con nodi già interrogati. Un altro interessante comportamento perpetuato da nodi malevoli è quello di effettuare comunicazioni con altri utenti offensori, il che può rappresentare l'avvenimento di due scenari non mutualmente esclusivi: un utente attaccante può attaccare un altro offensore, ignaro o meno di essersi messo in contatto con un altro nodo dello stesso tipo, oppure i due nodi attaccanti possono comunicare con lo scopo di condividere informazioni e dati ottenuti illecitamente o per coordinare forme di attacchi distribuiti su più dispositivi offensori.

Infine, di particolare interesse nell'analisi visiva di reti vi è l'osservazione dei cluster dei nodi, ovvero dei gruppi di vertici definiti secondo proprietà logiche legate agli archi, e non fisiche dovute al posizionamento geografico dei dispositivi. Per il raggiungimento di tale fine è stata praticata ciò che viene definita come *community detection*. Tale concetto si riferisce ad una metodologia per la scoperta di gruppi, o cluster, di individui all'interno di sistemi complessi interpretabili come grafi. Tra i possibili algoritmi utilizzabili per l'individuazione di cluster all'interno della rete, è stato praticato il cosiddetto *metodo di Louvain*. Tale algoritmo porta alla scoperta di gruppi di nodi tramite l'ottimizzazione della modularità, valore che misura la densità relativa di archi all'interno delle comunità di nodi rispetto agli archi esterni a tali comunità, che fungono quindi da collegamenti tra i gruppi. L'ottimizzazione di questa misura porta all'ottenimento teorico del miglior raggruppamento possibile dei nodi di una rete. Tuttavia, poiché raggruppare i nodi secondo tutte le possibili combinazioni di nodi esistenti risulta essere un'operazione estremamente poco pratica e dispendiosa, viene utilizzato un approccio euristico che consente di giungere al termine dell'algoritmo in tempi accettabili. La strategia proposta dal *metodo di Louvain* prevede una fase iniziale durante la quale vengono individuate piccole comunità ottimizzando la modularità localmente su tutti i nodi, quindi ogni piccola comunità viene raggruppata e interpretata come un singolo nodo, per poi ripetere il primo passo dell'algoritmo iterativamente. Le comunità di nodi così individuate sono osservabili nella rappresentazione grafica della rete riportata in Figura 3.5, dove i diversi gruppi sono distinguibili dalla colorazione, e ai quali gruppi è stato associato un valore numerico che ne renderà più semplice la discussione successivamente. Grazie alla rappresentazione grafica così prodotta, si può notare come i nodi attaccanti si trovino esclusivamente nei cluster 1, 2 e 4. Inoltre, è interessante osservare come il cluster 3 sia distinto dal cluster 4, seppure visivamente essi appaiono come un'unica comunità. Ciò sta ad indicare come l'algoritmo utilizzato sia in grado di catturare proprietà intrinseche della rete non direttamente osservabili. La suddivisione della rete in cluster risulterà cruciale per successive fasi dell'analisi.

Figura 3.5: Rappresentazione grafica della rete con distinzione visiva tra i cluster di nodi



3.4 Analisi strutturale della rete

Oltre all'analisi visiva della rete, è di interesse andare a studiare la stessa da un punto di vista logico, andando cioè ad individuare caratteristiche e proprietà delle connessioni tra i dispositivi che potrebbero non essere visivamente osservabili tramite la consultazione di rappresentazioni grafiche. Per tale fine, sono state inizialmente calcolate alcune misure che consentono di quantificare numericamente l'estensione e l'interconnessione della rete:

- Densità di archi. Questa misura corrisponde alla proporzione di archi osservati all'interno della rete rispetto al totale degli archi che potrebbero potenzialmente essere presenti. La formula matematica per effettuare il calcolo in questione è la seguente:

$$DA = \frac{n_a}{n_n \cdot (n_n - 1)}\,, \qquad n_a = |(u, v) \in G|\,, \qquad n_n = |u \in G|$$

  Dove, sia G il grafo corrispondente alla rete, u un generico nodo e (u, v) un generico arco, n_a corrisponde al numero di archi presenti nella rete, mentre n_n al numero di nodi. Il valore di tale misura osservato nella rete in questione è 0.0127. Un numero prossimo a 1 indicherebbe un'alta densità di archi, si evince quindi che la rete in esame sia caratterizzata da una scarsità di connessioni tra i dispositivi.

- Reciprocità di archi. Tale misura rappresenta la proporzione di archi bidirezionali rispetto al numero totale degli archi presenti nel grafo. In altre parole, essa corrisponde alla frazione di archi uscenti a cui è corrisposto un arco entrante, rispetto alla totalità degli archi. La formula equivalente è la seguente:

$$RA = \frac{n_{ar}}{n_a}\,, \qquad n_{ar} = |(u,v) \in G | (v,u) \in G|\,, \qquad n_a = |(u,v) \in G|$$

  Dove n_{ar} equivale al numero di archi bidirezionali, mentre n_a rappresenta il numero di archi totale. Si noti che, data la natura della rete in questione, ogni arco rappresenta effettivamente una comunicazione bidirezionale, dal momento che ad ogni pacchetto inviato da u verso v corrisponde un pacchetto inviato da v verso u. Dunque, in questo caso, la reciprocità stima la probabilità che, seguente ad una comunicazione iniziata da u e diretta a v, segua una comunicazione iniziata da v e diretta a u. Il valore di tale misura osservato nella rete in questione è 0.0864. Analogamente al caso della densità, un valore prossimo a 1 indicherebbe una forte propensione allo scambio dei ruoli nelle comunicazioni tra i dispositivi, cosa che non sembra dunque avvenire.
- Transitività. Per transitività si intende una misura della propensione dei nodi a perpetuare le loro connessioni a nodi adiacenti. In particolare, una condizione di perfetta transitività tra tre nodi comporta che, se il nodo u è collegato al nodo v, e il nodo v è collegato al nodo w, allora il nodo u è collegato al nodo w. Ciò si verifica raramente all'interno di reti reali, dal momento che la presenza di tale comportamento per ciascun nodo appartenente alla rete comporterebbe la presenza di un arco tra ogni coppia di nodi raggiungibili. Tuttavia, si noti che in un grafo orientato come quello corrispondente alla rete in esame, la transitività può essere studiata osservando diversi tipi di comportamenti assunti da triadi di nodi. Il cosiddetto *censimento delle triadi*, proposto da Davis e Leinhardt, risulta utile allo studio della transitività dal momento che classifica tutte le possibili triadi presenti in una rete secondo sedici categorie. Tali strutture, e l'associato numero di volte che esse si presentano nella rete in esame, sono riportate nella Tabella 3.3. Un'osservazione complessiva della tabella porta ad osservare che non sembra essere diffuso un comportamento influenzate tra nodi adiacenti che porterebbe al collegamento triangolare di una triade. Ciò si evince dal fatto che le forme di triadi con codice 030T e successivi, esclusa la forma identificata dal codice 201, sono quelle che indicano tale comportamento triangolare, e le frequenze associate a tali forme non risultano essere elevate.

Tabella 3.3: Tipi di comportamenti transitivi e numero di volte che ciascuno si presenta nella rete

Codice della triade	Forma della triade	Frequenza
003	u, v, w	225248
012	$u \rightarrow v, w$	12409
102	$u \leftrightarrow v, w$	691
021D	$u \leftarrow v \rightarrow w$	1806
021U	$u \rightarrow v \leftarrow w$	182
021C	$u \rightarrow v \rightarrow w$	25
111D	$u \leftrightarrow v \leftarrow w$	31
111U	$u \leftrightarrow v \rightarrow w$	35
030T	$u \rightarrow v \leftarrow w, u \rightarrow w$	19
030C	$u \leftarrow v \leftarrow w, u \rightarrow w$	0
201	$u \leftrightarrow v \leftrightarrow w$	9
120D	$u \leftarrow v \rightarrow w, u \leftrightarrow w$	1
120U	$u \rightarrow v \leftarrow w, u \leftrightarrow w$	7
120C	$u \rightarrow v \rightarrow w, u \leftrightarrow w$	1
210	$u \rightarrow v \leftrightarrow w, u \leftrightarrow w$	0
300	$u \leftrightarrow v \leftrightarrow w, u \leftrightarrow w$	0

- Diametro della rete. Sia il percorso geodetico, o percorso più breve, il minore numero di archi necessari a collegare due nodi, il diametro della rete corrisponde al più lungo percorso geodetico che si presenta all'interno di essa. In questo caso tale percorso ha una lunghezza di 4 archi, e in particolare, servendosi degli indirizzi IP dei dispositivi come identificativo dei nodi, esso è il seguente:

  0.0.0.0 → 255.255.255.255 → 172.24.1.1 → 172.24.1.101 → 224.0.0.252

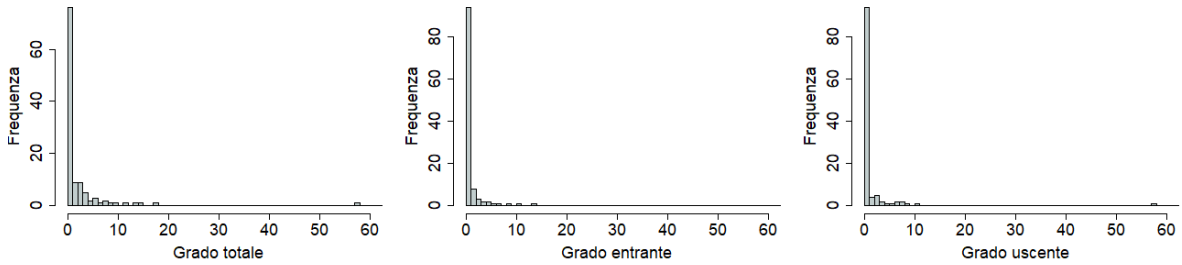
  Si noti, inoltre, che tale lunghezza equivale circa al doppio della lunghezza media dei percorsi geodetici presenti all'interno della rete, che è pari a 2.0549. Un diametro di 4 archi e una lunghezza media di percorsi approssimabile a 2 indicano che la rete in questione assume le caratteristiche di un cosiddetto *small-world*, termine che indica la tendenza di una rete ad avere un rapido flusso di transizione delle informazioni, che trova, da un punto di vista logico, tutti i nodi della rete "vicini".

Sono, poi, state calcolate una serie di misure utili a valutare la centralità e la centralizzazione della rete. Per centralità si intende la caratteristica di un particolare nodo che lo denota come "centrale", ovvero lo identifica come un nodo particolarmente influente relativamente alla porzione di rete che lo circonda. Il termine centralizzazione, invece, ha una connotazione più generale, si riferisce cioè ad una proprietà che caratterizza l'intera rete, e non una particolare frazione della stessa. La

procedura generale coinvolta in qualsiasi misura di centralizzazione del grafo è quella di osservare le differenze tra i punteggi di centralità del punto più centrale e quelli di tutti gli altri punti. La centralizzazione, quindi, è il rapporto tra la somma effettiva delle differenze e la somma massima teoricamente possibile delle differenze. Sono, quindi, state calcolate una serie di misure che permettono di osservare tali proprietà da diversi punti di vista:

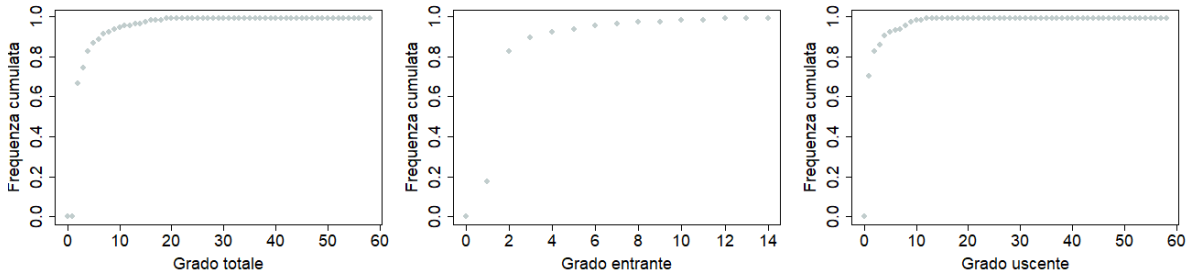
- Grado. Il grado di un nodo, citato precedente nel paragrafo 1.3, corrisponde al numero di archi incidenti nello stesso. Gli istogrammi presentati in Figura 3.6 rappresentano le frequenze dei valori osservati dei tre tipi di grado presenti in un grafo orientato come quello associato alla rete in esame, ovvero il grado totale, quello entrante e quello uscente.

  Figura 3.6: Istogrammi del grado totale, entrante e uscente



  Tali rappresentazioni grafiche indicano un andamento unimodale con moda pari a 0 o 1 per ciascun tipo di grado, denotando la natura individualista della maggior parte dei nodi appartenenti alla rete. In Figura 3.7, invece, si possono notare le frequenze cumulate di tali valori.

  Figura 3.7: Frequenze cumulate del grado totale, entrante e uscente

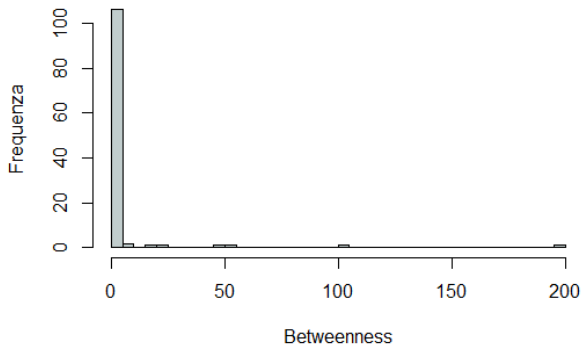


  Anche in questo caso i grafici assumono forme simili, denotando un significativo salto tra i primi valori e i successivi, per poi assumere un comportamento apparentemente logaritmico. Si nota, inoltre, come la stazionarietà della curva

sia raggiunta in corrispondenza di valori delle ascisse significativamente bassi, almeno per quanto riguarda il grado totale e quello uscente.

- Centralizzazione del grado. Valori prossimi a 1 potrebbero indicare una rete che presenta alcuni nodi fortemente influenti sugli altri, almeno dal punto di vista del grado. Tale proprietà non sembra verificarsi per la rete in esame, dal momento che la centralizzazione dei valori del grado totale, di quello entrante e di quello uscente ha prodotto, rispettivamente, i seguenti valori: 0.2461, 0.1112 e 0.5005. Tuttavia, è importante notare che le misure risultanti dalla centralizzazione forniscono un maggior valore informativo quando usate per confrontare diverse reti della stessa tipologia, e non considerando i valori calcolati come misura assoluta della proprietà che si sta osservando.
- Betweenness. La misura in questione rappresenta il numero totale di volte in cui un nodo si trova all'interno del percorso più breve che collega due nodi. Anche in questo caso è stato prodotto un istogramma che raffigura le frequenze dei valori di tale misura, riportato in figura 3.8.

Figura 3.8: Istogramma dei valori della betweenness

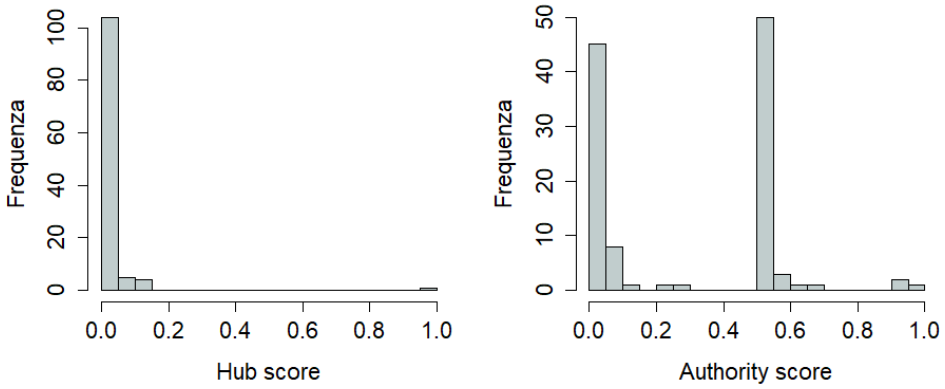


Dall'osservazione di tale grafico si può notare come, analogamente a quanto osservato per il grado, la moda della distribuzione si trovi in corrispondenza dei primi valori delle ascisse. Ciò è dovuto all'importante numerosità di nodi nei quali incide un solo arco, in particolare per quanto riguarda il cluster numero 4, osservabile in Figura 3.5.

- Centralizzazione della betweenness. La centralizzazione di tale misura ha portato al valore 0.0156, conforme con i risultati ottenuti dalla centralizzazione del grado, indicando un ancora più estremo scostamento dall'ipotesi che la rete presenti nodi significativamente influenti globalmente.
- Hub e authority score. Il termine hub è associato ad un nodo il quale grado è positivamente sproporzionato rispetto alla media dei gradi calcolati sugli altri nodi della rete, mentre per authority si intende un nodo che condivide archi con

molteplici hub. Un comportamento comune che si osserva generalmente tra i nodi di una rete vede che un attore fortemente considerato hub tende a puntare a molte buone autorità e un attore con alta autorità tende a ricevere da molti buoni hub. Il punteggio di autorità di un vertice è quindi proporzionale alla somma dei punteggi hub dei vertici sui legami in entrata e il punteggio hub è proporzionale ai punteggi di autorità dei vertici sui legami in uscita. Gli istogrammi relativi a tali punteggi sono riportati in Figura 3.9.

Figura 3.9: Istogrammi dei valori di hub score e authority score



Il grafico relativo all'hub score presenta un andamento unimodale con moda in prossimità di valori molto bassi, infatti i nodi con punteggio hub maggiore di 0.1 risultano essere solamente 5, il 4.38% del numero totale di nodi presenti sulla rete. Ciò indica che sembra esserci una scarsa numerosità di hub all'interno della rete. Per quanto riguarda la distribuzione osservata del punteggio authority, essa presenta invece un comportamento bimodale, con mode in prossimità di valori nulli e di valori a metà della scala di riferimento. In questo caso, infatti, il numero di nodi con authority score maggiore di 0.1 è pari a 61, ovvero il 53.51% del totale. Sembra quindi esserci un numero importante di nodi considerabili come autorità.

Capitolo 4

Modelli di classificazione per i nodi

4.1 Suddivisione della rete

Per gli scopi dell'analisi, ovvero la classificazione di nodi come attaccanti o normali, è stato stimato un opportuno modello statistico. Precedentemente a tale operazione, è stato di fondamentale importanza organizzare i dati a disposizione per ottenere un'affidabile previsione del tipo di nodo. A tale scopo è stato praticato il cosiddetto *train-test split*, una tecnica per valutare la prestazione di un modello statistico o di un algoritmo di apprendimento automatico. Tale procedura può essere utilizzata per problemi di classificazione e di regressione, e prevede la suddivisione in due sottoinsiemi del set di dati a disposizione. Il primo sottoinsieme è usato per adattare il modello e viene chiamato dataset di addestramento, o training set. Mentre il secondo, chiamato test set, viene fornito al modello già stimato, per poi calcolare delle previsioni che vengono comparate ai valori effettivamente osservati. Questa tecnica simula l'utilizzo reale di un modello di regressione, dal momento che esso viene stimato sui dati a disposizione, i quali sono etichettati da una variabile che classifica le osservazioni, in questo caso tale variabile è *label*, variabile dicotomica responsabile della discriminazione tra nodi comuni e malevoli. Successivamente, vengono calcolate delle previsioni utilizzando il modello su dati non classificati, di cui cioè non si conosce la reale appartenenza ai due gruppi. Trattandosi di una simulazione della realtà, in questo caso si dispone dell'etichetta dei nodi di entrambi i set di dati, il che risulterà utile nel momento in cui verranno confrontate le etichette reali dei nodi appartenenti al dataset di testing con quelle stimate dal modello, andando a verificarne le capacità previsive. Si noti che la suddivisione della rete in train set e test set è stata praticata rispetto ai nodi, e non rispetto agli archi, per via del fatto che il soggetto della classificazione sono i vertici della rete, e non le connessioni tra gli stessi. La procedura di suddivisione del dataset presenta un parametro in particolare, ovvero la dimensione dei due subset. La decisione sulle dimensioni dei due sottoinsiemi di dati deve solitamente prendere in considerazione i seguenti concetti: costo computazionale dell'addestramento del modello, costo computazionale della valutazione del modello, rappresentatività del set di addestramento, rappresentatività del test set. Ciò considerato, sono comunque presenti delle configurazioni di uso comune per problemi analoghi a quello in esame, le quali, espresse in percentuali rispetto alla totalità del numero di soggetti, sono riportate di seguito:

- Train 80%, test 20%.
- Train 67%, test 33%.
- Train 50%, test 50%.

La configurazione scelta è la prima, che vede quindi l'80% circa dei dispositivi selezionati per l'addestramento del modello, e i rimanenti utilizzati per valutarne le capacità previsive. Si noti che, in questo caso, tali percentuali si riferiscono alla rete una volta rimossa la comunità numero 6, dal momento che essa risulta sconnessa dal resto del grafo e non include al suo interno alcun nodo attaccante, non rappresentando dunque un utile gruppo di vertici per gli scopi dell'analisi. Risulta cruciale adottare un'opportuna strategia per la selezione specifica dei nodi da destinare ai due subset. Infatti, dal momento che i dati in questione rappresentano una rete, si vuole mantenere una struttura rappresentativa dei comportamenti comunicativi all'interno del train set e del test set. I vari tentativi effettuati per l'individuazione dell'opportuna strategia di suddivisione del dataset sono riportati di seguito:

- Suddivisione temporale. Secondo questo approccio, la selezione è stata effettuata secondo gli archi: sono stati selezionati i nodi partecipanti alle comunicazioni avvenute entro una determinata data ed essi sono stati assegnati al dataset di training, mentre i rimanenti hanno costituito il test set. I problemi legati a questo approccio sono i seguenti:
    - I due dataset risultanti non sono mutualmente esclusivi, vi è cioè la possibilità di osservare il medesimo nodo all'interno di entrambi i subset. Ciò è dovuto al fatto che la selezione è stata effettuata secondo gli archi, e non secondo i nodi, infatti non è ragionevole sostenere che un dato nodo interrompa le sue attività di comunicazione con gli altri nodi da un determinato istante temporale in poi, e che l'istante di interruzione delle comunicazioni di ciascun nodo di un subset della rete avvenga entro una certa soglia utilizzabile per la suddivisione del dataset.
    - La variabile associata alla data, corrispondente all'avvenuta comunicazione tra due dispositivi, presenta valori mancanti. Per utilizzare questa strategia bisognerebbe quindi andare a rimuovere gli archi per i quali non è nota la corrispondente data, il che risulterebbe essere una decisione imperfetta dal momento che il nodo hub della comunità numero 4 è un attaccante, i quali attacchi presentano informazioni mancanti circa la data in cui essi sono avvenuti. Per tale ragione, inoltre, la variabile data è stata rimossa dal dataset come le altre variabili presentanti valori assenti.
- Suddivisione risultante dall'ottimizzazione dei rapporti di attacchi sul totale degli archi. Questa strategia prevedeva la selezione arbitraria dei nodi che consentisse di ottenere simili proporzioni di attacchi all'interno delle due sottoreti risultanti dalla suddivisione del grafo in train set e test set e all'interno

della rete originale. L'obiettivo di tale approccio era quello di ottimizzare la rappresentatività dei due subset, tentando di ottenere all'interno degli stessi un comportamento relativo alla proporzione degli attacchi simile a quello osservato nella rete originale. Tuttavia, oltre ad essere computazionalmente inefficiente, dal momento che prevedeva la selezione manuale dei nodi, questa strategia comportava una significativa alterazione della struttura topografica della rete, andando a costruire due sottoreti fortemente sconnesse internamente.

- Suddivisione per comunità. L'approccio in questione consisteva nell'associare determinati cluster individuati dall'*algoritmo di Louvain* al subset di addestramento, utilizzando i restanti per la valutazione del modello. Questa strategia è risultata essere vincente, non dimostrando problematiche di sorta e consentendo di mantenere inalterate le strutture comunicative tra nodi relativamente alla porzione di rete più prossima agli stessi.

In particolare, con il fine di suddividere il dataset secondo la proporzione 80/20% citata precedentemente, i nodi appartenenti ai cluster 2, 3, 4, e 5 sono stati assegnati al dataset di training, mentre la comunità 1 corrispondeva al test set. Dunque, il set per l'addestramento contava un totale di 82 nodi rispetto ai 99 appartenenti alla rete originale, costituendo l'82.83% del totale, mentre il test set conteneva 17 nodi, ovvero il 17.17% del totale dei nodi. Informazioni più dettagliate riguardo nodi e archi appartenenti ai due subset sono contenute nella Tabella 4.1.

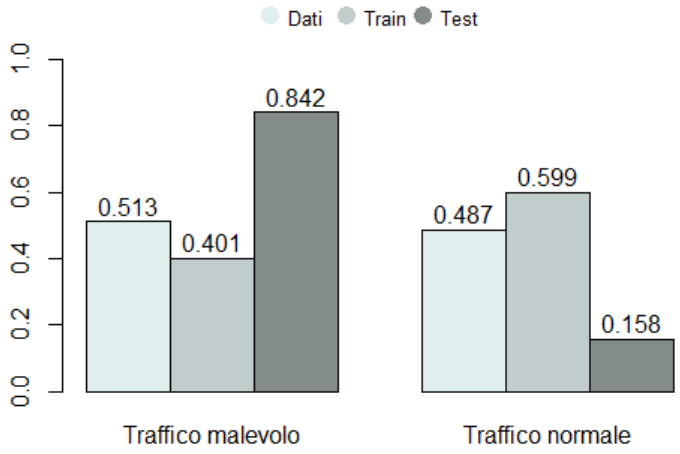
Tabella 4.1: Informazioni riguardanti nodi e archi dei tre dataset

	Dataset completo	Training set	Testing set
Numero di nodi origine	34	15	11
Numero di nodi attaccanti	10	6	4
Numero di nodi vittime	7	6	2
Numero totale di archi	779047	581121	197334
Numero di attacchi	399263	233154	166109
Numero di archi normali	379784	347967	31225

Si noti che per nodo origine si intende un dispositivo responsabile di almeno una comunicazione, ovvero un nodo presentante almeno un arco uscente. Tale valore sembra essere particolarmente basso all'interno dei tre dataset, quando comparato col rispettivo totale dei nodi appartenente a ciascuna rete o sottorete, indicando, ancora una volta, la natura generalmente passiva dei nodi appartenenti alla rete. Inoltre, si nota come la somma del numero di vittime del training set e del testing set sia maggiore al valore osservato nel dataset completo. Ciò è dovuto al fatto che un nodo può essere vittima di attacchi provenienti da più offensori, e per via del fatto che la suddivisione del dataset è stata operata rispetto ai nodi origine, assicurandosi cioè che essi non appartenessero ad entrambi i subset. Dunque, si è verificata la presenza di alcuni nodi

appartenenti sia al dataset destinato all'addestramento che al dataset per la valutazione. Tale avvenimento non inficia, tuttavia, la stima e la valutazione del modello, dal momento che i nodi da classificare sono esclusivamente i nodi origine, per via del fatto che un nodo passivo non può essere responsabile di un attacco. Un'osservazione più approfondita del ruolo dei nodi attaccanti permette di notare che, sia per quanto riguarda il train set che il test set, essi talvolta risultano essere anche vittime di attacco, confermando la presenza sulla rete del primo scenario citato nella descrizione della Figura 3.4, il quale vede la presenza di nodi offensori che operano attacchi nei confronti di altri nodi malevoli. Infine, si può osservare come il numero di archi normali, per quanto riguarda il test set, sia significativamente inferiore al numero di attacchi. Tale comportamento, più facilmente apprezzabile tramite la visione della Figura 4.1, non sembra essere rappresentativo di ciò che accade nel dataset completo. Tuttavia, ciò non comporta un sostanziale problema per il fine dell'analisi, dal momento che il problema sorgerebbe se tale sbilanciamento fosse dovuto ad un numero di archi normali particolarmente maggiore al numero di attacchi, per via del fatto che in tal caso la sottorete potrebbe non consentire di catturare il comportamento degli offensori una volta stimato il modello.

Figura 4.1: Frequenze relative di archi normali e attacchi all'interno delle tre reti



4.2 Costruzione della matrice del modello

Per costruire la matrice del modello di regressione utilizzato per la classificazione dei nodi è stato utilizzato un approccio analogo a quello descritto nel paragrafo 3.2 per la costruzione della matrice del modello comprendente le sole variabili problematiche. In particolare, considerando le informazioni contenute nel dataset di training, si è costruito un nuovo data frame che vede nelle righe i nodi origine appartenenti alla

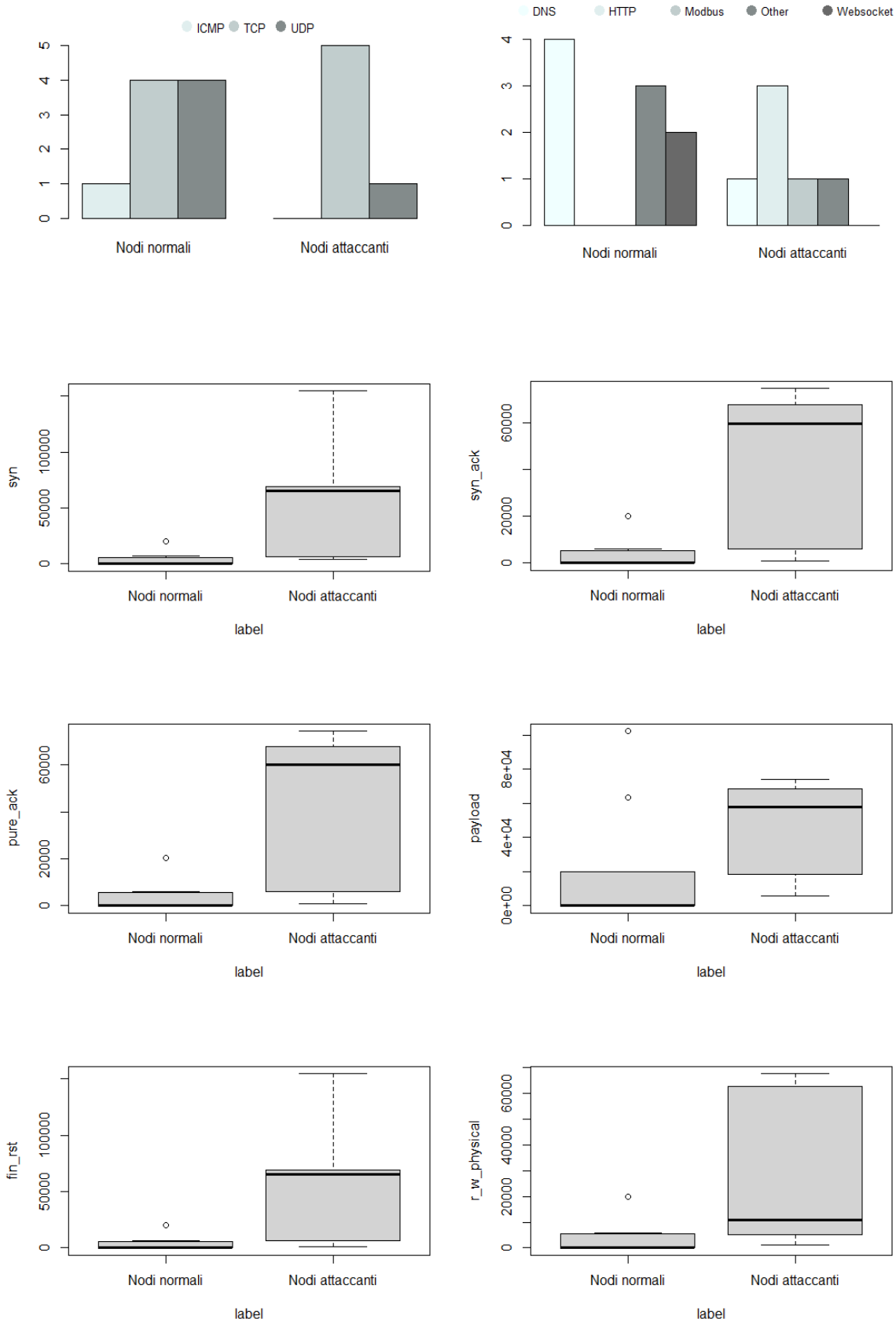
sottorete, mentre nelle colonne è riportato il risultato di un'elaborazione praticata sulle variabili descriventi gli archi del grafo. I dettagli sulle modalità secondo le quali le variabili sono state elaborate sono riportati di seguito:

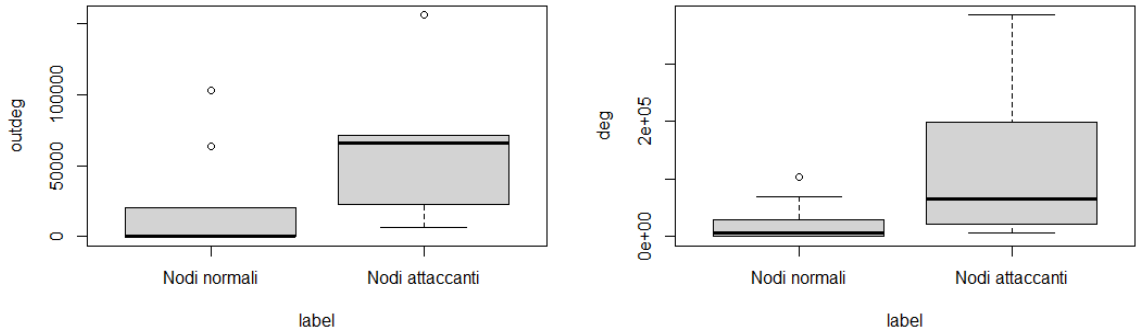
- La variabile *src_ip*, descrivente l'indirizzo IP del nodo origine dell'arco, è stata utilizzata per identificare i nodi origine.
- Le variabili *des_ip*, *des_port*, *src_port* e *timestamp* sono state ignorate per le motivazioni riportate nel seguito. Per quanto riguarda la variabile *des_ip*, essa descrive l'indirizzo IP del dispositivo ricevente la comunicazione, è quindi stata ignorata dal momento che il dataset finale riporterà esclusivamente nodi origine. Una motivazione analoga ha, quindi, portato all'esclusione di *des_port*, variabile che corrisponde alla porta del dispositivo ricevente nella quale la comunicazione è stata recepita. La variabile *src_port*, descrivente la porta del dispositivo origine dalla quale la comunicazione è stata inviata, è invece stata esclusa per il fatto che essa si riferisce ad una proprietà specifica del tipo di dispositivo, e non può, quindi, essere rielaborata senza tenere conto di tale proprietà, informazione non presente all'interno del dataset. Per quanto riguarda *timestamp*, codice relativo all'istante temporale nel quale la comunicazione è stata inviata, esso risulterebbe difficilmente interpretabile, ed è per tale ragione stato rimosso.
- Le variabili *protocol* e *service*, essendo qualitative sconnesse, sono state rielaborate in modo da ottenere, per ciascun nodo origine, il livello più frequentemente osservato di tali variabili sull'intera collezione di archi uscenti da tale nodo. Dunque, ad ogni nodo origine, è stato associato il protocollo comunicativo maggiormente utilizzato dallo stesso per connettersi agli altri dispositivi presenti sulla rete, per quanto riguarda l'elaborazione della variabile *protocol*. Rispetto a *service*, invece, ad ogni nodo origine è stato associato il protocollo applicativo maggiormente utilizzato dai destinatari per la ricezione delle comunicazioni da lui praticate.
- La variabile *conn_state*, qualitativa ordinale, è stata rielaborata in maniera analoga a *protocol* e *service*. Essa rappresenta lo stato della connessione tra i due dispositivi interessati dall'arco, ed è espressa secondo i tre livelli: completa, stazionaria e parziale. Ad ogni nodo origine è stato, dunque, associato il livello più frequentemente osservato di tale variabile rispetto alla totalità degli archi uscenti da tale nodo.
- La variabile *ossec_alert_level*, dal momento che descrive un determinato livello di allerta segnalato da un sensore OSSEC e associato ad una particolare comunicazione, è stata rielaborata ottenendo, per ciascun nodo origine, il più alto livello di allerta osservato all'interno della collezione di archi uscenti da tale nodo. Ad ogni nodo mittente è stato dunque associato un valore da 0 a 10, corrispondente al massimo livello di allerta registrato che è stato attribuito alle comunicazioni da esso effettuate.

- Le restanti variabili, essendo dicotomiche, sono state elaborate in modo da ottenere, per ciascun nodo origine, il numero di volte in cui si è osservato il valore 1 di ciascuna variabile rispetto alla totalità degli archi uscenti da tale nodo. Considerando l'esempio della variabile *login_attmp*, essa rappresenta, per ciascun arco, la risposta alla domanda "il nodo origine ha effettuato un tentativo di autenticazione per poter accedere a dati presenti nella memoria del dispositivo contattato?". Dunque, ad ogni nodo origine della rete è stato associato il numero di volte in cui a tale domanda la risposta è stata "sì" nel totale delle comunicazioni da esso effettuate.

Si veda l'Appendice C per osservare il codice relativo alla rielaborazione delle variabili. Inoltre, in aggiunta alle variabili osservate sulla struttura fisica della rete, sono state calcolate una serie di variabili tratte dall'osservazione topologica del grafo. In primo luogo, sono state calcolate alcune misure di centralità, descritte nel paragrafo 3.4, in particolare si tratta del grado totale, del grado entrante, del grado uscente e della betweenness di ciascun nodo origine presente nella sottorete. Successivamente, servendosi del *metodo di Louvain* descritto nel paragrafo 3.3, sono stati individuati i cluster nei quali i nodi risultano raggruppati. Si noti che tali cluster non corrispondono necessariamente ai cluster 2, 3, 4 e 5 della rete originaria che compongono la sottorete di training, dal momento che il grafo in questione può presentare proprietà differenti da quelle della rete originale, per via del fatto che sono stati rimossi i nodi raggruppati nella comunità 1, destinati alla sottorete di testing. Infatti, il clustering sul grafo in esame ha permesso di riconoscere la presenza di tre comunità, e non quattro. Si noti che, tuttavia, tale variabile non può essere utilizzata direttamente, dal momento che non è possibile aspettarsi con certezza che futuri grafi osservati sulla stessa rete o su altre reti presentino lo stesso numero di comunità. Per tale ragione, sono state calcolate le seguenti misure, descritte nel paragrafo 3.4, per ciascuno dei cluster individuati: densità di archi, distanza media e diametro. Ad ogni nodo origine sono state, dunque, associate tali misure riferite alla comunità a cui esso appartiene, con il fine di osservare se le proprietà strutturali del cluster a cui un determinato nodo appartiene possano essere utili per determinarne il tipo di comportamento, malevolo o meno. Infine, alla matrice del modello è stata aggiunta la variabile *label*, descrivente il tipo di nodo come attaccante o normale. Tale variabile etichetta il nodo come offensore nel caso esso avesse effettuato almeno una comunicazione malevola. Con l'obiettivo di effettuare una prima valutazione visiva della possibile significatività delle variabili descritte per il fine di discriminare i nodi offensori da quelli comuni, sono stati prodotti diversi grafici. In particolare, per quanto riguarda le variabili qualitative, sono stati prodotti dei grafici a barre rappresentanti le frequenze dei vari livelli di tali variabili condizionatamente al tipo di nodo. Per quanto riguarda le variabili quantitative, invece, i grafici prodotti sono boxplot, rappresentanti la variabilità dei conteggi di esito pari a 1 di ciascuna variabile condizionatamente al tipo di nodo.

Figura 4.2: Grafici relativi alle variabili apparentemente rilevanti, rispetto alla variabile label





In Figura 4.2 si possono osservare i grafici relativi alle sole variabili che potrebbero apparentemente risultare utili ai fini dell'analisi.

Successivamente, rispettivamente alle variabili quantitative presentanti un comportamento interessante dal punto di vista grafico e in analogia a quanto fatto nel paragrafo 3.2 per lo studio delle variabili presentanti valori assenti, è stata studiata la presenza di omoschedasticità tra i gruppi dettati dalla variabile label rispetto alle variabili in esame, con l'obiettivo di effettuare un test ANOVA ad una via sulle stesse. I risultati relativi a tale studio, riportati in Tabella 4.2, indicano che, con l'eccezione della variabile *payload*, le varianze entro i gruppi sembrano essere significativamente diverse per ciascuna delle variabili sotto studio, non rispettando, dunque, l'ipotesi di omoschedasticità sottostante il test ANOVA ad una via.

Tabella 4.2: Analisi dell'omogeneità della varianza nei gruppi dettati dalla variabile label

Variabile	Label	Deviazione standard
syn	0	6821
	1	54991
syn_ack	0	6769
	1	32676
pure_ack	0	6776
	1	32634
payload	0	36798
	1	28136
fin_rst	0	6765
	1	55682
r_w_physical	0	6671
	1	30458
outdeg	0	37105
	1	52152
deg	0	37190
	1	144414

Simmetricamente a quanto svolto nel paragrafo 3.2, è stato quindi operato il meno esigente *test Kruskal-Wallis* su tali variabili, con l'obiettivo di valutare statisticamente la presenza di comportamenti diversi tra attaccanti e nodi normali rispetto alle variabili in esame. L'osservazione dei risultati prodotti da ciascun test, osservabili in Tabella 4.3, permette di appurare che le variabili *payload* e *degree* sono le uniche la cui utilità, se utilizzate per la stima di un modello di regressione, risulta essere dubbia, dal momento che il p-value dei test ad esse associato non sembra indicare la presenza di comportamenti diversi tra nodi attaccanti e normali rispetto a tali variabili, una volta confrontati tali valori con una soglia nominale del 10%. Per quanto riguarda le altre variabili, esse invece risultano potenzialmente significative per la stima di un modello anche se il corrispondente p-value del test associato è confrontato ad un valore soglia del 1%, tranne per quanto riguarda *outdegree* che richiederebbe una soglia del 5% per poter trarre tale conclusione.

Tabella 4.3: Risultati del test Kruskal-Wallis condotto rispetto alle variabili apparentemente rilevanti

Variabile	Statistica H	Gradi di libertà	P-value
syn	7.1704	1	0.007412
syn_ack	7.1704	1	0.007412
pure_ack	7.1704	1	0.007412
payload	3.1306	1	0.07684
fin_rst	7.837	1	0.005119
r_w_physical	5.1996	1	0.02259
outderee	4.5081	1	0.03374
degree	3.5619	1	0.05912

Per quanto riguarda le variabili qualitative *protocol* e *service*, non essendo possibile praticare il *test Kruskal-Wallis* data la loro natura, la loro significatività per gli scopi dell'analisi è stata valutata tramite il *test esatto di Fisher*, test usato per determinare la significativa presenza di una relazione di dipendenza tra due variabili qualitative, che prevede il confronto tra la frequenza di ogni categoria per una variabile con le categorie della seconda, che in questo caso è la variabile *label*. I p-value prodotti da tale test rispetto alle variabili *protocol* e *service* sono, rispettivamente, 0.4118 e 0.07546, indicando che la sola variabile *service* sembra dimostrare una relazione di dipendenza con la variabile *label*, come si evince confrontando tali valori con una soglia nominale del 10%. Si conclude, quindi, che le variabili utilizzate per l'adattamento dei modello di regressione sono *syn*, *syn_ack*, *pure_ack*, *payload*, *fin_rst*, *r_w_physical*, *outdegree*, *degree* e *service*.

4.3 Adattamento del modello

Per via della natura dicotomica della variabile risposta *label*, con valori 0 per i nodi normali e 1 per i nodi attaccanti, il modello adattato ai dati è il modello lineare generalizzato di regressione logistica. Adottando una differente nomenclatura per comodità, sia y_i = 1 se l' i-esimo nodo è attaccante e 0 altrimenti, il modello in questione assume che $y_1, \dots, y_{15}$ siano realizzazioni di variabili casuali indipendenti $Y_1, \dots, Y_{15}$ aventi distribuzione binomiale di parametri $\mu_1, \dots, \mu_{15}$, si ha quindi $Y_i \sim Bi(1, \mu_i)$, con $g(\mu_i) = \eta_i = \sum_{r=1}^{p} \beta_r x_{ir} = \boldsymbol{x}_i\beta$ predittore lineare, per i che va da 1 a 15, dove β_r è il valore stimato dal modello per il coefficiente associato all'r-esima variabile esplicativa presente nel modello, mentre x_{ir} è il valore dell'i-esima unità statistica in corrispondenza dell'r-esima variabile osservato nei dati. Infine, p corrisponde al numero di coefficienti associati alle variabili presenti nel modello. La funzione legame $g(\cdot)$ utilizzata è il legame canonico, ovvero la funzione logistica:

$$g(\mu_i) = \log\left(\frac{\mu_i}{1-\mu_i}\right) = \boldsymbol{x}_i\beta$$

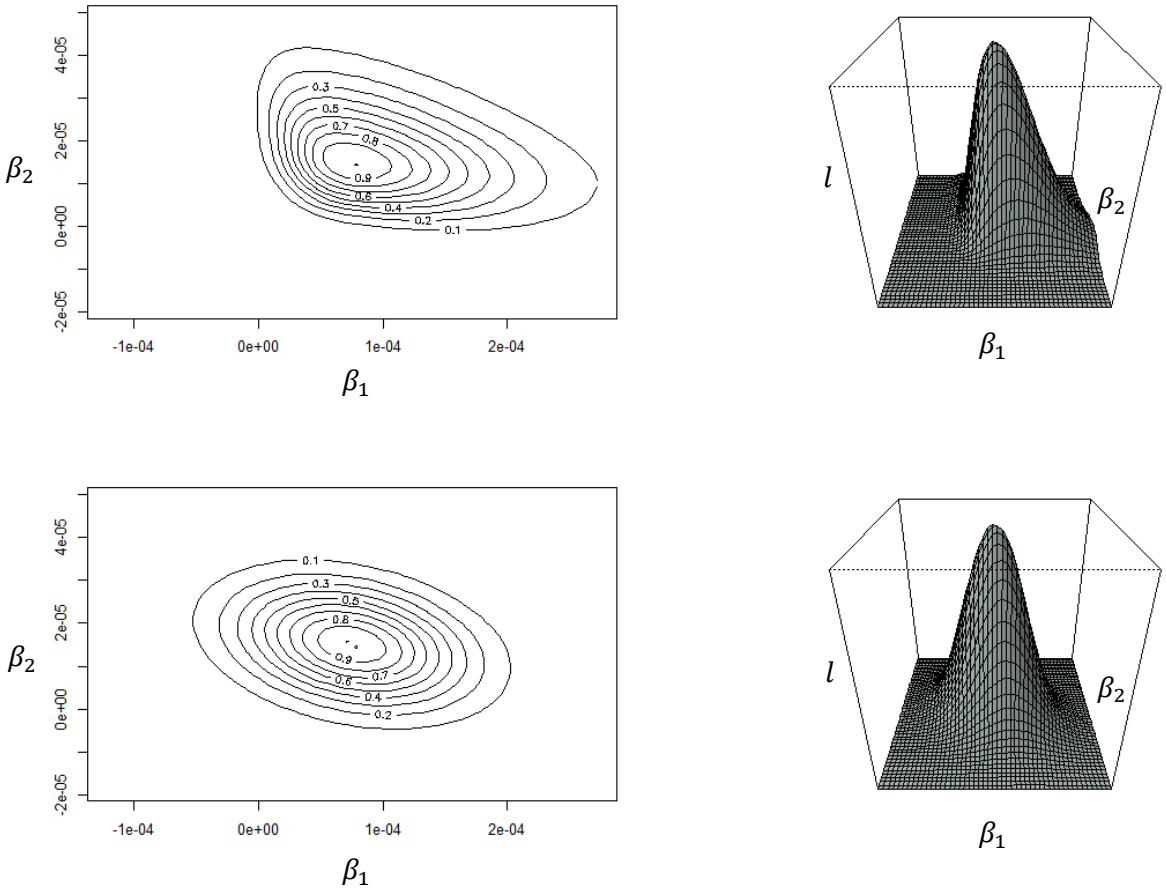
La quale inversa risulta essere la seguente:

$$\mu_i = \frac{e^{\boldsymbol{x}_i\beta}}{1 + e^{\boldsymbol{x}_i\beta}}$$

Vista la ridotta dimensione del campione a disposizione, dovuta al fatto che i nodi origine appartenenti al dataset di training sono solo 15, non è stato possibile stimare un modello che contenesse tutte le nove variabili individuate nel paragrafo precedente per poi adottare una strategia *backward selection*, che prevede cioè di rimuovere iterativamente la variabile meno significativa ottenendo infine un modello parsimonioso che consenta di prevedere correttamente i dati. È stata dunque utilizzata una metodologia opposta, che prevedeva l'adattamento di un modello comprendente una sola variabile, per poi valutare iterativamente l'apporto di informazione tratto dall'inserimento nel modello di un'altra variabile, fino ad ottenere il modello ottimale. Tale strategia prende il nome di *forward selection*. Questo metodo di selezione delle variabili per la stima del modello presenta, tuttavia, la caratteristica di essere dipendente dalla variabile utilizzata per iniziare l'algoritmo. Per ovviare a tale problema, la strategia è stata applicata nove volte, una per ogni variabile considerata, variando la prima variabile selezionata e ignorando le combinazioni di variabili osservate nelle precedenti implementazioni dell'algoritmo per evitare ripetizioni. Tale procedura ha consentito di ottenere due modelli candidati utili ai fini dell'analisi, i quali vedono l'utilizzo delle variabili *syn* e *fin_rst*, per il modello 1, e *r_w_physical* e *degree* per il modello 2. Prima di procedere all'analisi e al confronto dei due modelli ottenuti, è importante specificare che la selezione delle variabili non è stata effettuata sulla base

di test di significatività Wald, tipicamente utilizzato in contesti simili a quello in esame e considerato il metodo di default da diversi linguaggi di programmazione e funzioni dell'ambiente R quale la funzione `summary`, bensì è stato utilizzato il test del log rapporto di verosimiglianza. Infatti, entrambi i test prevedono l'approssimazione parabolica della funzione di verosimiglianza, ma le ipotesi utilizzate da i due test differiscono sensibilmente: mentre il test del log rapporto di verosimiglianza richiede che esista una trasformazione dei dati tale che la forma della verosimiglianza è ragionevolmente approssimabile ad un comportamento parabolico, il test Wald richiede che l'approssimazione parabolica sia ragionevole rispetto alla scala dei parametri di cui si testa la significatività. La rappresentazione di tale approssimazione parabolica per il modello 2 è osservabile in Figura 4.3. Si veda l'Appendice D per osservare il codice relativo alla visualizzazione di tali grafici.

Figura 4.3: Curve di livello (a sinistra) e rappresentazione 3D (a destra) della verosimiglianza (in alto) e della sua approssimazione parabolica (in basso)



Si noti che la log-verosimiglianza normalizzata è stata calcolata tramite la seguente formula:

$$l = \sum_{i=1}^{15} (y_i \, (\beta_0 + \beta_1 x_{i1} + \beta_2 x_{12}) - \log\left(1 + e^{\beta_0 + \beta_1 x_{i1} + \beta_2 x_{12}}\right))$$

Mentre la corrispondente approssimazione parabolica è stata calcolata come segue:

$$l \doteq -\frac{1}{2} (\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}})^t \, j\left(\widehat{\boldsymbol{\beta}}\right) (\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}})$$

Dove $j\left(\widehat{\boldsymbol{\beta}}\right)$ è la matrice di informazione osservata del modello, ottenuta tramite l'inversione della matrice di covarianza del modello stesso.

Dall'osservazione grafica della figura si evince, quindi, che non è ragionevole aspettarsi che la verosimiglianza presenti un comportamento parabolico, inficiando i risultati prodotti dal test Wald. Il problema potrebbe essere causato dalla ridotta dimensione del campione utilizzato per l'adattamento del modello. La discrepanza tra gli esiti prodotti dai due test, con riferimento ai modello 1 e 2, è osservabile nelle tabelle 4.4-4.7.

Tabella 4.4: Esiti del test di significatività Wald sul modello 1

Coefficiente	Stima	Standard error	Valore z	Pr(>|z|)
Intercetta	-2.889031	1.354641	-2.133	0.033
syn	0.001666	0.001186	1.405	0.160
fin_rst	-0.001563	0.001174	-1.331	0.183

Tabella 4.5: Esiti del test del log rapporto di verosimiglianza sul modello 1

	Gradi di libertà	Devianza	GdL residui	Devianza residua	Pr(>Chi)
NULL			14	20.1904	
syn	1	9.5445	13	10.6458	0.002005
fin_rst	1	3.3411	12	7.3047	0.067568

Tabella 4.6: Esiti del test di significatività Wald sul modello 2

Coefficiente	Stima	Standard error	Valore z	Pr(>|z|)
Intercetta	-2.086e+00	1.025e+00	-2.035	0.0419
r_w_physical	7.482e-05	6.560e-05	1.141	0.2540
deg	1.507e-05	1.126e-05	1.338	0.1810

Tabella 4.7: Esiti del test del log rapporto di verosimiglianza sul modello 2

	Gradi di libertà	Devianza	GdL residui	Devianza residua	Pr(>Chi)
NULL			14	20.190	
r_w_physical	1	4.9314	13	15.259	0.02637
deg	1	3.1599	12	12.099	0.07547

L'osservazione delle tabelle sopra riportate consente, dunque, di notare come il secondo test indichi la rilevanza dei coefficienti delle variabili in esame, mentre gli stessi coefficienti sono ritenuti non significativamente diversi da zero secondo il test Wald.

4.4 Interpretazione del modello

La funzione di legame logistica permette di interpretare il predittore lineare $\eta_i = \log\left(\frac{\mu_i}{1-\mu_i}\right)$ in termini di logaritmo della quota (*log-odds*). La quota in questione è, appunto, $\frac{\mu_i}{1-\mu_i}$, ovvero il rapporto tra probabilità di successo e insuccesso. Di conseguenza, il generico coefficiente β_r esprime l'effetto sul logaritmo della quota di un incremento unitario di x_{ir}, fermo restando il valore delle ulteriori variabili esplicative del modello, per r che va da 1 a p e che rappresenta l'indice dell'r-esimo coefficiente, e per i che va da 1 a n e che rappresenta l'indice dell'i-esima unità. Con due variabili esplicative quantitative, se si incrementa una variabile esplicativa di una unità, passando da x_{ir} a $x_{ir} + 1$, la variazione del predittore lineare è pari a β_r, fermo restando l'altra variabile esplicativa. La quota corrispondente risulta quindi moltiplicata per e^{β_r}. Per quanto riguarda il modello 1, si ha $\beta_1 = 0.001666$ e $\beta_2 = -0.001563$, valori ottenuti tramite le procedure illustrate nel paragrafo 4.3. Seppur tali valori siano prossimi allo zero, si noti che il test del log-rapporto di verosimiglianza ne ha saggiato la significativa diversità dal valore nullo, indicandone la rilevanza per la stima dei dati. Dunque, incrementando x_{i1} di una unità, fermo restando il valore di x_{i2}, la variazione del predittore lineare è pari a 0.001666, di conseguenza la quota corrispondente risulta moltiplicata per $e^{\beta_1} = e^{0.001666} = 1.00166739$. Mentre, per quanto riguarda β_2, per un incremento unitario di x_{i2} si ottiene una variazione del predittore lineare pari a -0.001563, e la quota corrispondente risulta moltiplicata per $e^{\beta_2} = e^{-0.001563} = 0.9984382$. Per quanto riguarda il secondo modello, si ha $\beta_1 = 0.00007482$ e $\beta_2 = 0.00001507$. Dunque, incrementando x_{i1} di una unità, fermo restando il valore di x_{i2}, la variazione del predittore lineare è pari a 0.00007482, di conseguenza la quota corrispondente risulta moltiplicata per $e^{\beta_1} = e^{0.00007482} = 1.00007482$. Mentre, per quanto riguarda β_2, per un incremento unitario di x_{i2} si

ottiene una variazione del predittore lineare pari a 0.00001507, e la quota corrispondente risulta moltiplicata per $e^{\beta_2} = e^{0.00001507} = 1.00001507$. Sembra, dunque, che l'incremento unitario del valore delle variabili non abbia un impatto significativo sulla quota $\frac{\mu_i}{1-\mu_i}$ in entrambi i modelli. È, tuttavia, rilevante notare il fatto che i valori delle variabili in questione osservati sul dataset variano di diversi ordini di grandezza. Con riferimento al dataset di addestramento, considerando le variabili utilizzate per la stima del modello 1, si ha che il range di valori della variabile *syn* è (0, 154518), mentre il range di valori associato alla variabile *fin_rst* è (0, 154669). Per quanto riguarda le variabili utilizzate per l'adattamento del secondo modello, invece, il range di valori per la variabile *r_w_physical* è (0, 67677), mentre per quanto riguarda *degree*, l'associato range di valori è (1, 384282). La variabilità dei valori osservati sembrerebbe, quindi, giustificare la significatività di tali variabili per il fine di discriminare gli utenti malevoli da quelli comuni.

4.5 Confronto dei modelli

Per il confronto dei due modelli in competizione sono state utilizzate diverse procedure e misure che sono illustrate nel seguito.

- Criteri di informazione. I criteri di informazione calcolati per il confronto dei due modelli sono l'*Akaike Information Criterion* (AIC) e il *Bayesian Information Criterion* (BIC), i quali identificano il modello preferibile tra i candidati come quello che minimizza il valore osservato derivante dal calcolo di tali criteri. La caratteristica fondamentale di tali criteri è il fatto che essi valutano come migliore il modello in grado di massimizzare la verosimiglianza penalizzando la presenza di un numero eccessivo di variabili. Le formule per il calcolo dell'AIC e del BIC sono riportate a seguire:

$$AIC = 2p - 2\ln(\hat{L})$$
$$BIC = p \cdot ln(n) - 2\ln(\hat{L})$$

  Dove: p è il numero di variabili esplicative utilizzate per l'adattamento del modello, n è la dimensione del campione e $\hat{L}$ è la stima di massima verosimiglianza del modello. Risulta quindi evidente come il BIC, qualora $ln(n)$ sia maggiore di 2, ovvero per n maggiore di 8 come in questo caso, propenda alla selezione di modelli più parsimoniosi, dando maggiore peso al numero di variabili utilizzate per la stima del modello rispetto all'indice AIC. I risultati prodotti dal calcolo dei due criteri per i due modelli in esame sono riportati nella Tabella 4.8.

Tabella 4.8: Valori osservati dei criteri AIC e BIC per i modelli 1 e 2

	AIC	BIC
Modello 1	13.30470	18.09901
Modello 2	15.42885	20.22316

L'osservazione dei risultati prodotti dal calcolo di tali indici in corrispondenza dei due modelli in esame porta, in entrambi i casi, a preferire il modello 1.

- Analisi grafica dei residui. La produzione dei quattro grafici mostrati nelle figure 4.4 e 4.5 rispettivamente ai modelli 1 e 2 sono utili allo studio dei seguenti comportamenti, riportati nello stesso ordine con cui compaiono i relativi grafici nelle figure:
  - *Residuals vs Fitted*. Questo grafico è utilizzato per valutare la presenza di comportamenti non lineari. Infatti, potrebbe essere presente una relazione non lineare tra delle variabili esplicative e la variabile risposta, se tale relazione non viene spiegata dal modello il pattern associato a tale relazione potrebbe comparire all'interno di questo grafico. Il risultato auspicabile è un grafico che vede i residui equamente distribuiti attorno ad una linea orizzontale.
  - *Normal Q-Q*. Tale grafico è utile per valutare se i residui presentano una distribuzione normale, in qual caso essi risulterebbero distribuiti lungo una linea retta. Se ciò non accade, potrebbe esserci la presenza di una diversa distribuzione dei residui da quella che ci si aspetta per il corretto adattamento del modello.
  - *Scale-Location*. Il grafico in questione è utile a valutare la presenza di omoschedasticità dei residui. Tale assunzione non è considerabile confutata se si nota una nuvola di punti casualmente distribuiti sul piano attorno ad una linea orizzontale.
  - *Cook's distance*. Questo grafico viene utilizzato per valutare la presenza di osservazioni con comportamenti significativamente differenti dal resto del campione, le quali tendono a influenzare le stime del modello. L'esclusione delle stesse potrebbero portare a risultati diversi nell'adattamento del modello.

Si noti che i residui di devianza, utilizzati nel primo grafico, sono ottenuti applicando la radice col segno al generico addendo della devianza residua, la quale è ottenuta tramite la seguente uguaglianza:

$$D = 2 \sum_{i=1}^{15} (y_i \cdot ln \frac{y_i}{\hat{\mu}_i} + (1 - y_i) \cdot ln \frac{1 - y_i}{1 - \hat{\mu}_i})$$

Mentre i residui di Pearson, utilizzati nel secondo e terzo grafico, si ottengono come segue:

$$r_i^P = \frac{y_i - \hat{\mu}_i}{\sqrt{\hat{\mu}_i(1-\hat{\mu}_i)}}, \qquad i = 1, \dots, 15$$

Dove $\hat{\mu}_i$ è il valore predetto dal modello della probabilità che l'i-esimo nodo sia un attaccante. Per entrambi i modelli, sembra che le assunzioni sottostanti il comportamento dei residui non siano rispettate, tuttavia, grazie al quarto grafico, si deduce la presenza di osservazioni anomale potenzialmente influenti, le quali sembrano essere le stesse responsabili dei comportamenti poco auspicabili osservabili negli altri tre grafici. Disponendo di un campione di dimensioni maggiori, si potrebbe decidere di rimuovere tali nodi dal dataset per poi adattare nuovamente i modelli sulla base delle osservazioni rimanenti, ciò non è praticabile nel caso in esame, vista la ridotta dimensione del campione.

Figura 4.4: Analisi grafica dei residui relativi al modello 1

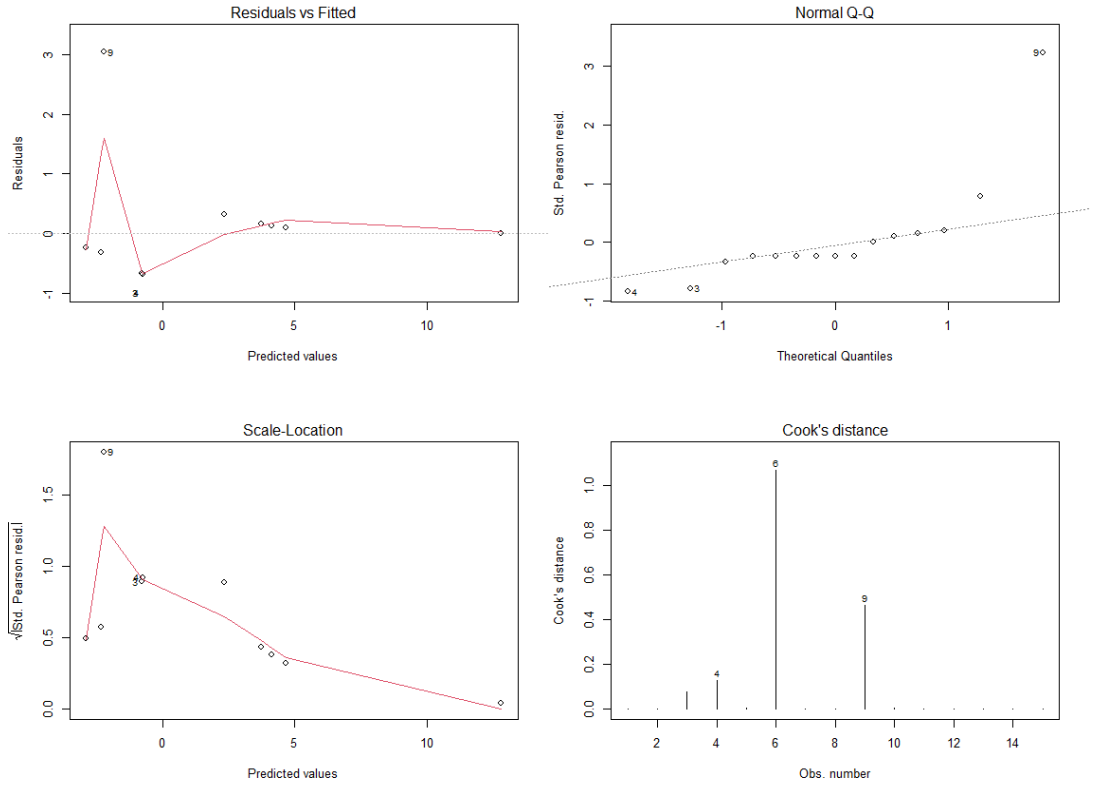
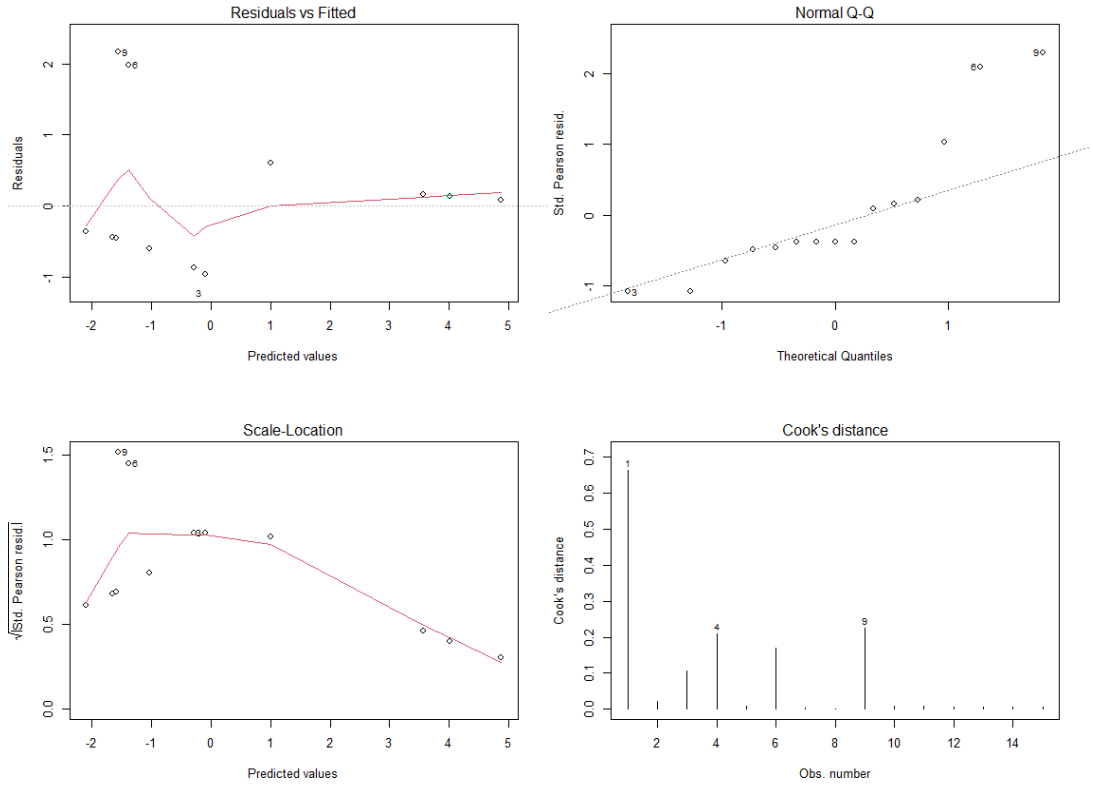


Figura 4.5: Analisi grafica dei residui relativi al modello 2



- Tabella di classificazione. A ciascun valore osservato y_i, uguale a 0 per i nodi comuni o 1 per gli attaccanti, si associa una previsione basata sul modello adattato. La previsione è così definita:

$$\hat{y}_i = \begin{cases} 0 \; se \; \hat{\mu}_i \leq \mu_0 \\ 1 \; se \; \hat{\mu}_i > \mu_0 \end{cases}$$

Dove μ_0 è un valore soglia, tipicamente scelto pari a 0.5. La definizione di $\hat{y}_i$ è tale per il fatto che il modello di regressione non produce stime dei valori della variabile risposta, ma la probabilità che i nodi assumano comportamenti offensivi. Le tabelle di classificazione calcolate rispetto ai due modelli sono riportate nelle tabelle 4.9 e 4.10.

Tabella 4.9: Tabella di classificazione associata al modello 1

Valori osservati	Valori predetti		
	Normale	Attaccante	Totale
Normale	9	0	9
Attaccante	1	5	6
Totale	10	5	15

Tabella 4.10: Tabella di classificazione associata al modello 2

Valori osservati	Valori predetti		
	Normale	Attaccante	Totale
Normale	9	0	9
Attaccante	2	4	6
Totale	11	4	15

Dall'osservazione delle tabelle di classificazione si può notare come entrambi i modelli non producano falsi positivi, ovvero classifichino correttamente i nodi normali. D'altra parte, il modello 1 classifica erroneamente un nodo offensore come normale, mentre il secondo modello commette tale errore due volte. Inoltre, è possibile ottenere una sintesi della tabella di classificazione tramite le stime delle due probabilità di classificazione corretta:

$$sensibilità = \Pr\left(\hat{Y}_i = 1 \mid Y_i = 1\right) = f_{11}/f_{1+}$$

$$specificità = \Pr\left(\hat{Y}_i = 0 \mid Y_i = 0\right) = f_{00}/f_{0+}$$

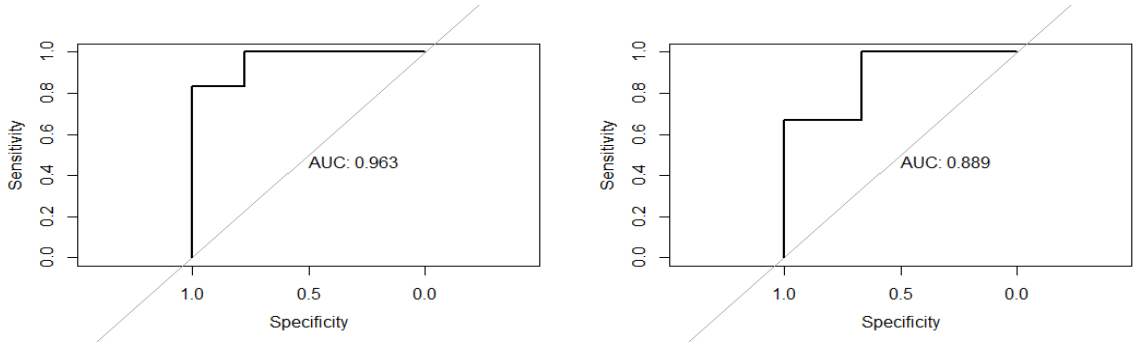
Dove f_{ii} corrisponde alla frequenza contenuta nella cella di coordinate $[i, i]$ all'interno della tabella di classificazione. Esse corrispondono, rispettivamente, alla probabilità di aver correttamente classificato i nodi malevoli e quella di aver classificato correttamente i nodi normali. I valori si sensibilità e specificità calcolati in corrispondenza dei due modelli stimati sono riportati in Tabella 4.11.

Tabella 4.11: Sensibilità e specificità calcolate rispetto ai modelli 1 e 2

	Sensibilità	Specificità
Modello 1	0.833	1
Modello 2	0.667	1

Tuttavia, la scelta di porre μ_0 pari a 0.5 è arbitraria e potrebbe non sempre essere opportuna. Dunque, una valutazione più articolata si ottiene considerando le stime di sensibilità e specificità per tutti i valori soglia $\mu_0 \in (0,1)$. Il grafico che vede la sensibilità come funzione della specificità per μ_0 che varia da 0 a 1 è detto curva ROC (Receiver Operating Charateristic). Quando μ_0 è prossimo a 1 la sensibilità è prossima a zero, mentre la specificità è prossima a 1. Per un valore fissato della specificità, la capacità predittiva del modello è tanto maggiore quanto più grande è la sensibilità. Pertanto, quanto più grande è l'area sotto la curva ROC (AUC), tanto migliore è la capacità predittiva. I grafici rappresentanti le stime delle curve ROC per i due modelli in esame sono riportati in Figura 4.6.

Figura 4.6: Curva ROC e valore AUC relativi al modello 1 (a sinistra) e al modello 2 (a destra)



In conclusione, per quanto è possibile osservare dalle stime delle tabelle di classificazione, della sensibilità, della specificità, della curva ROC e della relativa area sotto la curva, si conclude che, rispettivamente a tali misure, il modello 1 risulti preferibile rispetto all'alternativa.

Considerando, dunque, la totalità delle procedure effettuate per il confronto dei due modelli in competizione, si conclude che il modello 1 risulti essere il più promettente tra i candidati come stima della struttura generatrice dei dati, sebbene le differenze emerse tra i due modelli non siano drastiche.

4.6 Valutazione previsiva del modello

Per questa fase dell'analisi è stato fatto uso del dataset di testing. In particolare, è stata operata la rielaborazione delle variabili *syn*, *fin_rst*, *r_w_physical* e *degree* analogamente a quanto svolto nel paragrafo 4.2. Si veda l'Appendice E per osservare il codice relativo a tale rielaborazione. Una volta ottenuto il data frame risultante da tali elaborazioni, i modelli sono stati adattati ai nuovi dati con il fine di valutarne le capacità previsive. Considerando il primo modello, la Tabella 4.12 riporta, per ciascun nodo origine presente nel test set, il valore osservato della variabile *label*, che assume valore 1 se il nodo è attaccante e 0 se normale, la stima della probabilità che il nodo sia attaccante, che permette di discriminare gli utenti una volta confrontata con una soglia nominale arbitrariamente posta pari a 0.5, e lo standard error associato a tale stima.

Tabella 4.12: Valori osservati e predetti per la variabile label tramite il modello 1

Indirizzo IP del nodo	Valore osservato	Valore predetto	Standard error
172.24.1.244	1	0.052698	0.067626
172.24.1.213	0	0.122219	0.122185
172.24.1.33	1	0.052698	0.067626
172.24.1.233	1	0.052698	0.067626
172.24.1.1	1	0.157202	0.137974
172.24.1.101	0	0.052812	0.067719
172.24.1.199	0	0.052698	0.067626
172.24.1.100	0	0.052698	0.067626
0.0.0.0	0	0.052698	0.067626
255.255.255.255	0	0.052698	0.067626
169.254.211.113	0	0.052698	0.067626

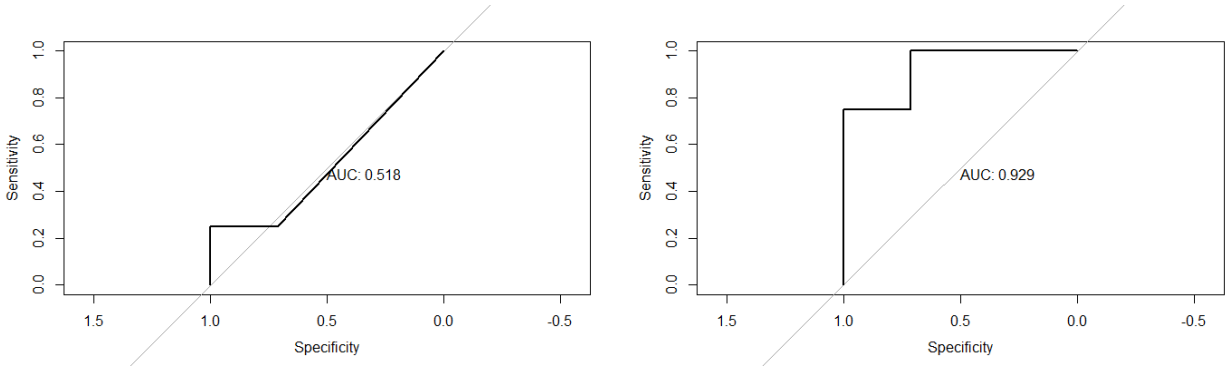
Come si può notare dall'osservazione della tabella, le prestazioni previsive del primo modello sembrano essere pessime. In particolare, seppur risulti che nessuno dei nodi normali sia stato erroneamente classificato, ciascuno dei nodi offensori è stato predetto come normale. Un'osservazione più approfondita del dataset utilizzato per operare le previsioni permette di notare la scarsa adozione delle flag SYN, FIN e RST all'interno delle comunicazioni propagate nella rete, il che può essere dovuto alla varia tipologia di dispositivi e comunicazioni presenti nella rete originale. Infatti, è importante ricordare che le flag SYN, ACK, FIN e RST sono supportare esclusivamente dal protocollo comunicativo TCP, dunque se i dispositivi appartenenti al dataset di testing tendessero ad utilizzare comunemente un diverso protocollo, tali variabili risulterebbero inefficaci per lo scopo di discriminare le tipologie di nodi. Questo sembra essere esattamente ciò che è accaduto, dal momento che nel dataset di training si sono osservate 398492 comunicazioni effettuate utilizzando il protocollo TCP, pari al 68.57% del totale delle comunicazioni avvenute tra i dispositivi assegnati al dataset di addestramento, mentre per quanto riguarda il dataset di testing il protocollo TCP è stato utilizzato in 7652 comunicazioni, ovvero il 3.88% dei casi. Dunque, sembra che l'importante frequenza di zeri rispetto alle variabili *syn* e *fin_rst* nel testing dataset abbia inficiato le previsioni della variabile risposta, dal momento che esse dipendono intrinsecamente dal protocollo utilizzato per la comunicazione. La discrepanza tra la valutazione del modello operata secondo il training set e quella svolta utilizzando il testing set dimostra come il modello 1 non sia adeguato per i fini dell'analisi, non essendo sufficientemente versatile rispetto all'eterogeneità dei dispositivi e delle comunicazioni presenti nella rete. Lo stesso non si può dire per il secondo modello stimato, i quali risultati previsivi sono riportati nella Tabella 4.13.

Tabella 4.13: Valori osservati e predetti per la variabile label tramite il modello 2

Indirizzo IP del nodo	Valore osservato	Valore predetto	Standard error
172.24.1.244	1	0.891141	0.232966
172.24.1.213	0	0.172224	0.123449
172.24.1.33	1	0.149593	0.114336
172.24.1.233	1	0.831533	0.292357
172.24.1.1	1	0.654864	0.376202
172.24.1.101	0	0.112502	0.101470
172.24.1.199	0	0.131829	0.108212
172.24.1.100	0	0.163635	0.121167
0.0.0.0	0	0.110446	0.100727
255.255.255.255	0	0.110450	0.100728
169.254.211.113	0	0.110437	0.100723

Dall'osservazione della Tabella 4.13 si può, infatti, notare come non sia stato prodotto alcun falso positivo, come per il modello 1, ma anche tre dei quattro nodi attaccanti risultano correttamente classificati. La differenza tra i risultati ottenuti dai due modelli dipende dal significato intrinseco delle variabili utilizzate per la stima degli stessi. Infatti, diversamente da quanto detto per il primo modello, le variabili utilizzate per l'adattamento del modello 2 non rappresentano una tipologia di comunicazione utilizzata, possibilmente dipendente dai tipi di dispositivi coinvolti nella comunicazione, bensì quantificano la propensione all'assunzione di un determinato comportamento. In particolare, la variabile *r_w_physical* descrive, in qualche misura, l'inclinazione di un dispositivo all'accesso e alla modifica di file, riguardanti sensori o attuatori, contenuti nella memoria del dispositivo destinatario della comunicazione, mentre *degree* descrive il numero di comunicazioni che incidono sul nodo mittente. Utilizzando, dunque, alcune delle misure descritte nel paragrafo 4.5, si osserva che entrambi i modelli presentano una specificità pari a 1, dal momento che non viene prodotto alcun falso positivo in alcuno dei modelli, mentre la sensibilità del modello 1 è pari a 0, per il fatto che nessuno degli attaccanti viene correttamente etichettato, e la stessa misura è pari a 0.75 per quanto riguarda il secondo modello. La Figura 4.7 riporta le rappresentazioni delle curve ROC e associati valori AUC dei due modelli in corrispondenza dei nuovi dai. Dall'osservazioni di tali grafici, e per quanto detto finora, si evince che il modello preferibile risulti essere il secondo.

Figura 4.7: Curva ROC e valore AUC relativi al modello 1 (a sinistra) e al modello 2 (a destra) rispetto al dataset di testing



A questo punto può risultare di interesse comprendere le debolezze del modello, in particolare studiare le motivazioni per le quali il secondo non sia stato in grado di classificare correttamente uno dei nodi attaccanti presenti nella rete. A tale proposito, l'osservazione della variabile *class2* dal dataset originale, variabile descrivente a quale fase del ciclo vitale dell'attacco l'arco corrispondente si trovi, ha permesso di valutare il livello di insediamento degli attacchi operati dai nodi attaccanti appartenenti alla rete. Tali informazioni sono reperibili dall'osservazione della Tabella 4.14.

Tabella 4.14: Tipologie di attacchi effettuati dai diversi nodi attaccanti appartenenti alla rete

Indirizzo IP	Subset	*Rec.*	*Wea.*	*Exp.*	*L.M.*	*C&C*	*Exf.*	*Tam.*	*C.R.*	*RDoS*
192.168.2.199	Train	×	×	×	×	×				
192.168.10.155	Train		×	×						
192.168.10.153	Train		×							
192.168.2.10	Train		×	×	×	×	×	×		
10.0.1.1	Train				×					
172.24.1.34	Train							×		
172.24.4.244	Test									×
172.24.1.33	Test	×								
172.24.1.233	Test									×
172.24.1.1	Test	×			×			×	×	

Dall'osservazione di tale tabella si possono trarre una serie di informazioni di importante rilevanza:

- I nodi attaccanti appartenenti al dataset di training sembrano presentare comportamenti diversi, rispetto agli offensori appartenenti all'altro dataset, per quando riguarda la tipologia e l'estensione attraverso il ciclo vitale degli attacchi operati sulla rete. Ciò può essere dovuto a diverse motivazioni, tra le quali la diversa tipologia dei dispositivi operanti gli attacchi, quella dei dispositivi

vittima o la selezione casuale degli archi sottostante la creazione del dataset originale. Questo, generalmente, non è necessariamente motivo di preoccupazioni per quanto riguarda la validità del modello, dal momento che le misure utilizzate dallo stesso, descritte dalle variabili *r_w_physical* e *degree*, sono generali e non strettamente collegate al tipo di dispositivo o di attacco.

- Derivante dal punto precedente, si può osservare che la tipologia di attacco RDoS non viene praticata dagli offensori appartenenti al dataset di training, mentre viene utilizzata nel secondo gruppo. Nonostante tale discrepanza, i nodi che operano tale tipo di attacco sono classificati correttamente dal modello, il che può essere dovuto al fatto che i comportamenti tipici di un nodo che effettua tale tipo di attacco, rispetto alle variabili utilizzate per l'adattamento del modello, sono riconducibili a quelli di un offensore che pratica altre tipologie di offesa, andando a confermare la natura generalizzata e largamente applicabile del modello.
- Il nodo identificato dall'indirizzo IP 172.24.1.33, ovvero l'unico attaccante appartenente al dataset di testing non correttamente classificato, sembra avere effettuato esclusivamente attacchi riconducibili alla fase di ricognizione. Sembra, dunque, che il suo comportamento non sia stato identificato come possibile minaccia da parte del modello per il fatto che esso non ha effettivamente praticato alcuna offesa all'interno della rete, ma si è limitato ad effettuare procedure di analisi delle risorse e identificazione delle vulnerabilità, assumendo quindi un comportamento apparentemente comune. Tale caratteristica delle comunicazioni effettuate dal nodo in esame potrebbero dunque essere il motivo per cui il modello non sia stato in grado di classificarlo correttamente, anche per il fatto che nessun nodo appartenente al dataset utilizzato per l'addestramento del modello presenta un comportamento di esclusiva ricognizione.

Capitolo 5

Conclusione

Il confronto dei due modelli ha prodotto risultati contrastanti per quanto riguarda l'adattamento ai dati di addestramento e ai dati utilizzati per la valutazione predittiva, tuttavia nel primo dei due casi le differenze tra i due modelli non sembravano particolarmente pronunciate, mentre nel secondo caso il primo modello è risultato fortemente inefficace per i fini dell'analisi, mentre il secondo è apparso decisamente promettente. Si può dunque concludere che il modello 2 risulti essere il modello preferibile per l'individuazione di utenti malevoli all'interno di una rete informatica di dispositivi IIoT. Tuttavia, il contesto attuale potrebbe non l'essere l'unico a favorire l'applicazione di tale modello. Infatti, per quanto detto riguardo la natura generale del modello, è possibile aspettarsi che esso possa essere adattato a dati proveniente da reti informatiche di altra tipologia. A sostegno di tale tesi, si noti che la varabile *degree* rappresenta il grado di un nodo, ovvero il numero di archi ad esso collegato, in altre parole essa rappresenta il numero di comunicazioni effettuate da o per tale nodo, dunque essa risulta essere rilevabile dallo studio di una qualsiasi rete informatica. Per quanto riguarda il variabile *r_w_physical*, invece, essa rappresenta il conteggio, rispetto alla totalità delle comunicazioni effettuate da un nodo, dell'avvenuto accesso da parte di esso alla memoria del dispositivo destinatario con fini di lettura e scrittura di file inerenti a sensori o attuatori. Generalizzando la tipologia di file, ci si può dunque attendere che un comportamento analogo sia rilevabile all'interno di reti informatiche denotate da una natura di dispositivi diversa da quella dei dispositivi IIoT presenti nella rete in esame, dal momento che informazioni di tale tipo possono essere ottenute interrogando i registri di accesso ai file presenti in svariate tipologie di dispositivi informatici.

Inoltre, in condizioni di maggiore numerosità campionaria e con il fine di tentare di discriminare i nodi malevoli i quali attacchi si trovano nella fase di ricognizione, si potrebbe pensare di arricchire il modello con altre variabili che si sono dimostrate essere potenzialmente utili al fine di classificare i dispositivi appartenenti ad una rete, come le variabili relative ai tempi di esecuzione di processi o di inattività del processore del dispositivo elettronico destinatario, le quali sono state rimosse dall'analisi per via della presenza di valori mancanti, come descritto nel paragrafo 3.2. Per quanto riguarda l'eterogeneità applicativa di un modello comprendente tali variabili, ci si aspetterebbe che le prestazioni del modello non variassero variando la tipologia di dispositivi appartenenti ad un'ipotetica rete, dal momento che le informazioni rappresentate da tali variabili non sembrerebbero essere dipendenti dal tipo di dispositivo, bensì sarebbe

necessario disporre di strumenti adatti alla loro rilevazione, il che sembra essere praticabile tramite la consultazione di opportuni registri contenuti nelle memorie dei dispositivi informatici. In merito a tali variabili, si noti che un approccio alternativo rispetto a quello adottato nell'analisi in questione avrebbe potuto essere quello di stimare tali dati mancanti tramite opportune procedure di interpolazione, il che non è stato fatto in questo caso vista la scarsa numerosità di variabili potenzialmente utili tra quelle presentanti valori assenti. In conclusione, si può quindi affermare che è possibile sfruttare la struttura della rete e i dati rilevati sui dispostivi in essa contenuti per valutare la presenza di utenti malevoli all'interno del sistema, con il fine di trarre informazioni da un avvenuto attacco informatico per poi precludere la possibilità che lo stesso si ripresenti. In particolare, è stato dimostrato che per tale scopo l'ammontare di informazioni necessarie alla classificazione dei nodi non risulta essere particolarmente esteso, pur considerando il fatto che il modello stimato non sembra essere in grado di catturare il comportamento assunto da un attaccante quando questo si trova nella fase iniziale della sua procedura offensiva nei confronti della rete.

Appendice A: Descrizione delle variabili rilevate dalla rete

Tabella A.1: Caratteristiche riguardanti il traffico di dati nella rete

Nome	Tipo	Descrizione
scr_ip	Qualitativa sconnessa	Indirizzo IP del dispositivo mittente
des_ip	Qualitativa sconnessa	Indirizzo IP del dispositivo destinatario
scr_port	Qualitativa sconnessa	Porta del dispositivo mittente
des_port	Qualitativa sconnessa	Porta del dispositivo destinatario
timestamp	Qualitativa sconnessa	Timestamp
date	Qualitativa sconnessa	Data
protocol	Qualitativa sconnessa	Protocollo di comunicazione
service	Qualitativa sconnessa	Protocollo applicativo della porta di destinazione
duration	Quantitativa continua	Tempo trascorso tra il primo e l'ultimo pacchetto ricevuti
src_bytes	Quantitativa discreta	Numero di byte inviati dal mittente al destinatario
des_bytes	Quantitativa discreta	Numero di byte inviati dal destinatario al mittente
missed_bytes	Quantitativa discreta	Numero di byte perduti nella comunicazione
src_pkts	Quantitativa discreta	Numero di pacchetti inviati
des_pkts	Quantitativa discreta	Numero di pacchetti ricevuti
src_ip_bytes	Quantitativa discreta	Numero di byte inviati nel campo della lunghezza totale dell'intestazione IP
des_ip_bytes	Quantitativa discreta	Numero di byte ricevuti nel campo della lunghezza totale dell'intestazione IP
conn_state	Qualitativa ordinale	Stato della connessione (1 completa, 2 stazionaria, 3 parziale)
total_bytes	Quantitativa discreta	Numero totale di byte scambiati tra mittente e destinatario
byte_rate	Quantitativa continua	Numero totale di byte scambiati al secondo
total_pkts	Quantitativa discreta	Numero totale di pacchetti scambiati tra mittente e destinatario
pkts_rate	Quantitativa continua	Numero totale di pacchetti scambiati al secondo

orig_bytes_ratio	Quantitativa continua	Percentuale di byte inviati rispetto al numero totale di byte
resp_bytes_ratio	Quantitativa continua	Percentuale di byte ricevuti rispetto al numero totale di byte
orig_pkts_ratio	Quantitativa continua	Percentuale di pacchetti inviati rispetto al numero totale di pacchetti
resp_bytes_ratio	Quantitativa continua	Percentuale di pacchetti ricevuti rispetto al numero totale di pacchetti
syn	Dicotomica	Se la connessione presenta pacchetti con flag SYN
syn_ack	Dicotomica	Se la connessione presenta pacchetti con flag SYN-ACK
pure_ack	Dicotomica	Se la connessione presenta pacchetti con flag ACK pura
payload	Dicotomica	Se la connessione presenta pacchetti con payload
fin_rst	Dicotomica	Se la connessione presenta pacchetti con flag FIN o RST
bad_checksum	Dicotomica	Se la connessione presenta pacchetti con bad checksum
syn_rst	Dicotomica	Se la connessione presenta pacchetti con entrambe le flag SYN e RST

Tabella A.2: Caratteristiche riguardanti i dispositivi

Nome	Tipo	Descrizione
avg_user_time	Quantitativa continua	Tempo medio di esecuzione di programmi o codici negli ultimi 10 secondi
std_user_time	Quantitativa continua	Deviazione standard del tempo di esecuzione di programmi o codici negli ultimi 10 secondi
avg_nice_time	Quantitativa continua	Tempo medio per la definizione della priorità di un processo negli ultimi 10 secondi
std_nice_time	Quantitativa continua	Deviazione standard del tempo per la definizione della priorità di un processo negli ultimi 10 secondi
avg_system_time	Quantitativa continua	Tempo medio di esecuzione di funzioni di sistema da parte del processore negli ultimi 10 secondi
std_system_time	Quantitativa continua	Deviazione standard del tempo di esecuzione di funzioni di sistema da parte del processore negli ultimi 10 secondi
avg_io_wait_time	Quantitativa continua	Tempo medio di attesa da parte della CPU delle operazioni I/O negli ultimi 10 secondi
std_io_wait_time	Quantitativa continua	Deviazione standard del tempo di attesa da parte della CPU delle operazioni I/O negli ultimi 10 secondi
avg_idle_time	Quantitativa continua	Tempo medio durante il quale la CPU non è occupata negli ultimi 10 secondi
std_idle_time	Quantitativa continua	Deviazione standard del tempo durante il quale la CPU non è occupata negli ultimi 10 secondi
avg_tps	Quantitativa continua	Numero medio di richieste di trasferimento al secondo negli ultimi 10 secondi
std_tps	Quantitativa continua	Deviazione standard del numero di richieste di trasferimento al secondo negli ultimi 10 secondi
avg_rtps	Quantitativa continua	Numero medio di transizioni di lettura al secondo negli ultimi 10 secondi

std_rtps	Quantitativa continua	Deviazione standard del numero di transizioni di lettura al secondo negli ultimi 10 secondi
avg_wtps	Quantitativa continua	Numero medio di transizioni di scrittura al secondo negli ultimi 10 secondi
std_wtps	Quantitativa continua	Deviazione standard del numero di transizioni di scrittura al secondo negli ultimi 10 secondi
avg_ldavg_1	Quantitativa continua	Media del carico medio del sistema durante l'ultimo minuto negli ultimi 10 secondi
std_ldavg_1	Quantitativa continua	Deviazione standard del carico medio del sistema durante l'ultimo minuto negli ultimi 10 secondi
avg_kbmemused	Quantitativa continua	Media della memoria utilizzata in kB negli ultimi 10 secondi
std_kbmemused	Quantitativa continua	Deviazione standard della memoria utilizzata in kB negli ultimi 10 secondi
avg_num_proc.s	Quantitativa continua	Numero medio di attività create al secondo negli ultimi 10 secondi
std_num_proc.s	Quantitativa continua	Deviazione standard del numero di attività create al secondo negli ultimi 10 secondi
avg_num_swch.s	Quantitativa continua	Numero medio di cambiamenti di contesto al secondo negli ultimi 10 secondi
std_num_swch.s	Quantitativa continua	Deviazione standard del numero di cambiamenti di contesto al secondo negli ultimi 10 secondi
anomaly_alert	Dicotomica	Se la connessione presenta un'allerta Zeek
ossec_alert	Dicotomica	Se la connessione presenta un'allerta OSSEC
ossec_alert_level	Quantitativa discreta	Livello di allerta OSSEC
r_w_physical	Dicotomica	Se viene praticata lettura o scrittura nel processo fisico
file_act	Dicotomica	Se sono state svolte attività su file (lettura, scrittura, cancellazione, copia, creazione, download)
proc_act	Dicotomica	Se è stato eseguito un nuovo processo

is_privileged	Dicotomica	Se l'attività svolta (login, creazione processo, attività su file) è privilegiata
login_attmp	Dicotomica	Se c'è stato un tentativo di login
succ_login	Dicotomica	Se c'è stato un login correttamente eseguito

Appendice B: Codice R per l'elaborazione delle variabili presentati valori assenti

```
#rimozione degli archi con valori assenti
IIoTID_na_omit <- na.omit(IIoTID)
#selezione delle sole variabili problematiche
IIoTID_na_omit <- IIoTID_na_omit[,c(1, 2, 3, 27:66)]
#selezione dei nodi origine del dataset
nodes_na_omit <- unique(IIoTID_na_omit$src_ip)

#variabile risposta
#inizializzazione della variabile risposta a zero
labels_na_omit <- rep(0, length(nodes_na_omit))
#creazione della matrice con nodi e etichette
response_vector_na_omit <- as.data.frame(matrix(c(nodes_na_omit, labels_na_omit),
                                                byrow = F, ncol = 2))
#impostazione dei nomi delle colonne
colnames(response_vector_na_omit) <- c("node", "label")

#matrice dei regressori
#inizializzazione a zero della matrice dei regressori
regressors_matrix_na_omit <- as.data.frame(matrix(rep(0, length(nodes_na_omit)*41),
                                                  length(nodes_na_omit), 41))
#impostazione dei nomi delle colonne
colnames(regressors_matrix_na_omit) <- c("node", colnames(IIoTID_na_omit[, 4:43]))
#inserimento dei nodi origine
regressors_matrix_na_omit$node <- nodes_na_omit

#valorizzazione del vettore della variabile risposta e delle colonne della matrice dei regressori
for(i in 1:dim(regressors_matrix_na_omit)[1]){ #per ogni nodo
    #se ha effettuato almeno un attacco
    if("Attack" %in% IIoTID_na_omit[IIoTID_na_omit$src_ip == regressors_matrix_na_omit[i, 1], 3]){
        response_vector_na_omit[i, 2] <- 1 #classificazione del nodo come attaccante
    }
    for(j in 2:41){ #per ogni variabile problematica
        #associazione al nodo i della media dei valori osservati della variabile j in
        corrispondenza degli archi di cui il nodo i è il mittente
        regressors_matrix_na_omit[i,j] <- mean(IIoTID_na_omit[IIoTID_na_omit$src_ip ==
        regressors_matrix_na_omit[i, 1], j+2])
    }
}
```

Appendice C: Codice R per l'elaborazione delle variabili associate agli archi per consentirne l'interpretazione dal punto di vista dei nodi - Train

```
#individuazione nodi origine nel train set
nodes <- unique(IIoTID_train$src_ip)
#variabile risposta
#inizializzazione a zero delle etichette
labels <- rep(0, length(nodes))
#creazione della matrice che conterrà nodi e etichette
response_vector <- as.data.frame(matrix(c(nodes, labels), byrow = F, ncol = 2))
colnames(response_vector) <- c("node", "label") #impostazione nomi colonne

#matrice delle variabili esplicative
#creazione matrice dei regressori
regressors_matrix <- as.data.frame(matrix(rep(0, length(nodes)*20), length(nodes), 20))
colnames(regressors_matrix) <- c("node", colnames(IIoTID_train[, 7:25])) #impostazione nomi colonne
regressors_matrix$node <- nodes #inserimento dei nodi nella matrice dei regressori
#valorizzazione del vettore della variabile risposta e delle colonne della matrice dei regressori
for(i in 1:dim(regressors_matrix)[1]){ #per ogni nodo
    #se ha effettuato almeno un attacco
    if("Attack" %in% IIoTID_train[IIoTID_train$src_ip == regressors_matrix[i, 1], 3]){
        response_vector[i, 2] <- 1 #classificazione del nodo come attaccante
    }
    for(j in 2:4){ #per le variabili protocol, service e conn_state
        #assegnazione alla cella (i,j) della matrice dei regressori del livello più comune osservato
        per ciascuna variabile j per ciascun nodo i
        regressors_matrix[i,j] <- names(sort(table(IIoTID_train[IIoTID_train$src_ip ==
        regressors_matrix[i, 1], j+5]),decreasing=TRUE)[1])
    }
    for(j in 5:11){ #per le variabili dicotomiche (con livelli TRUE e FALSE)
        #se il nodo presenta almeno un TRUE in corrispondenza della variabile
        if ("TRUE" %in% IIoTID_train[IIoTID_train$src_ip == regressors_matrix[i, 1], j+5]){
            #assegnazione alla cella (i,j) della matrice dei regressori della frequenza di TRUE per
            ciascuna variabile j per ciascun nodo i
            regressors_matrix[i,j] <-table(IIoTID_train[IIoTID_train$src_ip == regressors_matrix[i,
            1], j+5])["TRUE"]
        }
    }
    for(j in 12:dim(regressors_matrix)[2]){ #per le variabili dicotomice (con livelli 1 e 0)
        #se il nodo presenta almeno un 1 in corrispondenza della variabile
        if (1 %in% IIoTID_train[IIoTID_train$src_ip == regressors_matrix[i, 1], j+5]){
            #assegnazione alla cella (i,j) della matrice dei regressori della frequenza di 1 per
            ciascuna variabile j per ciascun nodo i
            regressors_matrix[i,j] <-table(IIoTID_train[IIoTID_train$src_ip == regressors_matrix[i,
            1], j+5])["1"]
        }
    }
    #per la variabile ossec_alert, assegnazione alla cella (i,j) della matrice dei regressori del
    livello più grave      osservato per ciascun nodo i
    regressors_matrix[i,20] <- max(IIoTID_train[IIoTID_train$src_ip == regressors_matrix[i, 1], 25])
}
```

Appendice D: Codice R per la rappresentazione grafica delle curve di livello e della forma 3D della verosimiglianza normalizzata e della relativa approssimazione parabolica

```
#modello 2: variabili: r_w_physical, deg
mod2 <- glm(label ~ r_w_physical + deg, data = model_matrix, family = binomial)
#rappresentazione grafica
b <- summary(mod2)$coef #ottenimento coefficienti del modello
#stima dell'intercetta
b0hat <- b[1, 1]
#stima e standard error di beta1
b1hat <- b[2, 1]; se1 <- b[2, 2]
#stima e standard error di beta2
b2hat <- b[3, 1]; se2 <- b[3, 2]

#calcolo della log-verosmiglianza
y <- model_matrix$label #valori osservati variabile risposta
x1 <- model_matrix$r_w_physical #valori osservati variabile esplicativa 1, r_w_physical
x2 <- model_matrix$deg #valori osservati variabile esplicativa 2, degree
#funzione per il calcolo della log-verosimigianza
loglik_fun <- function(b1, b2){
sum(y*(b0hat + b1*x1 + b2*x2) - log(1 + exp(b0hat + b1*x1 + b2*x2)))
}
#inizializzazione a zero del vettore contenente i valori della verosimiglianza
lik<- NULL
#range valori +-3 * se per beta1
bb1<- seq(b1hat - 3*se1, b1hat + 3*se1 , len = 50)
#range valori +-3 * se per beta2
bb2 <- seq(b2hat - 3*se2, b2hat + 3*se2, len = 50)
for (b1 in bb1){ #per tutti i valori di beta1
    for (b2 in bb2){ #per tutti i valori di beta2
        #calcolo della log-verosimiglianza in prossimità di tali valori
        lik <- c(lik,loglik_fun(b1, b2))
    }
}
#massima log-verosimiglianza
maxlik <- max(lik)
#normalizzazione della log-verosimiglianza
lik <- lik-maxlik
#ottenimento della verosimiglianza tramite funzione inversa
lik <- exp(lik)
#rappresentazione grafica delle curve di livello della verosimiglianza normalizzata
contour(bb1, bb2, matrix(lik, 50, byrow = T), level = seq(0, 1 , by = 1/10), xlab = NA, ylab = NA)
#rappresentazione grafica 3D della verosimiglianza normalizzata
persp(bb1, bb2, matrix(lik, 50, byrow = T), theta = 0, phi = 30, col = "azure3", shade = 0.2,
axes = F)
```

```
#confronto con approssimazione parabolica della verosimiglianza normalizzata
#funzione per il calcolo dell'approssimazione parabolica della log-verosimiglianza normalizzata
parab_approx <- function(b1, b2){
    -(1/2)*t(c(b0hat, b1, b2)-c(b0hat, b1hat, b2hat)) %*% solve(vcov(mod2)) %*% (c(b0hat, b1, b2)-
    c(b0hat, b1hat, b2hat))
}
#inizializzazione a zero del vettore che conterrà i valori dell'approssimazione parabolica della
verosimiglianza normalizzata
lik_app <- NULL
for (b1 in bb1){ #per ogni valore di beta1
    for (b2 in bb2){ #per ogni valore di beta 2
        #calcolo dell'approssimazione della log-verosimiglianza normalizzata in corrispondenza di
        tali valori
        lik_app <- c(lik_app,parab_approx(b1, b2))
    }
}
# ottenimento della verosimiglianza approssimata tramite funzione inversa
lik_app <- exp(lik_app)
#rappresentazione grafica delle curve di livello dell'approssimazione parabolica della
verosimiglianza normalizzata
contour(bb1, bb2, matrix(lik_app, 50, byrow = T), level = seq(0 , 1, by = 1/10), xlab = NA, ylab =
NA)
#rappresentazione grafica 3d dell'approssimazione parabolica della verosimiglianza normalizzata
persp(bb1, bb2, matrix(lik_app, 50, byrow = T), theta = 0, phi = 30, col = "azure3", shade = 0.2,
axes = F)
```

Appendice E: Codice R per l'elaborazione delle variabili associate agli archi per consentirne l'interpretazione dal punto di vista dei nodi - Test

```
#individuazione dei nodi origine appartenenti al test set
nodes_test <- unique(IIoTID_test$src_ip)
#variabile risposta
labels_test <- rep(0, length(nodes_test)) #inizializzazione a zero del vettore delle etichette
response_vector_test <- as.data.frame(matrix(c(nodes_test,labels_test), byrow = F, ncol = 2))
#creazione matrice di nodi e etichette
colnames(response_vector_test) <- c("node", "label") #impostazione nomi colonne


#variabili esplicative
#creazione della matrice dei regressori
regressors_matrix_test <- as.data.frame(matrix(rep(0, length(nodes_test)*5), length(nodes_test),
5))
#impostazione nomi colonne
colnames(regressors_matrix_test) <- c("node", "syn", "fin_rst", "r_w_physical", "deg")
regressors_matrix_test$node <- nodes_test #inserimento dei nodi nel data frame
#valorizzazione del vettore della risposta e della matrice delle variabili esplicative
for(i in 1:dim(regressors_matrix_test)[1]){ #per ogni nodo
   #se ha effettuato almeno un attacco
   if("Attack" %in% IIoTID_test[IIoTID_test$src_ip == regressors_matrix_test[i, 1], 3]){
      response_vector_test[i, 2] <- 1 #impostazione della relativa etichetta a 1
   }
   #se il nodo presenta almeno un TRUE
   if ("TRUE" %in% IIoTID_test[IIoTID_test$src_ip == regressors_matrix_test[i, 1], "syn"]){
      #assegnazione del numero di volte in cui la flag syn è pari a 1 nel totale delle comunicazioni
      effettuate da tale nodo
      regressors_matrix_test[i, 2] <- table(IIoTID_test[IIoTID_test$src_ip ==
      regressors_matrix_test[i, 1], "syn"])["TRUE"]
   }
   #se il nodo presenta almeno un TRUE
   if ("TRUE" %in% IIoTID_test[IIoTID_test$src_ip == regressors_matrix_test[i, 1], "fin_rst"]){
      #assegnazione del numero di volte in cui la flag fin_rst è pari a 1 nel totale delle
      comunicazioni effettuate da tale nodo
      regressors_matrix_test[i,3] <- table(IIoTID_test[IIoTID_test$src_ip ==
      regressors_matrix_test[i, 1], "fin_rst"])["TRUE"]
   }
   #se il nodo presenta almeno un TRUE
   if (1 %in% IIoTID_test[IIoTID_test$src_ip == regressors_matrix_test[i, 1], "r_w_physical"]){
      #assegnazione del numero di volte in cui il nodo ha accesso ai dati nel totale delle
      comunicazioni effettuate da tale nodo
      regressors_matrix_test[i,4] <- table(IIoTID_test[IIoTID_test$src_ip ==
      regressors_matrix_test[i, 1], "r_w_physical"])["1"]
   }
}
```

Bibliografia

Salvan, A., Sartori, N. & Pace, L. (2020). *Modelli Lineari Generalizzati*. Springer-Italia, Milano.

Pace, L., Salvan, A. (2001). *Introduzione alla Statistica - II. Inferenza, Verosimiglianza, Modelli*. Cedam, Padova.

Al-Hawawreh M., Sitnikova E., Aboutorab N. (2021). *X-IIoTID: A Connectivity-and Device-agnostic Intrusion Dataset for Industrial Internet of Things*, IEEE Internet of Things Journal, DOI:10.1109/JIOT.2021.3102056.

Sitografia

L'Italia è quarta al mondo per numero di attacchi informatici a tema Covid-19, Tom's Hardware.

https://www.tomshw.it/hardware/litalia-e-quarta-al-mondo-per-numero-di-attacchi-informatici-a-tema-covid-19/

Cos'è un attacco informatico e quali le diverse tipologie, UniverseIT.

https://universeit.blog/attacchi-informatici/#:~:text=Gli%20attacchi%20informatici%20sono%20azioni,e%20servizi%20digitali%20online%2C%20ecc.

Grafo, Treccani.

https://www.treccani.it/enciclopedia/grafo_%28Enciclopedia-della-Matematica%29/

What is IoT vs IIoT?, Copadata.

https://www.copadata.com/en/product/platform-editorial-content/what-is-the-iot-and-iiot/

What is IoT?, Oracle.

https://www.oracle.com/in/internet-of-things/what-is-iot/

Edge Gateway, IoTONE.

https://www.iotone.com/term/edge-gateway/t754

Dictionary Attack, HYPR.

https://www.hypr.com/dictionary-attack/#:~:text=A%20Dictionary%20Attack%20is%20a,of%20terms%20or%20other%20values.

Backdoor computing attacks, Malwarebytes.

https://www.malwarebytes.com/backdoor

What is the IoT? Everything you need to know about the Internet of Things right now, ZDNet.

https://www.zdnet.com/article/what-is-the-internet-of-things-everything-you-need-to-know-about-the-iot-right-now/

Cyber attack, TechTarget.

https://www.techtarget.com/searchsecurity/definition/cyber-attack

What is Computer Networking?, Amazon Web Services.

https://aws.amazon.com/what-is/computer-networking/#:~:text=Computer%20networking%20refers%20to%20interconnected,over%20physical%20or%20wireless%20technologies.

Kruskal-Wallis Test in R, STHDA.

http://www.sthda.com/english/wiki/kruskal-wallis-test-in-r#import-your-data-into-r

Kruskal-Wallis Test: Definition, Formula, and Example, Statology.

https://www.statology.org/kruskal-wallis-test/

Network Analysis and Visualization with R and igraph, Kateto.

https://kateto.net/netscix2016.html

Community Detection, ScienceDirect.

https://www.sciencedirect.com/topics/computer-science/community-detection#:~:text=The%20concept%20of%20community%20detection,through%20represented%20on%20a%20graph.

Louvain method, Wikipedia.

https://en.wikipedia.org/wiki/Louvain_method

Transitivity, Università degli Studi G. d'Annunzio Chieti.

https://www.google.com/search?q=unich&rlz=1C1ONGR_itIT945IT945&oq=unich&aqs=chrome..69i57j69i60.972j0j1&sourceid=chrome&ie=UTF-8&tbs=cdr:1,cd_min:1/1/0

Reciprocity, NetworkX.

https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.reciprocity.reciprocity.html#:~:text=Compute%20the%20reciprocity%20in%20a,%2C%20v%20)%20%E2%88%88%20G%20%7C%20.

Centrality and Centralization, Analytic Tech.

http://www.analytictech.com/mb119/chapter5.htm

Train-Test Split for Evaluating Machine Learning Algorithms, Machine Learning Mastery.

https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/#:~:text=The%20reason%20is%20that%20when,effectively%20evaluate%20the%20model%20performance.

Wald vs likelihood ratio test, The Stats Geek.

http://thestatsgeek.com/2014/02/08/wald-vs-likelihood-ratio-test/

What is an IP Address – Definition and Explanation, Kaspersky.

https://www.kaspersky.com/resource-center/definitions/what-is-an-ip-address

TCP flags, GeeksforGeeks.

https://www.geeksforgeeks.org/tcp-flags/

Ringraziamenti

Terrei particolarmente a ringraziare le persone che mi hanno accompagnato durante questo percorso universitario, consentendomi di arrivare alla produzione di questa tesi.

Vorrei ringraziare il mio relatore, Antonio Canale, per la sua immensa pazienza, per i suoi indispensabili consigli, per le conoscenze trasmesse durante tutto il percorso di stesura di questo elaborato.

Infinite grazie alla mia famiglia, Lorella, Antonello, Alberto e Mattia, che da sempre mi sostiene nel raggiungimento dei miei obiettivi. In particolare, voglio ringraziare i miei genitori, per avermi fatto da guida in questa avventura, insegnandomi l'importanza dell'impegno e della perseveranza, e i miei fratelli, per avermi accompagnato, proponendo soluzione ad ogni problema. A voi, la mia famiglia, sarò per sempre grato per avermi permesso di arrivare a questo traguardo.

Non potrò mai ringraziare abbastanza la mia fidanzata, Sara, per essere sempre stata pronta a darmi una spalla su cui piangere, un orecchio che sappia ascoltare, una voce che sappia guarire, di cui ho avuto bisogno un inquantificabile numero di volte durante questo percorso.

Non posso non ringraziare i miei amici più cari, Enrico e Luca, i quali sono stati vittima delle mie continue assenze dovute alla produzione di questa tesi e che sono sempre stati presenti per ascoltarmi e per interessarsi circa l'andamento del mio percorso di studi.

Infine, dedico questa tesi a me stesso, perché con probabilità pari a 1 sarò per sempre l'unico veramente cosciente di quante sfide questa esperienza mi abbia proposto e quanto queste mi abbiano cambiato.