POLITECNICO DI TORINO

Laurea Magistrale in Ingegneria Informatica

Tesi di Laurea Magistrale

Sviluppo e applicazione di un algoritmo di Machine Learning per l'identificazione di e-mail di phishing

Relatore:
Prof. Cataldo Basile

Candidato:
Vito Christian Carulli

Anno Accademico 2023/2024
Torino

Abstract

Il fenomeno delle e-mail di phishing rappresenta una delle principali minacce alla sicurezza informatica, mirata a ingannare gli utenti al fine di ottenere informazioni personali, come credenziali di accesso, informazioni finanziarie o altri dati sensibili. Spesso le e-mail di phishing contengono caratteristiche comuni come errori grammaticali, grafica di bassa qualità e richieste urgenti di fornire informazioni personali o cliccare su link. Un algoritmo di machine learning, sfruttando questi schemi ricorrenti, grazie alla sua capacità di apprendere da grandi quantità di dati, offre soluzioni efficaci per l'identificazione e la mitigazione di queste minacce.

L'obiettivo di questa tesi, svolta presso Spike Reply, è applicare tali algoritmi per individuare potenziali e-mail malevoli. Lo studio prende in analisi l'intera e-mail per garantire la robustezza del processo contro tutte le variazioni possibili in un contesto reale. Per ogni e-mail vengono considerati header, testo e allegati che subiscono un processo di preprocessing volto a semplificare e uniformare i dati da utilizzare per l'addestramento dell'algoritmo. Al termine dell'esecuzione vengono riportate predizioni sugli input forniti con relative metriche di valutazione che specificano quanto siano affidabili.

L'approccio seguito si è dimostrato estremamente efficace nell'identificare le e-mail di phishing, raggiungendo un buon livello di accuratezza. Un aspetto particolarmente rilevante è stato il limitato numero di falsi positivi, che è un elemento cruciale per garantire l'affidabilità del sistema in contesti reali poiché consente di evitare che e-mail legittime vengano erroneamente segnalate come pericolose.

Indice

Elenco delle figure

Capitolo 1

INTRODUZIONE

Negli ultimi anni, l'utilizzo crescente di Internet e delle comunicazioni digitali ha comportato un aumento esponenziale del traffico di posta elettronica, rendendo l'email uno degli strumenti di comunicazione più diffuso. A causa di tale diffusione e per la facilità di raggiungere in modo capillare gli utenti, la posta elettronica è divenuta un veicolo per possibili attacchi. Uno di questi è il phishing.

Le tecniche di machine learning hanno dimostrato di poter essere un mezzo per automatizzare l'identificazione di email malevoli, permettendo di analizzare grandi volumi di dati.

L'obiettivo di questa tesi, svolta presso la sede di Spike Reply di Torino, è la progettazione e lo sviluppo di un algoritmo di classificazione che sfrutta tecniche di Machine Learning per rilevare le email di phishing.

In particolare, il fine dello studio è quello di costruire un modello che integri informazioni provenienti da header, testo e allegati delle email, combinando dei modelli di Decision Tree allenati separatamente su ciascuna componente, per poi unire le previsioni in un classificatore finale che effettui la classificazione definitiva.

A tal fine, è stato costruito un dataset bilanciato contenente 2131 email appartenenti a entrambe le classi.

Le email, una volta scomposte utilizzando script Python, sono state analizzate per estrarre le features di ogni componente. I testi sono stati, in seguito, ulteriormente processati in RapidMiner per preparare i dati per la fase di addestramento.

RapidMiner è una piattaforma che consente di progettare flussi di lavoro di Machine Learning e data mining servendosi di un'interfaccia grafica intuitiva, offrendo strumenti per la preparazione dei dati, il preprocessing e la visualizzazione dei risultati.

Le performance del modello sono state valutate in seguito a Cross Validation, utilizzando metriche come accuratezza, precisione, recall e F1-Score.

L'approccio proposto si differenzia dalle tradizionali tecniche focalizzate su un'unica componente, sfruttando l'integrazione di features estratte da più parti dell'email per migliorare la precisione complessiva del modello, minimizzando il numero di falsi positivi e di falsi negativi.

Inoltre, l'utilizzo di RapidMiner consente una visualizzazione chiara del flusso di lavoro, permettendo l'identificazione di eventuali criticità e intervenire per migliorare il processo.

Capitolo 2

PHISHING

Il termine "phishing" è stato utilizzato per la prima volta nei primi anni '90. Secondo il CERT dell'Agenzia per l'Italia Digitale, "Il phishing è una frode informatica, realizzata attraverso l'invio di e-mail contraffatte, finalizzata all'acquisizione, per scopi illegali, di dati riservati oppure a far compiere alla vittima determinate operazioni o azioni" [1].

Uno dei primi esempi documentati coinvolgeva gli utenti di America Online (AOL): gli attaccanti utilizzavano software come AOHell per inviare agli utenti messaggi ben realizzati, ma ingannevoli, chiedendo loro di rivelare le proprie credenziali di accesso. Questi attacchi, sebbene rudimentali, come spiegato dallo stesso creatore del software Koceilah Rekouche, hanno posto le basi per le tecniche di phishing più sofisticate in uso oggi [2].

Con l'avvento delle e-mail negli anni '90, gli attacchi di phishing sono diventati più frequenti. Gli attaccanti inviavano e-mail che sembravano provenire da enti affidabili, come banche o istituzioni finanziarie, chiedendo agli utenti di confermare i propri dati personali. Frutto di questo periodo sono gli **spear phishing**: attacchi più mirati che utilizzavano informazioni personali per essere più credibili.

La crescita del crimine informatico organizzato ha ulteriormente amplificato la portata e la complessità degli attacchi di phishing. Le reti criminali si sono servite di strumenti avanzati per automatizzare e amplificare tali attacchi.

Secondo i report pubblicati da APWG, si è passati da circa 300.000 attacchi di phishing unici nel 2010 a quasi 1,5 milioni nel 2015 [3].

Negli ultimi anni, il phishing è diventato una minaccia globale, con attacchi che colpiscono vari settori. La pandemia di COVID-19 ha portato ad un aumento degli attacchi di phishing, con i criminali che sfruttano la crisi sanitaria per lanciare campagne mirate.

La pandemia, infatti, ha aumentato la tendenza del lavoro da remoto, rendendo

maggiormente vulnerabili gli utenti che non godono degli stessi sistemi di sicurezza e filtraggio delle e-mail delle reti aziendali.

L'Interpol, tra gennaio e aprile 2020, ha rilevato 907.000 e-mail spam e 48.000 URLs malevoli legati al COVID-19 [4]. Secondo lo studio "Phishing Insides 2021", il 90% degli intervistati in Israele ha segnalato un aumento degli attacchi di phishing, in Austria l'88% e in UK il 74% [5]. Tale tendenza è confermata dal "Cyber Security Breaches Survey 2021", che evidenzia un aumento degli eventi di phishing dal 72% all'83% tra il 2017 e il 2021 [6].

2.1 Tecniche di Phishing

Gli attacchi di phishing, con l'avvento degli smartphone e dei social media, sono diventati più sofisticati e utilizzano tecniche diverse.

Gli attacchi di **spear phishing** sono indirizzati contro dipendenti di aziende per ottenere accesso a dati sensibili o per installare malware. Un messaggio di questo tipo può sembrare legittimamente inviato dal datore di lavoro o un collega e indirizzato a impiegati, membri di una specifica organizzazione o gruppi di lavoro. La particolarità di questa tecnica è l'impiego di informazioni personali della vittima per rendere l'attacco più credibile. Infatti, se le tradizionali campagne di phishing sono volte a rubare informazioni degli individui, l'obiettivo dello spear phishing è quello di ottenere accesso ai sistemi informatici di un'azienda [7]. È la tecnica più diffusa su internet, che corrisponde al 91% degli attacchi [8].

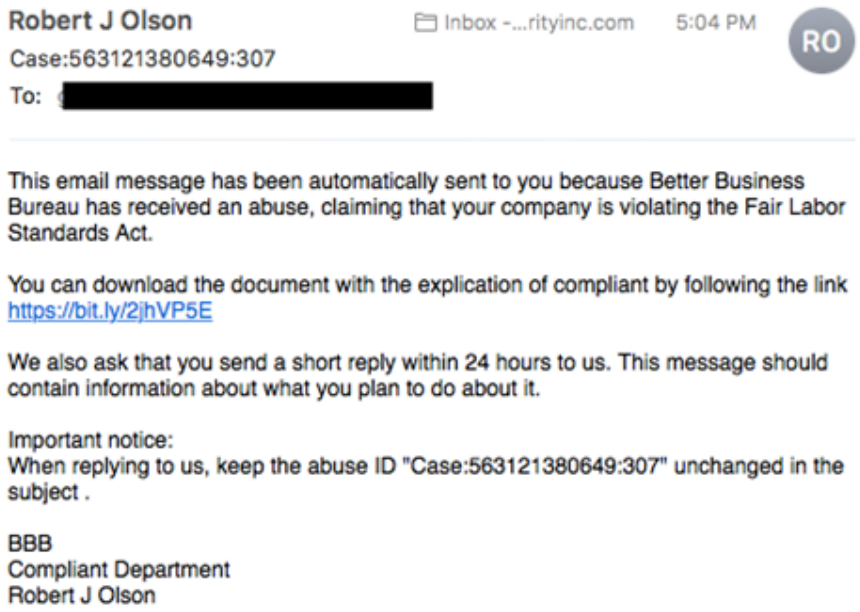
In figura 2.1 un esempio di spear phishing e-mail dove si può notare la richiesta urgente di risposta, l'attenzione all'uso di informazioni specifiche e la presenza di un link malevolo al quale la vittima è tenuta ad accedere per un download [9].

Se la vittima dell'attacco è una figura dirigenziale come un manager o un CEO, si parla di **whaling**. In questo caso le e-mail sono altamente personalizzate grazie a un precedente lavoro di raccolta di informazioni e solitamente hanno come tema problemi legali, amministrativi o lamentele di clienti, con lo scopo di ottenere credenziali critiche [9].

Il **clone phishing** corrisponde all'impiego di una e-mail legittima contenente un link o un allegato intercettata dall'attaccante. Quest'ultimo ne crea una copia esatta, modificandola in modo da renderla malevole. Ad esempio, il link potrebbe portare ad un sito fraudolento, anziché ad uno autentico. L'e-mail aggiornata, poi, viene rimandata al ricevente.

Si fa riferimento al **malware phishing**, se l'e-mail è utilizzata come mezzo di diffusione di malware attraverso gli allegati [10].

Se vengono usati mezzi alternativi come SMS o VoIP, l'attacco prende il nome di **Smishing** o **Vishing** [10].


Robert J Olson
Inbox -...rityinc.com 5:04 PM
RO
Case:563121380649:307
To: [redacted]

This email message has been automatically sent to you because Better Business Bureau has received an abuse, claiming that your company is violating the Fair Labor Standards Act.

You can download the document with the explication of compliant by following the link https://bit.ly/2jhVP5E

We also ask that you send a short reply within 24 hours to us. This message should contain information about what you plan to do about it.

Important notice:
When replying to us, keep the abuse ID "Case:563121380649:307" unchanged in the subject .

BBB
Compliant Department
Robert J Olson

Figura 2.1: Esempio di spear phishing [8]

2.2 Tecniche Utilizzate

Un attacco di phishing ingloba diverse tecniche volte a ingannare le vittime sfruttando la psicologia o le vulnerabilità dei sistemi di sicurezza e di comunicazione.

L'**ingegneria sociale** ha un ruolo importante nella riuscita dell'attacco. Le vittime, infatti, sono manipolate per ottenere informazioni utili alla riuscita dell'attacco e persuase a compiere azioni pericolose facendo leva sulla paura o sull'urgenza.

Attraverso l'**e-mail spoofing** l'attaccante può falsificare l'indirizzo e-mail del mittente utilizzandone uno simile a quello legittimo o manipolare i campi dell'header, sfruttando il Simple Mail Transfer Protocol (SMTP) per rendere l'email credibile [11].

Tecnica simile è applicabile ai siti web. Il **website spoofing**, infatti, prevede la creazione di copie di pagine di siti web legittimi, come quelle relative al login, includendo loghi, design e contenuti. Le vittime, attirate su queste pagine, sono invitate a inserire dati sensibili che possono dunque essere rubate dagli attaccanti [12].

La **manipolazione dei link** permette di mascherare collegamenti a siti malevoli con dei link familiari alla vittima. Inoltre, è possibile usare il DNS per reindirizzare il traffico da un sito legittimo verso uno fraudolento [13].

2.3 Fasi di un Attacco di Phishing

La Figura 2.2 offre una panoramica dettagliata delle fasi di un attacco di phishing tipico. Il processo inizia quando gli attaccanti ottengono l'infrastruttura necessaria, che può includere server, domini e strumenti software. Questa fase iniziale prevede l'acquisizione di risorse tecniche che consentono di ospitare un sito web di phishing, successivamente configurato su questa infrastruttura con l'aiuto di kit di phishing. Questi kit sono facilmente reperibili sul dark web e possono essereutilizzati anche da chi ha competenze tecniche limitate. Corrispondono a pacchetti preconfigurati di software e risorse, progettati per rendere più facile e veloce la creazione di un sito di phishing convincente e, di conseguenza, più efficiente. Questi kit sono facilmente reperibili sul dark web e possono essere utilizzati anche da chi ha competenze tecniche limitate.

Una volta che il sito web è operativo, gli attaccanti avviano la fase di distribuzione, inviando i link alle potenziali vittime tramite e-mail di phishing, messaggi sui social media, SMS, o persino annunci pubblicitari ingannevoli. Le vittime, inconsapevoli del pericolo e ingannate dall'aspetto autentico del sito, iniziano ad accedervi e vengono indotte a inserire informazioni sensibili come credenziali di accesso, numeri di carte di credito e altri dati personali.

A questo punto, le fasi successive possono non seguire un ordine preciso. Una volta che l'infrastruttura anti-phishing rileva l'attacco, vengono messe in atto una serie di misure di mitigazione che possono includere avvisi di phishing integrati nei browser, blocchi a livello di rete da parte dei provider di servizi internet e segnalazioni ai servizi di sicurezza informatica. Idealmente, questa mitigazione dovrebbe avvenire prima che le vittime inizino ad accedere al sito fraudolento, prevenendo, così, ulteriori danni. Tuttavia, se la mitigazione non avviene tempestivamente, le vittime possono continuare a visitare il sito di phishing per un periodo prolungato, permettendo agli attaccanti di raccogliere una quantità significativa di dati sensibili.

Gli attaccanti, una volta raccolti i dati, li utilizzano in vari modi per monetizzare. Questo può includere il test delle credenziali rubate sulle piattaforme corrispondenti per ottenere accesso non autorizzato, o l'uso dei dati finanziari rubati per effettuare transazioni fraudolente. In alcuni casi, i dati raccolti possono essere venduti nel dark web ad altri criminali informatici.

In seguito agli sforzi da parte delle autorità competenti o delle aziende di sicurezza informatica, il sito di phishing potrebbe essere disattivato. La rimozione potrebbe essere condotta anche deliberatamente dagli attaccanti stessi, per evitare di essere scoperti.

In conclusione, un attacco di phishing è un processo complesso e ben orchestrato, che richiede una combinazione di tecnologie sofisticate e tecniche di ingegneria sociale per avere successo. La chiave per difendersi da questi attacchi risiede nell'educazione degli utenti, nella tempestiva rilevazione degli attacchi e nell'implementazione di difese robuste a livello tecnologico [14].

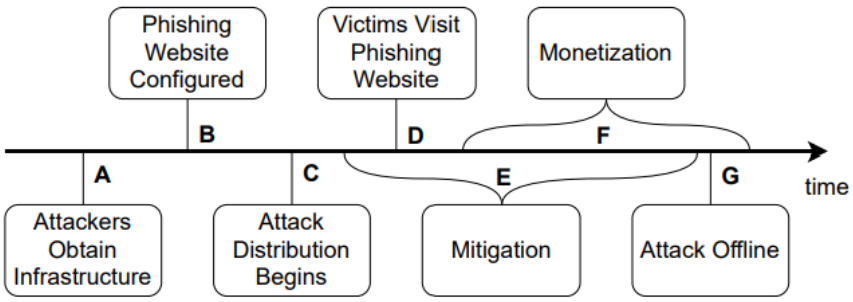


Figura 2.2: Fasi di un attacco di phishing [14]

2.4 Casi di Phishing Rilevanti e il Loro Impatto

Per evidenziare l'impatto significativo di tale fenomeno su aziende e individui e sottolineare l'importanza della sicurezza informatica e del costante monitoring, si riportano alcuni esempi di casi rilevanti di phishing.

Nel novembre del 2014 la **Sony Pictures** ha subito un attacco di phishing attraverso delle email fraudolente finalizzate ad ottenere l'accesso alle credenziali di diversi dipendenti. Sono state rubate, oltre alle informazioni personali dei dipendenti e delle loro famiglie, anche copie di film non ancora rilasciati, email, futuri piani aziendali e altri dati. Successivamente, nel 2015, oltre 30.000 documenti e oltre 170.000 indirizzi mail sono stati ripubblicati da WikiLeaks [15]. I danni stimati, oltre quelli a lungo termine, ammontano a 15 milioni di dollari nell'immediato [16].

Nello stesso anno, **eBay** è stata vittima di cybercriminali. Attraverso email di phishing rivolte agli utenti, è stato ottenuto accesso a passwords, nomi, indirizzi email, indirizzi fisici, numeri di telefono, date di nascita e altre informazioni personali, ma il database attaccato non conteneva informazioni finanziarie. Nonostante gli utenti siano stati prontamente invitati a reimpostare le passwords, il risultato è stato un'enorme perdita di fiducia da parte degli utenti eBay, con conseguente diminuzione di traffico sulla piattaforma e la necessità per l'azienda di implementare misure di sicurezza più rigorose per proteggere i dati degli utenti [17].

Tra il 2013 e il 2014, sono stati compromessi oltre 1 miliardo di account di **Yahoo** a causa di due data breaches effettuati attraverso tecniche di phishing. A questi attacchi hanno fatto seguito numerose cause legali e il pagamento di 117 milioni di dollari, contribuendo al declino della reputazione dell'azienda e del relativo valore di mercato che ha portato alla successiva vendita a Verizon nel 2017 [18].

Nel 2013 sono stati rubati dati relativi a 40 milioni di carte di credito e informazioni personali di 70 milioni di clienti per mezzo di phishing, infettando 40.000 POS, ai danni

di Target. L'impatto per l'azienda è stato devastante, causando una perdita finanziaria notevole, la riduzione del valore delle azioni e una spesa di 162 milioni di dollari.

2.5 Il Machine Learning per Identificazione di Email di Phishing

Le tecniche tradizionali di rilevamento e prevenzione del phishing, pur essendo efficaci, hanno dei limiti, specialmente di fronte a minacce in continua evoluzione. Il machine learning, con la sua capacità di analizzare grandi quantità di dati e identificare pattern nascosti, offre un approccio più robusto e adattivo.

Un esempio di approccio di questo tipo è presentato nel lavoro dal titolo "A Machine Learning Approach towards Phishing Email Detection.". Gli autori hanno studiato un sistema di classificazione che confronta diversi algoritmi, tra cui Decision Tree, Random Forest, Support Vector Machine e Naive Bayes. Il loro processo si articola in diverse fasi, a partire dalla raccolta dati, fino all'addestramento e alla valutazione dei modelli. Il Random Forest e il Decision Tree hanno mostrato le prestazioni migliori e vengono evidenziati, inoltre, l'importanza di una buona selezione delle features e un dataset bilanciato per ottenere una buona efficacia del modello [19].

Lo studio "A Comparison of Natural Language Processing and Machine Learning Methods for Phishing Email Detection" di P.Bountakas et al. confronta l'efficacia di diverse tecniche basate su algoritmi di Natural Language Processing (NLP) e Machine Learning (ML) per il rilevamento delle email di phishing attraverso l'analisi del testo. I risultati mostrano che l'uso di modelli ibridi che combinano NLP e ML offre buone prestazioni. In particolare, una buona fase di preprocessing del testo e selezione delle features risultano rilevanti per ottenere una buona accuratezza del modello [20].

L'articolo intitolato "Improving malicious email detection through novel designated deep-learning architectures utilizing entire email" di Trivikram Muralidharan e Nir Nissim propone soluzioni di deep learning che analizzano l'intero contenuto delle email, sostenendo che l'analisi completa del testo e delle strutture delle email può migliorare significativamente l'accuratezza della classificazione rispetto ai metodi tradizionali. L'architettura propone una prima fase di preprocessing di ogni sezione dell'email, seguita da una fase di rappresentazione delle features. Ciò prevede la trasformazione dell'header e degli allegati in immagini RGB e tokenizzazione attraverso BERT per l'header e testo. Infine, le classificazioni parallele vengono usate per la classificazione conclusiva. I risultati mostrano che l'analisi dell'intero contenuto delle email consente di ottenere prestazioni migliori rispetto ai metodi tradizionali [21].

Capitolo 3

MACHINE LEARNING

L'apprendimento automatico, o machine learning, è una branca dell'intelligenza artificiale (IA) che si concentra sulla costruzione di sistemi capaci di apprendere da dati e migliorare le proprie performance nel tempo senza essere esplicitamente programmati per ogni specifica azione.

Un algoritmo di apprendimento automatico è definito "soft coded" in quanto l'insieme di istruzioni matematiche e statistiche utilizzate al suo interno, gli permettono di adattarsi e migliorare basandosi sui dati che riceve. Nella fase di "training", infatti, l'algoritmo si autoconfigura in maniera ottimale in modo da poter generare il risultato atteso, non solo dall'input usato in questa fase, ma anche da nuovi dati [22].

Esistono diverse tipologie di apprendimento automatico:

- **Apprendimento supervisionato**: con questo approccio si crea un meccanismo che produce generalizzazioni utilizzando un insieme di dati di training già etichettati in modo che l'algoritmo apprenda a mappare gli input agli output corretti e a fare previsioni su nuovi dati.
- **Apprendimento non supervisionato**: vengono analizzati dati non etichettati per estrarne pattern e schemi ricorrenti senza avere informazioni sui risultati desiderati. È spesso utilizzato per il clustering, il cui fine è raggruppare dati simili tra loro.
- **Apprendimento semi-supervisionato**: questo metodo è utile quando non è facile ottenere dati etichettati, ma una piccola quantità di dati può migliorare le prestazioni. Per la fase di training, infatti, vengono usati dati di cui solo una parte è etichettata.
- **Apprendimento con rinforzo**: l'algoritmo interagisce con un ambiente dinamico attraverso un approccio "trial and error": riceve delle penalità o ricompense in base alle azioni eseguite e cerca di massimizzare le ricompense ottenute. Viene spesso impiegato per l'addestramento di robot o nell'ambito dei videogiochi.

Gli algoritmi di machine learning possono essere classificati anche in base alla natura del compito da svolgere. Tra i più comuni:

- **Classificazione**: tecnica di apprendimento supervisionato volta a mappare i dati a una delle diverse categorie predefinite. L'obiettivo della classificazione è trovare un profilo descrittivo, chiamato modello, che consente di assegnare etichette di classe a nuovi dati che non ne dispongono. Nella fase di training viene fornito un set di dati etichettati in modo tale che possa crearne un modello; nella fase di testing, il sistema classifica i nuovi dati basandosi sul modello precedentemente estratto.
- **Regressione**: tecnica ad apprendimento supervisionato che si occupa di predire valori continui, rappresentati da variabili dipendenti numeriche, in relazione con variabili indipendenti con valori noti. Molti algoritmi di regressione si basano sull'ipotesi che esista una relazione lineare tra la variabile dipendente e le variabili indipendenti; tuttavia, esistono anche metodi per modellare relazioni polinomiali. L'obiettivo della regressione è minimizzare l'errore di predizione, ossia la differenza tra il valore osservato e il valore previsto. Mentre la regressione predice un valore numerico, la classificazione si occupa del valore categorico. La regressione può essere usata, ad esempio, per la predizione del prezzo di una casa, cambiamenti climatici, previsione dei tassi di interesse.
- **Clustering**: tecnica di apprendimento non supervisionato in cui l'algoritmo raggruppa i dati in cluster basati su caratteristiche simili, senza avere etichette predefinite. Il clustering di partizionamento divide gli oggetti in partizioni in modo da massimizzare la distanza tra cluster differenti e minimizzare la distanza tra gli oggetti dello stesso cluster. Il clustering gerarchico crea un insieme di cluster organizzati come alberi gerarchici. Il clustering basato sulla densità, come il DBSCAN, considera aree ad alta densità di punti per formare i cluster.

3.1 Iperparametri

Nel machine learning, gli iperparametri giocano un ruolo cruciale nell'addestramento e nella performance dei modelli. A differenza dei parametri del modello, appresi durante il processo di addestramento, gli iperparametri sono configurazioni impostate dall'utente prima dell'addestramento. Questi determinano la struttura del modello e la modalità di apprendimento, influenzando direttamente l'efficacia e l'efficienza dell'algoritmo.

Esistono diverse tecniche per la ricerca degli iperparametri ottimali:

- **griglia di ricerca**: tecnica automatica utilizzata in questo studio, è la più diffusa e consiste nel provare tutte le possibili combinazioni di valori per i parametri selezionati. Questa tecnica ha un alto costo computazionale a causa dell'elevato numero di possibili combinazioni.

- **ricerca casuale**: consiste nella scelta di un valore in un insieme predefinito per ogni iperparametro, in modo da testare un numero limitato di combinazioni. Anche se presenta dei vantaggi rispetto alla griglia, come il minor costo computazionale e il minor tempo necessario, è possibile che non vengano esplorate alcune aree critiche dello spazio degli iperparametri.
- **ottimizzazione bayesiana**: si basa sul teorema di Bayes e prevede la costruzione di un modello probabilistico utile alla selezione dei punti da valutare durante ogni iterazione.

3.2 Algoritmi di Classificazione

I problemi di classificazione possono essere binari, se l'output è una variabile binaria, o multi-classe se l'output è un set di classi differenti. Gli algoritmi più comuni volti alla classificazione sono Decision Tree, Random Forest, Support Vector Machines, Logistic Regression, Naive Bayes, K-Nearest Neighbors.

3.2.1 Decision Trees

Utilizzati sia per problemi di classificazione che di regressione, gli alberi di decisione offrono una rappresentazione gerarchica delle decisioni prese utilizzando l'apprendimento supervisionato. L'albero viene costruito partendo dal **nodo radice** che rappresenta la decisione iniziale.

Le decisioni successive sono rappresentate dai nodi intermedi da cui si diramano altri nodi o foglie. La decisione finale è delineata dai nodi terminali. L'attributo che viene valutato ad ogni nodo è scelto in base a criteri che possono essere indicati dal Gini index, entropia, guadagno di informazione o riduzione della varianza nel caso di problemi di regressione.

Il Gini index viene usato per valutare la purezza di un nodo. Al diminuire del Gini index, aumenta la purezza del nodo.

$$\mathrm{GINI}(t) = 1 - \sum_{j=1}^{k} [p(j|t)]^2 \tag{3.1}$$

Dove $p(j|t)$ è la frequenza relativa della classe j nel nodo t. La somma è calcolata su tutte le classi k presenti nel nodo. Quando i record sono distribuiti equamente tra tutte le classi, l'impurità è massima e assume valore $1 - \frac{1}{n_c}$, dove n_c è il numero delle classi. Se tutti i record appartengono a una sola classe, il Gini index è pari a 0 ed indica un nodo completamente puro.

Un'altra misura di impurità di un nodo è l'entropia.

$$\text{Entropia}(t) = -\sum_{j=1}^{k} p(j|t) log_2(p(j|t)) \tag{3.2}$$

Dove $p(j|t)$ è la frequenza relativa della classe j nel nodo t e k è il numero delle classi. L'entropia assume valore massimo $\log n_c$ se tutti i record sono equamente distribuiti tra le classi, indicando un alto grado di impurità. Un basso grado di impurità indica che tutti i record appartengono a una sola classe.

Il guadagno di informazione misura la riduzione dell'entropia dopo una suddivisione. Viene scelta la suddivisione che massimizza il guadagno.

$$\text{GAIN} = Entropia(p) - \left(\sum_{j=1}^{k} \frac{n_i}{n} Entropia(i) \right) \tag{3.3}$$

Dove p è il nodo padre, diviso in k partizioni e n_i è il numero di record appartenenti alla partizione i.

Questa metrica, però, tende a preferire le divisioni che producono un gran numero di partizioni a condizione che siano pure. Per evitare tale tendenza, viene normalizzato il guadagno di informazione, dividendolo per la Split Information, grandezza che indica l'entropia dei partizionamenti.

$$\text{GAINratio}_{\text{split}} = \frac{\text{GAIN}_{\text{split}}}{\text{SplitINFO}} \tag{3.4}$$

$$\text{SplitINFO} = -\sum_{i=1}^{k} \frac{n_i}{n} log \frac{n_i}{n} \tag{3.5}$$

Dove il nodo padre, p è diviso in k partizioni e n_i è il numero di record nella partizione i.

I Decision Tree rientrano tra gli algoritmi più utilizzati perché sono facili da comprendere e interpretare, possono gestire vari tipi di dati sia categorici che continui, senza necessità di normalizzazione.

Tra gli iperparametri da variare, i più importanti sono la profondità massima dell'albero, il numero massimo di partizioni in caso di attributi continui e la soglia minima del guadagno di informazione dopo ogni split.

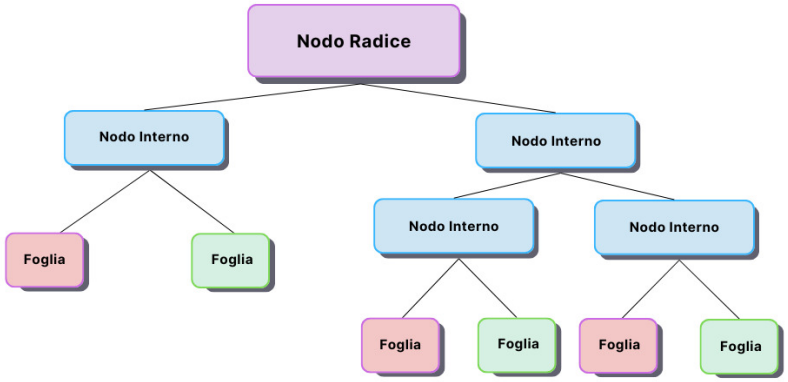


Figura 3.1: Decision Tree

3.2.2 Random Forest

Random Forest è un algoritmo supervisionato per problemi di classificazione e regressione, utilizzato per migliorare l'accuratezza e la stabilità delle predizioni, riducendo al contempo il rischio di overfitting. Questa metodologia si basa sulla combinazione di più alberi decisionali, per ottenere una maggiore robustezza nei risultati. Si parte dal dataset originale e si eseguono campionamenti casuali con sostituzione per generare diversi sottoinsiemi di dati. Da ciascun sottoinsieme di dati, si costruisce un albero di decisione indipendente. In caso di problemi di classificazione, una volta addestrati tutti gli alberi, i loro risultati vengono combinati attraverso il voto di maggioranza o attraverso la media delle predizioni in caso di problemi di regressione.

Il funzionamento del Random Forest per problemi di regressione può essere descritto nel modo seguente:

$$\hat{y} = \text{argmax}_c \sum_{t=1}^{T} I(h_t(x) = c) \tag{3.6}$$

Dove $h_t(x)$ corrisponde alla predizione dell'albero t-esimo per l'input x, c è una classe candidata e $I()$ è una funzione indicatrice che vale 1, se $h_t(x) = c$, e 0 altrimenti. La classe c, che massimizza il conteggio totale, è la predizione del Random Forest [23].

Combinando più alberi, la Random Forest tende ad avere un'alta accuratezza e riduce il rischio di overfitting, risultando, però, meno interpretabile e più costoso in termini di risorse.

Gli iperparametri sono gli stessi del decision tree ma è possibile anche impostare il numero di alberi.

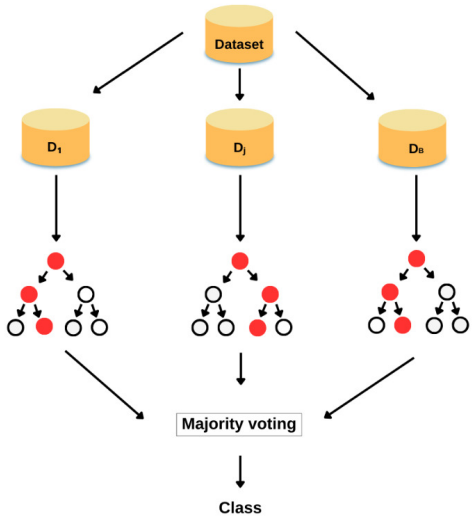


Figura 3.2: Random Forest

3.2.3 Support Vector Machine

L'obiettivo di una Support Vector Machine (SVM) è identificare un iperpiano che divida i dati appartenenti a due classi diverse con il maggior margine possibile tra di esse. In altre parole, l'SVM cerca di trovare una linea (in uno spazio bidimensionale), un piano (in uno spazio tridimensionale), o un iperpiano (in uno spazio con più dimensioni) che separi chiaramente i dati delle due classi. Per i dati che possono essere separati linearmente, la SVM determina un iperpiano che, non solo divide i dati, ma lo fa massimizzando la distanza (detta margine) tra i dati delle due classi più vicini all'iperpiano stesso.

Tuttavia, non tutti i dati possono essere separati linearmente. In questi casi, le SVM utilizzano le cosiddette funzioni kernel, che mappano i dati iniziali in uno spazio di dimensioni superiori, dove è possibile trovare un iperpiano che separi i dati linearmente. In questo modo, anche i dati complessi e non lineari possono essere classificati correttamente.

Attraverso l'iperparametro di penalizzazione, è possibile controllare il trade-off tra una classificazione corretta dei punti di training e la massimizzazione del margine di decisione. È possibile anche indicare il tipo di kernel utilizzato nella trasformazione dei dati.

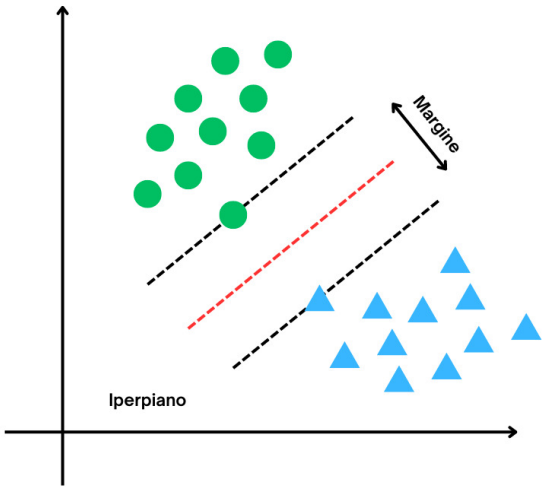


Figura 3.3: Support Vector Machines

3.2.4 Logistic Regression

Nella Logistic Regression viene applicata la funzione sigmoide ad una combinazione lineare delle features pesate del dato. Tale funzione risulta perfetta per modellare la probabilità, in quanto è definita nel range da 0 a 1. L'output è un valore di probabilità tra 0 e 1 che viene confrontato con una soglia per prendere la decisione finale [24].

È possibile indicare il numero massimo di iterazioni durante la fase di training, la soglia in caso di problemi di classificazione e un parametro di regolarizzazione.

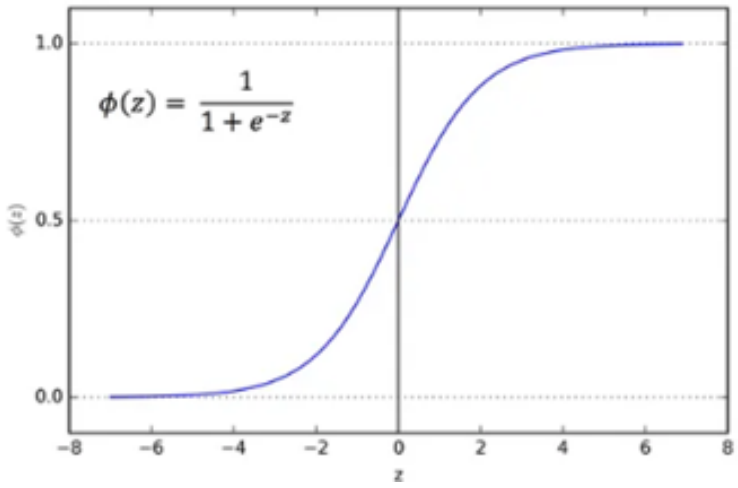


Figura 3.4: Funzione Sigmoide

3.2.5 Classificatore Bayesiano

Questo tipo di classificazione è basata sul teorema di Bayes e sull'indipendenza statistica degli attributi. Calcola la probabilità condizionata per ogni classe, cioè la probabilità che un dato appartenga ad una determinata classe.

3.2.6 K-Nearest Neighbors

Il K-Nearest Neighbors è un algoritmo che si basa sull'idea che oggetti simili si trovino vicini nello spazio. Data una nuova istanza da classificare, viene calcolata la distanza da tutti i record di training e assegnata la classe più frequente tra i k record più vicini. È fondamentale scegliere il parametro k in maniera accurata poichè un valore troppo piccolo può rendere l'algoritmo sensibile al rumore, mentre un valore troppo grande può far sì che vengano considerati punti non rilevanti o che facciano parte di altre classi.

Comunemente vengono utilizzate la Distanza Euclidea o la Distanza di Manhattan, rispettivamente: $d = \sqrt{\sum_{i=1}^{n}(x_i - y_j)^2}$ e $d = \sum_{i=1}^{n}|x_i - y_j|$.

In fase di tuning dell'algoritmo, è possibile impostare il numero di record vicini da considerare per la classificazione di un nuovo punto, il peso e il tipo di distanze da utilizzare.

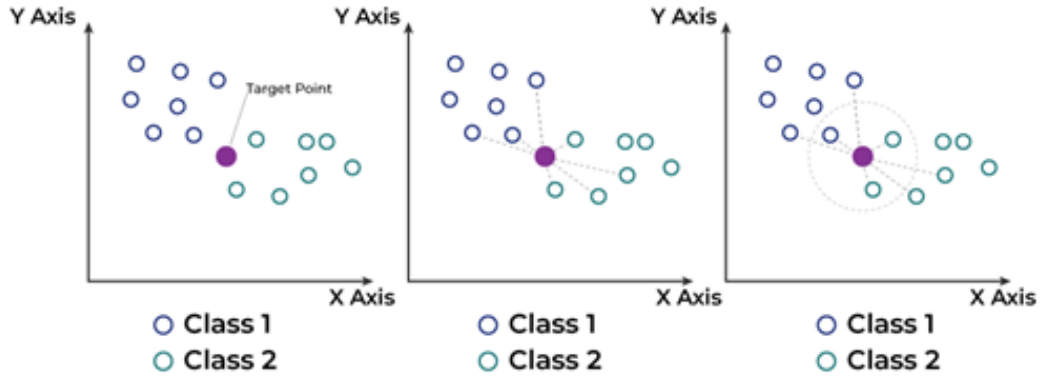


Figura 3.5: K-Nearest Neighbors

3.2.7 Metodi di Ensemble

I metodi di ensemble combinano le previsioni di diversi modelli di base per migliorare l'accuratezza e la robustezza delle predizioni finali. L'idea di base è quella secondo cui un gruppo di modelli possa produrre un risultato migliore di quello ottenuto da un singolo modello, riducendo l'errore complessivo e migliorando la capacità di generalizzazione.

I metodi paralleli allenano ogni classificatore individualmente e possono essere omogenei ed eterogenei.

I primi, utilizzano molteplici versioni dello stesso algoritmo di base, mentre i secondi usano algoritmi diversi che possono avere pesi distinti a seconda della qualità della stima o i risultati possono essere usati come metadati per allenare un predittore finale. Un esempio di impiego del metodo parallelo omogeneo è il Random Forest che utilizza una collezione di alberi di decisione addestrati su sottoinsiemi del dataset e combina le predizioni mediante il voto di maggioranza.

I metodi sequenziali utilizzano dei modelli che vengono addestrati in sequenza, ognuno sui risultati del precedente. Ogni modello cerca di correggere gli errori in modo da migliorare gradualmente il risultato complessivo. Un esempio di impiego del metodo sequenziale è AdaBoost. In questo caso, ad ogni iterazione vengono modificati i pesi dei campioni, sopperendo agli errori precedenti [25].

3.3 Metriche di Valutazione

Le metriche di valutazione sono fondamentali per determinare le prestazioni di un algoritmo di machine learning. Ogni tipo di problema richiede metriche specifiche e la scelta di quella più appropriata, dipende dal contesto e dagli obiettivi.

Nel contesto dei problemi di classificazione, **la matrice di confusione** permette di visualizzare i risultati di un algoritmo mostrando i conteggi delle predizioni corrette e non corrette per ciascuna classe. La suddetta matrice ha dimensioni pari al numero di classi del problema. Nel caso di classificazione binaria, la dimensione è di 2x2 e indica il numero di campioni della classe positiva correttamente classificati come tali (TP); il numero di campioni della classe negativa, erroneamente classificati come positivi (FP); il numero di campioni della classe positiva, erroneamente classificati come negativi (FN) e il numero di campioni della classe negativa correttamente classificati (TN).

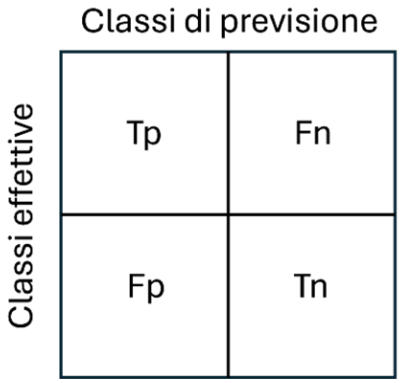


Figura 3.6: Matrice di confusione

La matrice di confusione permette di calcolare diverse metriche utili come [26]:

- **Accuratezza**: rapporto tra le previsioni corrette e quelle totali;

$$\text{Accuratezza} = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.7}$$

- **Precisione**: proporzione di previsioni positive corrette sul totale di quelle positive;

$$\text{Precisione}(p) = \frac{TP}{TP + FP} \tag{3.8}$$

- **Richiamo o sensibilità**: proporzione tra veri positivi e il totale dei positivi;

$$\text{Richiamo}(r) = \frac{TP}{TP + FN} \tag{3.9}$$

- **F1-Score**: media armonica tra precisione e richiamo;

$$\text{F1-Score} = \frac{2pr}{p+r} \tag{3.10}$$

- **False Positive Rate**: rapporto tra falsi positivi e totale dei negativi presenti

$$\text{FPR} = \frac{FP}{FP+TN} \tag{3.11}$$

- **Receiver Operating Characteristic (ROC)**: è una curva che traccia il rapporto tra TPR e FPR ed è utile per confrontare le prestazioni di diversi classificatori. L'area al di sotto della curva (AUC) riassume le performance del classificatore: un valore pari a 1 descrive un classificatore perfetto, mentre 0.5 indica un classificatore che non ha capacità discriminativa. I punti sulla diagonale, infatti, indicano prestazioni equivalenti al caso casuale; i punti vicini all'angolo superiore sinistro, al contrario, indicano buone prestazioni con alti TPR e bassi FPR.

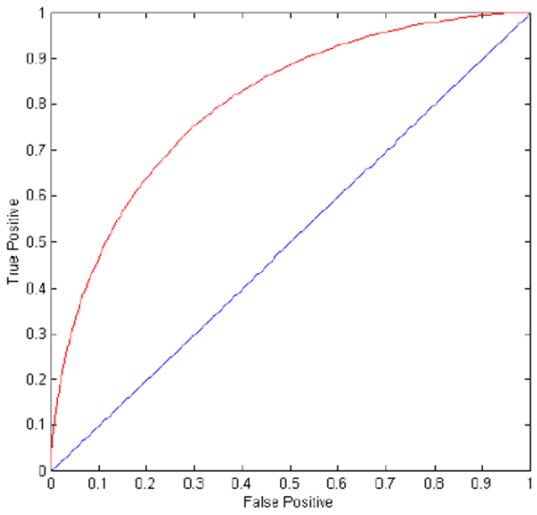


Figura 3.7: ROC

La **K-Fold Cross Validation** è la tecnica più comunemente usata per valutare le prestazioni di un modello di machine learning e stimarne l'efficacia in maniera affidabile. In questo metodo, il dataset viene diviso in k sottogruppi: k-1 corrispondono ai set di training, il rimanente è quello di test. Ad ogni iterazione viene cambiato a turno il set utilizzato come test. I risultati forniscono una stima delle prestazioni del modello.

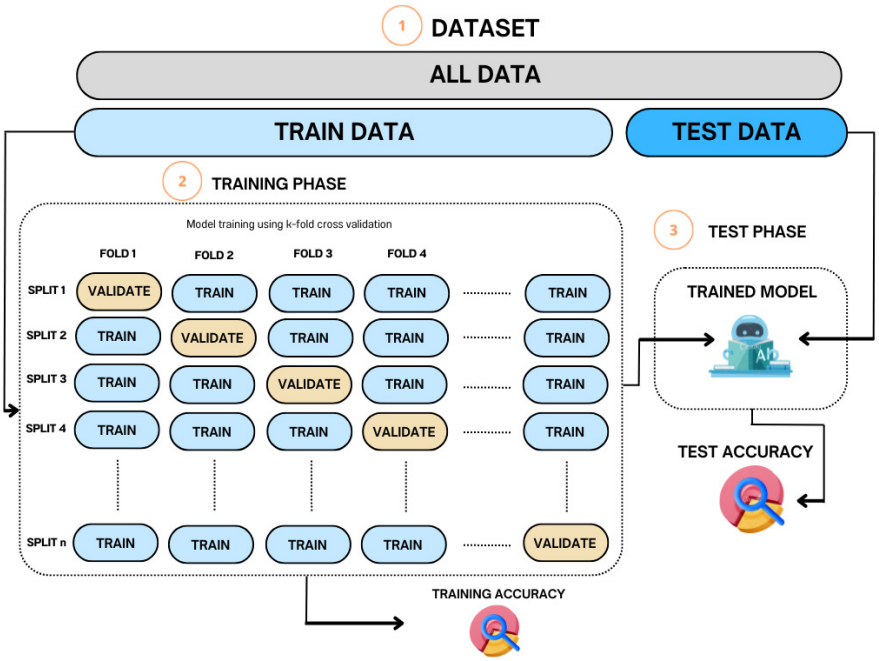


Figura 3.8: K-Fold Cross Validation

3.4 Natural Language Processing

Il Natural Language Processing (NLP) è un campo dell'intelligenza artificiale il cui obiettivo è consentire a computer e dispositivi digitali il riconoscimento, la comprensione e la generazione del testo e voce, combinando linguistica computazionale, statistica, deep learning e machine learning.

L'NLP ha diversi campi di utilizzo come trascrizione da linguaggio parlato in testo; la tradizione di testi tra lingue diverse, l'interazione automatica dei sistemi con gli utenti attraverso generazione del linguaggio naturale, l'estrazione delle informazioni e l'analisi delle opinioni ed emozioni espresse nei testi [27].

La funzione **term frequency – inverse document frequency (tf-idf)** è utilizzata per misurare l'importanza di un termine rispetto ad un documento o ad una collezione di documenti.

È composta dalla combinazione del **term frequency (tf)** che indica il numero di volte che una parola appare nel testo e l'**inverse document frequency (idf)** calcolato come il logaritmo del rapporto tra il numero totale di documenti e il numero di documenti in cui appare la parola:

$$\mathrm{tf}_{i,j} = \frac{n_{i,j}}{|d_j|} \qquad (3.12) \qquad\qquad \mathrm{idf}_i = \log_{10} \frac{|D|}{|d : i \epsilon d|} \qquad (3.13)$$

Dove $n_{i,j}$ è il numero delle occorrenze del termine i nel documento j; $|d_j|$ è la dimensione del documento j, espressa in numero di parole; $|D|$ è il totale dei documenti; $|\{d : i \in d\}|$ è il numero di documenti che contengono il termine i.

Quindi:

$$(\mathrm{tf} - \mathrm{idf})_{i,j} = \mathrm{tf}_{i,j} \times \mathrm{idf}_i \qquad (3.14)$$

Il **Word2Vec** è un metodo, sviluppato da Google, capace di capire il contesto di una parola nel testo, la sua relazione con le altre parole, la semantica e la somiglianza sintattica, attraverso una rete neurale. Crea una rappresentazione vettoriale delle parole in modo tale che termini simili siano vicini nello spazio.

Il **BERT**, sviluppato da Google, è un modello che consiste nella lettura del testo in entrambe le direzioni, in modo da capire il contesto di una parola, basandosi su quelle che la precedono e la susseguono. Durante la fase di pre-addestramento alcune parole vengono casualmente nascoste e indovinate dal modello oppure il modello impara la relazione tra due frasi per predire la frase successiva [28].

3.5 RapidMiner

RapidMiner è una piattaforma di data science e machine learning utilizzata per la creazione e la gestione di processi di analisi dei dati. Attraverso la sua interfaccia visiva a blocchi permette di progettare, testare e valutare i modelli predittivi senza dover necessariamente scrivere codice.

RapidMiner supporta un'ampia gamma di algoritmi di machine learning, regressioni, classificazioni e clustering. Permette di valutare di confrontare diversi modelli attraverso la gestione automatizzata della cross validation e fornendo delle metriche relative alla performance. Inoltre, permette di incorporare facilmente funzionalità aggiuntive e librerie esterne come strumenti per il deep learning o per il text mining.

Un altro aspetto importante è la capacità di interfacciarsi con diversi tipi di sorgenti dati come database relazionali, file Excel e tecnologie cloud, rendendo la piattaforma versatile per chi lavora con dataset eterogenei e complessi.

Capitolo 4

DATASET

Il dataset utilizzato in questa tesi è composto da due fonti principali, selezionate per garantire un bilanciamento tra email malevole (phishing) ed email legittime (non phishing). I dati sono stati raccolti combinando una repository pubblica per le email di phishing e una raccolta privata per le email legittime. La repository pubblica, appartenente a José Nazario [29], include email confermate e campioni di phishing inviati per truffare utenti o rubare credenziali, raggruppando email raccolte manualmente tra il 2021 e il 2024, in seguito anonimizzate.

Le email legittime sono state raccolte manualmente da account di posta elettronica personali e aziendali, anche queste anonimizzate, per simulare fedelmente le tipologie di comunicazioni che gli utenti ricevono quotidianamente.

Il dataset è stato creato includendo diverse categorie di email, come: email di lavoro inviate tra colleghi e superiori; newsletter contenenti informazioni di marketing o aggiornamenti di siti e servizi; email personali tra conoscenti e amici e email informative come conferme di prenotazioni, ricevute di acquisto, e comunicazioni ufficiali. Le email sono fornite in formato .mbox, il che ha permesso di accedere a tutti i dettagli strutturali e tecnici (header, testo, eventuali allegati).

La classe di phishing è composta da 1113 email e quella legittima da 1018 email, per un totale di 2131 campioni. Questa suddivisione bilanciata garantisce una base solida per il training dei modelli di machine learning e riduce il rischio di dataset sbilanciati.

4.1 Raccolta e Preparazione dei Dati

Le email sono state scaricate direttamente dalla repository di José Nazario, che fornisce campioni in formato standard .txt e convertiti in formato .mbox [29].

Le email legittime, contenute in dataset anonimizzati, sono state estratte manualmente da vari account di posta elettronica tramite operazioni di esportazione in formato .mbox. È stata prestata particolare attenzione a rimuovere qualsiasi informazione sensibile e a garantire la conformità con le normative sulla privacy.

Per ogni campione, sono stati estratti i seguenti elementi: header, testo e allegati, utilizzando la libreria mailbox di Python.

Prima di passare alla fase di estrazione delle features, è stata condotta un'analisi esplorativa per ottenere informazioni preliminari sul dataset. Tale indagine ha evidenziato l'utilizzo di domini sospetti e testi con codice html e urls nei campioni di phishing. Inoltre, sono stati individuati elementi utili alla fase successiva (es. la ripetizione nel corpo delle email di informazioni relative al proprietario della repository, la presenza di testi in lingue e codifiche diverse), in seguito eliminati per evitare bias in fase di training dell'algoritmo.

Capitolo 5

ARCHITETTURA DELL'ALGORITMO

L'architettura definita in questo studio prevede un sistema di classificazione che prende in considerazione le intere email scomponendole in header, testo e allegati, utilizzando le features estratte per allenare più modelli in parallelo e catalogarle in phishing o legittime.

A differenza di approcci che considerano queste parti in modo isolato o solo in parte, l'approccio proposto integra le caratteristiche estratte da ciascun componente al fine di formare un modello più robusto e preciso.

Gli attacchi di phishing, infatti, possono essere condotti con tecniche diverse. Pertanto, combinando le informazioni estratte da ogni componente, è possibile migliorare la capacità di classificazione, riducendo il rischio di falsi positivi e negativi e massimizzando l'uso delle informazioni disponibili.

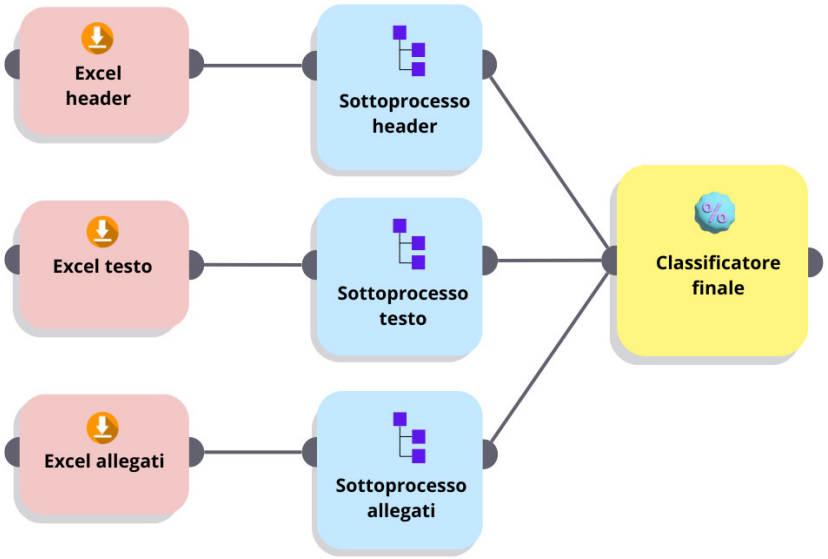


Figura 5.1: Architettura dell'algoritmo

L'algoritmo, nel dettaglio, è costituito da diverse fasi:

1. **Suddivisione dell'email**: ogni email viene divisa in header, testo e allegati utilizzando la libreria email di Python.

2. **Estrazione delle features**: le features vengono estratte da ciascuna delle tre componenti e salvate in tre files Excel distinti

3. **Importazione dei dati in RapidMiner**: i tre files Excel vengono importati in RapidMiner per la fase di preprocessing e addestramento dei modelli.

4. **Preprocessing del testo in RapidMiner**: il testo subisce una fase di preprocessing per generare le informazioni utili alla fase di addestramento.

5. **Addestramento dei Decision Tree**: in RapidMiner vengono addestrati tre modelli separati di Decision Tree, uno per ogni parte delle e-mail;

6. **Decisione finale**: i risultati dei tre Decision Trees vengono combinati attraverso un ulteriore Decision Tree che funge da classificatore prendendo la decisione finale basandosi sui tre modelli precedenti.

5.1 Estrazione delle Features

Nel contesto dell'analisi e classificazione delle email, la fase di estrazione delle features ha un ruolo fondamentale per identificare pattern e costruire, quindi, un modello efficace. Le features considerate riguardano sia attributi testuali che metadati, con il fine di consentire la distinzione tra email legittime ed email di phishing.

Le componenti di ogni email sono state trattate in maniera distinta tramite script Python volti a estrarre le features utili allo studio, in seguito salvate in tre file Excel diversi.

5.1.1 Estrazione delle Features dall'Header

Le informazioni estratte dall'header sono legate ai metadati tra cui mittente, destinatario, informazioni di autenticazione e strutture dei messaggi.

Il primo passo dello script consiste nel verificare la presenza o assenza dei campi principali (es. "From","To","Subject"). L'assenza di tali informazioni all'interno delle email potrebbe essere un indizio di tentativo di phishing. Il codice, quindi, controlla l'esistenza di tali campi e la registra attraverso variabili booleane.

In seguito, viene analizzata anche la coerenza tra i campi. Ad esempio, il "Return-Path", che indica l'indirizzo di consegna dei messaggi di errore, dovrebbe coincidere con il "From" contenente quello del mittente della mail. Lo stesso controllo viene effettuato sul campo "Reply-To" che indica l'indirizzo di risposta: se questo fosse diverso da quello del "From", l'attacco di phishing potrebbe avere come obiettivo quello di indurre il destinatario a rispondere a un altro indirizzo.

Tra le altre features considerate ci sono anche quelle legate alla sicurezza.

L'**SPF** (Sender Policy Framework) è un protocollo che permette di autorizzare un indirizzo all'invio da parte di un dominio specifico, aggiungendo l'informazione al DNS.

Il **DKIM** (DomainKeys Identified Mail) serve a verificare l'integrità della mail ed evitare l'email spoofing. Consiste nell'aggiunta nell'header di una firma digitale e della relativa chiave pubblica. Il destinatario dell'email può verificare, quindi, che la chiave indicata nell'header sia associata al dominio del mittente. In caso assuma il valore "fail" il messaggio potrebbe essere stato modificato.

Il **DMARC** (Domain-Based Message Authentication, Reporting & Conformance) è un sistema di verifica delle email che utilizza in maniera congiunta i campi precedenti. Pubblicato come un record DNS, il DMARC permette di specificare le politiche di gestione delle email che non superano i controlli precedenti, applicando una serie di criteri: "none" nei casi in cui il messaggio deve essere consegnato ugualmente ma, con l'invio al titolare

del dominio di un rapporto DMARC; "quarantine" se il messaggio deve essere consegnato nella cartella spam del destinatario e "reject" nel caso in cui non deve essere recapitato.

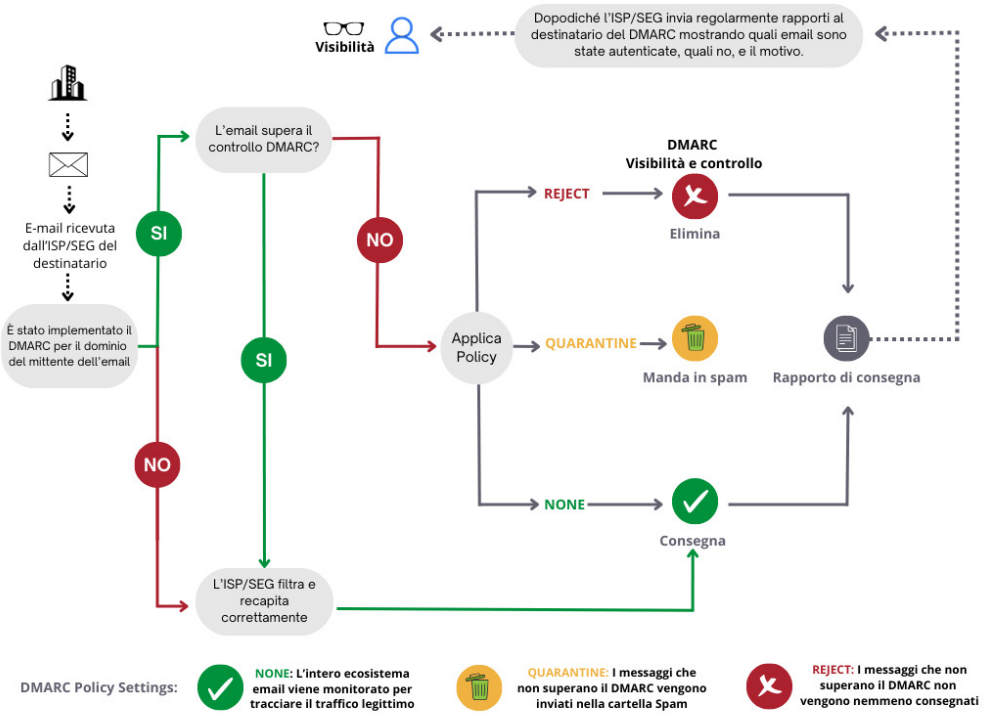


Figura 5.2: Funzionamento del DMARC

Vengono considerati anche l'uso di protocolli standard come ESMTP, IMAP e POP per identificare anomalie nel percorso delle email di phishing. Inoltre, si verifica che il campo "Received" non contenga la parola "unkown", indicante il passaggio delle email attraverso server non identificati. Le informazioni ricavate vengono salvate in variabili booleane.

Anche il campo "Subject" può contenere informazioni relative a possibili tentativi di phishing. Tra le features vengono tenute in conto, attraverso variabili booleane: l'uso di maiuscole per attirare l'attenzione del destinatario, la presenza di parole frequenti come "password", "verifica", "account", "gratis", "consegna" o la presenza dei simboli "!@ # $ % ^ & *()".

Infine, i campi "To" e "Cc" possono essere indicatori di tentativi di phishing se è presente un elevato numero di destinatari, valore che viene incluso tra le features. I valori generati vengono organizzati in un vettore che viene salvato in un file excel contenente una riga per ogni email, in modo da avere una rappresentazione utile al modello di machine learning.

Di seguito si riporta l'elenco delle features estratte:

- *subject_exists*, *from_exists*, *to_exists*, *replyTo_exists*, *returnPath_exists*, *messageid_exists*: sono variabili di tipo booleano e assumono valore "True" se l'email ha rispettivamente i campi "Subject", "From", "To", "Reply-To", "Return Path", "messageID";

- *replyTo_pass*: variabile di tipo booleano che assume valore "True" se il dominio del "Reply-To" è uguale al dominio del "From";

- *valore_dkim*, *valore_spf*, *valore_dmarc*: variabili di tipo Stringa che estraggono il valore di DKIM, SPF, DMARC se sono presenti in "Authentication-Results";

- *received_unknown*: variabile di tipo booleano che assume valore "True" se il campo "Received" contiene la parola "unknown";

- *protocol_known*: variabile di tipo booleano che assume valore "True" se il campo "Received" contiene protocolli noti (ESMTP, IMAP, SMTP, POP);

- *empty_date*: variabile booleana che assume valore "True" se il campo "Date" è vuoto;

- *subject_pass*, *subject_verify*, *subject_bank*, *subject_account*, *subject_business*, *subject_loan*, *subject_free*, *subject_delivery*, *subject_shipment*: variabili booleane che assumono valore "True" nel caso in cui il campo "Subject" contenga rispettivamente le parole "password", "verifica", "banca", "account", "business", "prestito", "free" o "gratis", "consegna", "spedizione";

- *subject_special*: variabile di tipo booleano che assume valore "True" nel caso in cui il campo "Subject" contenga caratteri speciali;

- *numTo_addresses*, *numCC_addresses*: variabili Intere contenenti il numero di indirizzi indicati nei campi "To" e "Cc";

- *subject_upper*, *subject_alpha*: variabili booleane che assumono valore "True" se il campo "Subject" è scritto completamente in maiuscolo o se contiene solo caratteri alfabetici.

```
Return-Path: <support@trusted-company.com>
Received:   from   mail.trusted-company.com   (mail.trusted-company.com
[203.0.113.45])
    by mail.recipientserver.com (Postfix) with ESMTPS id CBA789FED
    for <victim@recipient.com>;
    Tue, 15 Oct 2024 10:32:10 +0200 (CEST)
Received:    from    mx.trusted-company.com    (mx.trusted-company.com
[203.0.113.46])
    by mail.trusted-company.com (Postfix) with ESMTP id DEF123ABC;
    Tue, 15 Oct 2024 08:30:02 +0000 (UTC)
Received-SPF:  pass  (trusted-company.com:  domain  of  support@trusted-
company.com designates 203.0.113.45 as permitted sender)
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed;
    d=trusted-company.com; s=default; t=1697374200;
    bh=ZkzKL5UOr6fA7lgxWvGQgDSdWjb1cN9y/wAnqSeBknw=;
    b=P9k3XLJsbD3w...
DMARC-Filter:       pass       (trusted-company.com:       policy=none)
header.from=trusted-company.com
Message-ID: <123456789@trusted-company.com>
Date: Tue, 15 Oct 2024 10:30:01 +0200
From: "Support Team" <support@trusted-company.com>
Reply-To: "Support Team" <support@trusted-company.com>
To: victim@recipient.com
Subject: Important: Confirm Your Account
MIME-Version: 1.0
Content-Type: text/html; charset="UTF-8"
```

Figura 5.3: Header di una email di phishing (esempio)

5.1.2 Estrazione delle Features dal Testo

Un aspetto fondamentale per preparare il testo delle email alle fasi successive è la corretta estrazione e pulizia del testo contenuto nei messaggi, eliminando le componenti superflue e normalizzando le informazioni.

Innanzitutto, è necessario scomporre il messaggio analizzando ricorsivamente ogni parte di esso, estraendo i payload associati ai differenti MIME.

Le email che contengono codice HTML sono state gestite attraverso la libreria BeautifulSoup per estrarne il testo ed eliminare i tag HTML ed elementi indesiderati come script e style.

Il testo, così ricavato, è sottoposto a una serie di trasformazioni per normalizzarne il contenuto e rimuovere il rumore.

Viene eliminato il simbolo "=\n" usato per andare a capo alla fine di una riga; gli URL, identificati tramite una regex, vengono sostituiti con la stringa generica "url"; alcuni termini vengono eliminati in quanto strettamente legati alla provenienza del dataset, per evitare di generare bias nel modello rappresentando false correlazioni.

Il risultato finale è una versione normalizzata del testo dell'email priva di elementi

superflui, che viene salvata in un vettore in un file excel contenente in ogni riga il testo di ciascuna email.

5.1.3 Estrazione delle Features dagli Allegati

L'estrazione delle features degli allegati è importante poiché, spesso, vengono utilizzati nomi di file ingannevoli o estensioni insolite per mascherare l'obiettivo malevolo. Ogni email viene analizzata per accertare la presenza di immagini o file, verificando il MIME di ogni parte e considerare gli allegati di tipo "application" o "image". L'analisi si concentra sul nome del file prendendo in esame il numero di caratteri che lo compongono, il conteggio di lettere e numeri, la quantità di lettere maiuscole, spazi e caratteri speciali.

Per ogni file vengono calcolati i rapporti tra i tipi di caratteri e la lunghezza totale del nome. Il basso rapporto tra il numero delle lettere e la lunghezza complessiva potrebbe indicare una manipolazione del nome con altri tipi di caratteri, o la densità di cifre numeriche, la cui elevata presenza potrebbe segnalare una generazione automatica del nome del file.

Successivamente, viene considerata l'estensione del file che fornisce informazioni sul tipo di contenuto: i file eseguibili possono contenere codice malevolo o macro nel caso di documenti di tipo .pdf o .docx.

Infine, si calcola la dimensione del file in byte. Un file troppo piccolo potrebbe essere un vettore di script, mentre uno eccessivamente grande potrebbe mascherare dati o script malevoli.

Le features estratte, registrate in un vettore, vengono scritte su un file excel contenente per ogni riga le informazioni relative agli allegati di ciascuna email.

Di seguito si riportano le features definite al termine di questa fase:

- *filename_length*: lunghezza del nome dell'allegato;
- *nLetters*: numero totale di lettere presenti nel nome del file;
- *nDigits*: numero totale di cifre nel nome del file;
- *capitalLetters*: numero di lettere maiuscole presenti nel nome del file;
- *spaces*: numero totale di spazi nel nome del file;
- *dashes*: numero di caratteri "-" e "_" nel nome del file;
- *other_chars*: numero di caratteri che non sono lettere o numeri, come simboli e punteggiatura;

- *vowels*: numero di vocali presenti nel nome del file;
- *ratio_letters*: rapporto tra numero di lettere e lunghezza totale del nome del file;
- *ratio_digits*: rapporto tra il numero di cifre e la lunghezza totale del nome del file;
- *ratio_vowels*: rapporto tra il numero di vocali e la lunghezza totale del file;
- *first_letter*, *first_digit*: indicano se il primo carattere del nome è una lettera o un numero;
- *last_letter*, *last_digit*: indicano se l'ultimo carattere del nome del file è una lettera o un numero;
- *has_attachment*: indica se l'email ha un allegato;
- *file_extension*: indica l'estensione del file dell'allegato;
- *attachment_size*: indica la dimensione dell'allegato in byte.

5.2 Preprocessing del Testo

Una volta importati in RapidMiner i file excel generati nella fase precedente, il testo è sottoposto a una fase di preprocessing per estrarre le informazioni utili alla fase successiva.

Figura 5.4: Preprocessing del testo

Il primo passo consiste nella Tokenizzazione attraverso la quale i testi vengono suddivisi in parole, separandoli in base a delimitatori.

In seguito, i token vengono convertiti in minuscolo per uniformare i termini e ridurre la differenziazione causata dalla sensibilità delle maiuscole.

Vengono eliminate, poi, le stopwords, cioè le parole comuni di un testo che non aggiungono un significato rilevante e non hanno niente a che vedere con un argomento specifico.

Lo stemming, invece, consiste nella riduzione delle forme flesse delle parole alla loro radice morfologica.

Infine, vengono eliminati i termini inferiori ai 4 caratteri così da escludere termini non utili all'algoritmo ma frequenti come articoli e preposizioni, eventuali rumori o parole errate.

Al termine di queste operazioni, viene costruita la matrice TF-IDF che rappresenta i documenti come vettori numerici. In tale struttura, ogni riga rappresenta un'email e ogni colonna un termine del vocabolario. Ogni cella, quindi, contiene il valore che misura l'importanza del termine rispetto al documento.

Un valore TF-IDF elevato significa che il termine è presente frequentemente nella email, ma appare raramente negli altri testi. Questo indica che il termine risulta importante in quella specifica email.

5.3 Addestramento del Modello

Le features prodotte nelle fasi precedenti vengono passate a diversi sottoprocessi, ognuno dei quali è responsabile della classificazione iniziale effettuata in maniera distinta sulla base delle informazioni estratte da header, testo e allegati.

5.3.1 Sottoprocesso Header

Una volta indicata la variabile target da predire, le features estratte vengono usate per addestrare un Decision Tree e valutarlo attraverso Cross Validation con 10 folds. Questo approccio suddivide il dataset in dieci sottoinsiemi, utilizzandone nove per l'addestramento e uno per il test, iterando il processo su tutti i folds per ridurre il rischio di overfitting.

Per ottimizzare le prestazioni del Decision Tree è stata utilizzata una griglia di ricerca per identificare i migliori parametri di configurazione del modello. La profondità massima dell'albero è stata, quindi, impostata a 29 livelli e il Gini Index, che misura la purezza dei nodi durante la suddivisione, è stato indicato come criterio di split.

Al termine del sottoprocesso, il modello genera per ciascuna email la relativa previsione, etichettandola come phishing o legittima. Questo risultato viene memorizzato come una riga nella tabella finale associata all'email corrispondente per facilitare l'integrazione con i risultati degli altri sottoprocessi.

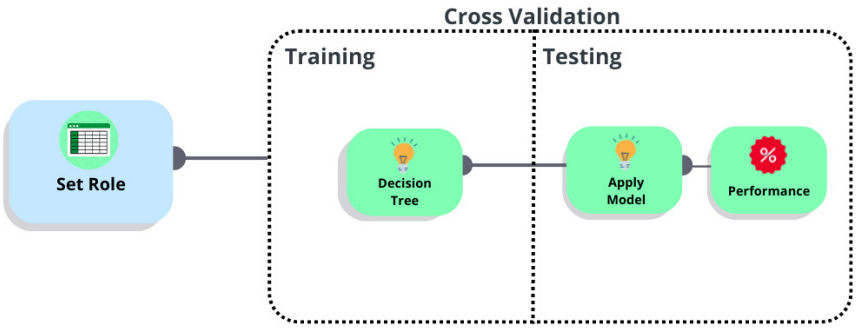


Figura 5.5: Sottoprocesso Header

5.3.2 Sottoprocesso Testo

Una volta prodotta la matrice TF-IDF, viene indicata la variabile target da predire, cioè l'etichetta che segnala un'email come phishing. Viene, dunque, calcolato per ogni termine il relativo Chi-Square rispetto alla variabile target, che misura l'associazione tra le frequenze di ogni termine e le classi del dataset in modo da identificare quali vocabili siano più indicativi nel distinguere le classi. Il Chi-Square si calcola tramite la formula:

$$X^2 = \sum \frac{(O - E)^2}{E} \tag{5.1}$$

"O" rappresenta la frequenza osservata del termine in una classe specifica, "E" designa la frequenza attesa.

I termini con punteggi più alti sono quelli che hanno una maggiore associazione con la classe target e sono, di conseguenza, più rilevanti.

I pesi, così calcolati, sono utilizzati per selezionare i 1000 termini più rilevanti ed eliminare quelli con pesi bassi che contribuiscono poco alla distinzione tra phishing e non phishing, riducendo il rumore e migliorando l'efficienza.

Un decision tree viene addestrato e valutato utilizzando la tecnica della Cross Validation con 10 folds con una profondità pari a 10 e come criterio il guadagno di informazione, la cui soglia minima è impostata a 0.01. Questi valori sono stati trovati attraverso il metodo della griglia di ricerca.

Infine, l'output di questo sottoprocesso equivale ad una riga per ogni email, contenente la previsione effettuata dal modello.

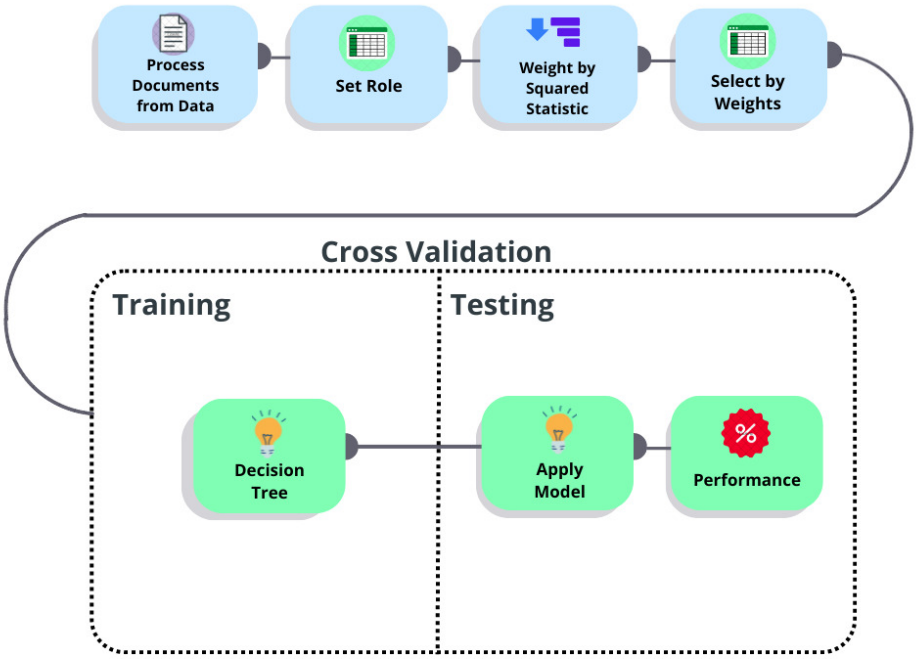


Figura 5.6: Sottoprocesso testo

5.3.3 Sottoprocesso Allegati

Analogamente all'header, anche le features estratte dagli allegati vengono usate per addestrare un Decision Tree indipendente. Una volta specificata la variabile target, l'addestramento viene eseguito attraverso la Cross Validation con 10 folds.

Vengono utilizzati come parametri ottimali quelli ottenuti attraverso la griglia di ricerca.La profondità dell'albero è pari a 25 livelli e il GINI Index viene usato come criterio di split.

Alla fine del sottoprocesso, l'output è una riga per ogni email contenente la previsione, che verrà, in seguito, combinata con le previsioni degli altri sottoprocessi per la classificazione finale

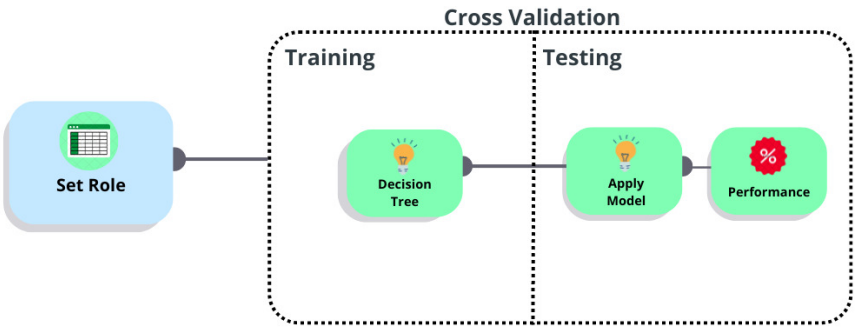


Figura 5.7: Sottoprocesso allegati

5.4 Classificatore Finale

Dopo aver completato l'addestramento e la valutazione dei tre modelli indipendenti, viene costruito un classificatore finale utilizzando un Decision Tree.

L'output di ciascun sottoprocesso, costituito da una riga contenente la previsione parziale per ogni email, viene utilizzato come feature di input di questo classificatore. Il Decision Tree viene addestrato utilizzando la tecnica di Cross Validation con 10 folds.

Anche per il classificatore finale viene utilizzata la griglia di ricerca per identificare i parametri ottimali. La profondità massima dell'albero è pari a 10 livelli, data la semplicità delle features di input e il guadagno di informazione è il criterio di split.

Il **Decision Tree finale** è progettato per combinare le informazioni derivate dai tre modelli iniziali, bilanciando le loro previsioni per determinare la classificazione complessiva dell'email. Questa configurazione consente di sfruttare i punti di forza di ciascun classificatore:

- **Il modello sull'header** si concentra su aspetti strutturali delle email (come campi "From" sospetti o modifiche nell'indirizzo).
- Il **modello sul testo** si occupa di individuare termini o frasi tipicamente associate agli attacchi di phishing.
- Il **modello sugli allegati** analizza le caratteristiche dei file allegati, come la presenza di estensioni dannose o anomalie nei metadati.

Questo approccio permette di creare un **sistema più affidabile** e robusto nella rilevazione del phishing. Grazie al **Decision Tree** globale, le relazioni tra le diverse previsioni vengono esplicitamente rappresentate e combinate in modo gerarchico, consentendo al modello di gestire casi ambigui o complessi. Il risultato è una maggiore accuratezza complessiva nel rilevamento delle email di phishing, con una riduzione dei falsi positivi e negativi rispetto all'utilizzo di singoli classificatori.

Uno dei principali vantaggi di un Decision Tree è la sua interpretabilità: è possibile visualizzare l'albero e seguire le logiche di classificazione, rendendolo comprensibile anche ai non esperti di machine learning. Inoltre, il processo di cross-validation consente di ottimizzare e ridurre la complessità del modello, producendo un albero che non solo classifica accuratamente, ma anche facile da comprendere e interpretare per valutare l'importanza delle caratteristiche.

Capitolo 6

ANALISI DEI RISULTATI

L'analisi dei risultati, ottenuti in seguito all'implementazione del modello di classificazione di email di phishing, consente di valutare l'efficacia del sistema e identificare i margini di miglioramento.

L'addestramento e la valutazione dei modelli sono stati effettuati utilizzando RapidMiner che ha reso possibile l'analisi dei dati in modo visivo e intuitivo, permettendo anche di testare rapidamente diverse configurazioni di modelli, ottimizzando le prestazioni generali del sistema.

Il modello finale selezionato è un Decision Tree, addestrato utilizzando la cross validation con 10 folds. La scelta è motivata dalla capacità del modello di fornire accuratezza e interpretabilità, risultata fondamentale per affinare il processo nelle fasi intermedie dello sviluppo.

6.1 Analisi dei Risultati Ottenuti

I risultati ottenuti si basano su un dataset che comprende un totale di 2131 campioni. Le classi sono distribuite in modo equilibrato, avendo 1113 campioni di phishing e 1018 classificati come non phishing. La matrice di confusione ottenuta durante la fase di valutazione del modello è riportata di seguito:

accuracy: 99.20% +/- 0.54% (micro average: 99.20%)

	true phishing	true no phishing	class precision
pred. phishing	1101	5	99.55%
pred. no phishing	12	1013	98.83%
class recall	98.92%	99.51%	

Da questa matrice di confusione è possibile ricavare le seguenti informazioni:

- **Vero Positivi (TP)**: 1101 email di phishing correttamente identificate come tali.
- **Falsi Negativi (FN)**: 12 email di phishing erroneamente classificate come legittime.
- **Vero Negativi (TN)**: 1014 email legittime correttamente identificate.
- **Falsi Positivi (FP)**: 5 email legittime erroneamente classificate come phishing.

6.1.1 Metriche di Valutazione

Attraverso i risultati ottenuti è possibile esaminare in dettaglio le principali metriche di valutazione del modello:

- **Accuratezza**: rappresenta la percentuale di classificazioni corrette sul totale di campioni

$$\text{Accuratezza} = \frac{TP + TN}{TP + FP + TN + FN} = \frac{1101 + 1013}{1101 + 1013 + 5 + 12} = \frac{2114}{2131} \approx 99.2\% \tag{6.1}$$

- **Precisione**: misura la percentuale di email classificate correttamente come phishing

$$\text{Precisione}(p) = \frac{TP}{TP + FP} = \frac{1101}{1101 + 5} = \frac{1101}{1106} \approx 99.5\% \tag{6.2}$$

- **Richiamo**: il richiamo indica la capacità del modello di identificare correttamente le email di phishing

$$\text{Richiamo}(r) = \frac{TP}{TP + FN} = \frac{1101}{1101 + 12} = \frac{1101}{1113} \approx 98.9\% \tag{6.3}$$

- **F1-Score**: media armonica tra precisione e richiamo

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Richiamo}}{\text{Precision} + \text{Richiamo}} = 2 \times \frac{0.995 \times 0.989}{0.995 + 0.989} \approx 99.2\% \tag{6.4}$$

- **False Positive Rate**: rapporto tra falsi positivi e totale dei negativi presenti

$$\text{FPR} = \frac{FP}{FP + TN} \tag{6.5}$$

- **ROC**: grafico che mostra il trade-off tra sensibilità e specificità. L'area sotto la curva (AUC, Area Under the Curve) è una misura di performance globale del classificatore

$$\text{AUC}{=}0.99 \tag{6.6}$$

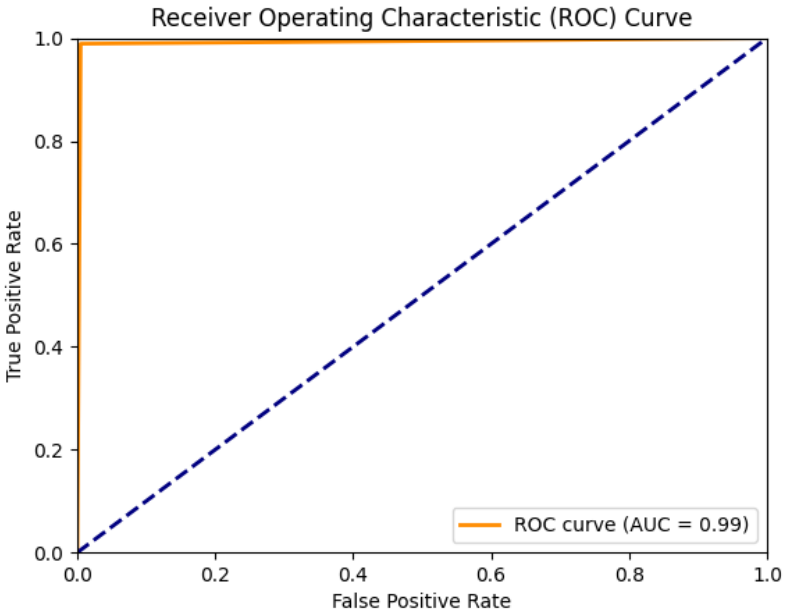


Figura 6.1: ROC

6.2 Risultati Intermedi

Durante il processo di sviluppo e addestramento del modello è stato fondamentale valutare i risultati intermedi provenienti dalle diverse componenti di analisi. Utilizzando RapidMiner è stato possibile osservare le prestazioni di ogni sottoprocesso, permettendo di identificare i punti di forza e le aree di miglioramento del modello. Questa sezione fornisce un'analisi delle prestazioni di ciascuno dei sotto processi.

6.2.1 Risultati Sottoprocesso Header

La componente di analisi dell'header si è rivelata accurata nel distinguere le email di phishing da quelle legittime.

La matrice di confusione per il sottoprocesso legato all'header è la seguente:

accuracy: 98.83% +/- 0.86% (micro average: 98.83%)

	true phishing	true no phishing	class precision
pred. phishing	1104	16	98.57%
pred. no phishing	9	1002	99.11%
class recall	99.19%	98.43%	

Le seguenti informazioni sono ricavabili dalla matrice di confusione:

- **Vero Positivi (TP)**: 1104 email di phishing correttamente identificate come tali.
- **Falsi Negativi (FN)**: 9 email di phishing erroneamente classificate come legittime.
- **Vero Negativi (TN)**: 1002 email legittime correttamente identificate.
- **Falsi Positivi (FP)**: 16 email legittime erroneamente classificate come phishing.

Inoltre, è possibile calcolare le seguenti metriche:

- **Accuratezza**:

$$\text{Accuratezza} = \frac{TP + TN}{TP + FP + TN + FN} = \frac{1104 + 1002}{1104 + 1002 + 16 + 9} \approx 98.83\% \quad (6.7)$$

- **Precisione**:
$$\text{Precisione}(p) = \frac{TP}{TP + FP} = \frac{1104}{1104 + 16} \approx 98.6\% \tag{6.8}$$
- **Richiamo**:
$$\text{Richiamo}(r) = \frac{TP}{TP + FN} = \frac{1104}{1104 + 9} \approx 99.2\% \tag{6.9}$$
- **F1-Score**:
$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Richiamo}}{\text{Precision} + \text{Richiamo}} = 2 \times \frac{0.986 \times 0.992}{0.986 + 0.992} \approx 98.9\% \tag{6.10}$$

6.2.2 Risultati Sottoprocesso Testo

L'analisi sul testo si è dimostrata più complessa rispetto all'header, evidenziando una maggiore difficoltà nel distinguere tra email di phishing e legittime.

La matrice di confusione ha riportato i seguenti risultati:

accuracy: 80.34% +/- 3.05% (micro average: 80.34%)

	true phishing	true no phishing	class precision
pred. phishing	914	220	80.60%
pred. no phishing	199	798	80.04%
class recall	82.12%	78.39%	

Dalla matrice di confusione è possibile evincere i seguenti risultati:

- **Vero Positivi (TP)**: 914 email di phishing correttamente identificate come tali.
- **Falsi Negativi (FN)**: 199 email di phishing erroneamente classificate come legittime.
- **Vero Negativi (TN)**: 220 email legittime correttamente identificate.
- **Falsi Positivi (FP)**: 798 email legittime erroneamente classificate come phishing.

Le metriche di valutazione, invece, assumono i seguenti valori:

- **Accuratezza**:
$$\text{Accuratezza} = \frac{TP + TN}{TP + FP + TN + FN} = \frac{914 + 798}{914 + 798 + 220 + 199} \approx 80.34\% \tag{6.11}$$

- **Precisione**:
$$\text{Precisione}(p) = \frac{TP}{TP + FP} = \frac{914}{914 + 220} \approx 80.6\% \tag{6.12}$$
- **Richiamo**:
$$\text{Richiamo}(r) = \frac{TP}{TP + FN} = \frac{914}{914 + 199} \approx 82.1\% \tag{6.13}$$
- **F1-Score**:
$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Richiamo}}{\text{Precision} + \text{Richiamo}} = 2 \times \frac{0.806 \times 0.821}{0.806 + 0.821} \approx 81.3\% \tag{6.14}$$

6.2.3 Risultati Sottoprocesso Allegati

L'analisi degli allegati ha mostrato una performance variabile, con la seguente matrice di confusione:

accuracy: 91.67% +/- 8.38% (micro average: 91.67%)

	true phishing	true no phishing	class precision
pred. phishing	69	6	92.00%
pred. no phishing	9	96	91.43%
class recall	88.46%	94.12%	

I risultati evidenziati dalla matrice di confusione sono i seguenti:

- **Vero Positivi (TP)**: 69 email di phishing correttamente identificate come tali.
- **Falsi Negativi (FN)**: 9 email di phishing erroneamente classificate come legittime.
- **Vero Negativi (TN)**: 96 email legittime correttamente identificate.
- **Falsi Positivi (FP)**: 6 email legittime erroneamente classificate come phishing.

I valori delle metriche di valutazione, invece, sono calcolati qui di seguito:

- **Accuratezza**:
$$\text{Accuratezza} = \frac{TP + TN}{TP + FP + TN + FN} = \frac{69 + 96}{69 + 96 + 6 + 9} \approx 91.67\% \tag{6.15}$$
- **Precisione**:
$$\text{Precisione}(p) = \frac{TP}{TP + FP} = \frac{69}{69 + 6} \approx 92.0\% \tag{6.16}$$

- **Richiamo**:

$$\text{Richiamo}(r) = \frac{TP}{TP + FN} = \frac{69}{69 + 9} \approx 88.5\% \tag{6.17}$$

- **F1-Score**:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Richiamo}}{\text{Precision} + \text{Richiamo}} = 2 \times \frac{0.920 \times 0.885}{0.920 + 0.885} \approx 90.2\% \tag{6.18}$$

6.3 Interpretazione dei Risultati

L'analisi delle metriche mostra che il modello di classificazione sviluppato ha ottenuto risultati positivi. Un'accuratezza del 99.2% indica che il modello riesce a classificare correttamente la maggioranza delle email, riducendo al minimo gli errori.

Inoltre, una precisione del 99.5% e 98.9% è indicativa sull'efficacia del sistema nel ridurre i falsi positivi. Solo 5 email legittime, infatti, sono classificate come phishing. Un falso positivo si verifica quando una mail legittima viene erroneamente classificata come phishing, questo comporta che venga persa o definita potenzialmente pericolosa.

I falsi negativi rappresentano una percentuale molto bassa rispetto al numero totale di email di phishing ma sono comunque rilevanti perché rappresentano un elevato rischio per l'utente.

Considerando i risultati dei modelli intermedi, quello basato sull'header ha un'elevata capacità di rilevare le email di phishing, con un basso numero di falsi negativi e un modesto numero di falsi positivi. Questo è un risultato accettabile poichè è ritenuto più rischioso etichettare una email di phishing come legittima che classificare erroneamente un'email legittima come phishing.

Il modello che analizza il testo ha ottenuto un numero significativamente più elevato di falsi negativi e falsi positivi. In particolare, un'identificazione maggiore di falsi positivi implica che un numero significativo di email legittime viene erroneamente considerato pericoloso. Questo indica che le caratteristiche linguistiche del testo possono sovrapporsi tra le email di phishing e quelle legittime, rendendo più difficile la classificazione.

L'analisi degli allegati ha mostrato una buona precisione nel rilevare le email di phishing con un contenuto numero di falsi positivi e falsi negativi. Tuttavia, è stata rilevata un'incertezza di 8.38% dovuta al ridotto numero di allegati presenti nel dataset.

L'approfondimento dei risultati dei tre componenti evidenzia come il processo di classificazione basato sull'header sia il più accurato e stabile. Al contrario, il modello basato su testo è il più esposto a errori, probabilmente a causa della varietà di stili linguistici che possono confondere l'algoritmo. L'analisi degli allegati, seppur efficace, è applicabile solo alle email contenenti effettivamente gli allegati, riguardando, quindi, solo una minima parte del dataset.

Capitolo 7

CONCLUSIONI

Il presente lavoro ha avuto come obiettivo principale l'analisi e la costruzione di un modello di machine learning per il rilevamento di email di phishing, evidenziando l'importanza di una combinazione di approcci per migliorare l'accuratezza e l'affidabilità del sistema.

Nel corso dello studio, i modelli sono stati valutati utilizzando diverse metriche tra cui accuratezza, precisione, richiamo e F1-score, al fine di comprendere a fondo le performance di ciascun classificatore.

L'analisi dei risultati ha dimostrato che, mentre singoli modelli possono avere buone performance, l'adozione di un approccio che integri le previsioni di diversi modelli porta a risultati significativamente migliori. Mentre l'analisi dell'header ha dimostrato prestazioni eccezionali, i sottoprocessi relativi al testo e agli allegati hanno evidenziato aree di vulnerabilità che possono essere colmate con un approccio integrato.

L'unione di diversi classificatori ha consentito di ottenere un modello finale che, non solo migliora l'accuratezza complessiva, ma riduce il numero di falsi positivi. Esaminando la matrice di confusione, infatti, è stato possibile calcolare l'accuratezza pari al 99.2% e notare la presenza di 5 falsi positivi.

Un basso numero di falsi positivi è un ottimo risultato in un contesto reale, poiché significa che il modello non classifica erroneamente molte email legittime come phishing, evitando, così, di interferire con le normali comunicazioni. In caso contrario, le email relative a comunicazioni aziendali, messaggi personali o notifiche andrebbero perse o classificate come potenzialmente pericolose. In senso pratico, ciò potrebbe significare che un'email legittima viene spostata nella cartella spam o scartata del tutto, generando problemi di comunicazione.

Un sistema con un basso tasso di falsi positivi significa, oltre a migliorare l'esperienza utente, una riduzione di lavoro manuale per la revisione e, di conseguenza un risparmio in

termini di tempo e risorse.

L'utilizzo di RapidMiner ha permesso di implementare il flusso di lavoro ottimizzando i modelli e testando diverse configurazioni. In tal modo, è stato possibile ottimizzare le prestazioni del sistema per mezzo delle funzionalità della griglia di ricerca e cross validation che ha permesso di confrontare le performance dei vari algoritmi. Inoltre, grazie alla sua interfaccia, RapidMiner ha permesso di visualizzare i vari passaggi del flusso di lavoro e di monitorare il processo di addestramento in tempo reale, facilitando l'identificazione di problemi quali l'overfitting o variazioni inattese delle prestazioni.

Un ulteriore contributo di questo lavoro è stato l'uso di tecniche di Natural Language Processing per l'analisi del testo delle email. In particolare, la pre-elaborazione del corpo dei messaggi attraverso l'uso di librerie come BeautifulSoup per la pulizia del testo e le funzioni messe a disposizione da RapidMiner, ha migliorato significativamente l'estrazione delle features e, quindi, le capacità del modello che risulta essere più robusto e affidabile.

L'uso del Decision Tree è risultato particolarmente utile non solo per la capacità di classificare correttamente le email, ma anche per la rappresentazione grafica e intuitiva. Questo approccio visivo permette di comprendere facilmente il percorso decisionale seguito dal modello per distinguere tra email legittime e di phishing, evidenziando quali features abbiano influenzato le scelte. La visualizzazione del Decision Tree ha permesso di identificare quali features fossero realmente rilevanti e quali responsabili di eventuali errori o falsi positivi. La possibilità di esplorare visivamente i rami dell'albero in RapidMiner ha aiutato a individuare le condizioni che portavano a previsioni errate e a sviluppare strategie per migliorare il modello finale.

I risultati ottenuti hanno rivelato non solo l'efficacia del modello, ma anche le aree che richiedono ulteriori miglioramenti.

I risultati dell'analisi del testo, con un'accuratezza del 80,34%, mostrano che il modello ha una performance inferiore rispetto all'analisi degli header con un numero significativo di falsi positivi. È possibile che i modelli di linguaggio utilizzati non siano stati adeguatamente addestrati su un campione rappresentativo e variegato di email di phishing, il che potrebbe aver limitato la capacità di generalizzazione del modello stesso.

Nonostante i risultati mostrino che il modello sia riuscito a identificare correttamente le minacce presentate dagli allegati, la scarsità di esempi potrebbe aver influenzato negativamente la capacità del classificatore di generalizzare in contesti reali, rendendo meno affidabili le previsioni effettuate.

L'ottimizzazione delle tecniche di preprocessing e un addestramento su dataset più ampi e diversificati potrebbero contribuire a migliorare la performance complessiva del modello ed evidenziare ulteriori punti deboli. In sinesi, sebbene il modello dimostri buone performance nella rilevazione di phishing, il miglioramento dell'analisi del testo e degli allegati è cruciale per garantire un sistema robusto e affidabile.

In futuro, i miglioramenti del modello dovrebbero concentrarsi sull'ottimizzazione delle componenti più deboli, quali l'analisi del testo e degli allegati. Una soluzione potrebbe essere quella di testare ulteriori algoritmi di NLP per il testo o utilizzare modelli di machine learning più potenti, a discapito delle risorse necessarie. Per quanto riguarda gli allegati, è necessario studiare il modello in presenza di un maggior numero di elementi per poter ottenere risultati più precisi e affidabili nell'interpretazione.

Infine, è possibile testare il modello su dataset più ampi, costituiti da fonti diverse in modo da rendere il campione più eterogeneo e approfondirne il comportamento su un maggior numero di elementi.

Un'area promettente di sviluppo futuro è l'integrazione di tecnologie di Generative AI. I modelli generativi, come quelli basati su reti neurali, potrebbero essere utilizzati per simulare attacchi di phishing più sofisticati e diversificati. Questo permetterebbe di addestrare il sistema su minacce simulate ma realistiche, aumentando la sua capacità di riconoscere anche le varianti di phishing meno comuni o inedite.

In conclusione, sebbene i filtri automatici e i modelli di machine learning possano ridurre drasticamente il numero di email di phishing che raggiungono gli utenti finali, affidarsi esclusivamente a tali strumenti non è sufficiente. È necessario, infatti, un approccio che preveda il continuo miglioramento degli algoritmi di rilevamento e un efficace programma di formazione che sensibilizzi gli utenti sui rischi del phishing in modo da ridurre il rischio dovuto alle minacce che riescano a eludere i controlli automatici.

Bibliografia

[1] *Phishing — cert-agid.gov.it*. https://cert-agid.gov.it/glossario/phishing/.

[2] Koceilah Rekouche. «Early Phishing». In: *CoRR* abs/1106.4692 (2011). eprint: 1106.4692. URL: http://arxiv.org/abs/1106.4692.

[3] *APWG | Phishing Activity Trends Reports — apwg.org*. https://apwg.org/trendsreports/.

[4] *INTERPOL report shows alarming rate of cyberattacks during COVID-19 — interpol.int*. https://www.interpol.int/en/News-and-Events/News/2020/INTERPOL-report-shows-alarming-rate-of-cyberattacks-during-COVID-19.

[5] Sophos. *Phishing Insights 2021*. https://assets.sophos.com/X24WTUEQ/at/2x7wmj8mf69r86fv3bgwc4tm/sophos-phishing-insights-2021-report.pdf.

[6] *Cyber Security Breaches Survey 2021 — gov.uk*. https://www.gov.uk/government/statistics/cyber-security-breaches-survey-2021.

[7] https://web.archive.org/web/20110806061136/http://www.microsoft.com/canada/athome/security/email/spear_phishing.mspx.

[8] Debbie Stephenson. *Spear Phishing: Who's Getting Caught? [Infographic] — firmex.com*. https://www.firmex.com/resources/infographics/spear-phishing-whos-getting-caught/.

[9] *What Are the Different Types of Phishing? — trendmicro.com*. https://www.trendmicro.com/en_us/what-is/phishing/types-of-phishing.html.

[10] *What Is Phishing? | Microsoft Security — microsoft.com*. https://www.microsoft.com/en-us/security/business/security-101/what-is-phishing.

[11] *What is email spoofing? A complete guide — us.norton.com*. https://us.norton.com/blog/online-scams/email-spoofing.

[12] Ankit Kumar Jain e BB Gupta. «A survey of phishing attack techniques, defence mechanisms and open research challenges». In: *Enterprise Information Systems* 16.4 (2022), pp. 527–565.

[13] Biju Issac, Raymond Chiong e Seibu Mary Jacob. «Analysis of phishing attacks and countermeasures». In: *arXiv preprint arXiv:1410.4672* (2014).

[14] Adam Oest, Penghui Zhang, Brad Wardman, Eric Nunes, Jakub Burgis, Ali Zand, Kurt Thomas, Adam Doupé e Gail-Joon Ahn. «Sunrise to sunset: Analyzing the end-to-end life cycle and effectiveness of phishing attacks at scale». In: *29th {USENIX} Security Symposium ({USENIX} Security 20)*. 2020.

[15] Brent Lang. *WikiLeaks Publishes Thousands of Hacked Sony Documents — variety.com*. https://variety.com/2015/film/news/wikileaks-sony-hack-1201473964/.

[16] *The Sony Pictures Breach: A Deep Dive into a Landmark Cyber Attack - Sep 15, 2023 — frameworksec.com*. https://www.frameworksec.com/post/the-sony-pictures-breach-a-deep-dive-into-a-landmark-cyber-attack.

[17] *eBay Sees Revenue Decline Due to Breach — bankinfosecurity.com*. https://www.bankinfosecurity.com/ebay-sees-fewer-sales-due-to-breach-a-7074.

[18] Shellmates Club. *Yahoo Data Breach: An In-Depth Analysis of One of the Most Significant Data Breaches in History — shellmates.medium.com*. https://shellmates.medium.com/yahoo-data-breach-an-in-depth-analysis-of-one-of-the-most-significant-data-breaches-in-history-ba5b46be560b.

[19] NB Harikrishnan, R Vinayakumar e KP Soman. «A machine learning approach towards phishing email detection». In: *Proceedings of the anti-phishing pilot at ACM international workshop on security and privacy analytics (IWSPA AP)*. Vol. 2013. 2018, pp. 455–468.

[20] Panagiotis Bountakas, Konstantinos Koutroumpouchos e Christos Xenakis. «A comparison of natural language processing and machine learning methods for phishing email detection». In: *Proceedings of the 16th International Conference on Availability, Reliability and Security*. 2021, pp. 1–12.

[21] Trivikram Muralidharan e Nir Nissim. «Improving malicious email detection through novel designated deep-learning architectures utilizing entire email». In: *Neural Networks* 157 (2023), pp. 257–279.

[22] Issam El Naqa e Martin J Murphy. *What is machine learning?* Springer, 2015.

[23] Leo Breiman. «Random forests». In: *Machine learning* 45 (2001), pp. 5–32.

[24] Eder S Gualberto, Rafael T De Sousa, P De B Thiago, João Paulo CL Da Costa e Cláudio G Duque. «From feature engineering and topics models to enhanced prediction rates in phishing detection». In: *Ieee Access* 8 (2020), pp. 76368–76385.

[25] X Dong, Z Yu, W Cao, Y Shi e Q Ma. «A survey on ensemble learning. Frontiers of Computer Science». In: (2020).

[26] Ietezaz Ul Hassan, Raja Hashim Ali, Zain Ul Abideen, Talha Ali Khan e Rand Kouatly. «Significance of machine learning for detection of malicious websites on an unbalanced dataset». In: *Digital* 2.4 (2022), pp. 501–519.

[27] *Cos'è l'NLP (elaborazione del linguaggio naturale)? | IBM — ibm.com*. https://www.ibm.com/it-it/topics/natural-language-processing.

[28] Panagiotis Bountakas, Konstantinos Koutroumpouchos e Christos Xenakis. «A comparison of natural language processing and machine learning methods for phishing email detection». In: *Proceedings of the 16th International Conference on Availability, Reliability and Security*. 2021, pp. 1–12.

[29] Jose Nazario. URL: `https://monkey.org/~jose/phishing/`.