



UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Laurea Triennale in Informatica

TESI DI LAUREA

# Quantum Machine Learning per un Sistema di Malware Detection

RELATORE

**Prof. Fabio Palomba**

Università degli studi di Salerno

CANDIDATO

**Luca Contrastø**

Matricola: 0512106490

Anno Accademico 2021-2022

*INSERIRE QUI UNA DEDICA O UNA CITAZIONE*

## Sommario

Il campo della Cybersecurity sta avendo uno sviluppo esponenziale negli ultimi anni, sia per ragioni politiche e sociali che per ragioni economiche. Sono infatti all'ordine del giorno le notizie che riguardano furto di dati, hacking o spionaggio; è necessario quindi da parte dei ricercatori e professionisti della Cybersecurity un lavoro continuo di aggiornamento e sviluppo delle tecniche volte a difendersi da questi attacchi. La causa principale è identificabile tecnicamente nella diffusione di Malware all'interno dei sistemi, gli attaccanti infatti attuano diverse tecniche, tra cui l'ingegneria sociale, per far in modo di introdurre del codice malevolo all'interno del sistema della vittima. Vista la necessità di studi sempre più approfonditi ed innovativi per il Malware Detection il seguente elaborato ha lo scopo di ampliare le sperimentazioni effettuate in questo ambito con una proposta diversa dalle tecniche più utilizzate. Negli ultimi anni il Machine Learning si è affermato nell'Informatica come l'insieme di algoritmi e tecniche capaci di fornire funzionalità intelligenti ad un prodotto software, sono infatti numerose le sue applicazioni tra cui troviamo anche il Malware Detection. La particolarità dell'elaborato e della sperimentazione associata è l'utilizzo del Quantum Machine Learning per il problema del Malware Detection, come anticipato il Machine Learning classico sta avendo ottimi risultati in tutti gli ambiti dell'Informatica tra cui anche il Malware Detection, il Machine Learning quantistico invece rappresenta una nuova frontiera dell'Informatica nata dall'unione del Machine Learning con il Quantum Computing. Gli algoritmi sviluppati saranno sia classici che quantistici e sarà analizzata la bontà di ognuno in termini di performance e tempo per il problema in analisi. Differentemente dalle aspettative, la promessa di un miglioramento netto degli algoritmi quantistici rispetto ai classici non risulta verificata per questo particolare tipo di applicazione.

---

## Indice

---

<b>Indice</b>	<b>ii</b>
<b>Elenco delle figure</b>	<b>iv</b>
<b>Elenco delle tabelle</b>	<b>v</b>
<b>1 Introduzione</b>	<b>1</b>
1.1 Motivazioni e Obiettivi . . . . .	1
1.2 Risultati . . . . .	3
1.3 Struttura della tesi . . . . .	3
<b>2 Background e Stato dell'arte</b>	<b>5</b>
2.1 Sicurezza di sistemi software . . . . .	5
2.1.1 Costo Data Breach nel 2021 . . . . .	5
2.1.2 Tipi di attacchi informatici . . . . .	6
2.1.3 Classificazione di Malware . . . . .	9
2.1.4 Gestione della sicurezza informatica . . . . .	12
2.2 Machine learning per la Cybersecurity . . . . .	16
2.2.1 Machine Learning . . . . .	16
2.2.2 Classificazione delle tecniche di Machine Learning . . . . .	17
2.2.3 Applicazioni del Machine Learning per la Cybersecurity . . . . .	19
2.2.4 Definizione dei principali algoritmi di Machine Learning . . . . .	20
2.3 Quantum Machine Learning . . . . .	23

2.3.1	Quantum Computing . . . . .	23
2.4	Stato dell'arte . . . . .	27
2.4.1	Malware Detection . . . . .	27
2.4.2	Machine Learning per Malware Detection . . . . .	30
2.4.3	Algoritmi di Quantum Machine Learning . . . . .	31
<b>3</b>	<b>Confronto tra Quantum Classifier</b>	<b>36</b>
3.1	Obiettivi . . . . .	36
3.2	Dataset utilizzato per lo studio . . . . .	37
3.2.1	Formato PE . . . . .	38
3.2.2	Descrizione delle Feature . . . . .	38
3.2.3	Gestione delle stringhe . . . . .	40
3.3	Definizione della pipeline . . . . .	41
3.3.1	Feature engineering . . . . .	42
3.3.2	Definizione dei modelli . . . . .	44
3.3.3	Valutazione . . . . .	45
3.4	Sperimentazione . . . . .	47
3.4.1	Metriche utilizzate . . . . .	47
3.4.2	Algoritmi classici . . . . .	48
<b>4</b>	<b>Risultati Ottenuti</b>	<b>49</b>
<b>5</b>	<b>Conclusioni</b>	<b>53</b>
<b>Bibliografia</b>		<b>55</b>
<b>Ringraziamenti</b>		<b>59</b>

---

## Elenco delle figure

---

2.1	Costo totale medio dei Data Breach . . . . .	6
2.2	Costo totale medio dei Data Breach diviso in categorie . . . . .	7
2.3	Classificazione degli attacchi . . . . .	7
2.4	Classificazione di Malware . . . . .	10
2.5	Classificazione degli algoritmi di machine learning . . . . .	17
2.6	Struttura di un qubit, sovrapposizione e correlazione quantistica . . . . .	24
2.7	Sfera di Bloch . . . . .	25
2.8	Metodi di rilevamento dei malware . . . . .	27
3.1	Struttura di un file PE . . . . .	37
3.2	Confronto tra Feature Extraction e Feature Selection . . . . .	43
3.3	Esempio matrice di confusione . . . . .	47
3.4	Calcolo metriche da matrice di confusione . . . . .	48
4.1	Grafico rappresentante l'accuracy dei modelli utilizzati . . . . .	50

---

## Elenco delle tabelle

---

2.1	Risultati della QSVM sul dataset Breast Cancer Wisconsin . . . . .	33
2.2	Risultati della QSVM sul dataset Wine . . . . .	33
2.3	Risultati ottenuti sul dataset ClaMP . . . . .	34
2.4	Risultati ottenuti sul dataset ReVeal . . . . .	34
4.1	Risultati ottenuti . . . . .	49

# CAPITOLO 1

---

## Introduzione

---

### 1.1 Motivazioni e Obiettivi

Dal rapporto Clusit 2022 [5] è emerso che nel 2021 gli attacchi informatici sono aumentati nel mondo del 10% rispetto all'anno precedente e mostrano un'importanza sempre più elevata, sul fronte economico sono state infatti stimate perdite nell'ordine di 6 trilioni di dollari, una cifra difficile da immaginare che è pari a 4 volte il PIL italiano. A livello geografico gli attacchi classificati dai ricercatori Clusit si sono verificati nel 45% nel continente americano (leggermente in calo rispetto al 2020), dati preoccupanti invece per l'Europa che registra un aumento del 5% rispetto all'anno precedente salendo al 21% e per l'Asia che passa dal 10% nel 2020 al 12% nel 2021. I 2049 attacchi gravi rilevati nel corso del 2021 costituiscono una cifra importante, ma molto più preoccupante è l'impatto che questi hanno comportato alle vittime, si possono infatti valutare i danni di ogni attacco facendo riferimento a 4 parametri: danno economico, danno di immagine, conseguenze sociali e implicazioni a livello geopolitico. Quest'ultimo parametro nel periodo storico che stiamo vivendo assume particolare rilevanza, sono infatti noti gli attacchi ad enti governativi e infrastrutture di tutto il mondo oltre al conflitto tra Russia e Ucraina che costituisce una cyberwar senza precedenti nella storia; dalle rilevazioni dei ricercatori infatti il 79% degli attacchi ha raggiunto la categoria di impatto elevato, dato preoccupante rispetto al 50% dell'anno precedente.

Osservando la tipologia di attaccanti quelli volti al cybercrime rappresentano l'86% della totalità, dato in salita del 5% rispetto all'anno precedente, per la restante parte l'11% è costi-

tuito da attività di spionaggio e le restanti quote marginali da information warfare<sup>1</sup> e cyber attivismo.

L'aumento del cybercrime e la tipologia di minacce diffuse, oltre allo scopo per le quali vengono rilasciate, evidenziano dinamiche di natura organizzata tra i cybercriminali. Ad avvalorare questa ipotesi vi è il cambiamento di obiettivi che gli attaccanti perseguono a partire dalle vittime scelte per gli attacchi. Queste infatti non sono più multiple targets ma vengono selezionate in particolari settori di interesse, migliorando l'efficienza e i danni causati in caso di successo dell'attacco, delineando un cambiamento di paradigma che ha come scopo l'ottenimento di vantaggi più consistenti colpendo vittime precise. Oltre alla scelta dei bersagli studiata ciò che fa ipotizzare la nascita di organizzazioni cybercriminali è la messa a disposizione sempre più rilevante in ambienti quali il dark web di servizi come il Ransomware-as-a-Service, che consente a chiunque di noleggiare un kit di tool che gli permette di effettuare attacchi mirati senza disporre delle competenze necessarie per poter effettuare un attacco del genere. A tal proposito è doveroso rilevare come i malware, ed in particolare i ransomware<sup>2</sup>, costituiscano le cause principali degli attacchi con il 41% [5].

A tal proposito è proprio nell'ambito del Malware Detection che la sperimentazione di questo elaborato ha lo scopo di inserirsi, infatti considerata la necessità di ampliare e migliorare questo campo della Cybersecurity viene analizzata un'alternativa innovativa al classico Malware Detection che prevede l'utilizzo di algoritmi di Machine Learning quantistico. Al di là della proposta quantistica della sperimentazione il Machine Learning, o apprendimento automatico, risulta essere un mezzo valido nel campo del Malware Detection poiché, come sarà anche analizzato successivamente, riesce ad apprendere correttamente e in modo efficiente i pattern esistenti tra i dati, riuscendo ad effettuare predizioni in grado di valorizzare un prodotto software andando ad aggiungere funzionalità intelligenti.

Avvalorata la potenza del Machine Learning in ambito Malware Detection, estendibile ad ogni campo dell'Informatica considerando la molteplicità di applicazioni intelligenti concrete usate tutti i giorni, si può pensare ad una proposta innovativa in questo ambito ossia, come anticipato precedentemente, il Quantum Machine Learning. Innanzitutto il campo del Quantum Computing<sup>3</sup> risulta essere un nuovo fronte dell'Informatica e della Fisica su cui fare ricerche al fine di migliorare gli algoritmi e i sistemi quantistici attuali che hanno la

---

<sup>1</sup>Letteralmente guerra di informazioni, si identifica nell'approccio ai conflitti tra Stati e/o organizzazioni tramite l'uso di informazioni in tutte le forme e a qualsiasi scopo per assicurarsi un vantaggio militare o politico

<sup>2</sup>Tipo di Malware che blocca l'accesso ai dati di un utente in cambio di un riscatto

<sup>3</sup>Tecnologia emergente che sfrutta la meccanica quantistica per approcciare in modo diverso la rappresentazione dell'informazione promettendo velocità e potenza di calcolo senza eguali

promessa di svoltare totalmente il calcolo computazionale fornendo una potenza di calcolo senza eguali. Il Quantum Machine Learning risulta essere l'unione del Quantum Computing con il Machine Learning che concretamente identifica l'insieme di algoritmi che hanno lo stesso scopo e funzionalità degli algoritmi di Machine Learning ma sono definiti appunto nell'ambito del Quantum Computing. Come sarà descritto e analizzato per poter rendere possibile una predizione di questo tipo è necessario convertire i dati classici in dati quantistici ed effettuare il training dei modelli su sistemi quantistici remoti.

## 1.2 Risultati

La promessa del Quantum Machine Learning è una maggiore potenza di calcolo rispetto agli algoritmi classici, di conseguenza poiché nella sperimentazione sono stati definiti modelli sia classici che quantistici, per avere una visione chiara di quanti e quali siano modelli applicabili ed efficienti in ambito Malware Detection, è ragionevole aspettarsi la superiorità degli algoritmi quantistici sui classici.

Come sarà analizzato nello Stato dell'Arte e discusso in modo più approfondito nel Capitolo 4 purtroppo i sistemi quantistici sia hardware che software attuali non sono ancora pronti per far apprezzare la potenza di questo nuovo paradigma computazionale. Nonostante ciò gli studi e le ricerche effettuate in questo ambito, compreso questo elaborato supportato dalla sperimentazione effettuata, delineano un miglioramento costante che, accompagnato dagli aggiornamenti promessi dalle aziende proprietarie dei sistemi quantistici, potrebbero portare nei successivi anni il Quantum Computing ad una platea di utenti sempre più ampia per gli utilizzi più disparati.

## 1.3 Struttura della tesi

L'elaborato sarà strutturato in diversi capitoli ognuno con lo scopo di presentare ed descrivere in modo approfondito diversi aspetti della sperimentazione effettuata. Nel Capitolo 2 sarà analizzato il Background e lo Stato dell'Arte ossia tutte le conoscenze ed informazioni raccolte in letteratura necessarie per capire tutti gli aspetti e le logiche impiegate nella sperimentazione, oltre ai lavori simili già effettuati e pubblicati in questo ambito su cui poter effettuare confronti o considerarli punti di partenza per uno studio ulteriore. Nel Capitolo 3 si entra nel fulcro dell'elaborato, viene infatti descritta la pipeline di Machine Learning seguita e tutte le tecniche di Preprocessing utilizzate oltre ad una spiegazione dei modelli quantistici

impiegati. Successivamente vengono presentati e discussi i Risultati Ottenuti nel Capitolo 4 presentando i possibili scenari e cause dei problemi riscontrati o degli ottimi risultati ottenuti. Infine nel Capitolo 5 si giunge alle Conclusioni dove si considerano tutti i dati raccolti e analizzati tramite la sperimentazione effettuata e si commentano i possibili sviluppi futuri.

# CAPITOLO 2

---

## Background e Stato dell'arte

---

### 2.1 Sicurezza di sistemi software

La cybersecurity, anche conosciuta come sicurezza informatica, è un settore dell'informatica che raggruppa tecnologie, metodi e protocolli che hanno lo scopo di proteggere i sistemi informatici dal punto di vista strutturale, oltre ad offrire privatezza e integrità di dati e risorse dei sistemi.

L'espansione di questo ramo dell'informatica è incrementata negli anni in modo proporzionale allo sviluppo di nuove tecnologie: con la crescita delle reti, la nascita dell'internet delle cose(IoT), i big data e il cloud computing sono stati definiti ulteriori sistemi hardware e software che di contro hanno offerto nuove potenziali minacce alla sicurezza.

In questa sezione saranno prima analizzati i costi che le aziende hanno dovuto affrontare per problemi di cybersecurity e successivamente saranno illustrati i principali tipi di attacchi, concludendo illustrando i principali metodi e pratiche da seguire per avere un buon grado di sicurezza per aziende e privati.

#### 2.1.1 Costo Data Breach nel 2021

Per avere una visione chiara dei problemi che gli attacchi informatici causano ad aziende ed istituzioni si può visionare il report stilato dall'IBM e dal Ponemon Institute "Cost of a Data Breach Report 2021" [22] . Quest'ultimo presenta un'analisi chiara dei costi che le aziende hanno dovuto affrontare nel 2021 a causa dei data breach, ossia una violazione della

sicurezza di un sistema informatico nella quale dati sensibili vengono copiati, rubati o resi indisponibili con lo scopo di causare danno all’azienda che subisce l’attacco o richiedere un riscatto in cambio dei dati.

Dal report è stata stimata una spesa media per azienda di 4,24 milioni di dollari con un aumento di quasi mezzo milione rispetto all’anno precedente, per quanto riguarda l’Italia il costo medio per azienda è stato di 3.61 milioni di euro.

Possiamo osservare dalla Figura 2.1[22] un grafico che mostra l’aumento della spesa media per azienda dal 2015 al 2021 a livello globale, diverse stime illustrano come purtroppo questa spesa sia destinata a salire.



**Figura 2.1:** Costo totale medio dei Data Breach

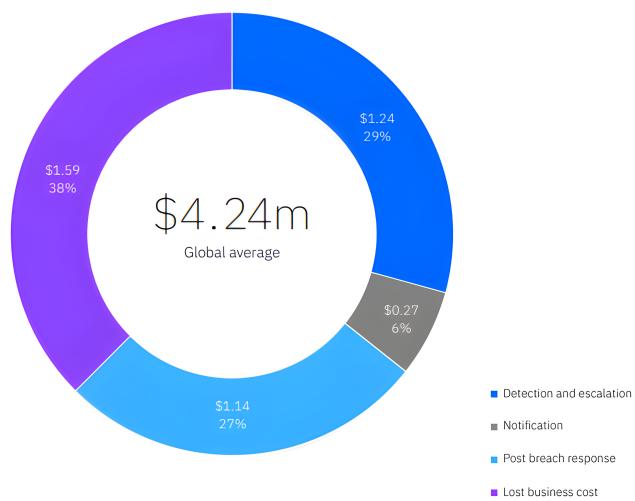
Il costo medio per azienda può essere diviso in quattro categorie che rappresentano le spese concrete che l’azienda deve affrontare per risolvere una violazione qualora si presenti, si osserva nella Figura 2.2[22] che la categoria predominante, quindi più dispendiosa, è quella delle possibili attività commerciali perse. In questa categoria infatti rientrano costi relativi all’interruzione dei servizi forniti dall’azienda, ai clienti persi e ai possibili clienti non acquisiti.

### 2.1.2 Tipi di attacchi informatici

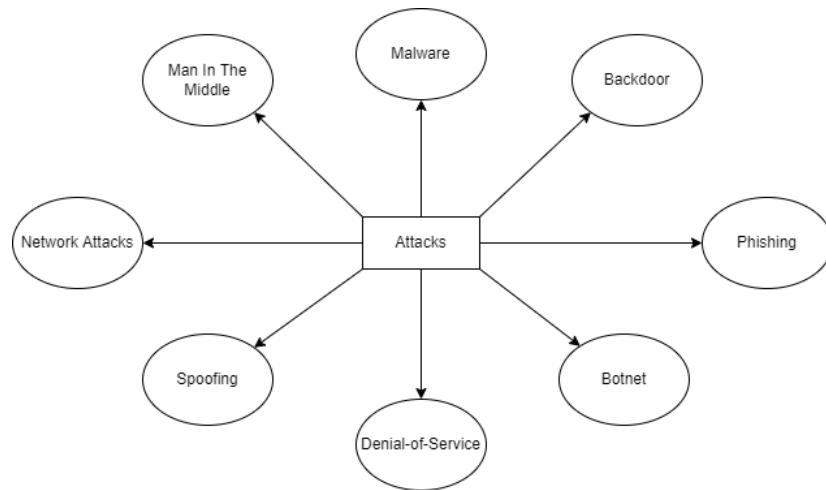
Saranno illustrati di seguito una serie attacchi mossi dai cybercriminali nel caso in cui un sistema informatico presenti vulnerabilità sfruttabili per causare danni al sistema. Della classificazione degli attacchi, basata anche sui dati collezionati da OWASP [13] e presentata in Figura 2.3, saranno descritti solo quelli più utilizzati o utili ai fini della tesi.

#### Attacchi alla rete

Questo categoria di attacchi può essere di tipo passivo o attivo: in quelli di tipo passivo l’attaccante può intercettare e decodificare i pacchetti trasmessi in rete per analizzarne il



**Figura 2.2:** Costo totale medio dei Data Breach diviso in categorie



**Figura 2.3:** Classificazione degli attacchi

contenuto o i metadati allo scopo di ricavare informazioni a lui utili; negli attacchi di tipo attivo invece si ha come obiettivo il danneggiamento o l'interruzione di un'infrastruttura di rete tramite l'inserimento di malware.

### Attacchi Man In The Middle

Una particolare categoria di attacco alla rete è l'attacco Man In The Middle, questo prevede che l'attaccante sia posizionato nel mezzo della comunicazione tra due endpoint, quindi ogni pacchetto inviato sarà ricevuto ed inoltrato dall'attaccante che avrà la possibilità di interrompere o alterare la comunicazione.

### Attacchi Malware

Un malware fa riferimento a diverse categorie di software malevoli: trojan, ransomware, spyware, worm e altri(vedi 2.1.3); tutti creati e utilizzati con lo scopo di danneggiare o alterare il comportamento di un sistema oppure compiere diverse operazioni sui dati. I data breach descritti precedentemente si verificano spesso tramite l'iniezione di un ransomware nel sistema di un'azienda. Quest'ultimo è un software che critta i dati e solo l'attaccante possiede la chiave per decriptarli, viene quindi chiesto un riscatto alla vittima per poter accedere nuovamente ai dati rubati.

### Botnet

Le botnet sono delle reti di dispositivi infettati da malware che possono essere utilizzate per diversi scopi come ad esempio attacchi DDoS, inviare spam o diffondere virus, il tutto senza che i veri proprietari dei dispositivi siano consapevoli. Questo tipo di attacco è oggi il principale attacco alla sicurezza dei sistemi in rete, possiamo infatti ricordare la botnet Mirai che nel 2016, al suo apice, è arrivata ad infettare oltre 600.000 dispositivi per compiere un attacco DDoS ai danni della rete internet del sud-est degli Stati Uniti.

### Backdoor

Gli attacchi tramite backdoor sono molto popolari e consistono nell'esecuzione di un malware, precedentemente immesso nel sistema tramite altre tecniche, sul dispositivo della vittima senza che essa ne sia a conoscenza. Il codice malevolo sfrutta la porta di un'applicazione aperta, o cerca di aprirne qualcuna, con lo scopo di dare l'accesso al dispositivo all'attaccante.

### Phishing

Uno degli attacchi più famosi perché il più efficiente è il phishing, questo consiste in una truffa online realizzata spesso tramite messaggi di posta elettronica dove si spinge l'utente a fornire i propri dati. Solitamente le email di phishing sono accompagnate da un link o da un documento, il primo porta ad una pagina web copia di un altro sito dove è richiesto l'accesso o l'inserimento di dati personali, il secondo invece è legato alla diffusione di malware.

### Denial-of-Service

Questo tipo di attacco ha lo scopo di esaurire le risorse di un server per renderlo inaccessibile, vengono infatti inviati molti pacchetti di richieste ad un server in un intervallo di tempo molto breve in modo da causare un crollo del sistema. Nel caso in cui le numerose richieste vengano inviate da due o più client si parla di Distributed Denial-of-Service(DDoS), talvolta le macchine che inviano le richieste appartengono ad una botnet, quindi i proprietari non sono a conoscenza dell'accaduto.

### Spoofing

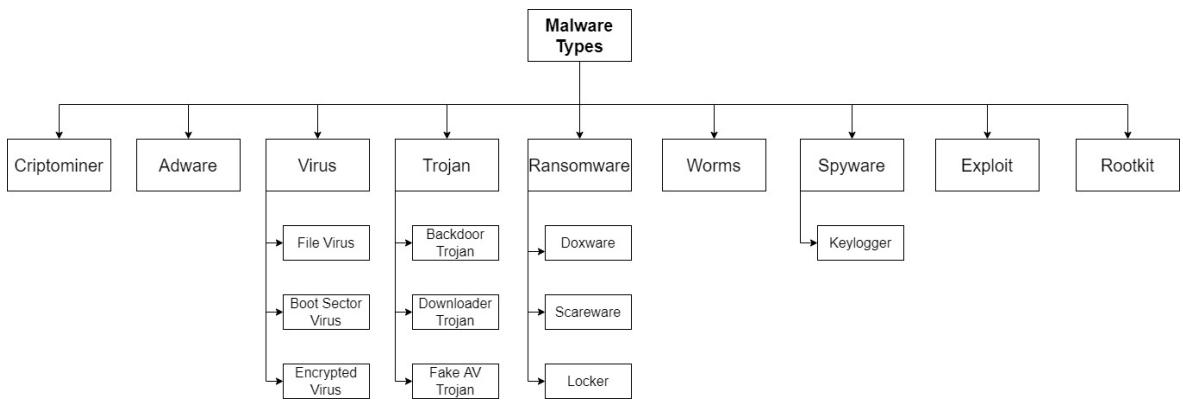
Lo Spoofing è rappresentato da tutti i metodi con cui un attaccante si finge di essere un'altra persona oppure dispositivo, esistono infatti diverse categorie di Spoofing: IP Spoofing, Email Spoofing, Man-In-The-Middle Spoofing, DNS Spoofing. Questa tecnica viene utilizzata per diversi scopi che vanno dall'ottenere un guadagno immediato dalla vittima all'avere accesso a sistemi privati[14].

Gli attacchi descritti sono i più utilizzati che nel corso degli anni sono diventati famosi, è però doveroso sottolineare come questi siano solo una parte dei possibili attacchi attuabili contro un sistema. Negli ultimi anni infatti le tecniche di attacco stanno diventando più sofisticate e strutturate, facendo uso di tecnologie come l'Intelligenza Artificiale e Machine Learning per creare programmi più precisi e robusti.

Poiché l'argomento malware è molto ampio e complesso e i codici malevoli vengono eseguiti per diversi obiettivi saranno successivamente descritti i principali tipi di malware.

#### 2.1.3 Classificazione di Malware

Come accennato in precedenza un malware è un particolare tipo di software che ha scopi deleteri per il sistema su cui è eseguito, infatti il termine stesso significa software dannoso (malicious software). Questo tipo di codice viene creato da persone che hanno intenzioni non etiche, in genere hacker black hat, per diffonderlo autonomamente oppure venderlo sul Dark Web[33]. I malware sono classificabili in diverse categorie, come si può osservare anche dalla figura 2.4, a seconda di diversi parametri e caratteristiche:



**Figura 2.4:** Classificazione di Malware

### Trojan

Questa categoria di malware, insieme ai Worm, viene utilizzata per indurre gli utenti ad installare o diffondere altro codice malevolo, sono infatti il mezzo con cui ci si può introdurre in un sistema. Un Trojan viene diffuso attraverso l'ingegneria sociale, un metodo per manipolare le persone al fine di ottenere informazioni o compiere determinate azioni, infatti si maschera sotto forma di software affidabile o viene installato da uno di questi. Questo codice può compromettere la sicurezza del sistema in diversi modi, principalmente opera sui dati infatti copia, modifica o elimina gli stessi. Un Trojan può essere a sua volta categorizzato in: Backdoor Trojan ossia un codice che installa una backdoor al fine di fornire un accesso al sistema dall'esterno, Downloader Trojan che invece scarica nuove versioni di Trojan o Adware precedentemente installati sul sistema ed infine una versione molto particolare di Trojan è il Fake AV Trojan che appare come un antivirus presentando all'utente false minacce al sistema risolvibili a pagamento[33][12].

### Virus

È il tipo di malware più diffuso, appare sotto forma di codice eseguibile e opera sul codice di altri programmi o su file. Un virus può essere di diverso tipo: File Virus è un programma che aggiunge del codice alla fine del codice di un altro programma in esecuzione per fare in modo che il controllo non torni al sistema operativo ma al programma principale, Boot Sector Virus è un codice che prima di avviare il settore per il caricamento del sistema operativo avvia altri supporti come dischi o drive USB infetti, Encrypted Virus è un virus che per non essere riconosciuto ed isolato da un antivirus è crittografato, quindi una volta superato il controllo dell'antivirus esegue l'algoritmo di decriptazione associato ed esegue il codice malevolo[33].

### Ransomware

Come descritto precedentemente il Ransomware è un malware che cifra alcuni o tutti i dati su un sistema che può anche essere bloccato, per sbloccare il sistema o l'accesso ai dati viene richiesto un riscatto tipicamente in Bitcoin. Una volta avvenuto il pagamento l'attaccante fornisce la chiave di decrittazione dei dati ed eventualmente sblocca il sistema. Un Ransomware può essere categorizzato a sua volta in: Doxware dove il codice sorgente è quello di un Ransomware classico ma l'attaccante minaccia di pubblicare i dati sensibili sottratti alla vittima, Scareware è molto simile ad un Fake AV poiché anch'esso appare come un antivirus ma blocca il sistema e richiede un pagamento per eliminare la finta minaccia, infine il Locker è un codice che blocca completamente il sistema operativo rendendo file ed applicazioni inaccessibili[33][12].

### Adware

Il nome è la contrattura di advertisement-supported software, ossia un software che genera annunci pubblicitari indesiderati ingombrando il sistema. Questo tipo di malware può anche aggiungere ulteriore codice dannoso per il sistema e per la privacy degli utenti o causare altri problemi come ad esempio rallentamento o crash del sistema[33].

### Worm

Un Worm è un codice che viene utilizzato principalmente per diffondere altri software malevoli, esso funziona generalmente inserendosi nel sistema tramite email, messaggi e file apparentemente sicuri. Successivamente il comportamento del Worm può essere differente, esso può infatti semplicemente replicarsi molte volte esaurendo le risorse del sistema oppure, oltre a replicarsi, può installare altro codice malevolo sulla macchina come ad esempio una backdoor in grado di dare accesso al sistema dall'esterno[33][12].

### Exploit

Un Exploit è una vulnerabilità del software o del sistema operativo, spesso causata da mancati aggiornamenti delle patch di sicurezza, che può far compiere operazioni indesiderate al sistema con lo scopo di rubare dati oppure compromettere le prestazioni o il funzionamento del sistema attaccato[12].

### Spyware

Questa categoria di malware, come si capisce dal nome, ha come scopo spiare e rubare le informazioni di un utente su un sistema. Generalmente l'attaccante vuole rubare dati come nomi utente e password oppure altri codici privati tramite un Keylogger, quest'ultimo è un particolare tipo di Spyware che legge tutti i tasti premuti sulla tastiera. Altri tipi di Spyware hanno invece come scopo carpire informazioni come ad esempio le pagine web visitate più spesso oppure i file aperti recentemente dall'utente e utilizzare queste informazioni per costruire una strategia di attacco[33].

### Rootkit

Un rootkit è un malware che permette all'attaccante di ottenere i permessi di amministratore del sistema operativo su cui viene eseguito e quindi infettato, questo tipo di malware è pericoloso poiché è difficile accorgersi della sua presenza infatti, nel caso sia ben progettato, nemmeno il sistema operativo riesce ad identificarlo come processo in esecuzione[12].

### Criptominer

Questo nuovo e particolare tipo di malware è nato nell'ultimo decennio a causa del crescente sviluppo ed utilizzo di criptovalute. Il programma viene utilizzato insieme ad una botnet e, nel caso in cui infetti un sistema, ne utilizza tutte le risorse disponibili per minare criptovalute e quindi portare guadagno all'attaccante[8].

Sono stati descritti i principali tipi di attacchi mossi contro un sistema che può essere aziendale o privato e i danni economici, commerciali e morali che ne derivano. Successivamente saranno introdotte le principali tecniche per difendersi dagli attacchi informatici con particolare riferimento alla gestione della sicurezza aziendale e i metodi di protezione per un privato.

#### 2.1.4 Gestione della sicurezza informatica

La sicurezza di un sistema, sia aziendale che privato, è difficile da garantire e controllare poiché le variabili da tenere in considerazione sono tante e di natura strutturale, logica e sociale. Saranno analizzati quindi i metodi più utilizzati per proteggere un sistema e le norme europee ed internazionali da seguire per avere un sistema sicuro.

Nell'ambiente molto complesso e variegato delle aziende la sicurezza dei dati e dei sistemi

viene a volte trascurata o affidata al reparto IT che però spesso deve occuparsi di altri progetti quindi non viene data la giusta attenzione alla protezione delle risorse aziendali. Nel corso degli anni è infatti emerso come le aziende con un reparto assegnato alla sicurezza informatica e dei dati abbiano riscontrato una quantità di problemi molto minore rispetto alle imprese che sottovalutano questo aspetto dell'organizzazione aziendale. È molto importante infatti avere un reparto Cybersecurity in azienda con una sua gerarchia ed organizzazione, solitamente è il CISO(Chief Information Security Officer) a fare da manager del reparto, questa posizione si occupa di gestire, organizzare e prendere decisioni su tutti gli aspetti relativi alla sicurezza e alla protezione dei sistemi e dei dati, il tutto comunicando con i manager di altri reparti o direttamente con il CIO(Chief Information Officer)[9]. Il settore Cybersecurity avrà inoltre un compito molto importante e purtroppo spesso sottovalutato che è la formazione. Citando Kevin Mitnick "il fattore umano è veramente l'anello più debole della sicurezza", il noto hacker e imprenditore nel suo libro "L'arte dell'inganno" espone quanto sia fondamentale avere dei dipendenti formati in azienda, consapevoli delle informazioni che possono rilasciare ma soprattutto a chi e a quali condizioni[16]. Inoltre è molto importante, come sarà descritto anche in seguito, avere una gestione del rischio aziendale ottimale ossia l'identificazione e valutazione dei rischi, considerando tutti i settori in cui l'azienda opera, da cui poter elaborare piani che consentano di controllare i rischi individuati e come agire nel caso in cui ci siano delle intrusioni o furti di dati.

Esistono aziende in cui non è possibile avere un settore dedicato alla cybersecurity per motivazioni economiche o organizzative per cui ci si può rivolgere ad aziende che offrono diversi servizi di sicurezza informatica a terzi prendendo le dovute precauzioni in ambito di condivisione di dati sensibili.

Un'altra buona pratica per garantire una sicurezza accettabile in un contesto aziendale è quella di definire anticipatamente i membri che dovranno occuparsi di risolvere un eventuale problema e soprattutto, come anticipato in precedenza, i protocolli da seguire. Dovranno quindi essere definiti due documenti fondamentali dai manager aziendali: la policy di sicurezza e il regolamento utente. Il primo documento ha lo scopo di definire ad alto livello le regole di gestione della sicurezza, con particolare riferimento alle figure che le hanno in carico; il secondo invece definisce le azioni legittime attuabili dagli utenti del sistema informativo, spesso suddivise per classe di utente. A questi due documenti ne saranno poi aggiunti altri che definiranno la gestione dei backup e degli incidenti, il tutto strutturato ed organizzato in modo più o meno complesso a seconda delle caratteristiche dell'impresa[7].

Descrivendo i metodi da attuare per garantire la sicurezza di un sistema aziendale è doveroso

introdurre lo standard ISO/IEC 27001. Questo standard è una norma internazionale contenente i requisiti per impostare e gestire un sistema di gestione della sicurezza delle informazioni comprendendo, oltre alla sicurezza logica, quella fisica/ambientale e la sicurezza organizzativa. L'azienda deve rispettare tutti i requisiti per poter ottenere la certificazione associata alla norma da un ente indipendente. Alla 27001 è inoltre collegata la norma ISO 27002 che è una raccolta di accorgimenti che possono essere adottati per rispettare i requisiti della norma 27001, l'obiettivo di queste due leggi è garantire l'integrità, riservatezza e disponibilità dei dati e sono applicabili ad aziende operanti nei settori commerciali e industriali come finanza, telecomunicazioni, logistica e settori governativi.

Nello specifico i requisiti dello standard, in relazione alle strategie da attuare, prevedono:

- Pianificazione e Progettazione
- Implementazione
- Monitoraggio
- Mantenimento e Miglioramento

Inoltre nella fase di progettazione vi sono diversi requisiti da dover rispettare che sono l'identificazione, analisi e valutazione dei rischi.

Riguardo la privacy, nonostante la norma 27001 preveda che i dati personali siano tutelati, quest'ultima si interessa anche dei dati dell'organizzazione. Ogni azienda però, come stabilito nel GDPR nel 2018, deve rispettare vincoli più restrittivi in materia di dati personali per poter gestire dati da e verso l'Europa.[7]

Si può notare come la progettazione e messa in opera di una strategia di cybersecurity sia un compito difficile per un'azienda a prescindere dal settore in cui opera e dalla sua complessità, risulta ancora più impegnativo rendersi conto quando è necessario cambiare le strategie in opera in seguito ad una validazione [32].

Oltre alla sicurezza dei sistemi aziendali è necessario individuare degli accorgimenti da attuare anche per i privati, infatti attacchi come il phishing, finalizzati al furto di dati o denaro, e le botnet sono molto comuni[6]. Risultano efficienti strategie per la protezione dei dispositivi e account personali:

- usare password forti cambiandole con cadenza costante
- evitare di aprire link da mail non verificate
- controllare i link di siti che richiedono l'inserimento di dati personali

- effettuare backup dei dati costantemente
- mantenere aggiornati i software soprattutto gli antivirus
- evitare di connettersi a reti Wi-Fi pubbliche

Nel corso degli ultimi anni è aumentato in modo esponenziale lo sviluppo di nuove tecnologie che possono essere usate anche per la definizione di strategie di cybersecurity efficaci. Tra le tante le più importanti sono l'utilizzo di infrastrutture Cloud per superare i vari rischi legati alla centralità dei sistemi(es. salvare copie di backup su cloud) e la definizione di algoritmi di Machine Learning che sono in grado di automatizzare alcuni processi di rilevamento e gestione dei rischi.

## 2.2 Machine learning per la Cybersecurity

In questa sezione sarà introdotto il Machine Learning, un sottoinsieme dell’Intelligenza Artificiale, attraverso cui è possibile definire algoritmi in grado di apprendere dai dati. Le tecniche di apprendimento possono essere anche utilizzate per problemi relativi alla sicurezza dei sistemi informativi.

### 2.2.1 Machine Learning

Così come le nuove tecnologie, sviluppate nel corso degli ultimi decenni, offrono diversi modi e ambienti attraverso cui è possibile intendersi illecitamente, queste ultime offrono nuovi tipi di dati utili da raccogliere e analizzare. L’analisi dei dati viene utilizzata in diversi contesti e per diversi obiettivi, come ad esempio lo sviluppo di algoritmi di apprendimento automatico. Quest’ultimo, meglio conosciuto con il termine di machine learning, è una sezione dell’Intelligenza Artificiale che sta avendo negli ultimi anni una crescita esponenziale e si occupa di definire algoritmi in grado di apprendere dai dati e fare delle valutazioni sulla base degli stessi[27].

I dati possono essere divisi in diverse categorie e in base al loro tipo essere utilizzati per diversi scopi. Le categorie dei dati sono:

- Dati Strutturati
- Dati Non Strutturati
- Dati Semi-Strutturati

#### Dati Strutturati

Hanno una forma ben definita e conforme ad uno standard(formato) accessibile da un software. Esempi di questo tipo possono essere i record di una tabella di un database nel quale vengono archiviate informazioni come nome, indirizzi, numeri di carte di credito, ecc.

#### Dati Non Strutturati

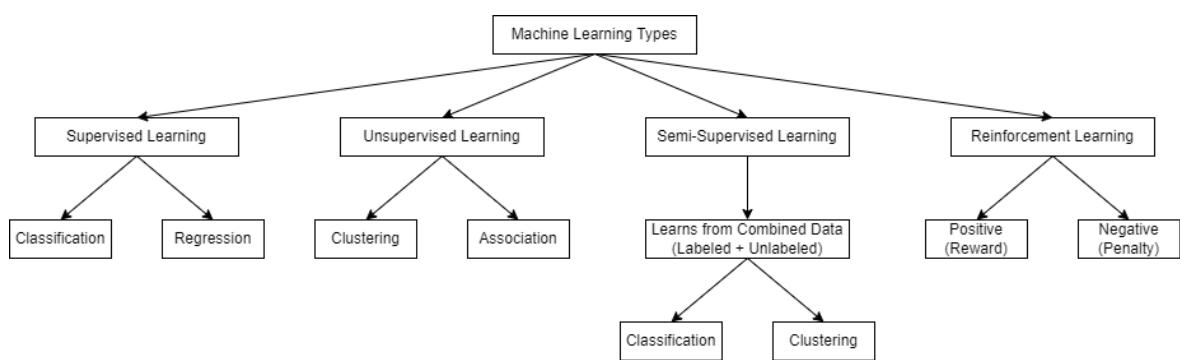
Per i dati non strutturati non esiste un formato standard, questo rende più difficile l’acquisizione ed elaborazione degli stessi. Esempi di dati non strutturati sono file multimediali come audio, video, immagini, ecc.

### Dati Semi-Strutturati

I dati semi-strutturati non hanno una organizzazione rigida come i dati strutturati ma rispettano un formato che ha alcune proprietà per facilitarne l’analisi. Esempi sono file HTML, XML, JSON, database NoSQL, ecc.

Per analizzare i dati in un particolare dominio applicativo ed estrarre informazioni utili da essi per la creazione di algoritmi intelligenti, possiamo definire diverse tecniche di apprendimento automatico in base alle loro caratteristiche.

#### 2.2.2 Classificazione delle tecniche di Machine Learning



**Figura 2.5:** Classificazione degli algoritmi di machine learning

### Supervisionato

L’apprendimento supervisionato è quello in cui un agente apprende usando dei dati etichettati. L’etichetta è definita come variabile dipendente dai dati di input, quando un algoritmo apprende dai dati deve quindi avere per ogni osservazione sia i dati di input che l’etichetta associata.

Le attività di apprendimento supervisionato più comuni sono la classificazione che suddivide i dati di input in due o più classi e la regressione le cui etichette sono numeri reali.

### Non Supervisionato

L’apprendimento non supervisionato viene utilizzato per problemi più complessi rispetto all’apprendimento supervisionato, un agente di questo tipo utilizza dati non etichettati per apprendere dai dati.

Gli algoritmi di apprendimento non supervisionato vengono utilizzati per attività di clustering, feature learning, riduzione di dimensionalità, stima di densità, regole di associazione tra dati, ecc.

### Semi-Supervisionato

L'apprendimento semi-supervisionato è un ibrido tra l'apprendimento supervisionato e non-supervisionato perché opera con dati sia etichettati che non etichettati. Questo tipo di apprendimento viene utilizzato per contesti del mondo reale nel quale i dati etichettati sono pochi, quindi questi algoritmi cercano di fornire risultati migliori rispetto all'utilizzo del solo apprendimento supervisionato.

Attività di apprendimento semi-supervisionato sono rilevamento di frodi, etichettatura dei dati e classificazione del testo.

### Apprendimento per Rinforzo

Gli algoritmi di apprendimento per rinforzo valutano il loro comportamento in un ambiente per migliorare l'efficienza delle decisioni future. Questi agenti si basano sull'assegnazione di una ricompensa o penalità alla decisione presa e in base a questa valutazione agire sull'algoritmo migliorandone l'efficienza. L'apprendimento per rinforzo è uno strumento molto potente per l'automazione di sistemi come la guida autonoma o robotica.

### Active Learning

L'Active Learning è un sottoinsieme dell'apprendimento supervisionato nel quale si ha una mole di dati non etichettati molto grande nella quale le etichette devono essere aggiunte manualmente dal data scientist. Un algoritmo di Active Learning serve in questo processo per diminuire il numero di entry del dataset da etichettare per far risparmiare tempo al data scientist, l'algoritmo avrà quindi il compito di trovare le entry le cui etichette portino più informazione e quindi aumentino l'affidabilità del modello supervisionato che si costruirà successivamente [28].

### Deep Learning

Il Deep Learning è un campo di ricerca del machine learning che tramite l'utilizzo di reti neurali impara dai dati in modo iterativo. Le reti neurali sono ispirate al funzionamento del cervello umano: così come i neuroni del cervello umano sono collegati tra loro per scambiarsi

informazioni, le reti neurali vengono progettate in livelli dove in ognuno di essi c'è un numero predefinito di nodi, definiti appunto neuroni, che comunicano con il livello successivo. Il deep learning è utilizzato in campi come il riconoscimento di immagini, computer vision e anche classificazione, infatti le reti neurali, con le giuste configurazioni, possono fornire risultati migliori rispetto agli algoritmi di classificazione più generali.

### 2.2.3 Applicazioni del Machine Learning per la Cybersecurity

Come accennato in precedenza il machine learning può essere utilizzato nel campo della cybersecurity e, in base al tipo di apprendimento utilizzato, definire modelli capaci di raggiungere un determinato obiettivo.

Con l'apprendimento supervisionato si possono definire classificatori o regressori in grado, per esempio, di stabilire se c'è stata un'intrusione in un sistema, determinare se un codice è malevolo e che tipo di minaccia rappresenta per il sistema, oppure applicazioni che operano nel dominio delle reti(network security). Gli algoritmi di apprendimento supervisionato più utilizzati per questi scopi sono: Decision Tree, Support Vector Machine(SVM), Naive Bayes e K-nearest neighbors(KNN).

L'apprendimento non supervisionato invece viene principalmente utilizzato per problemi di clustering e associazione nelle reti. Si cerca infatti di trovare dei pattern tra gli attacchi verificati in una rete per poterli prevedere e attuare le giuste misure di sicurezza. Algoritmi non supervisionati sono K-Means, clustering gerarchico e K-Medoids.

Altre tecniche di machine learning per la cybersecurity possono essere l'apprendimento semi-supervisionato che viene utilizzato quando si ha conoscenza dell'etichetta solo di un sottoinsieme dei dati e anche la riduzione di dimensionalità del dataset utilizzato per il train del modello. Quest'ultima tecnica è realizzata tramite gli algoritmi PCA ed SVD.

Oltre al machine learning classico è possibile utilizzare anche il Deep Learning per problemi di cybersecurity. Quest'ultimo presenta alcuni vantaggi come ad esempio l'esecuzione automatica della feature extraction nei livelli profondi della rete neurale che invece deve essere eseguita manualmente nel machine learning classico. Altro vantaggio fondamentale è la maggiore precisione delle reti neurali che quindi consente di sviluppare un modello più efficiente[25].

### 2.2.4 Definizione dei principali algoritmi di Machine Learning

Di seguito saranno descritti gli algoritmi di machine learning introdotti precedentemente per comprenderne il funzionamento e la struttura.

#### Decision Tree

Gli alberi decisionali appartengono alla categoria di algoritmi utilizzati per la classificazione, quindi apprendimento supervisionato. L'algoritmo è basato sulla relazione tra entropia e information gain associata ad una feature del dataset con lo scopo di creare un albero i cui nodi sono sottoinsiemi del dataset e gli archi delle decisioni. Per costruire l'albero si parte dal dataset completo e lo si suddivide iterativamente considerando le feature con maggior information gain fino ad arrivare ai nodi foglia in cui tutte le entry del dataset hanno lo stesso valore nella colonna label(sottoinsieme puro). In presenza di una nuova il modello assegna la label semplicemente navigando l'albero fino ad arrivare ad una foglia[10].

#### Random Forest

Questo algoritmo è in stretta correlazione al precedente Decision Tree in quanto il Random Forest considera un insieme di alberi generati a partire dalle stesse feature prese in modo casuale. Esso viene utilizzato in quanto minimizza il rischio di overfitting rispetto al Decision Tree.

#### Support Vector Machine

Il Support Vector Machine è un modello di apprendimento supervisionato, che in genere viene associato ad algoritmi di classificazione e regressione. Esso è basato sull'idea di costruire degli iperpiani che dividano al meglio un set di dati in due o più classi, a questo scopo sono utilizzati i vettori di supporto, che rappresentano i data points più vicini all'iperpiano, questi varieranno in base al set di dati che si sta analizzando e, nel caso in cui vengano rimossi o modificati, alterano la posizione dell'iperpiano divisorio[34].

#### Naive Bayes

I classificatori probabilistici come il Naive Bayes effettuano la predizione basandosi sulla probabilità che il valore delle caratteristiche appartengano ad una determinata classe, in particolare si considera la probabilità condizionata di ogni caratteristica con la variabile indipendente e quando si presenta una nuova entry si sceglie la classe con la probabilità

di appartenenza più alta. Questo tipo di classificatori ha lo svantaggio di non considerare eventuali relazioni tra le caratteristiche a differenza dei classificatori basati sull'entropia[10].

### K-nearest neighbors

Questo algoritmo viene utilizzato per problemi di classificazione, il suo funzionamento è molto semplice: si assegna una classe ad un oggetto scegliendo la classe più comune tra i  $k$  vicini nello spazio delle feature. Risulta quindi rilevante la scelta di un parametro  $k$  ottimale[10].

### K-means

Il K-means è l'algoritmo a partizionamento iterativo ad errore quadratico più famoso per problemi di clustering(raggruppamento di oggetti con pattern simili). L'algoritmo si dice iterativo poiché parte dal dataset completo e individua casualmente  $k$  centroidi, assegna ogni campione al centroide più vicino e successivamente ricalcola un nuovo centroide(punto medio) per ogni gruppo; il procedimento viene eseguito iterativamente fin quando i centroidi di due iterazioni coincidono. L'algoritmo è inoltre ad errore quadratico poiché mira a minimizzare l'errore di ogni campione rispetto ai centroidi[10].

### Clustering gerarchico

Il clustering gerarchico considera dei raggruppamenti multilivello dove in ognuno c'è un diverso grado di similarità tra i campioni. Durante la definizione dei diversi cluster è possibile collegarli per formare un albero, definito dendogramma, che ha il dataset completo come radice e i singoli campioni come foglie. La costruzione dell'albero dipende dall'algoritmo utilizzato: nel caso in cui il clustering sia agglomerativo si parte dai singoli campioni e si raggruppano i due più simili fino ad arrivare ad avere un unico cluster, altrimenti il clustering è divisivo quindi si parte dal dataset intero e si procede in modo inverso[10].

### K-medoids

L'algoritmo K-medoids è strettamente correlato al K-means, infatti entrambi sono degli algoritmi partizionali ad errore quadratico e il procedimento per costruire i  $k$  cluster è molto simile. K-medoids ha però un vantaggio: considera come centroidi dei campioni effettivi presenti nel dataset individuando il campione più vicino al centroide calcolato; questo procedimento rende il K-medoids più robusto al rumore e agli outlier rispetto al K-means[31].

**PCA e SVD**

Sono entrambi algoritmi che hanno lo scopo di ridurre la dimensionalità del set di dati utilizzato per aumentare l'affidabilità di un modello oppure diminuirne la complessità. Questi algoritmi operano entrambi con l'algebra lineare e si differenziano per il metodo di scomposizione della matrice associata al dataset.

Dopo aver introdotto le principali applicazioni del Machine Learning per la Cybersecurity si può pensare a diverse soluzioni per problemi concreti in ambito di sicurezza informatica. Il Machine Learning classico può però richiedere molte risorse computazionali e un ampio intervallo temporale per addestrare e validare un modello, con l'avvento del Quantum Computing sono stati sperimentati degli algoritmi di Machine Learning eseguiti su computer quantistici, che saranno analizzati nella prossima sezione, i quali cercano di superare o migliorare queste limitazioni.

## 2.3 Quantum Machine Learning

Il Quantum Machine Learning nasce dall’intersezione del Quantum Computing con il Machine Learning. Questo approccio diverso nasce dalla necessità di eseguire gli algoritmi di ML più velocemente, inoltre è stato osservato che i computer quantistici sono in grado di riconoscere alcuni pattern nei dati, difficilmente osservabili tramite un approccio classico[3]. Di seguito sarà illustrato il Quantum Computing in modo da comprendere meglio il concetto di Quantum Machine Learning.

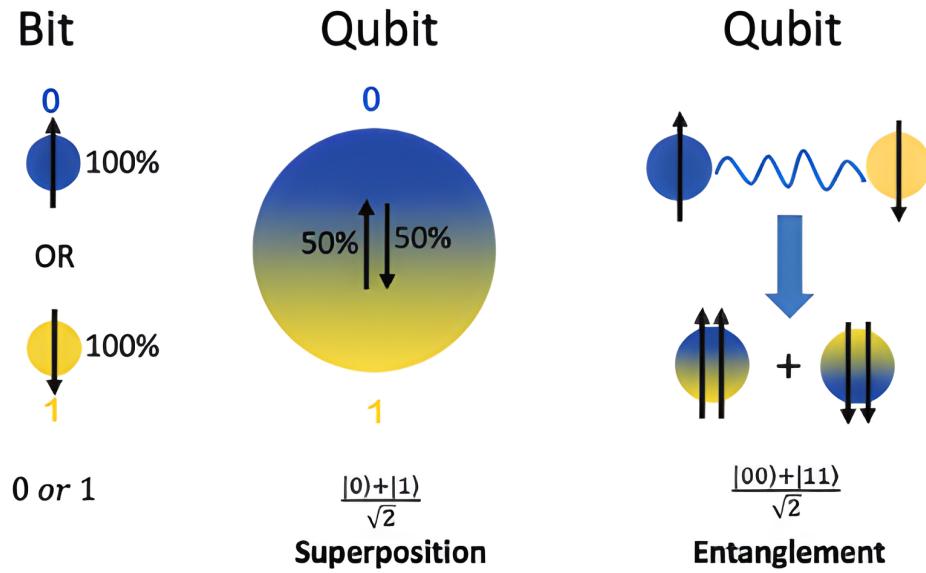
### 2.3.1 Quantum Computing

La differenza sostanziale tra la computazione classica e la computazione quantistica è la rappresentazione dell’informazione. Nella computazione classica l’unità minima di informazione è il bit, codificabile in qualsiasi sistema fisico che può trovarsi in uno dei due possibili stati, esso può infatti assumere due diversi valori in modo esclusivo ossia 0 e 1. Nella computazione quantistica l’unità minima di informazione è il qubit che ha una struttura e proprietà più complesse rispetto al bit(vedi Figura 2.5), si penserà al qubit in modo astratto senza preoccuparsi del sistema quantistico fisico che lo codifica, infatti questo potrebbe essere un atomo, un fotone, un elettrone o altro.

Sulla base della meccanica quantistica il qubit può assumere lo stato 0, 1 o, a differenza dei bit, entrambi gli stati contemporaneamente. Questo fenomeno è definito come sovrapposizione quantistica, la caratteristica che consente di risolvere classi di problemi con una complessità computazionale elevata, intrattabili con la computazione classica.

Per meglio capire questo concetto e il vantaggio in termini di velocità che ne deriva consideriamo l’esempio di avere 3 bit, gli stati che questi possono assumere sono otto, d’altra parte lo stato di 3 qubit può essere nella sovrapposizione di tutti e otto gli stati. Questo implica che raddoppiare il numero di bit in una macchina classica raddoppierebbe anche la complessità computazionale, con l’utilizzo dei qubit per avere la stessa complessità computazionale basta aggiungere un bit.

Un’altra proprietà fondamentale del Quantum Computing è la correlazione quantistica, più conosciuta come entanglement, la quale permette a due o più qubit di essere in uno stato particolare che collega i qubit, come si può osservare in Figura 2.6[11]. Questo significa che ogni osservazione fatta su un qubit del gruppo sarà uguale a tutti i qubit del gruppo anche se vi è una grande distanza fisica tra essi[11].



**Figura 2.6:** Struttura di un qubit, sovrapposizione e correlazione quantistica

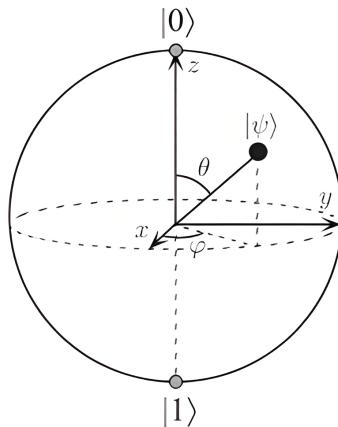
Si può descrivere matematicamente un qubit attraverso un vettore nello spazio di Hilbert, con due stati base mutuamente ortogonali che in generale vengono etichettati con  $|0\rangle$  e  $|1\rangle$ :

$$|\psi\rangle = a|0\rangle + b|1\rangle, \quad a, b \in \mathbb{C}$$

Tuttavia non sono necessari due numeri complessi per descrivere lo stato di un qubit ma bastano due numeri reali, questo è possibile perché non servono la normalizzazione complessiva e la fase complessiva del vettore ai fini della misurazione, si può osservare questa rappresentazione attraverso la sfera di Bloch presentata in Figura 2.7[24]. Fatte queste assunzioni si ottiene:

$$\begin{aligned} |a|^2 + |b|^2 &= 1 \\ |\psi\rangle &= \cos \frac{\theta}{2} + e^{i\phi} \sin \frac{\theta}{2} \quad 0 \leq \theta \leq \pi, \quad 0 \leq \phi < 2 \end{aligned}$$

Un altro fenomeno molto importante della meccanica quantistica, in particolare quando facciamo riferimento alla misurazione di un qubit, è l'interferenza quantistica. Così come nella fisica classica l'interferenza di due onde sinusoidali che si trovano in uno stesso punto causa un'interferenza costruttiva(somma delle onde) o distruttiva(annullamento delle onde) anche nella fisica quantistica due particelle sono in grado di interferire tra loro sommandosi o annullandosi. Più in particolare si somma l'ampiezza di probabilità di ottenere una determinata misura che, in riferimento all'ampiezza, elevata al quadrato dà la probabilità dello stato di un qubit in quel momento. Considerando la notazione matematica definita



**Figura 2.7:** Sfera di Bloch

precedentemente per il qubit si può ottenere per esempio:

$$\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

Dalla la precedente formula possiamo elevare al quadrato l'ampiezza di probabilità associata a ciascun stato per ottenere, come illustrato in precedenza, la probabilità dello stato associato quando si effettua la misurazione sul qubit:

$$\left(\frac{1}{\sqrt{2}}\right)^2 = 0.5$$

Quindi ci sarà il 50% di probabilità che al momento della misurazione il qubit abbia lo stato 0 oppure 1.

É molto importante notare come l'osservazione, quindi lo stato, dei qubit non sia deterministica come nel caso della computazione classica, ma probabilistica. Questa caratteristica è legata principalmente al tempo di coerenza di un qubit: a livello fisico, a prescindere da come il qubit sia realizzato, si ha un tempo nel quale lo stato del qubit è coerente ossia l'osservazione che si effettua è affidabile. A causa dell'interazione di un sistema quantistico con l'esterno il qubit può perdere coerenza e quindi falsare l'osservazione. Per diminuire questa interazione, quindi massimizzare il tempo di coerenza, il sistema quantistico deve essere mantenuto ad una temperatura vicina allo zero assoluto[21].

Tuttavia avere il sistema isolato rende difficile l'accesso ai qubit e quindi eseguire operazioni con essi proprio a causa della decoerenza. Si deve quindi trovare il giusto equilibrio tra il tempo di coerenza e il tempo impiegato per eseguire delle operazioni sui qubit. In generale l'obiettivo da perseguire è quello di massimizzare il tempo di coerenza e minimizzare il tempo per eseguire un'operazione su un qubit.

Riguardo le operazioni eseguibili sui qubit, queste permettono l'inizializzazione e la trasformazione degli stessi e sono definite come porte quantistiche(quantum gates). Come nella computazione classica esistono delle porte nella computazione quantistica come AND, OR, NOT e XOR, in particolare possiamo avere delle porte relative sollo alla fisica quantistica che permettono ad esempio l'entangling di due o più qubit[23].

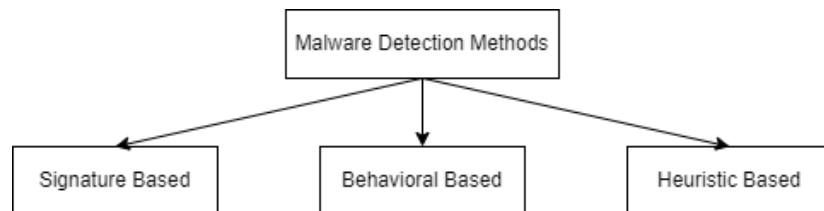
Avendo chiaro come il Quantum Computing è definito e come opera sui dati si può analizzare in modo più approfondito il Quantum Machine Learning. Questo questo nuovo campo, essendo in una fase iniziale della sua sperimentazione e definizione, non è supportato da una teoria ben definita che ne spieghi completamente il comportamento, infatti non è ancora ben delineato se e come gli algoritmi di machine learning quantistico siano in grado di elaborare i dati in modo più preciso rispetto agli algoritmi di machine learning classico. Tuttavia, in particolari condizioni, gli algoritmi per il Quantum Machine Learning sono una proposta valida soprattutto per problemi di classificazione.

## 2.4 Stato dell’arte

In questa sezione saranno descritti i sistemi, metodologie e studi, compiuti e pubblicati in letteratura, riguardo i principali argomenti affrontati dalla tesi ossia Malware Detection e Quantum Machine Learning.

### 2.4.1 Malware Detection

Attualmente il problema del Malware Detection è una sfida aperta nel campo della cybersecurity poiché, così come i progettisti dei moderni antivirus cercano nuove tecniche di rilevamento e perfezionano le attuali, gli sviluppatori di malware cercano nuovi metodi per aggirare gli stessi sistemi. Nello specifico le tecniche di rilevamento di malware sono categorizzate, così come in Figura 2.8[2], in:



**Figura 2.8:** Metodi di rilevamento dei malware

- Metodi basati sulla firma (Signature Based)
- Metodi basati sul comportamento (Behavioral Based)
- Metodi euristici (Heuristic Based)

I metodi basati sulla firma effettuano un’analisi statica del codice binario identificando tutti i possibili percorsi di esecuzione, tutto senza che il codice venga eseguito. Questo tipo di analisi del codice risulta essere obsoleto vista la mole dei programmi odierni e anche per altre motivazioni che saranno descritte successivamente. Di conseguenza si è dovuti passare ad un’analisi del codice dinamica che quindi effettua l’analisi a runtime osservando il flusso di controllo e le chiamate API[18].

#### Metodi basati sulla firma

Questa tipologia di analisi è stata largamente utilizzata negli antivirus più famosi, la ragione è essenzialmente la velocità ed efficienza nella classificazione di un nuovo software preso in analisi. La firma è metaforicamente l’impronta digitale di un software, è quindi

unica ma presenta un piccolo tasso di errore nella sua definizione ed è stata proprio questa la discriminante di un buon antivirus. Il processo di classificazione di un codice consiste nel confrontare la sua firma con quelle dei malware precedentemente calcolate, se si trova una corrispondenza il codice in analisi è dannoso.

I principali svantaggi del Signature Based Method sono la quantità di tempo e denaro utilizzati per estrarre le firme dei malware, l’aggiornamento del database di firme dannose ma soprattutto algoritmi di questo tipo non riescono a riconoscere come dannosi i malware che cambiano il loro codice ad ogni infezione oppure crittano lo stesso per non farlo riconoscere come dannoso. Di conseguenza sono stati presi in considerazione dai ricercatori nuovi metodi di identificazione di software dannosi[2].

### Metodi basati sul comportamento

Questa tipologia di rilevamento consiste nell’analizzare le azioni, le risorse e i servizi richiesti al sistema da un software per deciderne l’attendibilità. A differenza del Signature Based Method i metodi basati sul comportamento prendono in considerazione le azioni compiute dal programma a tempo di esecuzione piuttosto che analizzarne la sua definizione. Il processo di rilevamento consiste nel definire una firma di comportamento per ogni tipologia di malware e quindi, allo stesso modo del Signature Based Method, calcolare la firma del codice da analizzare e trovare, se esiste, una corrispondenza con le precedenti.

I vantaggi di questo tipo di rilevamento sono principalmente la capacità di riconoscere come dannosi i malware in cui i Signature Based Methods non riescono, purtroppo però riconoscono con troppa frequenza come non dannosi dei programmi che in realtà sono malware, hanno quindi un alto False Positive Ratio[2].

### Metodi Euristici

I metodi euristici effettuano la classificazione del codice utilizzando modelli di Machine Learning, quindi devono essere definiti dei dataset tramite cui allenare i modelli e, in presenza di un nuovo codice da analizzare, estrarre gli opportuni dati per poter effettuare la predizione. I dataset possono essere costruiti sulla base di:

- chiamate API, utilizzate per dare comandi al sistema operativo
- OpCode, istruzioni Assembly
- N-Grams, sottostringhe di codice binario

- grafi che rappresentano il flusso di controllo
- features ibride tra API call e Control Flow Graph

### Allineamento di Sequenze

Purtroppo anche i metodi euristici non sono esenti da errori nonostante siano molto utilizzati negli antivirus odierni, così i ricercatori hanno sperimentato nuove tecniche di rilevamento osservando che l'estrazione di pattern simili tra le chiamate API di diverse categorie di malware porta ad ottimi risultati[18].

Analogamente alle tecniche di rilevamento esistono diversi metodi di occultamento del malware, che gli sviluppatori utilizzano al fine di eludere gli antivirus e quindi riuscire a compiere l'attacco. I malware sono classificati in due generazioni, la prima non cambia il suo corpo, ossia il codice sorgente, mentre nella seconda generazione il corpo cambia lasciando però inalterate le azioni che il codice compie[30]. A seconda di come il corpo del codice viene trasformato si può a sua volta classificare la seconda generazione di malware in:

- Malware Crittografato
- Malware Oligomorfico
- Malware Polimorfico
- Malware Metamorfico

### Crittografia del codice

Gli sviluppatori usano questa tecnica per cifrare l'intero programma o parte del codice. Per poter funzionare questa tecnica necessita che il nuovo codice del malware sia composto da due parti: il codice cifrato tramite uno specifico algoritmo e il codice usato per decifrarlo(decrypter). Questo metodo è efficace ma può non funzionare poiché l'algoritmo di decriptazione rimane invariato per ogni esecuzione, quindi un antivirus può riconoscere facilmente la firma di quest'ultimo invece della firma del codice malevolo[30].

### Strategia oligomorfica

Questa categoria rappresenta un'evoluzione dei malware che utilizzano la crittografia per non essere rilevati, infatti differiscono da questi ultimi poiché cambiano decrypter ad ogni

nuovo utilizzo. Il numero di codici di decriptazione è però limitato nell’ordine delle centinaia, quindi gli antivirus, anche se questo processo impiega molto tempo, sono comunque in grado di riconoscere la signature di ogni algoritmo di decriptazione associato al malware[30].

### Strategia polimorfica

Anche in questa categoria viene utilizzato un algoritmo di crittografia che però, oltre a nascondere il codice del malware, cifra il codice di un motore(engine) che ha il compito di scegliere casualmente ad ogni esecuzione un algoritmo crittografico diverso da utilizzare. Concretamente è possibile avere per un singolo malware un numero illimitato di decrypters per il suo codice sorgente, di conseguenza diventa difficile la rilevazione per un antivirus[30].

### Strategia metamorfica

I malware metamorfici invece di generare un nuovo decrypter ad ogni esecuzione cambiano il corpo del loro codice sorgente utilizzando metodi di offuscamento come salti non necessari oppure istruzioni inutili ai fini del funzionamento. I malware metamorfici sono impossibili da rilevare da sistemi basati sulla firma poichè questa cambia per ogni istanza di malware, tuttavia sono molto difficili da scrivere ed è possibile rilevarli tramite tecniche di Machine Learning[30].

Avendo illustrato le principali metodologie di rilevamento ma anche le tecniche di occultamento si può osservare come le strategie basate su firma, largamente utilizzate negli antivirus passati e odierni inizino ad avere problemi nel rilevamento di alcune particolari tipologie di malware, dove invece metodi euristici come il Machine Learning riescono[15].

#### 2.4.2 Machine Learning per Malware Detection

Soluzioni di apprendimento automatico realizzabili attraverso l’utilizzo del Machine Learning per problemi di Malware Detection sono divenute necessarie negli ultimi anni in quanto la crescita esponenziale di Internet ha causato un altrettanto marcato aumento delle minacce informatiche, quindi metodi di rilevamento manuali come quelli basati su signature risultano essere troppo dispendiosi in termini di tempo e costi.

Le diverse tipologie di Machine Learning vengono impiegate nell’ambito del Malware Detection per diversi scopi.

- Il Machine Learning non supervisionato viene impiegato nella fase di raccolta dei dati. In particolare quando si ha disponizione un largo set di dati non etichettato viene utilizzato il clustering per trovare i pattern tra i campioni delle varie classi e quindi questo consente di assegnare manualmente solo poche label per ogni classe invece di assegnarle tutte manualmente.
- Il Machine Learning supervisionato viene usato quando si ha a disposizione un set di dati abbastanza grande da poter allenare un modello che sia in grado di predire se un codice è malevolo nel caso di una classificazione binaria, oppure il tipo di malware(es. Trojan, Worm, ecc.) nel caso di classificazione multilabel.
- Il Deep Learning che è utilizzato in larga scala in applicazioni come riconoscimento di immagini o voce viene impiegato per la classificazione nel caso in cui si hanno a disposizione dati a basso livello per arrivare a dati ad alto livello, quindi è necessaria un’analisi diversa, più approfondita, da quella fornita dal Machine Learning supervisionato. Esso può infatti essere applicato nel contesto del Malware Detection in quanto i dati con cui vengono costruiti i dataset sono molto a basso livello(si pensi alle system call) e quindi ha senso definire reti neurali per una classificazione di questo tipo.

Il focus nella definizione di modelli di Machine Learning per Malware Detection deve essere posto quindi sul set di dati di partenza, essendo infatti una metodologia data-driven si devono utilizzare le caratteristiche più importanti e rilevanti che quindi renderanno il modello capace di assegnare la giusta etichetta ai nuovi campioni. Inoltre nella fase di valutazione del modello è doveroso minimizzare il False Positive Ratio poiché anche un solo campione valutato in modo errato come benigno può avere conseguenze serie per gli utenti[15].

Avendo chiare le diverse tecniche di rilevamento di Malware il Machine Learning risulta essere promettente nel campo della Malware Detection per cui nella sperimentazione effettuata in questa tesi saranno utilizzati diversi approcci al Machine Learning per la Malware Detection con lo scopo di verificare quale metodo sia migliore.

#### 2.4.3 Algoritmi di Quantum Machine Learning

Oltre alle normali strategie di Machine Learning è possibile, grazie ad algoritmi proprietari sviluppati da diverse aziende operanti nel campo del Quantum Computing, addestrare un modello di Machine Learning ed eseguirlo su un sistema quantistico reale. Esistono più librerie che mettono a disposizione degli sviluppatori algoritmi quantistici che possono avere

lo scopo di convertire dati classici in dati quantistici oppure di effettuare una classificazione/regressione su dati classici o quantistici. Sono infatti diverse le proposte ibride nel mondo del Quantum Machine Learning poiché sfruttano sia la potenza del Quantum Computing che può trattare problemi più complessi che la versatilità degli algoritmi classici nell'eseguire task come Preprocessing dei dati oppure calcoli aritmetici.

Le librerie più utilizzate e famose nel campo del Quantum Machine Learning sono:

- Cirq, libreria Python sviluppata da Google permette la creazione e ottimizzazione di circuiti quantistici eseguibili su sistemi e simulatori quantistici;
- TensorFlow Quantum, sviluppata sempre da Google è una libreria Python che si concentra su dati quantistici e sulla costruzione di modelli ibridi, inglobando algoritmi e logica progettati in Cirq;
- Pennylane è la principale libreria, sviluppata in Python, utilizzata nel mondo del Quantum Computing, essa infatti offre funzioni, simulatori, hardware e risorse nel campo del Quantum Machine Learning ma anche in Quantum Chemistry e algoritmi di ottimizzazione;
- infine Qiskit sviluppata in Python da IBM permette di definire algoritmi quantistici ottimizzati per Machine Learning, Finance, Chemistry e Optimization e poi eseguirli su simulatori e sistemi quantistici; a differenza delle altre librerie Qiskit offre una gamma più ampia di algoritmi tra cui i paralleli quantistici di alcuni algoritmi di Machine Learning classico.

### Confronto di prestazioni

Dato il crescente sviluppo di algoritmi quantistici ne deriva un utilizzo maggiore da parte degli sviluppatori di modelli di Machine Learning per cui diviene necessario capire se e come i modelli quantistici siano migliori dei classici. A tal proposito si considerino gli esperimenti [4] e [19] che utilizzano gli algoritmi della libreria Qiskit poiché, come già detto precedentemente, è più semplice la comparazione in quanto sono stati sviluppati da IBM algoritmi paralleli a quelli usati nel Machine Learning classico.

Dai confronti effettuati è emerso come la bontà degli algoritmi quantistici dipenda molto dal dataset utilizzato, con dataset in cui vi è un numero elevato di campioni o feature su cui effettuare il training del modello è stato osservato che gli algoritmi quantistici non sono all'altezza dei classici. Nell'esperimento [4] sono stati effettuati due esperimenti confrontando

Classifier	Backend	Time	Accuracy(%)
QSVM kernel-based	Statevector Simulator	5.96µs	100%
QSVM variational	Statevector Simulator	5.96µs	95%
Classic SVM	Local CPU	6.20 µs	85%

**Tabella 2.1:** Risultati della QSVM sul dataset Breast Cancer Wisconsin

Classifier	Backend	Time	Accuracy(%)
QSVM variational	Statevector Simulator	5.96µs	100%
Classic SVM	Local CPU	6.20 µs	90%

**Tabella 2.2:** Risultati della QSVM sul dataset Wine

l’algoritmo SVM con la sua controparte quantistica QSVM kernel-based e variazionale(due implementazioni diverse dell’algoritmo quantistico) su dataset di piccole dimensioni: Breast Cancer Wisconsin dataset con 569 campioni e 32 feature su cui è possibile effettuare una classificazione binaria; UCI ML Wine dataset formato da 13 feature e 178 campioni che possono appartenere a 3 classi differenti. La Tabella 2.1[4] mostra i risultati ottenuti dalla classificazione usando i due differenti algoritmi sul primo dataset, analogamente la Tabella 2.2[4] mostra i risultati ottenuti sul secondo.

Dai risultati risalta che su accuratezza e tempo gli algoritmi quantistici sono superiori rispetto al classico SVM. È importante evidenziare che gli ottimi valori ottenuti derivano da due dataset di dimensioni relativamente ridotte che non sono comparabili a quelli usati nella maggior parte delle applicazioni concrete del Machine Learning.

Di conseguenza saranno analizzati altri due dataset utilizzati in applicazioni concrete del Machine Learning, in particolare in ambito Cybersecurity. Essi sono ClaMP, dataset utilizzato in ambito Malware Detection in quanto formato da header PE(Portable Executable) che contengono tutte le informazioni necessarie al sistema operativo per eseguire un programma, e ReVeal che traccia le vulnerabilità di sistemi operativi open-source quali Debian e Chromium[19].

In termini di dimensioni ClaMP è formato da 69 feature e 5184 campioni, invece ReVeal contiene un totale di 22734 campioni ognuno con 100 feature; per poter essere effettuato il train di un modello quantistico è stato necessario diminuire le dimensioni dei dataset in quanto il numero di qubit disponibile è limitato a 16(al momento della sperimentazione

Classifier	Time(s)	Accuracy(%)
QNN	2698	52.1%
Hybrid-QNN	2507	52.27%
QSVM	10000	73.5%
Classical SVM	1	93.5%
Classical NN	22	92.7%

**Tabella 2.3:** Risultati ottenuti sul dataset ClaMP

Classifier	Time(s)	Accuracy(%)
QNN	3006	52.71%
Hybrid-QNN	2563	52.71%
QSVM	16682	58.26%
Classical SVM	2	60.34%
Classical NN	41	55.7%

**Tabella 2.4:** Risultati ottenuti sul dataset ReVeal

effettuata) ed esiste una corrispondenza tra quest’ultimo e le feature quindi con tecniche di Preprocessing dei dati quali Feature Extraction o Feature Selection sono stati modificati i dataset per renderli compatibili[19]. Le tabelle 2.3[19] e 2.4[19] mostrano i risultati ottenuti dal training dei diversi modelli usando i dataset precedentemente descritti.

Dalle tabelle dei risultati si può osservare la marcata differenza tra le due coppie di dataset e come le performance sia in termini di tempo che di accuratezza siano notevolmente peggiorate. Le motivazioni sono diverse: in primo luogo si può osservare che a causa delle limitazioni hardware dei computer quantistici o simulatori quantistici riguardo il numero di qubit utilizzabili portano ad effettuare il training del dataset con meno feature quindi informazioni utili su cui poter effettuare una valutazione, inoltre altrettanto importante è la presenza su sistemi quantistici rumorosi di una percentuale di decoerenza che può alterare il valore di una valutazione. Per quanto riguarda i tempi lunghi sono dovuti sia alla coda per accedere al sistema o simulatore quantistico che al tempo necessario per l’inizializzazione dei qubit.

Nonostante queste limitazioni è stato dimostrato che gli algoritmi quantistici sono ottimi con semplici dataset e che sono almeno in grado di gestire dataset complessi e con un elevato

numero di campioni come quelli in ambito cybersecurity.

Lo scopo di questo elaborato sarà quello di effettuare un’analisi ed un confronto in termini di prestazioni e tempo tra algoritmi quantistici e classici utilizzando un dataset con caratteristiche simili al dataset ClaMP su cui saranno eseguiti diversi algoritmi di Preprocessing per cercare di migliorare le prestazioni ottenute, il tutto per ampliare lo stato dell’arte presente in letteratura.

# CAPITOLO 3

---

## Confronto tra Quantum Classifier

---

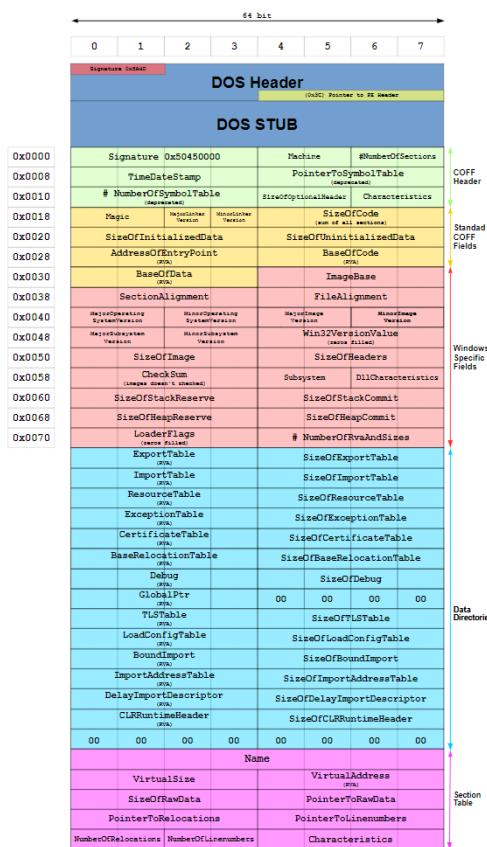
### 3.1 Obiettivi

L'obiettivo della sperimentazione è quello di osservare il comportamento degli algoritmi quantistici su un problema di Malware Detection e successivamente di confrontarli con i principali algoritmi utilizzati nel Machine Learning classico per valutare se e quanto sia vantaggiosa una possibile applicazione concreta degli algoritmi quantistici.

La valutazione della bontà dei modelli dovrà quindi essere basata su alcuni parametri di riferimento, oltre alle metriche per valutare le performance dei modelli sarà considerato anche il tempo di training e testing del modello proprio per valutare l'effettiva applicabilità dei modelli quantistici. È importante sottolineare che la valutazione sarà effettuata su un particolare tipo di dataset, successivamente descritto, e che i risultati ottenuti non valgono di norma per ogni tipo di dataset utilizzato.

### 3.2 Dataset utilizzato per lo studio

Il dataset utilizzato in questa sperimentazione è EMBER<sup>1</sup>, il primo dataset di grandi dimensioni a contenere campioni benigni e maligni in ambito Malware Detection. Sono pochi i dataset utilizzabili per questo scopo poiché ci sono diversi ostacoli nella raccolta e pubblicazione di questo tipo di dati: in primo luogo ci sono motivazioni legali che impediscono la pubblicazione dei file binari, per i binari maligni siti come VirusShare<sup>2</sup> si occupano della loro condivisione, è però vietata una condivisione successiva così come per i binari benigni che a causa di leggi sul copyright non possono essere pubblicati; un'altra difficoltà risiede nell'etichettatura dei campioni poiché, a differenza di altri tipi di file, per stabilire la classe di un binario è necessario molto tempo e alcuni risultati già disponibili sono protetti dai vendor antimalware che li hanno classificati. Per superare tutte queste difficoltà i creatori del dataset EMBER hanno estratto tutte le feature utili alla classificazione di un eseguibile dai file Portable Executable(PE) di Windows che consentono la pubblicazione di campioni sia benigni che maligni senza vincoli[1].



**Figura 3.1:** Struttura di un file PE

<sup>1</sup><https://iee-datalab.org/open-access/malware-analysis-datasets-top-1000-pe-importsfiles>

<sup>2</sup><https://virusshare.com/>

### 3.2.1 Formato PE

Il formato PE viene usato nei sistemi Windows in associazione a qualsiasi file eseguibile, quindi anche file di librerie e file oggetto, viene definito portable poiché non dipende dall'architettura su cui è eseguito ma dipende esclusivamente dal binario eseguibile. Un file PE infatti è un header, quindi una struttura dati, che contiene tutte le informazioni necessarie al loader di Windows per gestire il codice eseguibile. La Figura 3.1[1] mostra la struttura di un header PE, si può innanzitutto osservare che ci sono diversi header ognuno contenente informazioni riguardo una funzione del programma da caricare in memoria, per esempio la sezione che fa riferimento al codice da eseguire viene allineata con l'inizio di una pagina di memoria e i permessi vengono impostati read-only. Le sezioni del file PE sono le seguenti:

- intestazione in formato COFF che contiene il tipo di macchina a cui è destinato il file, la natura del file, il numero di sezioni ed altre informazioni generali;
- l'intestazione opzionale include la versione del linker, la dimensione del codice, la dimensione dei file inizializzati e non inizializzati ed altre informazioni riguardo l'eseguibile compresa la directory dei dati al cui interno vi sono i puntatori alle sezioni della memoria che contengono i dati da utilizzare durante l'esecuzione;
- infine la tabella delle sezioni contiene i puntatori a tutte le sezioni del file PE.

Nonostante la struttura e l'impostazione dei permessi dei file PE questi non garantiscono che il programma associato sia legittimo, infatti programmi dannosi non vengono riconosciuti come tali e quindi vengono eseguiti dalla macchina che, per poter eseguire il codice, legge il file PE. Non è rara, ma anche semplice da riconoscere per un moderno antivirus, l'aggiunta di codice dannoso all'interno dei file PE di servizi messi a disposizione da Windows, quindi sicuramente esenti da codice malevolo. In ultima analisi un file di questo tipo può anche essere utilizzato come vettore di malware[1].

### 3.2.2 Descrizione delle Feature

Da un file PE si possono ricavare tutte le informazioni necessarie per effettuare un'analisi accurata del programma ad esso associato, quindi si possono aggirare le difficoltà riscontrate nella condivisione del file binario di un programma analizzate precedentemente e quindi poter costruire un dataset su cui allenare modelli di Machine Learning.

Il dataset è costituito da un oggetto JSON per ogni campione che comprende i seguenti dati:

- l'hash sha256 del file originale utilizzato come id del campione;

- una stima approssimativa della data in cui il campione è stato osservato per la prima volta;
- una label che ha valore 1 quando il programma è dannoso, 0 altrimenti;
- un set di 8 gruppi di caratteristiche grezze(cioè non utilizzabili direttamente per l'addestramento).

Le feature descritte successivamente possono essere unite in due gruppi più grandi, i primi 5 gruppi costituiscono le feature ricavabili dopo il parsing del file PE da parte del sistema mentre i restanti 3 gruppi sono indipendenti dal parsing.

- Informazioni generali, comprende la dimensione del file e le informazioni di base ottenute dall'intestazione PE come il numero di funzioni importate ed esportate, la presenza di una sezione di debug, la memoria locale degli eventuali thread, risorse e il numero di simboli.
- Informazioni sull'intestazione, prese dall'intestazione del COFF contengono il timestamp, la macchina di destinazione in formato stringa e un elenco di caratteristiche che riguardano il sistema.
- Funzioni importate, si legge la tabella degli indirizzi di importazione e se ne ricavano le relative funzioni importate per libreria.
- Funzioni esportate, vengono ricavate allo stesso modo delle precedenti.
- Informazioni sulle sezioni, vengono fornite informazioni su ogni sezione riguardo nome, dimensione, entropia ed elenco di stringhe che rappresenta le caratteristiche della sezione.
- Istogramma dei byte, contiene 256 valori interi che rappresenta il conteggio di ogni valore di byte all'interno del file.
- Istogramma dell'entropia dei byte, approssima la distribuzione congiunta dell'entropia e del valore dei byte.
- Informazioni sulle stringhe, contengono semplici statistiche sulle stringhe come la loro lunghezza media, un istogramma dei caratteri stampabili all'interno di una stringa e l'entropia dei caratteri in tutte le stringhe.

### 3.2.3 Gestione delle stringhe

Il dataset utilizzato ha un numero molto ampio di stringhe poiché sono definite come caratteristiche le funzioni chiamate dal programma, di conseguenza poiché i modelli di Machine Learning lavorano bene con dati numerici è necessaria una codifica in grado di convertire le stringhe in numeri. Una possibile conversione, più semplice, consiste nell'assegnare ad ogni stringa un valore numerico e quindi la feature che originariamente era un vettore di stringhe diventa un vettore di interi. Questo metodo di conversione viene definito `OrdinalEncoder` ma non fornisce un buon grado di entropia in quanto vi è semplicemente un mapping tra funzione e id quindi funzioni con caratteristiche opposte potrebbero avere id molto vicini e quindi l'algoritmo le interpreta come simili. Un altro metodo di conversione che permette di fornire il giusto grado di entropia è la `One-Hot Encoding` che consiste nel creare una nuova feature per ogni valore che compare nella feature di partenza e, per la nuova feature, impostare a 1(hot) solo i campioni che hanno quel determinato valore associato e tutti gli altri a 0 [10]. In questo modo si riesce a convertire le feature grezze presenti nel dataset in feature interpretabili da un algoritmo di Machine Learning.

### 3.3 Definizione della pipeline

Avendo descritto il dataset utilizzato per la sperimentazione e le tecniche utilizzate per renderlo effettivamente interpretabile da un algoritmo di Machine Learning adesso sarà definita e descritta la pipeline seguita per allenare i modelli ed effettuare le valutazioni.

Risulta però necessario definire una pipeline di Machine Learning e capirne la logica e il funzionamento. Essa non è altro che una successione di operazioni che partono dall'analisi del problema fino ad arrivare ad un modello utilizzabile ed applicabile ad un sistema software. I modelli di Machine Learning nascono infatti per essere affiancati a sistemi software esistenti in modo da aggiungere una funzionalità intelligente, si pensi alle auto a guida autonoma oppure ai suggerimenti forniti da e-commerce o piattaforme di streaming.

Così come l'ingegneria del software racchiude le metodologie e i processi da seguire per lo sviluppo di un software, l'ingegneria del Machine Learning è una disciplina analoga basata sullo sviluppo di moduli intelligenti. Sono stati definiti diversi approcci per la creazione di modelli di Machine Learning tra cui il modello CRISP-DM e un approccio agile chiamato TDSP. Questi modelli sono composti da diverse fasi simili tra loro e differiscono nel numero di rilasci o dalla possibilità di modificare una parte del progetto quando si è in un'altra fase della progettazione. In ogni modello vengono però identificati dei passi comuni da seguire per lo sviluppo di un modello di Machine Learning qualitativamente valido.

- **Business Understanding** è la prima fase da seguire che racchiude la raccolta e analisi dei requisiti oltre agli obiettivi di business e success criteria che stabiliscono sostanzialmente cosa deve fare il modello e i criteri secondo cui quest'ultimo è in linea con gli obiettivi prefissati.
- **Data Acquisition** è una fase che riguarda i dati attraverso cui il progettista identifica e acquisisce il set di dati da utilizzare, inoltre vengono risolte problematiche relative alla mancanza di dati se presenti.
- La **Data Preparation** è una fase molto importante poiché serve a modificare il set di dati individuato precedentemente in modo da migliorarlo qualitativamente e renderlo interpretabile dal modello. In questa fase vengono applicati processi di data cleaning(pulizia da dati mancanti o rumorosi) e successivamente si passa al feature engineering attraverso il quale vengono effettuate delle operazioni di selezione o estrazione sulle feature, inoltre i dati vengono formattati in modo da poter essere interpretati dal modello selezionato.

- Nella fase di **Modeling** si effettua la configurazione e il train del modello scelto in Business Understanding sul dataset risultante dalla precedente fase.
- L'**Evaluation** è un passo fondamentale nella definizione di un buon modello di Machine Learning perché in essa possiamo valutare i risultati ottenuti dal modello e vedere quanto quest'ultimo sia consistente e coerente con gli obiettivi prefissati quindi, in base alla valutazione effettuata, modificare le fasi precedenti in modo da migliorare il modello.
- Una volta concluso il processo di definizione e valutazione del modello si passa all'ultima fase del processo ovvero il **Deployment** attraverso cui si definisce come integrare il modello in un prodotto software e renderlo usabile[26].

Poiché la fase di Business Understanding e Data Acquisition sono già state affrontate precedentemente si passa direttamente alla descrizione della Data Preparation.

### 3.3.1 Feature engineering

Dato che è stata già precedentemente descritta la formattazione del dataset tramite la One-Hot Encoding in modo da renderlo interpretabile dai modelli e considerando anche che il dataset di partenza non presentava dati mancanti o rumorosi sarà successivamente descritto il processo di feature engineering, in particolare da quali tecniche è costituito e quali sono state applicate nel sistema sviluppato.

#### Feature Scaling

Il Feature Scaling è una tecnica applicata alle feature del dataset che consente di normalizzare o scalare l'insieme dei valori della feature in analisi. La motivazione di questa operazione nasce dal fatto che feature con un range di valori molto diverso potrebbero far confondere l'algoritmo, ovvero esso potrebbe sovrastimare o sottostimare l'importanza di una caratteristica se ha una scala di valori inferiore/superiore alle altre.

I principali algoritmi di Scaling sono Min-Max Normalization che normalizza i dati in un range da 0 a 1 tramite una formula che considera appunto i valori di massimo e minimo presenti e il più utilizzato Z-Score che considera la media della distribuzione e la deviazione standard per assegnare i nuovi valori alla feature in analisi.

Poichè il dataset in analisi è stato formattato tramite la One-Hot Encoding tutte le feature avranno valore 0 o 1 di conseguenza non è stato applicato nessun algoritmo di scaling[10].

### Feature Selection

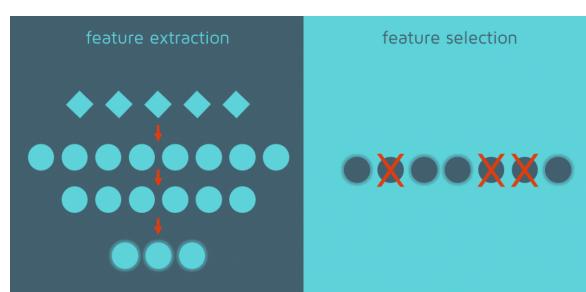
Tramite questo processo vengono selezionate le caratteristiche più correlate al problema in analisi. La metodologia più seguita per l'eliminazione di feature segue un approccio non supervisionato, in particolare si eliminano feature con bassa varianza, ossia hanno valori simili nel dataset di conseguenza non ha senso mantenerle entrambe; inoltre si può utilizzare un altro metodo più elaborato che prevede la selezione delle variabili sulla base di test statistici: vengono effettuati dei test di correlazione tra tutte le variabili indipendenti e la variabile dipendente ottenendo una classifica delle feature basata sulla correlazione, di conseguenza è possibile selezionare le  $k$  migliori. Quest'ultima tecnica è stata utilizzata all'interno della sperimentazione (SelectKBest della libreria sklearn) poiché il numero di feature è molto elevato quindi si potrebbe avere un peggioramento in termini di tempo soprattutto per gli algoritmi quantistici[10].

### Feature Extraction

La Feature Extraction, o estrazione delle caratteristiche, può essere utilizzata nel contesto dell'Intelligenza Artificiale per ridurre la complessità e fornire una rappresentazione più semplice dei dati. Questo processo può sembrare simile alla Feature Selection ma presenta differenze molto importanti poiché con la Feature Extraction vengono generate delle caratteristiche diverse da quelle del dataset di partenza espresse come combinazioni lineari delle originali, l'algoritmo più utilizzato in questo ambito (utilizzato anche nella sperimentazione) è PCA.

Di conseguenza con la Feature Extraction abbiamo un nuovo set di caratteristiche mentre con la Feature Selection abbiamo un sottoinsieme delle caratteristiche originali.

Questo processo è stato utilizzato nella sperimentazione nella fase successiva alla Feature Selection poiché lo scopo è stato sempre quello di diminuire di complessità il dataset ma non escludendo la capacità informativa di molte feature[17].



**Figura 3.2:** Confronto tra Feature Extraction e Feature Selection

### Data balancing

Il data balancing comprende l'insieme di tecniche che hanno come obiettivo convertire un set di dati sbilanciato in uno bilanciato. Ciò vuol dire avere lo stesso numero di campioni per ogni classe della variabile dipendente. I metodi per bilanciare un dataset sono due: Undersampling con cui vengono eliminate casualmente un numero di istanze della classe di maggioranza dal dataset e Oversampling con cui invece vengono aggiunte casualmente delle istanze della classe minoritaria al dataset. Ci sono degli svantaggi da tenere in considerazione in entrambi i metodi: per quanto riguarda l'Undersampling è possibile che alcuni campioni rilevanti vengano eliminati oppure se abbiamo un dataset con poche istanze eliminandone altre non avremo un modello adeguato al problema; nel caso dell'Oversampling l'unico problema possibile è quello dell'overfitting ossia che il modello non è elastico nel riconoscere l'appartenenza di un campione ad una classe ma si comporta come se avesse imparato "a memoria" i valori di una classe [10]. Nonostante il problema dell'overfitting l'Oversampling resta il metodo più utilizzato, nella sperimentazione è stato utilizzato l'algoritmo SMOTE che utilizza a sua volta l'algoritmo KNN per poter creare nuove istanze dei campioni ed aggiungerle al dataset.

#### 3.3.2 Definizione dei modelli

I modelli quantistici utilizzati sono stati sviluppati parametrizzando quelli offerti dalla libreria Qiskit di IBM.

Per tutti i modelli, come è ovvio che sia, è necessario effettuare un mapping tra dati classici, quindi binari, e quantistici ottenuti tramite l'utilizzo di una Feature Map che ci permette appunto di effettuare il mapping dei dati ed eseguire gli algoritmi quantistici. Per ogni modello saranno illustrate le principali componenti necessarie per l'esecuzione ed il corretto funzionamento.

#### Quantum Support Vector Classifier

Il QSVC, o anche Quantum Support Vector Classifier, è la versione quantistica del SVC con kernel "linear". A tal proposito è stato necessario definire un kernel quantistico per il funzionamento dell'algoritmo, quest'ultimo come il kernel classico ha lo scopo di aumentare la dimensione dello spazio delle feature in modo da poter effettuare la classificazione.

### Pegasos QSVC

Questo tipo di algoritmo è una variante del QSVC che risulta essere più veloce in fase di train del modello. Infatti studi condotti da ricercatori sono riusciti ad ottenere una complessità lineare per l’addestramento indipendente dalla dimensione del dataset, questo algoritmo è infatti ottimo per dataset di grandi dimensioni[29]. Allo stesso modo è necessario definire la Feature Map, il Kernel quantistico e successivamente si può proseguire con la fase di classificazione.

### Quantum Neural Network

La QNN è una rete neurale quantistica. Esistono diverse versioni nella libreria di riferimento ma è stata utilizzata la CircuitQNN che è la più versatile e parametrizzabile. Nel contesto delle reti neurali la definizione è leggermente più complessa poiché è necessario definire un ansatz, che avrà lo scopo di assegnare casualmente dei pesi ai campioni alla prima iterazione della rete neurale. Successivamente viene definito un circuito quantistico che prende in input sia l’ansatz che la Feature Map. Viene successivamente definita la rete neurale che ha già di default le funzioni di forward e backward; la rete neurale basata su un circuito quantistico viene poi data in input al NeuralNetworkClassifier, un oggetto utilizzato da tutte le versioni di reti neurali per poter eseguire effettivamente la classificazione, si può infatti specificare: l’ottimizzatore, il numero di iterazioni che la rete dovrà eseguire prima di arrestarsi e anche la loss function.

#### 3.3.3 Valutazione

Un’operazione molto importante nella progettazione di un modello di Machine Learning è la Valutazione del modello. Oltre agli obiettivi descritti in precedenza in questa fase si divide il dataset in due parti chiamate Train-Set e Test-Set, in base a dei criteri descritti successivamente, in modo da effettuare il train del modello con il Train-Set e valutarlo sul Test-Set. La motivazione di questa divisione nasce dal fatto che valutare un modello sugli stessi dati usati per il train è un grosso errore che porta a dei risultati totalmente inaffidabili, tecnicamente questo fenomeno è definito data leakage che si verifica quando un modello è molto accurato in fase di training ma non al rilascio.

È importante notare che la divisione del dataset deve essere effettuata prima che quest’ultimo subisca modifiche da parte del progettista per fare in modo di testare il modello con dati che potrebbe effettivamente dover valutare dopo il deploy e che non ha mai incontrato

in precedenza. Esistono principalmente due tecniche per la Validazione di un modello: Simple-Split e K-fold Validation.

### Simple Split

Questo metodo ha un funzionamento molto semplice: si prende casualmente una percentuale di campioni del dataset per il train e la restante per il test, ovviamente ha senso avere la parte del train più numerosa rispetto all'altra, è stato infatti stimato che i valori ottimali di queste percentuali sono in generale 70-30 e 80-20.

### K-Fold validation

La K-Fold Validation o convalida incrociata opera mischiando i dati di partenza in maniera casuale e poi dividerli in  $k$  gruppi, successivamente in modo iterativo viene scelto uno dei  $k$  gruppi e lo si considera test-set, i restanti  $k-1$  gruppi rappresentano il train-set quindi si continua addestrando il modello e valutandolo. Questa tecnica ha senso perché considerando diversi gruppi di train e test-set otterremo modelli con prestazioni diverse partendo dallo stesso dataset, di conseguenza al termine delle iterazioni si va a considerare il modello che ha ottenuto le prestazioni migliori.

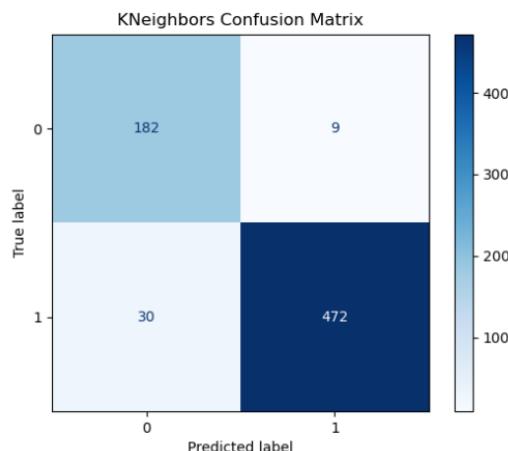
Nella sperimentazione sono state usate entrambe le tecniche poiché per gli algoritmi quantistici la convalida incrociata risultava troppo dispendiosa in termini di tempo per cui quest'ultima è stata usata solo per i modelli classici.

## 3.4 Sperimentazione

Per poter valutare il sistema di Malware Detection quantistico sviluppato sono stati utilizzati diversi approcci, oltre al normale ed ovvio calcolo delle metriche di classificazione ottenute dai diversi modelli sono stati definiti degli algoritmi di Machine Learning classici che consentono un effettivo confronto di prestazioni tra gli algoritmi classici e quantistici basati sullo stesso dataset su cui sono state applicate tecniche di Preprocessing simili.

### 3.4.1 Metriche utilizzate

Per quanto concerne le metriche si considerano ovviamente solo quelle relative alla classificazione quindi senza addentrarsi nell'esposizione di metriche riguardanti clustering o regressione. Le seguenti sono state calcolate per ogni modello sia classico che quantistico utilizzando la classica matrice di confusione. Come è possibile osservare dalla Figura 3.4 tramite la confusion matrix è possibile valutare la bontà di un classificatore andando a studiare True Positive(TP), False Positive(FP), True Negative(TN) ed infine False Negative(FN). Questi ultimi vengono utilizzati per il calcolo delle metriche che saranno presentate successivamente.



**Figura 3.3:** Esempio matrice di confusione

### Accuracy

L'accuracy è la metrica più utilizzata per valutare la bontà di un modello ma è necessario fare una precisazione riguardo le informazioni che può dare. Dalla Figura 3.5 si osserva che essa rappresenta la percentuale di campioni correttamente valutati sia positivi che negativi. Diventa quindi importante osservare il bilanciamento del dataset poiché se quest'ultimo

fosse sbilanciato il modello potrebbe essere molto bravo a riconoscere una classe mentre invece un'altra no, quindi avere una bassa precision.

### Precision

La precision rappresenta la capacità di un modello di valutare correttamente i campioni di una determinata classe rispetto a tutte le predizioni fatte dal classificatore.

### Recall

La recall indica il numero di predizioni valutate correttamente per la classe rispetto a tutte le istanze calcolate per quella classe.

### F1 Score

L'f1 score viene utilizzata per combinare sia precision che recall in un'unica metrica soprattutto nel caso in cui ci sia un dataset sbilanciato.

Le precedenti metriche sono state utilizzate nella sperimentazione per valutare i modelli e nella Figura 3.5 si può osservare la formula per ognuna di esse.

$$\begin{aligned}precision &= \frac{TP}{TP + FP} \\recall &= \frac{TP}{TP + FN} \\F1 &= \frac{2 \times precision \times recall}{precision + recall} \\accuracy &= \frac{TP + TN}{TP + FN + TN + FP}\end{aligned}$$

**Figura 3.4:** Calcolo metriche da matrice di confusione

### 3.4.2 Algoritmi classici

Per quanto riguarda gli algoritmi classici sono stati utilizzati e parametrizzati i seguenti offerti dalla libreria sklearn: Random Forest, Decision Tree, KNeighbors, Naive Bayes ed infine LinearSVC per il quale esiste la controparte quantistica quindi offre un buon metro di giudizio. I risultati ottenuti da tutti gli algoritmi, sia classici che quantistici, con le relative tecniche di validazione e preprocessing saranno presentati e commentati nel Capitolo 4.

## CAPITOLO 4

---

### Risultati Ottenuti

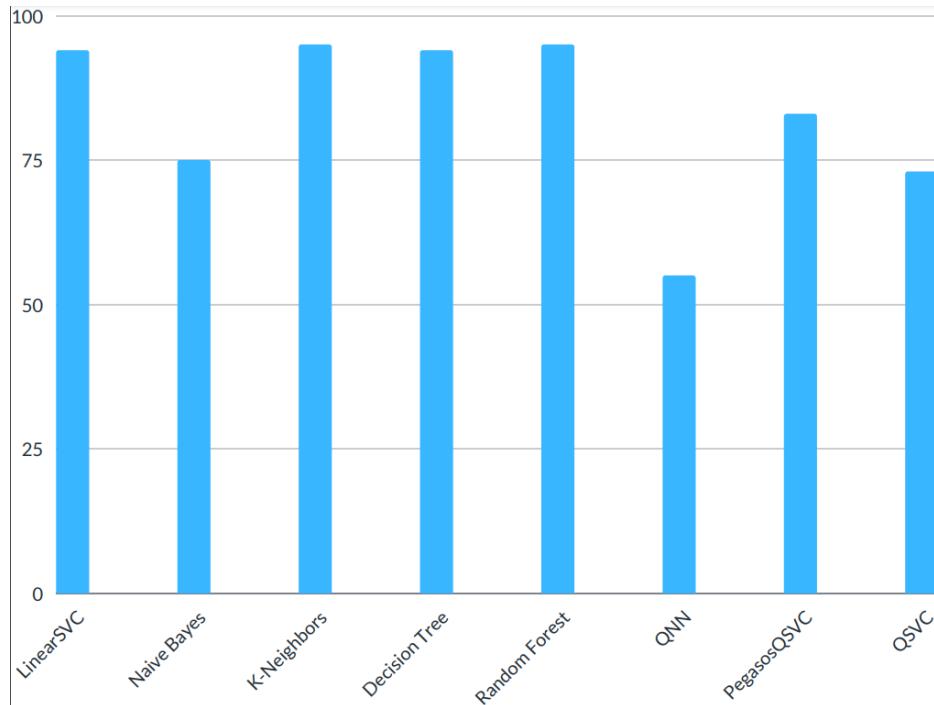
---

In questo capitolo, in Tabella 4.1, saranno presentati tutti gli esperimenti effettuati con le relative tecniche di Preprocessing e l'algoritmo di classificazione utilizzato.

Classifier	Validazione	FS	FE	Time	Accuracy	Precision	Recall	F1
Naive Bayes	10Fold	750	250	0.3	75	70	69	70
K-Neighbors	10Fold	750	250	2.03	95	93	96	95
Decision Tree	10Fold	750	250	2.48	94	92	93	93
Linear SVC	10Fold	750	250	6.70	94	91	93	92
Random Forest	10Fold	750	250	10.3	95	94	95	94
QNN	SimpleSplit	750	250	36946	55	58	57	54
PegasosQSVC	SimpleSplit	750	250	10821	83	81	76	78
QSVC	SimpleSplit	750	250	25349	73	72	74	71

**Tabella 4.1:** Risultati ottenuti

Dai risultati ottenuti, interpretabili anche dalla Figura 4.1, si osserva che gli algoritmi quantistici sono in grado di gestire questo tipo di dataset ottenendo performance mediocri e tempi purtroppo molto lunghi per una possibile applicazione concreta. Di notevole interesse risulta l'esperimento effettuato utilizzando l'algoritmo PegasosQSVC, questo infatti è stato



**Figura 4.1:** Grafico rappresentante l'accuracy dei modelli utilizzati

il più veloce degli algoritmi quantistici utilizzati ed ha ottenuto anche le metriche migliori rispetto sempre ai modelli quantistici. La motivazione deriva dalla complessità lineare del training del modello che quindi ha impiegato la maggior parte del tempo per effettuare il testing. Osservando in particolare l'accuracy il PegasosQSVC risulta essere un modello valido nonostante la complessità del dataset utilizzato e con le dovute migliorie potrebbe pareggiare gli algoritmi classici ed avere un'accuracy accettabile per lo scopo della sperimentazione.

Gli algoritmi classici invece, tranne per il Naive Bayes che è un algoritmo di tipo probabilistico quindi non prende in considerazione la correlazione e i pattern tra le feature, ottengono prestazioni interessanti sia in termini di tempo che in termini di performance. Infatti il Naive Bayes è l'unico algoritmo classico ad aver ottenuto un'accuracy inferiore all'80%, per quanto riguarda i restanti algoritmi hanno invece ottenuto metriche buone poiché tutti con un'accuracy pari al 94% o 95%. Oltre all'accuracy è da tenere in considerazione anche la percentuale di False Positive Ratio, discussa nel capitolo 2, di particolare rilevanza per sistemi di Malware Detection; questa risulta essere per tutti gli algoritmi sia classici che quantistici, tranne per QNN, mediamente pari all'1% un dato interessante ma mediocre poiché per sistemi di questo tipo deve essere più vicina possibile allo 0. Tuttavia è interessante notare come su questo fronte gli algoritmi classici e quantistici abbiano ottenuto risultati quasi simili. Riguardo le altre metriche si possono analizzare la Precision e la Recall che ci consentono di capire

---

la percentuale di campioni rappresentanti malware riconosciuti correttamente dal modello calcolati però sul totale di istanze positive predette dal modello (True Positive + False Positive) per quanto riguarda la Precision e sulle istanze positive reali (True Positive + False Negative), quindi presenti nel dataset, nel caso della Recall. Entrambe queste metriche hanno valori piuttosto vicini in quasi tutti i modelli, risulta rilevante però la differenza del 5% che è presente nel caso del PegasosQSVC, questo significa che il modello ha prodotto meno False Positive rispetto ai False Negative. Per avere un'idea più chiara e precisa del rapporto tra i risultati di Precision e Recall si può far riferimento all'F1 che è una metrica creata proprio a questo scopo infatti rappresenta una media armonica delle due. Nella maggior parte dei casi questa risulta essere molto vicina all'accuracy del modello tranne per il Naive Bayes e PegasosQSVC dove l'accuracy è maggiore del 5%, di conseguenza questo significa che il modello ha ottenuto un'accuracy maggiore grazie alle istanze negative predette correttamente.

Per quanto riguarda i modelli classici possono essere confrontati osservando il tempo impiegato per effettuare train e test: rientrano tutti nell'ordine dei secondi ma è possibile notare una differenza con LinearSVC e Random Forest che risultano essere i più lenti. La motivazione per quanto riguarda il LinearSVC deriva dal calcolo del kernel mentre invece il Random Forest per definizione è costituito da diverse iterazioni di Decision Tree di conseguenza è inevitabile un aumento del tempo di esecuzione.

Visti i risultati ottenuti si può affermare che per questo particolare tipo di dataset gli algoritmi quantistici non sono ancora in grado di pareggiare i classici ma hanno dimostrato almeno di poter gestire dataset grandi. La motivazione per cui gli algoritmi quantistici ottengano prestazioni mediocri è attribuibile al fatto che per poter funzionare sono state tagliate fuori molte feature da un dataset che presentava 1000 feature totali. Questo taglio così massiccio non ha fatto altro che togliere informazioni rilevanti per una classificazione performante, nonostante ciò gli algoritmi quantistici di questa sperimentazione allenati su un dataset complesso e numeroso risultano essere migliori rispetto a quelli presentati nello stato dell'arte. Infatti è possibile notare che in letteratura gli esperimenti effettuati su dataset complessi e di grandi dimensioni in ambito Cybersecurity abbiano ottenuto tutti un'accuracy inferiore al 60% (il migliore ha ottenuto 58%), di conseguenza si può affermare che la sperimentazione effettuata abbia portato valore alla tesi sull'applicabilità degli algoritmi quantistici in ambito Cybersecurity.

Per quanto riguarda i tempi di esecuzione molto lunghi sono dovuti innanzitutto ai tempi di coda per l'accesso all'hardware quantistico e ai tempi di esecuzione effettiva che comunque superano di gran lunga l'ordine dei secondi degli algoritmi classici. Inoltre si deve tener

---

conto anche della decoerenza, un fenomeno fisico presente in sistemi quantistici rumorosi che altera lo stato di un qubit e rende falsata l'osservazione quindi il calcolo del valore del qubit è da ripetere causando un prolungamento del tempo di esecuzione.

Nonostante le limitazioni dell'attuale calcolo quantistico e delle difficoltà di accesso ai sistemi quantistici questo studio conferma e avvalora l'ipotesi che il Quantum Machine Learning sia in grado di trattare dataset di grandi dimensioni nell'ambito Cybersecurity.

# CAPITOLO 5

---

## Conclusioni

---

Questa tesi ha analizzato diversi ambiti dell'informatica tra cui la Cybersecurity che sta avendo un'importanza sempre maggiore in ambito sia aziendale che privato, il Machine Learning grazie al quale i sistemi software riescono ad effettuare predizioni intelligenti e il Quantum Computing, un'area non ancora del tutto ben definita e accessibile alla maggior parte delle persone. In ambito Cybersecurity è stato osservato come una mancata gestione del rischio possa causare danni economici o perdite di dati soprattutto a causa di intrusioni nei sistemi e diffusione di malware. A tal proposito una delle principali difficoltà della Cybersecurity è il Malware Detection, che mette alla prova ricercatori e professionisti per perfezionare i sistemi di rilevamento. Il Machine Learning, campo altrettanto importante, ha portato una grande innovazione e un forte miglioramento dei sistemi software rendendoli intelligenti e capaci di effettuare predizioni e personalizzazioni. Non meno importante il Quantum Computing propone un'evoluzione radicale dell'informatica promettendo una capacità di calcolo senza eguali rendendo eseguibili problemi difficili dell'informatica classica.

Lo scopo dell'elaborato, oltre a presentare questi tre grandi e diversi settori e vedere come questi possono interagire tra di loro, è stato quello di testare algoritmi di Quantum Machine Learning su un dataset per la Malware Detection. Guardando lo stato dell'arte presente in letteratura si è potuto osservare che i suddetti algoritmi si comportano bene e meglio rispetto agli algoritmi classici con dataset di dimensioni ridotte, quando invece si ha a che fare con dataset di taglia più grande sono stati riscontrati problemi in termini di performance ma soprattutto di tempo che li rendono inapplicabili in un contesto reale. Le motivazioni, già di-

---

scusse in precedenza, sono attribuibili alla poca facilità di accesso e ai problemi di calcolo che ci sono in ambito quantistico entrambi derivanti dal fatto che il Quantum Machine Learning sia un campo recente e quindi ancora con delle limitazioni. A tal proposito è doveroso pensare ai possibili sviluppi futuri della sperimentazione effettuata, infatti con un aggiornamento dei sistemi quantistici disponibili ed utilizzabili, principalmente con un incremento dei qubit disponibili, sarà possibile allenare i modelli quantistici con un numero maggiore di feature e quindi si può pensare ad un incremento delle performance degli algoritmi utilizzati. Tuttavia è stato dimostrato che ciò nonostante gli algoritmi quantistici sono stati in grado di trattare dataset di grandi dimensioni e che quindi in un futuro non troppo lontano, con i dovuti miglioramenti e accorgimenti, si potrebbe pensare ad un'applicazione di algoritmi quantistici in un contesto reale.

---

## Bibliografia

---

- [1] Hyrum S. Anderson and Phil Roth. EMBER: an open dataset for training static PE malware machine learning models. *CoRR*, abs/1804.04637, 2018. (Citato alle pagine 37 e 38)
- [2] Zahra Bazrafshan, Hashem Hashemi, Seyed Mehdi Hazrati Fard, and Ali Hamzeh. A survey on heuristic malware detection techniques. In *The 5th Conference on Information and Knowledge Technology*, pages 113–120, 2013. (Citato alle pagine 27 e 28)
- [3] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, Sep 2017. (Citato a pagina 23)
- [4] Swami Chandrasekaran Christopher Havenstein, Damarcus Thomas. Comparisons of performance between quantum and classical machine learning. *SMU Data Science Review*, 1(4), 2018. (Citato alle pagine 32 e 33)
- [5] Clusit. Rapporto clusit 2022 sulla sicurezza ict in italia, 2022. (Citato alle pagine 1 e 2)
- [6] CyberDivision. Suggerimenti per difendersi dagli attacchi informatici. (Citato a pagina 14)
- [7] Cybersecurity360. Cyber security: cos'è e come garantire la sicurezza dei sistemi informatici e delle reti, 2018. (Citato alle pagine 13 e 14)
- [8] Cybersecurity360. Malware: cosa sono, come riconoscerli e come rimuoverli, 2019. (Citato a pagina 12)

- [9] Digital4. Cyber security, i vantaggi dei servizi di sicurezza gestiti (managed services), 2021. (Citato a pagina 13)
- [10] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems.* " O'Reilly Media, Inc.", 2019. (Citato alle pagine 20, 21, 40, 42, 43 e 44)
- [11] Sukhpal Singh Gill, Adarsh Kumar, Harvinder Singh, Manmeet Singh, Kamalpreet Kaur, Muhammad Usman, and Rajkumar Buyya. Quantum computing: A taxonomy, systematic review and future directions. *Software: Practice and Experience*, 52(1):66–114, 2022. (Citato a pagina 23)
- [12] Red Hat. What is malware?, 2018. (Citato alle pagine 10, 11 e 12)
- [13] A. Mani Kandan, G. JaspheWillsie Kathrine, and Alfred Raja Melvin. Network attacks and prevention techniques - a study. In *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pages 1–6, 2019. (Citato a pagina 6)
- [14] Kaspersky. What is spoofing – definition and explanation. (Citato a pagina 9)
- [15] Kaspersky-Lab. Machine learning for malware detection. 2021. (Citato alle pagine 30 e 31)
- [16] William L. Simon Kevin D. Mitnick. *L'arte dell'inganno.* Feltrinelli, 2019. (Citato a pagina 13)
- [17] Samina Khalid, Tehmina Khalil, and Shamila Nasreen. A survey of feature selection and feature extraction techniques in machine learning. In *2014 Science and Information Conference*, pages 372–378, 2014. (Citato a pagina 43)
- [18] Youngjoon Ki, Eunjin Kim, and Huy Kang Kim. A novel approach to detect malware based on api call sequence analysis. *International Journal of Distributed Sensor Networks*, 11(6):659101, 2015. (Citato alle pagine 27 e 29)
- [19] Mohammad Masum, Mohammad Nazim, Md Jobair Hossain Faruk, Hossain Shahriar, Maria Valero, Md Abdullah Hafiz Khan, Gias Uddin, Shabir Barzanjeh, Erhan Saglamyurek, Akond Rahman, and Sheikh Iqbal Ahamed. Quantum machine learning for software supply chain attacks: How far can we go?, 2022. (Citato alle pagine 32, 33 e 34)
- [20] Microsoft. Formato pe. 2022.

- [21] Microsoft. Understanding quantum computing. 2022. (Citato a pagina 25)
- [22] IBM Security Ponemon Institute. Cost of a data breach report 2021. Technical report, Ponemon Institute, IBM Security, 2021. (Citato alle pagine 5 e 6)
- [23] John Preskill. Quantum computing and the entanglement frontier. Technical report, California Institute of Technology, 2011. (Citato a pagina 26)
- [24] John Preskill. Quantum computing 40 years later. Technical report, California Institute of Technology, 2021. (Citato a pagina 24)
- [25] Azar Salih, Subhi T. Zeebaree, Sadeeq Ameen, Ahmed Alkhyyat, and Hnan M. Shukur. A survey on the role of artificial intelligence, machine learning and deep learning for cybersecurity attack detection. In *2021 7th International Engineering Conference "Research Innovation amid Global Pandemic" (IEC)*, pages 61–66, 2021. (Citato a pagina 19)
- [26] Jeff Saltz. 3 steps to define an effective data science process, 2020. (Citato a pagina 42)
- [27] Iqbal H. Sarker. Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, 2(3), 2021. (Citato a pagina 16)
- [28] Burr Settles. From theories to queries: Active learning in practice. In Isabelle Guyon, Gavin Cawley, Gideon Dror, Vincent Lemaire, and Alexander Statnikov, editors, *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, volume 16 of *Proceedings of Machine Learning Research*, pages 1–18, Sardinia, Italy, 16 May 2011. PMLR. (Citato a pagina 18)
- [29] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming*, 127(1):3–30, 2011. (Citato a pagina 45)
- [30] Ashu Sharma and Sanjay Kumar Sahay. Evolution and detection of polymorphic and metamorphic malwares: A survey. *arXiv preprint arXiv:1406.7061*, 2014. (Citato alle pagine 29 e 30)
- [31] Kalpit G Soni and Atul Patel. Comparative analysis of k-means and k-medoids algorithm on iris data. *International Journal of Computational Intelligence Research*, 13(5):899–906, 2017. (Citato a pagina 21)
- [32] Carol Woody and Rita Creel. Lessons learned in building and implementing an effective cybersecurity strategy. Acquisition Research Program, 2021. (Citato a pagina 14)

- [33] Nishant Yadav, Gagandeep Kaur, Sukhwinder Kaur, Anshu Vashisth, and Cheerala Rohith. A complete study on malware types and detecting ransomware using api calls. In *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pages 1–5, 2021. (Citato alle pagine 9, 10, 11 e 12)
- [34] Yongli Zhang. Support vector machine classification algorithm and its application. In Chunfeng Liu, Leizhen Wang, and Aimin Yang, editors, *Information Computing and Applications*, pages 179–186, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. (Citato a pagina 20)

---

## Ringraziamenti

---

INSERIRE RINGRAZIAMENTI QUI