RdS
RICERCA DI SISTEMA

ENEA
AGENZIA NAZIONALE PER LE NUOVE TECNOLOGIE, L'ENERGIA E LO SVILUPPO ECONOMICO SOSTENIBILE

MINISTERO DELL'AMBIENTE E DELLA SICUREZZA ENERGETICA

Ricerca di Sistema elettrico

Studio di modelli di ML per la detezione di cyber-attacchi in sistemi energetici attraverso l'analisi statistica del dato misurato su nodi cyber-fisici (LA3.11)

S. De Vito, E. Esposito

RdS_PTR 22-24_PR 2.1_LA3.11_070

STUDIO DI MODELLI DI ML PER LA DETEZIONE DI CYBER-ATTACCHI IN SISTEMI ENERGETICI ATTRAVERSO L'ANALISI STATISTICA DEL DATO MISURATO SU NODI CYBER-FISICI

Autori Saverio De Vito, Ph.D. (ENEA); Elena Esposito, Ph.D. (ENEA)

09/2023

Report Ricerca di Sistema Elettrico

Accordo di Programma Ministero dell'Ambiente e della Sicurezza Energetica - ENEA
Piano Triennale di Realizzazione 2022-2024

Obiettivo: *Decarbonizzazione/Digitalizzazione ed evoluzione delle reti*
Progetto: *tema di ricerca (come da capitolato)*
Linea di attività: *3.11 (come da capitolato)*
Responsabile del Progetto: Maria Valenti, ENEA
Responsabile Linea di Attività: Saverio De Vito, ENEA
Mese inizio previsto: 06/2022
Mese inizio effettivo: 06/2022
Mese fine previsto: 06/2023
Mese fine effettivo: 06/2023

Indice

1 Risultati attesi

Si riporta di seguito la lista dei risultati attesi come da capitolato vigente:

- Studio e descrizione di almeno 10 modelli per applicazioni di detezione cyber attacchi descritti negli ultimi 15 anni al fine della selezione degli algoritmi per il toolset da sviluppare e testare nella LA3.12 e nella LA3.13.

2 Risultati ottenuti

Studio e descrizione di una lista di modelli recenti per applicazioni di anomaly detection in sistemi cyberfisici e successivo focus su algoritmi di detezione cyber attacchi. Per la selezione degli studi ci si è concentrati su risultati molto recenti per l'accelerazione decisa che la produzione scientifica ha registrato in questi ultimi cinque anni nel dominio di riferimento. In particolare, per gli utenti che insistono su microgriglie di tipo isolato o meno, i risultati ottenuti concorreranno all'incremento della resilienza agli attacchi cyber in termini di capacità di detezione precoce e quindi di reazione temporalmente efficace per limitare danni e disservizi.

3 Prodotti sviluppati

La LA3.11 non prevede lo sviluppo di prodotti hardware/software.

4 Analisi degli scostamenti su attività e risultati

Non si sono registrati scostamenti tecnico nell'ambito della LA3.11. Dal punto di vista economico ed in particolare in termini di assegnazione e rendicontazione del personale coinvolto è da segnalarsi l' impiego di personale tecnico. Diversamente da quanto inizialmente previsto, personale appartenente al ruolo tecnico è stato impiegato per l' implementazione di specifiche azioni relative alla ricerca, alla verifica tecnica preliminare (incluse analisi della strutturazione di formato e, successivamente, di copertura e verifica dell' etichettatura) e all' acquisizione dei dataset pubblicamente disponibili oltre che per la realizzazione fisica della repository "data lake" da utilizzarsi per lo sviluppo delle attività successive.

5 Sintesi delle attività svolte

Nella LA3.11 è stato condotto uno studio selettivo di algoritmi di anomaly detection su nodi cyber fisici di architetture a microgriglia. Gli algoritmi, le tecnologie e le architetture proposti dalla letteratura, sono stati analizzati in chiave critica ponendo in risalto le proposte più recenti basate fondamentalmente su tecniche statistiche e machine learning. Sono stati identificati i tipi di attacchi riportati come più frequenti o pericolosi in base alle analisi di rischio. Sono state analizzate le basi di dati utilizzati ed in particolare la metodologia di raccolta degli stessi, distinguendo tra dataset simulati, ottenuti da sistemi di simulazione in tempo reale anche HIL ed infine ottenuti utilizzando almeno in parte misure in campo su sottosistemi rilevanti di microgriglia reali. Infine, sono state identificate le architetture tipiche di microgriglia tipicamente utilizzate in questi studi ed in particolare in quelli in cui si simulavano queste architetture in tempo reale.

6 Dettaglio delle attività svolte

6.1 Studio critico degli algoritmi di machine learning per la detezione di anomalie legate a cyber-attacchi

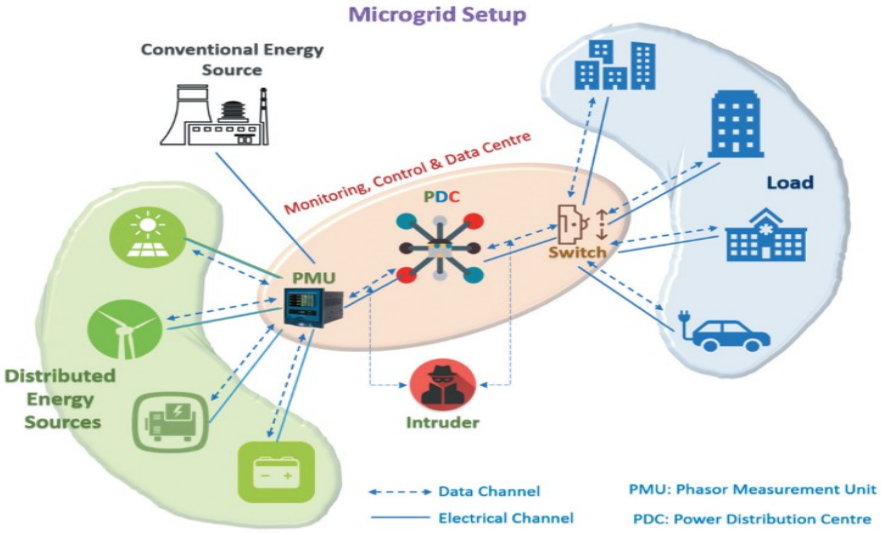
Una definizione semplicistica di microgriglia (microgriglia) riflette un piccolo sistema di potenza che copra sia gli aspetti di generazione che di consumo lavorando in modalità connessa oppure isolata. In realtà tali sistemi da tempo sono evoluti in complessi sistemi dove singole componenti di potenza interagiscono scambiandosi informazioni vitali per il mantenimento del bilancio tra consumi e generazione/inception mentre ottimizzano il funzionamento globale dell'intero sistema seguendo requisiti multi-obiettivo. Pertanto, una microgriglia può essere sicuramente visualizzata con funzionalità multilivello in cui le differenti appliances comunicano efficacemente per realizzare obiettivi specifici mediante l'azione di coordinamento di un livello decisionale anch'esso, almeno in potenza, distribuito. Quest'ultimo governa l'intera rete e ne regola le interazioni con l'esterno rendendo de facto possibile l'interazione attiva sul mercato dell'energia. L'elevato grado di coordinazione necessario al funzionamento ottimale e la criticità per i sistemi di potenza attuali ne fanno un obiettivo moto appetibile in termini di cyberattacchi. Gli smart meters in particolare catturano informazioni vitali riguardo i consumi e sono un punto ad elevata vulnerabilità e, se compromessi, possono portare la microgriglia al collasso o, in casi meno severi, ad operare lontano dal punto ottimale. Oltre che in base alla gravità gli attacchi possono essere suddivisi in attacchi volti ad operare un completo arresto in maniera immediata portando il sistema ad un rapido collasso attraverso la compromissione dei meccanismi primari di decisione e di bilancio energetico (e.g. portando ad un rapido shutdown per impossibilità di risposta alla domanda o per la generazione di set point impossibili da implementare) oppure attacchi stealth con meccanismi di compromissione lenti che possono evolvere in eventi catastrofici oppure rimanere silenti con perdite economiche rilevabili nel lungo periodo. In ogni caso, l'elevata dipendenza dagli smart meters, l'eterogeneità delle sorgenti dati, l'alto throughput trasmissivo in correlazione con un'alta sensitività alla sincronizzazione temporale costituiscono delle criticità e delle debolezze intrinseche nei confronti di queste minacce.

Di fatto, con la diffusione delle microgriglia sono emerse infatti preoccupazioni significative riguardo i possibili cyber attacchi cui possono essere sottoposti. In effetti. attacchi di tipo worm (e.g. Stuxnet) hanno già mostrato la loro capacità di danneggiare parti significative della rete di potenza industriale. Attacchi della tipologia False data injection nei sistemi di potenza sono d'altra parte riferiti già dal 2007. Pertanto, è ragionevole aspettarsi una crescita significativa di cyberattacchi alle microgriglie che stanno evolvendo in sistemi critici per l'infrastruttura inerentemente distribuita di cui fanno parte. L'identificazione precoce degli attacchi e l'implementazione di strategie di resilienza sono dunque obiettivi primari per garantire l'ottimale utilizzo delle risorse della microgriglia. L'attività si è dunque focalizzata sull'analisi degli approcci esistenti in letteratura per effettuare una scrematura volta all'implementazione di algoritmi specifici da integrare nella catena di elaborazione prevista nel progetto.

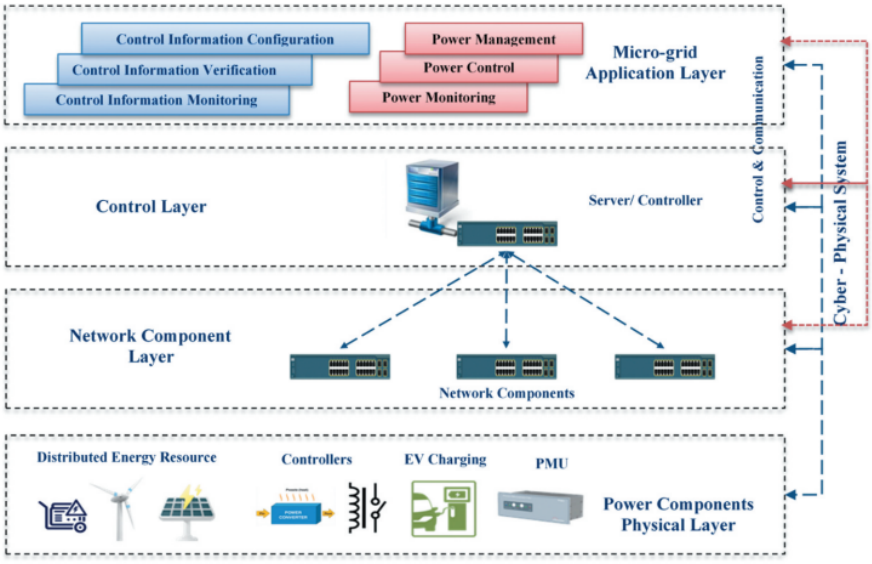
A priori sono stati analizzati gli effetti degli attacchi. Analizzare a fondo gli effetti di attacchi all'integrità dei dati trasmessi a sistemi di potenza distribuiti aiuta in effetti a neutralizzare gli effetti più severi dei cyber attacchi (Duan et al., 2018). Caratteristiche intrinseche di resilienza possono essere potenziate con l'utilizzo di hardware o processi di controllo volte a migliorare le performance dei sistemi di intrusion detection. Liu et al., utilizzano regolatori secondari di frequenza appunto per migliorare la resilienza quando gli IDS mostrano un eccessivo tasso di falsi positivi (Liu et al., 2021). In caso di attacchi DoS è possibile migliorare la resilienza in maniera simile ma utilizzando approcci di reinforcement learning distribuito che cercando di massimizzare un ottimo globale in tempo reale permettono di ottenere il risultato voluto anche sotto attacco.

Come vedremo gran parte degli approcci utilizzati hanno come obiettivo la rilevazione di attacchi False Data Injection (FDI) considerati generalmente la base fondamentale di tutti gli attacchi con effetti potenzialmente più gravi. Gli attacchi FDI manipolano i dati effettivamente trasmessi dai sistemi di monitoraggio

concentrandosi in particolare sui sottosistemi di smart metering ma in realtà è facile prevedere una crescita di attacchi aventi come obiettivo il numero crescente di sensori e attuatori presenti in tali architetture. La detezione di tali attacchi può essere condotta sulla base di componenti per la rilevazione di anomalie. Un approccio forse semplicistico ma relativamente efficace è quello presentato in (Duraijai et al, 2020) dove basandosi su di un modello di rilevazione gaussiano si introducono dei limiti ai valori dei parametri misurati. Mentre dunque il panorama della letteratura offre copiosi spunti all'analisi di tali problemi per i sistemi di potenza e le reti di trasporto in generale, gli approcci specifici alle applicazioni delle microgriglia sono significativamente meno diffusi. Un esempio di interesse è riportato in (Beg et al., 2017), dove appunto lo scenario di un attacco FDI è riportato in una DC microgriglia. In pratica le anomalie sono basate sul confronto tra valori attesi e valori effettivamente trasmessi e catturati in rete. L'approccio è limitato in quanto non permette la valutazione della severità e la profondità degli attacchi rilevati.



(a)



(b)

Figura 1: Esempio di architettura microgriglia con esempio di superficie di attacco (a) e livelli architetturali (b).

In Kavousi et al., 2021,si analizzano invece i diversi sottosistemi target in modo tale da valutare la severità dell'attacco subito e la criticità per l'intero sistema. Potenzialmente questo potrebbe portare a reazioni target specifiche rendendo in questo modo la microgriglia più resiliente. Anche in questo approccio la differenza tra

valori predetti in termini di intervalli valoriali e valori effettivamente rilevati e trasmessi porta alla rilevazione di anomalie. La predizione è effettuata con un approccio Symbiotic Organisms Search di chiara ispirazione biomimetica utilizzato per il tuning di reti neurali feed forward miranti a definire intervalli di predizione di consumi. Se il valore misurato (false data) risulta esterno all'intervallo di predizione viene attivata l'allerta di cyberattacco.

In (Danalakshmi et al., 2021) un approccio misto basato sulla concatenazione di una Deep Belief Network e un sistema a regole viene utilizzato per l'identificazione di attacchi FDI usando il Dataset Industrial Control Cyberattacks Dataset. L'approccio viene testato in contrasto con approcci CNN mostrando un generale allineamento prestazionale o un leggero vantaggio in taluni casi.

In (Sadi et al., 2020), gli autori analizzano un caso basato su IEEE bus 39 simulato in Matlab/Simulink con la connessione di due turbine eoliche come generatori. Attacchi di tipo FDI e DoS (in termini di hacking dei set point di impianto) vengono simulati al fine di costituire un dataset per l'addestramento di approcci machine learning o statistici-adattativi. Una soluzione basata su reti tapped delay operanti sulle correlazioni tra i segnali di voltaggio è risultata più efficace di un sistema encoder-decoder operanti sugli stessi ingressi e di un sistema statistico operante sulle oscillazioni delle frequenze generate dagli attacchi. È interessante notare che il dataset comprende anche casi di alterazioni temporanee dei valori nominali dovuti a contingenze del funzionamento ordinario.

Nell'interessante contributo di (Ma et al.,2022) gli autori realizzano un simulatore hw di microgriglia caratterizzato da diversi sistemi di generazione e consumo DC basati su Raspberry4 e controllori *Arduino* che pilotano coppie di motori dc usati da generatori ed un carico resistivo. Un attacco Man in the Middle è utilizzato per modificare i set point di Voltaggio dei controllori rendendoli simili tra diversi sistemi di generazione e consumo. L'attacco è naturalmente stealth. Differenti sistemi ML sono utilizzati per l'analisi diretta di caratteristiche di voltaggio e delay tra i pacchetti ipotizzando effetti sia sul sistema cyberfisico che su quello di comunicazione. L'algoritmo Random Forest è risultato il migliore considerando i classici indici di riferimento di Precision, Recall e Accuracy nel problema di detezione dell'attacco. Benché semplice nell'approccio globale, il lavoro è un interessante esempio di analisi simultanea degli effetti e delle firme di attacco sia sul sottosistema di comunicazione che su quello cyberfisico.

Al di fuori del contesto specifico della cybersecurity Khaledian et al. propongono l'utilizzo di diversi sistemi di clustering non supervisionato e classificatori basati su correlazioni statistiche per l'identificazione di anomalie nei dati sincrofasoriali da PMU distribuiti. Essi ne immaginano l'utilizzo anche nel contesto in cui quest'attività si muove. L'identificazione e la gestione di anomalie viene effettuata attraverso numerosi steps che coinvolgono sia aspetti di identificazione non supervisionata che l'applicazione di conoscenze dell'infrastruttura quando si cercano correlazioni tra i dati e le anomalie identificate in diversi PMU. Queste correlazioni permettono di isolare PMU malfunzionanti o identificare eventi a scala di smart grid (vedi figura).

Gallo et al. e (Tang et al., 2018) affrontano il problema del FDI attraverso l'istituzione di tecniche di watermarking distribuito e rispettivamente di identificazione attraverso GANs. Utilizzando un approccio simulativo e, rispettivamente di misura, di misura su dati sincrofasoriali o di consumo costruiscono dei dataset in grado di addestrare e validare le loro metodologie. Il confronto è semanticamente interessante poiché il primo che garantisce una efficacia maggiore necessita dell'implementazione a bordo del singolo PMU si un meccanismo di watermarking del dato.

In conclusione, si produce una tabella (cfr. Tabella 1) di analisi tassonomica degli approcci considerati utilizzando codici colore per facilitare la comprensione della situazione allo stato dell'arte e delle caratteristiche di rilievo che hanno fatto includere il lavoro nella selezione. Quasi la totalità dei lavori utilizza simulatori, rarissimamente di tipo HIL, per la produzione di dataset in cui vengono ricreate condizioni ed effetti di un attacco. Sono estremamente rari i dati da componenti reali di microgriglie, in questo caso poi, gli attacchi sono simulati iniettando anomalie nel dataset. La maggior parte degli attacchi simulati sono di tipo FDI o in alternativa di tipo DoS. Rarissimamente si affronta il caso di iniezione di codice malevolo in generale avente come target componentistica inverter. L'utilizzo di tecniche machine learning è pressoché ubiquitario almeno negli aspetti di detezione, mentre lo è in maniera moderata per gli approcci di resilienza. Sistemi basati su tecniche tradizionali vengono di recente affiancati da sistemi deep learning come autoencoders, convolutional neural network e in qualche sporadico caso di tipo transformers. A seguito

dell'analisi delle risultanze fin qui ottenute si procederà alla selezione degli algoritmi da implementare nella successiva attività in programma. Infine, in tabella 2 vengono riportati alcuni lavori di review a beneficio del lettore. Allo stesso scopo in appendice è riportato un breve excursus sugli algoritmi di detezione delle anomalie.

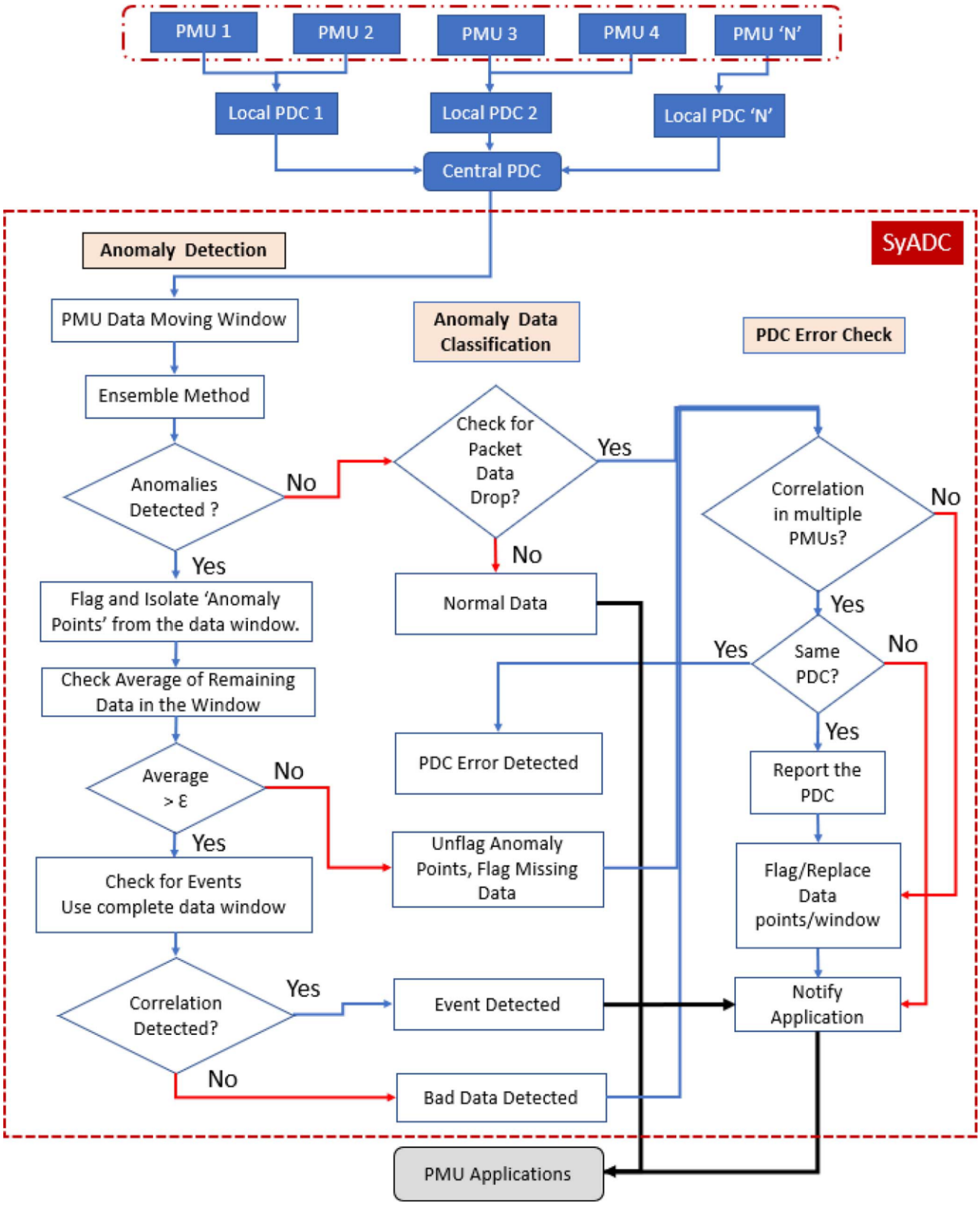


Figura 2: Schema funzionale di tratttamento anomalie in (KHALEDIAN et al.: 2020)

TABELLA 1: SCHEMA TASSONOMICO DEGLI APPROCCI ANALIZZATI IN LETTERATURA

Paper	Metodo	Tipologia di attacco	Tipologia di dataset
Toker et al. 2022	Utilizzo di autoencoder come strumento per l' anomaly detection.	False Data Injection	Utilizzo di dati da controllore di frequenza carico (Load Frequency Control) in microgriglie isolate. Dataset da misure reali.
Takiddin et al., 2022	Utilizzo di autoencoder ricorsivi su dati di corrente e voltaggio.	False Data Injection/Replay	Dati da testbed con microgriglia DC, Dataset da misure reali.
Panthi et al., 2020	Utilizzo di RF, Naïve Bayes, J-Ripper.	False Data Injection, Hijacking di sub componenti.	IEEE 3 Bus System dataset da simulazione.
Marino et al., 2019	Framework basato su sensori virtuali	Diverse tipologie di attacco.	Test su CPS SCADA Cybersecurity Testbed (ISAAC) dataset da simulazione
Choi et al., 2023	Utilizzo di una combinazione di Gaussian Processes Regression e 1-Class SVM su microgriglia simulata HIL	False Data Injection	Utilizzo dati di generazione distribuita (frequenza). Il classificatore lavora su pattern di differenza tra valori predetti e misurati. Il generatore è simulato tramite Opal-RT. dataset da simulazione
Durairaj et al., 2022	Approccio misto Anomaly detection DL/Rule Based. La parte DL è costituita da una DBN addestrata in modalità non supervisionata.	False Data injection (and DoS)	Dataset pubblico Industrial Intrusion Detection Dataset + Alcuni IEEE Bus Systems dataset da simulazione+reali
Kavousi-Fard et al., 2020	FFNN per prevedere l' incertezza di un metodo LUBE. Aggregazione e training con differenti metodologie.	Data injection attacks in AMI	Dataset con aattacchi simulate superimposti a dati aggregati da power meters in centinaia di utilizzatori Dataset da misure reali.
Tang et al., 2021	GAN incentrata su un metodo LUBE	False Data injection attacks in AMI	Dataset da 300+ abitazioni con focus sulla parte consumi (smart meters) Dataset da misure reali.
Cui et al., 2020	Ensemble basato su auto encoders che analizzano dati	False Data Injection in AMI	Focalizzato su MG di tipo DC. Anomalie identificate sulla base dei segnali di

	da trasformata Hilbert del segnale monitorato.		voltaggio di riferimento da singoli componenti AMI. Veloce (10 ms detection time) e preciso (95% cc rate) . Estremamente generico nella definizione del dataset sebbene mostri il sistema simulato e la tipologia di attacchi FDI. Dataset non adeguatamente descritto
Dehgani et al., 2021	Ensemble basato su auto encoders che analizzano dati SVD da trasformata wavelet del segnale monitorato.	False Data Injection in AMI	Focalizzato su MG di tipo DC. Anomalie identificate sulla base dei segnali di voltaggio di riferimento da singoli componenti AMI. Veloce (10 ms detection time) e preciso (95% cc rate) . Estremamente generico nella definizione del dataset sebbene mostri il sistema simulato e la tipologia di attacchi FDI. Dataset non adeguatamente descritto
Ma et al., 2020	Algoritmi ML classici operanti su segnali di voltaggio e caratteristiche dello stream dati (latenza pacchetti) in caso di attacco MiM volto alla replica di segnali di voltaggio acquisiti in una piccola microgriglia (4 DGs)	Stealth FDI	Dataset creato con simulazione HW in the loop di microgriglia DC con generatori simulati da coppie di motori elettrici DC, Dataset da misure reali.
Sadi et al., 2022	Rete shallow di tipo tapped delay operante su segnali di voltaggio in una rete power grid, confrontato con approcci auto-encoder e basati su clustering.	Stealthy FDI	Dataset simulativo su modello IEEE Bus System 39. Dataset da simulazione
Gallo et al., 2018	Detezione di anomalie basato su un metodo di watermarking distribuito con segnali randomici per evitare attacchi replay.	Data integrity attacks	Dataset da simulatore real time di singole unità di generazione distribuita. Dataset da simulazione
Xi et al., 2020	Framework integrale per la detezione di intrusion e classificazione delle tipologie di attacco basato su una CNN.	Data integrity and DOS	Semplice proposta di utilizzo di strutture CNN per la rilevazione attacchi. La proposta include la generazione di firme

			sensoriali da attacchi ma non propone la costruzione di un dataset ne presenta risultati Dataset non adeguatamente descritto
Kuruvila et al., 2021	Differenti metodologie di machine learning basate su una feature selection di tipo PCA	Malicious code injection	Usa un Sistema HW adHoc con microcontroller isolato con firmware da inverter TI solar. Quattro differenti modifiche al sw relative a diversi tipi di attacco generano firme differenti in un HPC simulato connesso al microcontrollore sotto attacco. Queste vengono identificate da componenti ML (DT, RF, NN).
Khaledian et al., 2021	Framework integrale per la detezione di anomalie basate su differenti tipologie di machine learning incluse Isolation Forests, KMeans, LoOP con combinazione e analisi di correlazione.	Fault detection	Basato su simulazione del bus IEEE 688 Dataset da simulazione

TABELLA 2: LAVORI DI REVIEW RECENTE DI INTERESSE PER L'APPLICAZIONE SPECIFICA.

De Dutta and Prasad, 2020	Review con focus architetturale/SCADA e riferimenti a protocolli, standards e tipologie di vulnerabilità identificate.
Nejabatkhah et al., 2021	Review focalizzata su sistemi di rilevazione FDI con attacchi perpetrati verso diverse funzionalità di microgriglia (Voltage set points, Frequency, State Estimation, ecc.)
Jamil et al., 2021	Review con focus architetturale/SCADA e riferimenti a protocolli, standards e tipologie di vulnerabilità identificate. Discute i possibili divari di ricerca.

RIFERIMENTI BIBLIOGRAFICI

(Beg et al., 2017) O.A... Beg; T.T. Johnson; A. Davoudi, "Detection of False-Data Injection Attacks in Cyber-Physical DC Microgrids", IEEE Trans. Industrial Informatics, vol. 13, no. 5, pp. 2693-2703, 2017.

(Cui et al., 2021) H. Cui, X. Dong, H. Deng, M. Dehghani, K. Alsubhi and H. M. A. Aljahdali, "Cyber Attack Detection Process in Sensor of DC Micro-Grids Under Electric Vehicle Based on Hilbert–Huang Transform and Deep Learning," in IEEE Sensors Journal, vol. 21, no. 14, pp. 15885-15894, 15 July15, 2021, doi: 10.1109/JSEN.2020.3027778.

(Choi et al., 2023) Choi, J., Roshanzadeh, B., Martínez-Ramón, M., Bidram, A.: An unsupervised cyberattack detection scheme for AC microgrids using Gaussian process regression and one-class support vector machine anomaly detection. IET Renew. Power Gener. 17, 2113–2123 (2023). https://doi.org/10.1049/rpg2.12753

(Durairaj et al, 2022) Danalakshmi Durairaj, Thiruppathy Kesavan Venkatasamy, Abolfazl Mehbodniya, Syed Umar & Tanweer Alam (22 Jan 2022): Intrusion detection and mitigation of attacks in microgrid using enhanced deep belief network, Energy Sources, Part A: Recovery, Utilization, and Environmental Effects

(De Dutta and Prasad, 2020) S. De Dutta and R. Prasad, "Cybersecurity for Microgrid," 2020 23rd International Symposium on Wireless Personal Multimedia Communications (WPMC), Okayama, Japan, 2020, pp. 1-5, doi: 10.1109/WPMC50192.2020.9309494.

(Dehghani et al., 2021) Dehghani, M.; Niknam, T.; Ghiasi, M.; Bayati, N.; Savaghebi, M. Cyber-attack detection in dc microgrids based on deep machine learning and wavelet singular values approach. Electronics 2021, 10, 1914.

(Durairaj et al., 2022) Durairaj, D.; Venkatasamy, T.K.; Mehbodniya, A.; Umar, S.; Alam, T. Intrusion detection and mitigation of attacks in microgrid using enhanced deep belief network. Energy Sources Part A Recover. Util. Environ. Eff. 2022, 44, 1–23.

(Duan et al., 2018) J. Duan; W. Zeng ; M.Y. Chow, "Resilient Distributed DC Optimal Power Flow Against Data Integrity Attack", IEEE Trans. Smart Grid, vol. 9, no. 4, pp. 3543 – 3552, 2018.

(Yang et al., 2017) Q. Yang, D. Li ; W. Yu ; Y. Liu ; D. An ; X. Yang ; J. Lin, "Toward Data Integrity Attacks Against Optimal Power Flow in Smart Grid", IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1726 – 1738, 2017

(Gallo et al., 2018) Gallo, A.J.; Turan, M.S.; Boem, F.; Ferrari-Trecate, G.; Parisini, T. Distributed watermarking for secure control of microgrids under replay attacks. IFAC-PapersOnLine 2018, 51, 182–187.

(Jamil et al., 2021) Jamil, N.; Qassim, Q.S.; Bohani, F.A.; Mansor, M.; Ramachandaramurthy, V.K. Cybersecurity of Microgrid: State-of-the-Art Review and Possible Directions of Future Research. Appl. Sci. 2021, 11, 9812. https://doi.org/10.3390/app11219812

(Khaledian et al., 2021) KHALEDIAN et al.: REAL-TIME SYNCHROPHASOR DATA ANOMALY DETECTION AND CLASSIFICATION USING ISOLATION FOREST, KMEANS, AND LoOP, IEEE Transactions on Smart Grid, 2021, doi:10.1109/tsg.2020.3046602

(Kavousi-Fard et al., 2021) A. Kavousi-Fard, W. Su and T. Jin, "A Machine-Learning-Based Cyber Attack Detection Model for Wireless Sensor Networks in Microgrids," in IEEE Transactions on Industrial Informatics, vol. 17, no. 1, pp. 650-658, Jan. 2021, doi: 10.1109/TII.2020.2964704.

(Kuruvila, A.P. et al., 2021) Kuruvila, A.P.; Zografopoulos, I.; Basu, K.; Konstantinou, C. Hardware-assisted detection of firmware attacks in inverter-based cyberphysical microgrids. Int. J. Electr. Power Energy Syst. 2021, 132, 107150.

(Liu et al., 2021) Liu, S.; Siano, P.; Wang, X. Intrusion-detector-dependent frequency regulation for microgrids

(Ma et al., 2020) Ma, M.; Lahmadi, A.; Chrisment, I. Detecting a Stealthy Attack in Distributed Control for Microgrids using Machine Learning Algorithms. In Proceedings of the 2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS), Tampere, Finland,10–12 June 2020; IEEE: Piscataway, NJ, USA, 2020; Volume 1, pp. 143–148.

(Marino et al., 2019) D. L. Marino et al., "Cyber and Physical Anomaly Detection in Smart-Grids," 2019 Resilience Week (RWS), San Antonio, TX, USA, 2019, pp. 187-193, doi: 10.1109/RWS47064.2019.8972003.

(Nejabatkhah et al., 2021) Nejabatkhah, F.; Li, Y.W.; Liang, H.; Reza Ahrabi, R. Cyber-Security of Smart Microgrids: A Survey. Energies 2021, 14, 27. https://doi.org/10.3390/en14010027

(Panthi et al., 2020) M. Panthi, "Anomaly Detection in Smart Grids using Machine Learning Techniques," 2020 First International Conference on Power, Control and Computing Technologies (ICPC2T), Raipur, India, 2020, pp. 220-222, doi: 10.1109/ICPC2T48082.2020.9071434.

(Sadi et al., 2022) Sadi, M.A.H.; Zhao, D.; Hong, T.; Ali, M.H. Time Sequence Machine Learning-Based Data Intrusion Detection for Smart Voltage Source Converter-Enabled Power Grid. IEEE Syst. J. 2022 , 16.

(Takiddin et al., 2022) A. Takiddin, S. Rath, M. Ismail, and S. Sahoo, "Data-Driven Detection of Stealth Cyber-Attacks in DC Microgrids," in IEEE Systems Journal, vol. 16, no. 4, pp. 6097-6106, Dec. 2022, doi: 10.1109/JSYST.2022.3183140.

(Tang et al., 2018) Tang, Z.; Lin, Y.; Vosoogh, M.; Parsa, N.; Baziar, A.; Khan, B. Securing microgrid optimal energy management using deep generative model. IEEE Access 2021, 9, 63377–63387., Gallo, A.J.; Turan, M.S.; Boem, F.; Ferrari-Trecate, G.; Parisini, T. Distributed watermarking for secure control of microgrids under replay attacks. IFAC-PapersOnLine 2018, 51, 182–187.

(Toker et al., 2022) O. Toker and M. R. Khalghani, "Cyber Anomaly Detection Design for Microgrids using an Artificial Intelligent-Based Method," 2022 North American Power Symposium (NAPS), Salt Lake City, UT, USA, 2022, pp. 1-5, doi: 10.1109/NAPS56150.2022.10012203.

(Xi et al., 2020) Xi, W.; He, S.; Chen, R.; Xu, Y.; Li, W.; Zhou, G.; Yu, W.; He, H.; Huang, Z.; Yu, Y.; et al. Research on attack detection method of microgrid central controller based on convolutional neural network. In Journal of Physics: Conference Series; IOP Publishing: Bristol, UK, 2020; Volume 1646, p. 012076.

Contributo delle eventuali consulenze alle attività sopra descritte

7 Pubblicazioni scientifiche

Elenco delle pubblicazioni scientifiche eventualmente risultanti dall'attività svolta

N/A

8 Eventi di disseminazione

Lista degli eventi di disseminazione eventualmente scaturiti dall'attività svolta

N/A

Appendice: Studio dei processi di Anomaly detection

L'identificazione e la classificazione delle anomalie sono processi fondamentali nell'analisi dei dati sia come funzionalità proprie sia come base per differenti applicazioni. Il dato anomalo, denominato spesso come outlier, può rappresentare eventi rari, comportamenti anomali o errori nei dati che differiscono significativamente dal normale pattern di comportamento. L'obiettivo principale di queste metodologie è individuare e caratterizzare tali anomalie per comprendere meglio i dati. L'anomalia può essere innescata nel confronto tra pattern temporali, ad esempio nel caso di andamenti marcatamente differenti dalla normale evoluzione di uno o più segnali caratterizzata da indicatori statistici che ne catturino il comportamento, ad esempio, cicliostazionario della v.a. in analisi. Nel campo specifico di applicazione la rilevazione di pattern anomali è messa in relazione. Altresi essa può essere caratterizzata in relazione a pattern di risposta anche istantanei di differenti v.a. fra di loro correlate o comunque connesse.
Più specificatamente: una anomalia può essere definita come un pattern nei dati che non è conforme al normale andamento del resto dei dati (Figura 1).

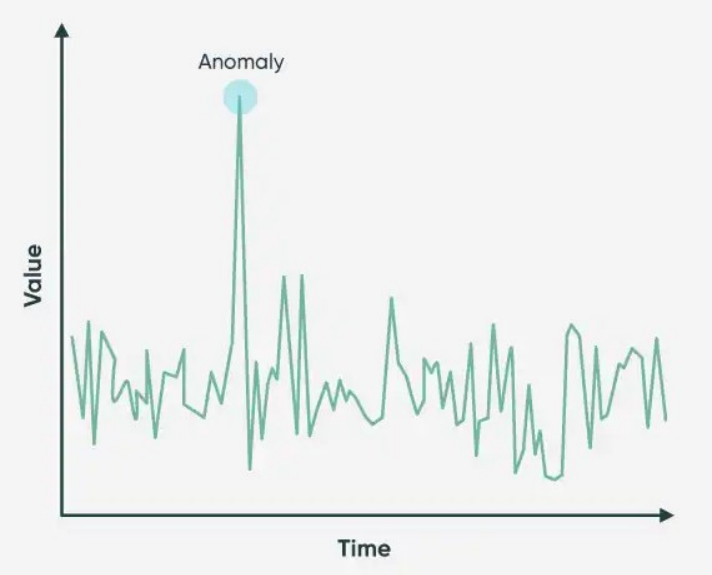


Figura 1: Esempio di anomalia o outlier presente in un dataset.

Nell'ambito della nostra attività si decide per effettuare la seguente classificazione:

Anomalie puntuali: è il tipo di anomalia più semplice, in quanto consiste nello stabilire se un singolo dato risulta anomalo rispetto al resto del dataset. Un esempio di anomalia puntuale all'interno del contesto applicativo di riferimento può essere l'alterazione del pattern specifico di un sincrofasore in una microrete causato da un attore che voglia imporre una diversa fasatura causando eventuali malfunzionamenti.

Anomalie Contestuali: un dato può risultare anomalo in un determinato contesto, ma normale in contesti diversi. Dunque, questo tipo di anomalia tiene conto delle interazioni o delle relazioni tra le diverse variabili presenti nel dataset. A differenza delle anomalie puntuali, le anomalie contestuali si verificano quando una osservazione è anomala rispetto alle interazioni o alle relazioni attese tra le variabili coinvolte.

Anomalie collettive: un sottoinsieme di dati è anomalo rispetto all'intero dataset. Il singolo dato in una anomalia collettiva potrebbe non essere anomalo, ma il verificarsi di esso insieme ad altri correlati costituisce un'anomalia.
Queste anomalie possono essere più complicate da identificare rispetto quelle puntuali, in quanto richiedono la rilevazione di una deviazione collettiva piuttosto che di singoli punti. Ad esempio, un cluster di osservazioni

che si discosta significativamente dalla distribuzione principale può essere considerato un'anomalia collettiva. Nel caso in esame, potrebbe consistere nella rilevazione di una produzione energetica significativamente inferiore rispetto alle aspettative, che potrebbe essere causata da un guasto indotto da un attacco o una iniezione di dati fasulli nel sistema di collezione e analisi dati.

Requisito fondamentale per la validazione di algoritmi per la identificazione delle anomalie è l'etichettatura dei dati. Le etichette che vengono associate ad un'istanza di dati da un soggetto esterno e indicano se quell'istanza è normale o anomala. L'etichettatura, dunque, spesso è eseguita manualmente da un esperto e richiede un enorme sforzo in termini di risorse umane. Etichettare i dati in modo da coprire tutti i tipi di comportamenti anomali è difficile, dato che il comportamento anomalo è spesso di natura dinamica. In base alla misura in cui le etichette sono disponibili, le tecniche di rilevamento delle anomalie possono operare in tre diverse modalità.
La prima è la cosiddetta Supervised anomaly detection, in cui le tecniche di apprendimento supervisionato necessitano di un dataset di training etichettato per classi normali e anomale. L'approccio utilizzato solitamente è consiste nel costruire un modello predittivo per le due classi e confrontare ogni istanza di dati nuova con il modello per determinare la classe di appartenenza. Ciò comporta l'insorgere di due problemi, ovvero la numerosità ridotta delle istanze anomale rispetto a quelle normali e l'ottenimento di etichette accurate e precise. CI si trova dunque in un contesto con dati inerentemente sbilanciati (unbalanced)
La seconda modalità è il Semi-Supervised anomaly detection, in cui si assume che il dataset di training sia etichettato solo per la classe normale, per questo motivo sono più utilizzate delle tecniche supervisionate. L'approccio usato solitamente è quello di costruire un modello per la classe normale e, tramite esso, identificare le anomalie nei dati di test.
L'ultima modalità è l'Unsupervised anomaly detection, che non richiede dati di training etichettati e dunque, risulta essere l'approccio più utilizzato. Tali tecniche operano sotto l'assunzione che le istanze normali siano molto più frequenti di quelle anomale. Se questa assunzione è falsa, allora tali tecniche generano un alto tasso di falsi allarmi. Molte delle tecniche semi-supervisionate possono essere adattate per operare in modo non supervisionato utilizzando un dataset non etichettato per il training.
La scelta della modalità dipende ovviamente dal tipo di dataset su cui si deve operare e dunque, dal tipo di sistema al quale si deve applicare l'AD. In ogni caso, fondamentale e il modo in cui le anomalie vengono segnalate. In generale, l'output prodotto può essere di due tipi:

- Score: a ogni istanza dei dati di test viene assegnato un punteggio di anomalia in base al grado in cui quell'istanza è considerata un'anomalia. Si potrebbe analizzare le prime anomalie o utilizzare una soglia per selezionarle.
- Etichette: a ogni istanza dei dati di test viene assegnata un'etichetta (normale o anormale).

Si può quindi utilizzare una soglia per selezionare le anomalie più rilevanti. In alcune tecniche, l'output è solo un'etichetta binaria e si possono modificare i parametri per selezionare le anomalie.

Tecniche di anomaly detection

Le tecniche di anomaly detection possono essere divise in diverse tipologie, ma negli ultimi anni si seguono principalmente due approcci entrambi di tipo statistico ma caratterizzate da un differente complessità delle ipotesi di partenza:

Tecniche di tipo statistico con ipotesi generative strutturate

Nelle tecniche di tipo statistico *un'anomalia è un'osservazione che si sospetta essere parzialmente o totalmente irrilevante perché non generata dal modello stocastico assunto*. Queste tecniche si basano sull'assunzione che le istanze di dati normali si verificano nelle regioni ad alta probabilità di un modello stocastico, mentre le anomalie si verificano nelle regioni a bassa probabilità. Queste tecniche adattano un modello statistico ai dati e applicano un test di inferenza statistica per determinare se un'istanza non

esaminata precedentemente appartenga a questo modello oppure no. Le istanze con bassa probabilità di essere generate dal modello appreso sono dichiarate come anomalie.
Per adattare un modello statistico esistono tecniche parametriche e non parametriche. Le prime presuppongono la conoscenza della distribuzione sottostante e stimano i parametri dai dati forniti; le seconde, invece, non presuppongono la conoscenza della distribuzione sottostante. La complessità computazionale dipende dal modello statistico che deve essere adattato ai dati. Per singole distribuzioni parametriche della famiglia esponenziale la complessità è tipicamente lineare nella dimensione dei dati e nel numero di attributi. Per le distribuzioni complesse essa è tipicamente lineare per iterazione anche se tali distribuzioni potrebbero essere lente a convergere. Queste tecniche hanno i seguenti vantaggi e svantaggi:
Vantaggi: forniscono una soluzione statistica giustificabile; l'anomaly score fornito è associato ad un intervallo di confidenza utilizzabile come informazione aggiuntiva quando si prende una decisione sul test; infine, possono operare in un ambiente non supervisionato.
Svantaggi: si basano sull'assunzione che i dati sono generati da una particolare distribuzione, ma questo, nella maggior parte dei casi, non è vero o risulta complesso da verificare.

Tecniche di Machine Learning

Come già introdotto, le tecniche di ML sono basate sull'apprendimento automatico, che si divide in due grandi categorie: l'apprendimento supervisionato (Supervised Learning) e l'apprendimento non supervisionato (Unsupervised Learning). Un altro campo che negli ultimi anni sta dando un enorme contributo è l'apprendimento per rinforzo (Reinforcement Learning).
Nell'apprendimento supervisionato, l'obiettivo è inferire una funzione o mappatura dai dati di allenamento (training data) etichettati. I dati di allenamento sono costituiti dal vettore di input X e dal vettore di output Y di etichette o tag. Un'etichetta o tag dal vettore Y è la spiegazione del suo rispettivo esempio di input dal vettore X. Insieme formano un training example. Per provare il risultato di un modello e le sue performance, si utilizza un altro sottoinsieme di dati dal dataset chiamato test set. Training set e test set sono quindi sottoinsiemi dei dati iniziali (cfr. Mohammed et al., 2016) . Due gruppi o categorie di algoritmi rientrano nell'apprendimento supervisionato: la classificazione e la regressione. Nella classificazione l'obiettivo è quello di costruire un modello conciso della distribuzione delle etichette di classe in termini di features predittive. Il classificatore risultante viene quindi utilizzato per assegnare le etichette di classe alle istanze di test in cui sono noti i valori delle features predittive, ma il valore dell'etichetta di classe è sconosciuto (cfr. Kotsiantis et al., 2007). Nel nostro contesto applicativo la rilevazione anomalie è fondamentalmente un processo di classificazione a due classi.
Nell'ambito dell'apprendimento supervisionato rientrano le tecniche di AD di tipo classification-based: Support Vector Machine (SVM), gli alberi decisionali (Decisione Tree), le reti Bayesiane e le Neural Networks (NN). Le Support vector machines (SVM) è un algoritmo che ruota attorno alla nozione di "margine", ai lati di un iperpiano che separa due classi di dati. È stato dimostrato che l'ottimizzazione del margine e la creazione della massima distanza possibile tra l'iperpiano di separazione e le istanze su entrambi i lati, riducono il limite superiore dell'errore di generalizzazione previsto. Se è possibile separare linearmente due classi, è possibile trovare un iperpiano di separazione ottimale minimizzando la norma quadrata dell'iperpiano di separazione. Nel caso di dati separabili linearmente, una volta trovato l'iperpiano di separazione ottimale, i punti dati che giacciono sul suo margine sono noti come punti vettore di supporto e la soluzione è rappresentata come una combinazione lineare di solo questi punti. Gli alberi decisionali (Decision Tree) sono alberi che classificano le istanze ordinandole in base ai valori delle features. Ogni nodo in un albero decisionale rappresenta una feature in un'istanza da classificare e ogni ramo rappresenta un valore che il nodo può assumere. Le istanze sono classificate a partire dal nodo principale e ordinate in base ai loro valori delle features. Il classificatore Naive Bayes è uno dei più rappresentativi algoritmi di statistical learning. Le Naive Bayes network (NB) sono reti bayesiane molto semplici che sono composte da grafici aciclici diretti con un solo genitore (che rappresenta il nodo non osservato) e diversi nodi figli (corrispondenti a nodi osservati) con una forte

assunzione di indipendenza tra i nodi figli nel contesto del loro genitore. Il modello di indipendenza (Naive Bayes) si basa sulla stima:

$$R = \frac{P(i \vee X)}{P(j \vee X)} = \frac{P(i)P(X \vee i)}{P(j)P(X \vee j)} = \frac{P(i)\prod P(X_r \vee i)}{P(j)\prod P(X_r \vee j)}.$$

Confrontando queste due probabilità, la probabilità maggiore indica il valore dell'etichetta di classe più probabile che sia l'etichetta effettiva (se R > 1: predice i altrimenti predice j). Poiché questo algoritmo di classificazione utilizza una moltiplicazione per calcolare le probabilità P(X, i), è particolar mente incline a essere influenzato da una probabilità 0. L'ipotesi di indipendenza tra i nodi figli è chiaramente quasi sempre sbagliata e per questo motivo i classificatori Bayesiani sono di solito meno accurati rispetto ad altri algoritmi di apprendimento.Una rete neurale artificiale (ANN) è un modello di calcolo ispirato alla struttura delle reti neurali nel cervello. L'idea alla base delle reti neurali artificiali è che molti neuroni possono essere uniti tramite collegamenti di comunicazione per eseguire calcoli complessi. È comune descrivere la struttura di una rete neurale come un grafo i cui nodi sono i neuroni e ogni arco (diretto) nel grafo collega l'output di

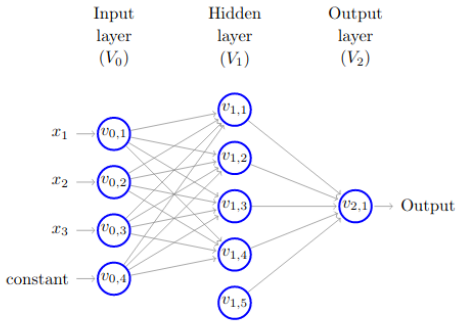


Figure 2: Esempio di una ANN feed-forward: V0 ,V1 ,V2 sono i tre strati della rete, vi,j rappresenta il neurone j-esimo dello strato i e xn i valori d'ingresso all'input layer

alcuni neuroni all'ingresso di un altro neurone. L'apprendimento di una ANN si divide in due parti: una fase di training e una di inferenza. Nella fase di training la rete è addestrata su una serie di dati per determinare la mappatura input-output. I pesi delle connessioni tra i neuroni vengono quindi fissati e la rete viene utilizzata per determinare le classificazioni di un nuovo set di dati. Durante la classificazione, il segnale sulle unità di input si propaga completamente attraverso la rete per determinare i valori di attivazione in tutte le unità di output. Pertanto, la ANN dipende da tre aspetti fondamentali: funzioni di input e attivazione dell'unità, architettura di rete e peso di ciascuna connessione di input. Dato che i primi due aspetti sono fissi, il comportamento della ANN è definito dagli attuali valori dei pesi. I pesi della rete da addestrare vengono inizialmente impostati su valori iniziali spesso casuali; quindi, le istanze del set di allenamento vengono ripetutamente esposte alla rete. I valori per l'input di un'istanza vengono posizionati sulle unità di input e l'output della rete viene confrontato con l'output desiderato per questa istanza utilizzando una funzione errore. Quindi, tutti i pesi nella rete vengono regolati leggermente nella direzione che avvicinerebbe i valori di uscita della rete ai valori per l'uscita desiderata. Nel caso applicativo specifico riteniamo di interesse le architetture convoluzionali e quelle ricorrenti. Le prime specializzate per l'elaborazione di dati multidimensionali (grid data). Esse utilizzano la convoluzione con una matrice chiamata kernel al posto della matrice generale moltiplicativa in almeno uno dei loro strati (cfr. Goodfellow et al., 2016). Le seconde sono una classe di rete neurale artificiale in cui i valori di uscita di uno strato di un livello superiore vengono utilizzati come ingresso ad uno strato di livello inferiore (cfr. Elman et a., 2016). Quest'interconnessione tra strati permette l'utilizzo di uno degli strati come memoria di stato, e consente, fornendo in ingresso una sequenza temporale di valori, di modellarne un comportamento dinamico temporale dipendente dalle informazioni ricevute agli istanti di tempo precedenti (cfr AA.VV., 2018). Tale capacità è ritenuta di estremo interesse per l'analisi di anomalie temporali.

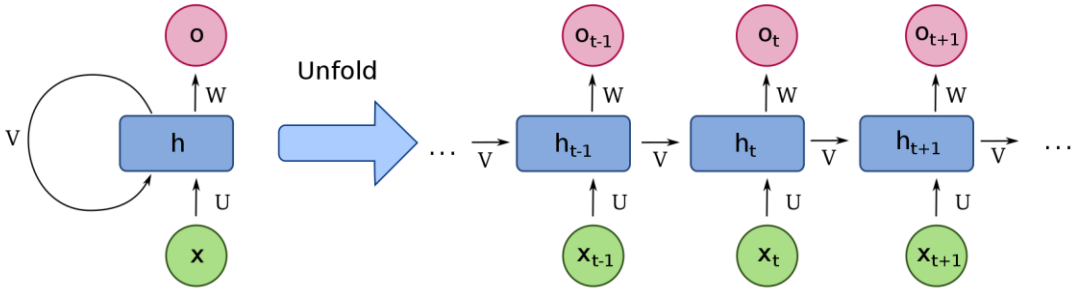


Figure 3: Architettura di una rete neurale ricorrente.

Riguardo le tecniche non supervisionata è bene ricordare come esso miri a rappresentare particolari modelli di input in un modo che rifletta la struttura statistica della raccolta complessiva di pattern di input. Diversamente dall'apprendimento supervisionato o dall'apprendimento rafforzato, non vi sono output target espliciti o valutazioni associate a ciascun input. Tra gli algoritmi non supervisionati vi sono le tecniche di clustering, tecniche statistiche e le reti neurali artificiali (Autoencoder).
I metodi di clustering prevedono la decomposizione diretta di un set di dati in una partizione composta da k cluster disgiunti, indicata con C = C1 , ..., Ck . Questi metodi generalmente cercano di produrre un'approssimazione locale a un globale funzione obiettivo, che viene identificata perfezionando iterativamente una soluzione iniziale. Il k-means è l'algoritmo di clustering partizionale più utilizzato. Impiega uno schema di trasferimento iterativo per produrre un clustering k-vie che minimizza localmente la distorsione tra gli oggetti dati e un set di k rappresentanti dei cluster. Ogni rappresentante, indicato come centroide, è calcolato come vettore medio di tutti gli oggetti assegnati a un determinato cluster
Invece di generare una partizione piatta di dati, può spesso essere utile strutturare una gerarchia di concetti producendo un insieme di cluster nidificati che potrebbero essere disposti a formare una struttura ad albero attraverso gli algoritmi gerarchici. Essi sono generalmente organizzati in due categorie distinte:
• Agglomerativo: iniziare con ciascun oggetto assegnato a un cluster singleton. Viene applicata una strategia dal basso verso l'alto in cui, ad ogni passo, la coppia di cluster più simile è unita.
• Divisive: inizia con un singolo cluster contenente tutti gli n oggetti. Applica una strategia verso il basso in cui, ad ogni passaggio, un cluster scelto viene suddiviso in due sotto cluster.
In entrambi i casi, la gerarchia risultante può essere presentata visivamente usando una struttura ad albero chiamata dendrogramma, che contiene nodi per ciascuno cluster costruito dall'algoritmo, insieme a cluster-relazione che illustrano le operazioni di unione o divisione eseguite durante il processo di clustering (cfr. Green et al., 2008).

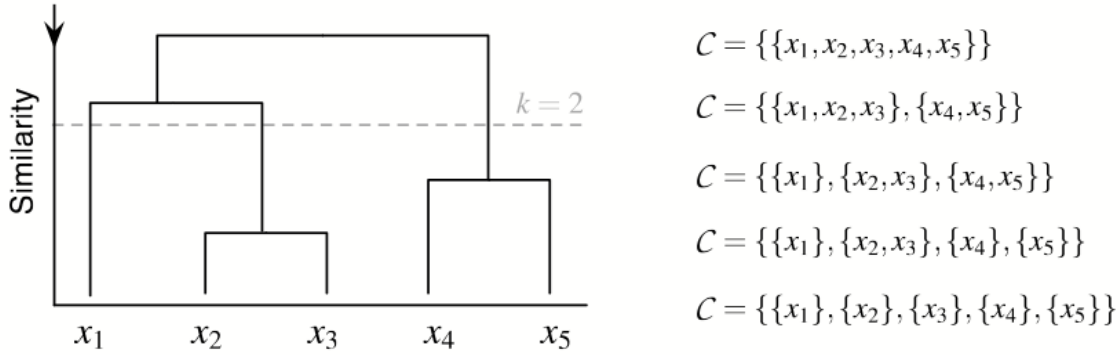


Figura 4: Esempio di algoritmo di clustering agglomerativo.

Gli algoritmi di clustering basati sulla densità (DBSCAN) sono progettati per rilevare cluster di forma arbitraria. Un cluster viene definito come una regione ad alta densità partizionata da regioni a bassa densità nello spazio dei dati. DBSCAN può scoprire cluster di forma arbitraria ma è sensibile ai parametri di input, specialmente quando la densità dei dati non è uniforme. Un tipico algoritmo DBSCAN richiede due parametri: ε e il numero minimo di punti richiesti per formare un cluster. Inizia con un punto di partenza arbitrario che non è stato visitato. Questo punto di ε-vicinato viene recuperato e, se contiene abbastanza punti, viene avviato un cluster, altrimenti il punto è etichettato come rumore. Si noti che questo punto potrebbe essere successivamente trovato in un ε-ambiente sufficientemente dimensionato di un punto diverso e quindi essere inserito in un cluster. Se si trova che un punto è una parte densa di un cluster, anche il suo ε-vicinato fa parte di quel cluster . Quindi, vengono aggiunti tutti i punti che si trovano all'interno del ε-vicinato, così come il loro stesso vicinato quando sono anche densi. Questo processo continua fino a quando non viene trovato completamente il cluster collegato alla densità. Quindi, un nuovo punto non visitato viene recuperato ed elaborato, portando alla scoperta di un ulteriore cluster o rumore (cfr. Suthar et al., 2013)
I vantaggi del DBSCAN sono i seguenti :

- Il DBSCAN non richiede di specificare a priori il numero di cluster nei dati, a differenza di k-means.
- Il DBSCAN cerca cluster di forma arbitraria.
- È robusto al rumore dei dati.
- Il DBSCAN richiede due parametri ed è per lo più insensibile all'ordinamento dei punti nel dataset.

È spesso conveniente ed efficace presumere che i dati siano stati generati a seguito di un processo statistico e quindi descrivere i dati trovando il modello statistico che si adatta meglio ai dati, in cui il modello è descritto in termini di distribuzione e un insieme di parametri per quella distribuzione. Ad alto livello, questo processo prevede la decisione su un modello statistico per i dati e la stima i parametri di quel modello dai dati. Queste tecniche si basano su modelli di Mixtures, che modellano i dati utilizzando una serie di distribuzioni statistiche. Ogni distribuzione corrisponde a un cluster e i parametri di ciascuna distribuzione forniscono una descrizione di cluster corrispondente, in genere in termini di centro e diffusione. Tra le tecniche statistiche troviamo: il maximum likelihood estimation (MLE) e l'algoritmo Expectation-Maximization (EM) (cfr. Tan et al., 2016).
I modelli "Mixtures" visualizzano i dati come un insieme di osservazioni da una mix di distribuzioni di probabilità differenti. Le distribuzioni di probabilità possono essere qualsiasi cosa, ma sono spesso considerati normali multivariati, poiché questo tipo di distribuzione è ben compreso, matematicamente facile da lavorare ed è stato mostrato per produrre buoni risultati in molti casi. Concettualmente, i modelli Mixtures corrispondono al seguente processo di generazione dei dati. Date diverse distribuzioni, di solito dello stesso tipo, ma con parametri diversi, selezionare casualmente una di queste distribuzioni e generare un oggetto da esso. Si Ripete il processo m volte, dove m è il numero di oggetti. Utilizzando metodi statistici, possiamo stimare il parametro di queste distribuzioni dai dati e quindi descrivere queste distribuzioni (cluster). Possiamo anche identificare quali oggetti appartengono a quali cluster. Tuttavia, la modellazione mista non produce una chiara assegnazione di oggetti a cluster, ma piuttosto dà la probabilità a cui appartiene un oggetto specifico a un particolare cluster. Dato un modello statistico per i dati, è necessario stimare i

parametri di quel modello. Un approccio standard utilizzato per questa attività è il maximum likelihood esimation (MLE). Si nota che i valori dei parametri che massimizzano la probabilità della funzione log likelihood massimizzano anche la probabilità poiché il log è una funzione monotona crescente. Possiamo anche utilizzare l'approccio MLE per stimare il modello parametri per un modello di mixture. Nel caso più semplice, sappiamo quali dati provengono da quali distribuzioni e la situazione si riduce a stimare i parametri di una singola distribuzione data da quella distribuzione. In una situazione più generale e più realistica, non sappiamo quali punti sono stati generati da quale distribuzione. Pertanto, non possiamo direttamente calcolare la probabilità di ciascun punto dati e, quindi, sembrerebbe che non è possibile utilizzare il MLE per stimare i parametri. La soluzione a questo problema è l'algoritmo EM. Data un'ipotesi per i valori dei parametri, l'algoritmo EM calcola la probabilità che ciascun punto appartenga a ciascuna distribuzione e quindi utilizza queste probabilità per calcolare una nuova stima per i parametri. (Questi parametri sono quelli che massimizzano la probabilità). Questa iterazione continua finché le stime dei parametri non cambiano o cambiano molto poco. Pertanto, utilizziamo la stima della massima probabilità, ma tramite una ricerca iterativa. Infine, gli autoencoder giocano un ruolo fondamentale per l'apprendimento non supervisionato e soprattutto nell'anomaly detection. Un autoencoder è una rete neurale addestrata a tentare di copiare il suo input al suo output. Internamente, ha uno strato nascosto h chiamato encoder layer il quale descrive una codifica dell'input. La rete può essere vista come composta da due parti: un funzione encoder h = f (x) e un decoder che produce una ricostruzione r = g(h) (in figura 5 vi è illustrato un esempio di architettura tipica di un autoencoder). Pertanto, gli autoencoder sono progettati per non essere in grado di imparare a copiare perfettamente. Di solito lo sono in modi che consentono loro di copiare solo approssimativamente e solo di copiare l'input che assomiglia ai dati di allenamento. Perché il modello è costretto a stabilire delle priorità su quali aspetti dell'input debbano essere copiati, questo fa sì che spesso apprenda proprietà utili dei dati.

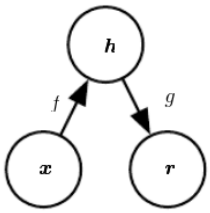


Figura 5: La struttura funzionale generale di un autoencoder con f che trasforma la configurazione di ingresso nella codifica h e g che riporta la configurazione nel dominio iniziale

Esistono diversi tipi di autoencoder:

• Un autoencoder la cui dimensione del code layer è inferiore della dimensione dell'input viene chiamato undercomplete. L'obiettivo è acquisire le caratteristiche più salienti dei dati di allenamento.

• Un autoencoder, invece, che ha il code layer maggiore dell'input, è chiamato overcomplete.

• Uno Sparse Autoencoder è un autoencoder il cui criterio di addestramento prevede una penalità di sparsità Ω(h) sul layer code h oltre all'errore di ricostruzione:

$$L(x, g(f(x))) + \Omega(h)$$

dove g(h) è il decoder output e h = f (x) l'input. Essi vengono in genere utilizzati per apprendere le funzionalità di un altro task, ad esempio come classificazione .

• Un Denoising Autoencoder permette di ricostruire dati da quelli corrotti, minimizzando la loss function L(x, g(f (x0))) dove x0 è nella stessa forma di x corrotta, però, dal rumore.
Un Variational Autoencoder (VAE) è un autoencoder composto sia da un encoder che da un decoder e che viene addestrato per ridurre al minimo l'errore di ricostruzione tra i dati codificati e decodificati e i dati iniziali. Tuttavia, al fine di introdurre una certa regolarizzazione dello spazio latente, procediamo a una leggera modifica del processo di codifica-decodifica: invece di codificare un input come un singolo punto, lo codifichiamo come distribuzione nello spazio latente (figura 6) (cfr. Prakash et al., 2014 e Rocca et al., 2019):
1. l'input viene codificato come distribuzione nello spazio latente.
2. un punto dallo spazio latente viene campionato da quella distribuzione.
3. il punto campionato viene decodificato e l'errore di ricostruzione può essere calcolato.
4. infine, l'errore di ricostruzione viene riprogrammato attraverso la rete.

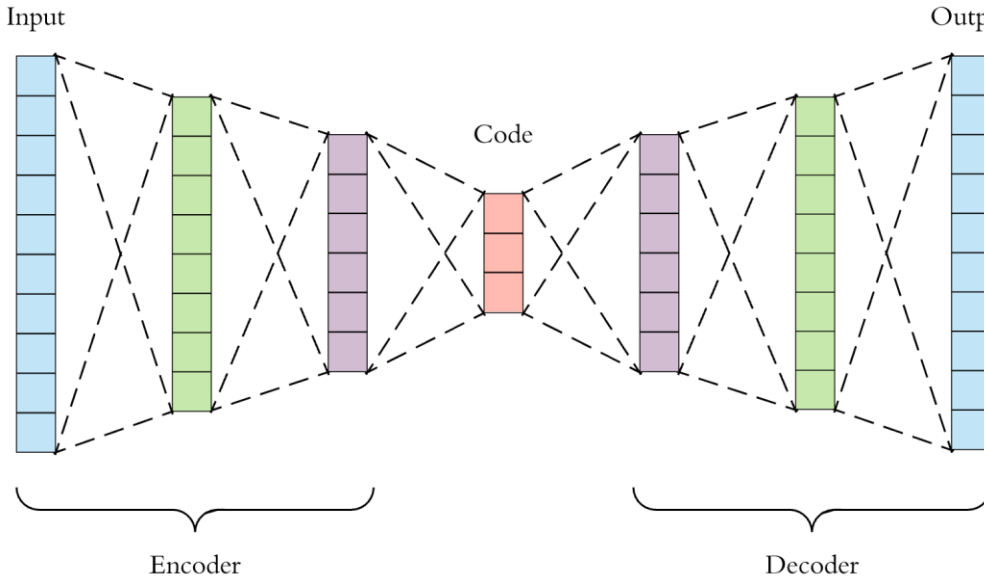


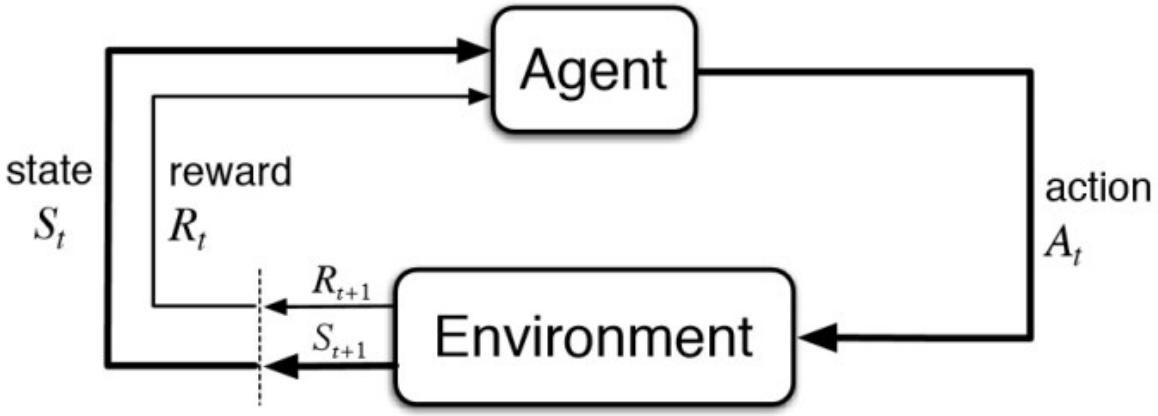
Figura 6: Architettura di un Autoencoder.

La metodologia di apprendimento semi-supervisionato può fornire elevate prestazioni di classificazione utilizzando dati senza etichetta. La metodologia può essere utilizzata per adattarsi a una varietà di situazioni identificando invece di specificare una relazione tra dati etichettati e non etichettati dai dati. Può produrre un miglioramento quando i dati senza etichetta possono ricostruire il limite di classificazione ottimale. Alcuni popolari modelli di apprendimento semi-supervisionato includono self-training, modelli di mixture, metodi basati su grafici, co-training e apprendimento multiview (cfr. An et al., 2015).
Il rilevamento di anomalie basato sull'apprendimento semi-supervisionato può essere costruito mediante gli autoencoder, utilizzando l'errore di ricostruzione come punteggio di anomalia. Infatti, i punti dati con elevata ricostruzione sono considerati anomalie poichè solo i dati con istanze normali vengono utilizzati per addestrare l'autoencoder. Dopo l'allenamento, l'autoencoder ricostruirà bene i dati normali, ma non riuscirà a farlo con i dati di anomalia che non ha riscontrato in fase di addestramento. Data una soglia, quindi, se l'errore di ricostruzione del dato, in fase di inferenza, è superiore alla soglia allora possiamo classificare quel dato come anomalo (cfr. Borghesi et al., 2019) altrimenti come normale (fig. 6). La soglia può essere scelta con molti metodi. Ad esempio, il valore può essere deciso osservando l'n-esimo percentile della distribuzione degli errori del normale set di dati. Infatti, l'n-esimo percentile è una statistica che indica il valore al di sotto del quale una determinata percentuale di osservazioni in un gruppo di dati cadono le osservazioni (prendendo, ad esempio, il novantesimo percentile è il valore al di sotto del quale è possibile trovare il 90% delle osservazioni) (cfr. Kaelbing et al., 1996).

Nell'apprendimento per rinforzo il problema di classificazione è affrontato da un agente che deve imparare il comportamento attraverso interazioni di prova ed errore con un ambiente dinamico. Esistono due strategie principali per risolvere i problemi di apprendimento di rinforzo. Il primo è cerca nello spazio dei comportamenti per trovare quello che si comporta bene nell'ambiente. Il secondo è usare tecniche statistiche e metodi di programmazione dinamica per stimare l'utilità dell'assunzione azioni negli stati del mondo. Nel modello standard di apprendimento del rinforzo, un agente è collegato al suo ambiente tramite percezione e azione. Ad ogni fase di interazione l'agente riceve come input, i, qualche indicazione dello stato o degli stati correnti dell'ambiente; l'agente quindi sceglie un'azione, a, da generare come output. L'azione cambia lo stato di ambiente e il valore di questa transizione di stato viene comunicato all'agente tramite un segnale di rinforzo scalare, r. Il comportamento dell'agente, B, dovrebbe scegliere le azioni che tendono per

Figura 7: Architettura funzionale del processo di apprendimento per rinforzo e suoi principali componenti

aumentare la somma a lungo termine dei valori del segnale di rinforzo.



Uno dei più conosciuti algoritmi di apprendimento per rinforzo è il Q- Learning. Esso fa parte della famiglia di algoritmi adottati nelle tecniche delle differenze temporali, adottate nel caso di modelli a informazione incompleta. Uno dei suoi maggiori punti di rilievo consiste nell'abilità di comparare l'utilità 'aspettata' delle azioni disponibili senza richiedere un modello completo dell'ambiente. Il suo obiettivo è quello di permettere ad un sistema di apprendimento automatico di adattarsi all'ambiente che lo circonda migliorando la scelta delle azioni da eseguire. Per giungere a questo obiettivo, cerca di massimizzare il valore del successivo premio per sconto . Pertanto, dato un insieme di stati S, un insieme di azioni per stato A, per ogni azione a ∈ A l'agente va da uno stato all'altro. Ogni stato fornisce all'agente una ricompensa. L'obiettivo dell'agente è massimizzare la ricompensa appena ricevuta. L'agente fa questo apprendendo quali sono le azioni ottimali associate ad ogni stato. Quindi l'algoritmo è provvisto di una funzione per calcolare la Qualità di una certa coppia stato-azione: Q : S × A → R. Prima che l'apprendimento inizi, Q restituisce un valore fisso, scelto dal progettista. Poi, ogni volta che l'agente riceve una ricompensa (lo stato è cambiato) vengono calcolati nuovi valori per ogni combinazione stato-azione. Il cuore dell'algoritmo fa uso di un processo iterativo di aggiornamento e correzione basato sulla nuova informazione

$$Q(st , at) \leftarrow Q(st , at) + at (st , at) \times (rt + \lambda * maxat+1 Q(st+1 , at+1) - Q(st , at))$$

dove Q(st , at) alla destra dell'equazione è il vecchio valore, at (st , at) il tasso di apprendimento, rt la ricompensa, λ il fattore di sconto, maxat+1 Q(st+1 , at+1) il valore futuro massimo. L'algoritmo termina quando lo stato st+1 è uno stato finale (o stato di assorbimento). Una semplice implementazione di Q-learning usa tabelle per memorizzare i dati. Tuttavia, questo approccio perde fattibilità al crescere del livello di complessità del sistema. Una possibile soluzione a questo problema prevede l'uso di una rete neurale artificiale come approssimatore di funzione, soluzione in rapido consolidamento allo stato dell'arte.

RIFERIMENTI BIBLIOGRAFICI IN APPENDICE

(AA.VV., 2018) MindOrks, "Understanding the recurrent neural network," 2018, available on line at https://medium.com/mindorks/understanding-the-recurrent-neural-network-44d593f112a2

(An et al., 2015) J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," Special Lecture on IE, vol. 2, no. 1, 2015.

(Borghesi et al., 2019) Borghesi, A. Bartolini, M. Lombardi, M. Milano, and L. Benini, "A semisupervised autoencoder-based approach for anomaly detection in high performance computing systems," Engineering Applications of Artificial Intelligence, vol. 85, pp. 634–644, 2019

(Elman et al., 2016) J. L. Elman, "Finding structure in time," Cognitive science, vol. 14, no. 2, pp. 179–211, 1990.

(Goodfellow et al., 2016) I Goodfellow, Y. Bengio, and A. Courville, Deep learning. MIT press, 2016.

(Green et al., 2008) D. Greene, P. Cunningham, and R. Mayer, "Unsupervised learning and clustering," in Machine learning techniques for multimedia. Springer, 2008, pp. 51–90

(Kaelbing et al., 1996) L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," Journal of artificial intelligence research, vol. 4, pp.237–285, 1996.

(Kotsiantis et al,, 2007) S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," Emerging artificial intelligence applications in computer engineering, vol. 160, pp. 3–24, 2007

(Mohammed et al., 2016) M. Mohammed, M. Khan, and E. Bashier, Machine Learning: Algorithms and Applications, 07 2016.

(Prakash et al., 2014) V. J. Prakash and D. L. Nithya, "A survey on semi-supervised learning techniques," International Journal of Computer Trends and Technology, vol. 8, no. 1, p. 25–29, Feb 2014.

(Rocca et al., 2019) B. R. Joseph Rocca, "Understanding variational autoencoders (vaes)," 2019, Available at https://towardsdatascience.com/ understanding-variational-autoencoders-vaes-f70510919f73

(Suthar et al., 2013) N. Suthar, P. Indr, and P. Vinit, "A technical survey on dbscan clustering algorithm," Int. J. Sci. Eng. Res, vol. 4, pp. 1775–1781, 2013.

(Tan et al., 2016) P.-N. Tan, M. Steinbach, and V. Kumar, Introduction to data mining. Pearson Education India, 2016.