



UNIVERSITÀ POLITECNICA DELLE MARCHE  
FACOLTÀ DI INGEGNERIA

---

Corso di Laurea Magistrale in Ingegneria Elettronica

**Tesi di Laurea**

**Sviluppo di Sistemi Edge per Earthquake  
Early Warning**

**Development of Earthquake Early Warning  
Edge Systems**

**Relatore:**

Prof. Paola Pierleoni

**Candidato:**

Marco Esposito

**Correlatore:**

Dott. Lorenzo Palma

*Anno Accademico 2020/2021*



# Abstract

I sistemi di Early Warning (EW) o sistemi di allerta precoce sono estremamente importanti per la prevenzione di danni a cose o persone in seguito a disastri naturali. Lo sviluppo tecnologico che negli ultimi anni ha interessato ambiti quali l'*Internet of Things* (IoT), le comunicazioni e l'*Edge Computing* rappresenta un'opportunità per migliorare e creare nuovi sistemi di EW che siano in grado di predire il verificarsi di un evento naturale, generare e disseminare allarmi. In questa tesi si analizzerà la letteratura riguardo i sistemi di Early Warning per tre tipologie di disastro naturale (terremoti, alluvioni e tsunami) allo scopo di determinare quali sono i principali requisiti di un sistema di allerta precoce basato sul paradigma dell'IoT. A partire da questi requisiti, tra i quali latenza, consumi, copertura ed affidabilità, si focalizzerà l'attenzione sul problema dell'allerta precoce *on site* per eventi sismici e sui principali algoritmi utilizzati in questo contesto, con lo scopo di progettare un dispositivo per la rilevazione dell'onda P basato sull'Edge Computing, ossia in grado di generare un allarme in prossimità di un terremoto senza aver bisogno di trasmettere dati verso un Cloud o un server centrale. La tesi quindi dimostrerà come implementare una procedura di ottimizzazione tramite il *Global Optimization Toolbox* di Matlab per l'algoritmo *Filter Picker*, basato sull'algoritmo di Allen per il "picking" dell'onda P, affinché sia in grado di rilevare tutti gli eventi sismici da una dataset di forme d'onda locali al luogo d'installazione, analizzando poi i risultati in termini di "missed detection" e numero di pick effettuati. Sempre con lo scopo di progettare un dispositivo di tipo Edge che sia in grado di trasmettere allarmi in maniera tempestiva e che rispetti i vincoli già citati, si è effettuata una prova di copertura per il protocollo di comunicazione LoRaWAN, con l'obiettivo di individuare un set di parametri LoRa che consentano al dispositivo scelto (STM32 B-L072Z-LRWAN1 Discovery kit) di operare con bassi consumi, data rate adeguati e basse durate della trasmissione LoRa (o *Time on Air*). I risultati ottenuti sia per quel che riguarda l'algoritmo di rilevazione dell'onda P che le prestazioni LoRaWAN dimostrano come la realizzazione di un dispositivo di tipo "Edge" per l'Early Warning sismico che usa questo protocollo e l'algoritmo ottimizzato possa essere una valida scelta, essendo in grado di rilevare il terremoto senza dover ricorrere ad elaborazioni remote dei dati accelerometrici utilizzati dall'algoritmo, così da generare allarmi in maniera immediata in prossimità di un sisma.



# Indice

<b>1</b>	<b>Sistemi di Early Warning</b>	<b>1</b>
1.1	Introduzione . . . . .	1
1.2	Architettura di un Sistema IoT per l'Early Warning . . . . .	3
1.2.1	Perception Layer . . . . .	3
1.2.2	Network Layer . . . . .	4
1.2.3	Edge/Fog Layer . . . . .	6
1.2.4	Application Layer e Middleware Layer . . . . .	7
1.3	Review dello Stato dell'Arte dei Sistemi di EW . . . . .	8
1.3.1	Sistemi di EW per Flash Floods . . . . .	8
1.3.2	Sistemi di EW per Earthquake . . . . .	12
1.3.3	Sistemi di EW per Tsunami . . . . .	16
1.3.4	Discussione . . . . .	20
<b>2</b>	<b>Early Warning per Earthquake Detection</b>	<b>25</b>
2.1	Terremoti: onde sismiche, Strong Motion Parameters e strumenti di misura. . . . .	25
2.2	Algoritmi per Earthquake Early Warning . . . . .	28
2.3	Algoritmo Filter Picker . . . . .	33
2.4	Testing e Ottimizzazione dell'Algoritmo Filter Picker . . . . .	36
2.4.1	Algoritmo 'Genetico' per l'ottimizzazione . . . . .	40
2.4.2	Dataset Utilizzato . . . . .	42
2.4.3	Risultati dell'Ottimizzazione . . . . .	43
<b>3</b>	<b>Nodo Edge per Earthquake Early Warning</b>	<b>47</b>
3.1	Introduzione a LoRa e LoRaWAN . . . . .	47
3.1.1	La Modulazione LoRa . . . . .	47
3.1.2	Topologia di una rete LoRaWAN . . . . .	52
3.1.3	Tipologia di Nodo LoRaWAN . . . . .	55
3.2	Valutazione della Connessione LoRa . . . . .	56
3.2.1	Hardware e Servizio Cloud Utilizzati . . . . .	56
3.2.2	Test di Copertura . . . . .	59
3.3	Dispositivo LoRaWAN per la trasmissione di allarmi. . . . .	62
	<b>Risultati</b>	<b>69</b>
	<b>Conclusioni</b>	<b>73</b>



## Elenco delle figure

1.1	Architettura Generica di Early Warning per alluvioni proposta in [1]	9
1.2	EEW basato sul paradigma di Mobile Crowd-Sensing proposto in [2]	14
2.1	Rappresentazione di un terremoto con evidenziati l'ipocentro e l'epicentro. Dal sito dell'Istituto Nazionale di Geologia e Vulcanologia. . .	26
2.2	Accelerazione misurata durante un evento sismico con evidenziate le finestre temporali relative all'Onda P e all'Onda S. (da <i>Performance Evaluation of a Low-Cost Sensing Unit for Seismic Applications: Field Testing During Seismic Events of 2016-2017 in Central Italy</i> [3]) . .	27
2.3	Diverse variazioni fisiche che precedono il verificarsi di un terremoto (da [4]). . . . .	29
2.4	Schema del Sistema di Early Warning <i>ShakeAlert</i> , basato sul rilevamento di onde P (da [5]). . . . .	29
2.5	Accelerazione e spettrogramma di un evento sismico (da <i>An Automatic P-Phase Picking Algorithm Based on Adaptive Multiband Processing</i> [6]). Si nota come attorno all'istante $t=150$ s si ha un picco di energia su diverse bande, in corrispondenza dell'arrivo dell'onda P. . . . .	32
2.6	Uscita del Filter Picker utilizzando la forma d'onda disponibile nel relativo pacchetto Matlab. . . . .	37
2.7	Uscita del Filter Picker utilizzando un sisma di magnitudo $4.1M_L$ [7]	38
2.8	Uscita del Filter Picker utilizzando un sisma di magnitudo $5M_L$ [7] .	39
2.9	Esempi di Popolazione (da <a href="https://it.mathworks.com/help/gads/how-the-genetic-algorithm-works.html">https://it.mathworks.com/help/gads/how-the-genetic-algorithm-works.html</a> ). . . . .	41
2.10	Statistiche dell'algorithm dopo l'ottimizzazione e usando invece i parametri standard. . . . .	44
2.11	Numero di Pick $N_P$ per eventi che superano la soglia massima ( $N_P = 4$ ) e relative percentuali degli eventi che presentano quel valore di $N_P$ . Si nota come l'algorithm ottimizzato, seppur non abbia molti più eventi con picking eccessivo rispetto a quello standard, presenta alcuni eventi con un numero molto alto di pick. . . . .	45
2.12	Uscita dell'Algorithm Filter Picker dopo l'ottimizzazione usando in ingresso la stessa forma d'onda riportata in Figura 2.8. In questo caso il picking dell'onda P avviene correttamente. . . . .	45

## Elenco delle figure

3.1	Stack Protocollore LoRaWAN [8]. LoRaWAN si trova al secondo livello della pila ISO/OSI. Si basa a livello fisico sullo schema di modulazione LoRA. . . . .	48
3.2	Andamento nel tempo (sopra) e spettrogramma (sotto) di un up-chirp cosinusoidale con frequenza che aumenta linearmente. Tramite lo spettrogramma, possiamo vederlo come una rampa che va da $f_{low}$ a $f_{high}$ , o viceversa per i down-chirp, in un tempo $t_{chirp}$ . . . . .	49
3.3	Confronto di chirp con differenti Spreading Factor [9]. All'aumentare di SF, aumenta la durata del singolo chirp e quindi il numero di bit che possono essere trasmessi in un tempo di simbolo. Al contempo, il tempo di simbolo risulta maggiore (rate diminuiscono). . . . .	50
3.4	Sensibilità ottenuta al ricevitore con diversi SF fissata una banda fissa di 125 kHz [10]. . . . .	51
3.5	Trama LoRa a Livello Fisico [8]. . . . .	52
3.6	Sequenza di simboli LoRA [9]. Si individua chiaramente l'inizio di una trama per il susseguirsi di 6 up-chirp e 2 down-chirp (preambolo). . . . .	52
3.7	Esempi di topologia di rete (da <a href="https://datarespons.com/industrial-or-commercial-wireless-mesh-technologies">https://datarespons.com/industrial-or-commercial-wireless-mesh-technologies</a> ). Una rete LoRa può essere vista come una rete 'a stella di stelle' in cui i coordinatori sono i gateway, che comunicano verso gli End Device e il Network Server. . . . .	53
3.8	Topologia di una rete LoRaWAN [8]. Ogni gateway riceve connessioni da più nodi come in una rete a stella, ma uno stesso nodo è connesso a più gateway. . . . .	54
3.9	Funzionamento di un Device di Classe A [11]. . . . .	55
3.10	Funzionamento di un Device di Classe B [11]. . . . .	55
3.11	Funzionamento di un Device di Classe C [11]. . . . .	56
3.12	B-L072Z-LRWAN1 LoRa®Discovery Kit. [12] . . . . .	57
3.13	Sentrius RG1xx LoRa-Enabled Gateway. [13] . . . . .	57
3.14	Stack di TTN [14] . . . . .	58
3.15	Identificativi di un Device LoRaWAN sulla Console TTN. . . . .	58
3.16	Valori dei Data Rate e corrispondenti SF. . . . .	59
3.17	Rappresentazione grafica dell'RSSI [11]. . . . .	59
3.18	Risultati della Prova di Copertura. L'andamento del segnale è stato osservato in 4 punti di interesse, e sono stati misurati più valori di RSSI per ogni zona per valutare l'oscillazione del segnale. . . . .	60
3.19	Diagramma di Flusso dell'Applicazione per la Trasmissione di Allarmi. . . . .	64
3.20	Confronto delle uscite del Filter Picker per la stessa forma d'onda di un sisma di magnitudo 5.1 $M_L$ rilevata dalla stazione GUMA, con i parametri standard (a) e dopo l'ottimizzazione (b) . . . . .	67
3.21	Statistiche dell'algorithmo dopo l'ottimizzazione e usando invece i parametri standard. . . . .	67

## Elenco delle tabelle

1.1	Confronto tra i Protocolli di Comunicazione più usati nell'EW. . . . .	6
2.1	Algoritmi e metodi per Earthquake Early Warning (tabella ricavata da [15]). . . . .	30
2.2	Parametri di Filtraggio. . . . .	34
2.3	Parametri dell'Algoritmo e loro valori standard. . . . .	37
2.4	Stazioni utilizzate nel formare il dataset per l'ottimizzazione. . . . .	43
2.5	Parametri dell'Algoritmo ottimizzato e loro valori standard. . . . .	43
3.1	Risultati del test per i diversi scenari. La soglia di disconnessione di $-120dBm$ non viene mai raggiunta per entrambi gli SF. . . . .	61
3.2	Confronto dei diversi Data Rate che possono essere ottenuti con SF=8 ed SF=12. SF=8 fornisce il data rate migliore. . . . .	63
3.3	Confronto dei diversi ToA e consumi ottenibili con SF=12 ed SF=8. Anche in questo caso SF=8 risulta essere più conveniente. La dimensione del payload è fissata a 8 byte. . . . .	63
3.4	Parametri dell'Algoritmo ottimizzato e loro valori standard. $\Delta T$ indica il tempo di campionamento. . . . .	66
3.5	Risultati del test per i diversi scenari. La soglia di disconnessione di $-120dBm$ non viene mai raggiunta per entrambi gli SF. . . . .	68
3.6	Confronto dei diversi Data Rate, ToA e consumi che possono essere ottenuti con SF=8 ed SF=12. . . . .	68
3.7	Caratteristiche della comunicazione LoRa ottenuti con SF=8, BW=125 kHz, banda EU868, $P_{TX}=14$ dBm, pacchetti da 8 byte. . . . .	69



# Capitolo 1

## Sistemi di Early Warning

### 1.1 Introduzione

Negli ultimi anni diverse calamità e disastri hanno colpito il globo, rendendo evidente la necessità di migliorare la nostra capacità di predire e rispondere a questi eventi. In Italia e nel mondo da diversi anni si sono verificati fenomeni sismici anche di grande importanza, ed alluvioni, frane e altri eventi connessi a problematiche di dissesto idrogeologico sono molto frequenti, anche a causa degli effetti del riscaldamento globale, che renderà questi eventi sempre meno sporadici. Allo stesso tempo, lo sviluppo tecnologico nell'ambito delle comunicazioni, dell'intelligenza artificiale e dell'*Internet of Things* (IoT) è sempre più veloce ed apre una importante prospettiva per poter migliorare e creare nuovi sistemi di allerta e previsione, così da consentire di limitare gli effetti di questi eventi su utenti ed infrastrutture, nonché di poter monitorare e caratterizzare meglio i danni dovuti al disastro naturale. Ne segue che l'ambito dell'allerta precoce o *Early Warning* (EW) è di estremo interesse e ci sono sforzi costanti nel migliorare le prestazioni e definire nuove architetture e tecnologie in questo contesto.

Un sistema di Early Warning è in generale un sistema integrato per il monitoraggio, la predizione, la valutazione del rischio e la comunicazione di allarmi che consentono a individui, comunità, enti e governi di svolgere azioni tempestive per consentire di ridurre il rischio relativo a un disastro naturale o in genere un evento pericoloso o catastrofico. Un sistema di EW presenta i seguenti elementi chiave: i) conoscenza e valutazione dei rischi; ii) capacità di monitorare i parametri che possono migliorare o permettere di effettuare predizioni, iii) mezzi per disseminare allarmi in maniera tempestiva, iv) preparazione adeguata per la risposta al disastro [16][2]. Un sistema di EW quindi comprende sia sistemi di misura e monitoraggio ambientale che modelli predittivi, ma anche server, infrastrutture di comunicazione e di risposta agli eventi. Mentre esistono già molti sistemi di EW già funzionanti, i mezzi per consentire di soddisfare al meglio gli aspetti di monitoraggio, processamento ed analisi dei dati, predizione e generazione e disseminazione degli allarmi possono essere forniti da tecnologie come l'IoT, il Cloud Computing e l'intelligenza artificiale, che risultano estremamente adatti a questo genere di task e sono ormai scelte consolidate per quel che riguarda il monitoraggio dell'ambiente o la creazione di sistemi di allarme.

L'Internet of Things è l'insieme delle infrastrutture che consentono di interconnettere e controllare oggetti, sensori ed attuatori, ottenere informazioni da essi e manipolare i dati che generano [17]. Un'architettura IoT basilare include 4 livelli: i) l'ambiente locale che contiene gli smart object o i sensori che interagiscono con l'ambiente e misurano grandezze di interesse, ii) un livello di trasporto che consente agli end-node dal primo livello di comunicare con i livelli superiori, iii) un livello di immagazzinamento, processing ed analisi dei dati, di solito implementato tramite un servizio Cloud, iv) utenti o altri sistemi che consentono di accedere e visualizzare i dati o controllare gli smart object. Mentre le *Wireless Sensor Networks (WSNs)* sono spesso una componente essenziale nell'IoT, fornendo un'interfaccia tra utente ed ambiente locale, l'IoT è una soluzione più globale che consente la coesistenza di oggetti e protocolli diversi, applicazioni in tempo reale, capacità di memorizzazione dei dati e di analisi, migliore sicurezza [18], da cui le reti di sensori "classiche" possono beneficiare. Nel contesto della gestione dei disastri e dell'Early Warning, l'IoT consente di ottimizzare il monitoraggio ambientale da diverse fonti, basse latenze e processamento dei dati in tempo reale per generare allerte tempestive in caso di avvenuta predizione del disastro.

Recentemente si è inoltre spostata l'attenzione verso sistemi di tipo *Edge*. Per *Edge Computing* si intende l'avvicinare le capacità computazionali, i servizi e la memoria di un sistema verso l'utente o più genericamente un end-node, piuttosto che su livelli più alti dell'architettura del sistema IoT. Questo ha diversi scopi, uno tra i quali la diminuzione della latenza dovuta alla trasmissione dei dati verso, ad esempio, un server centrale o un Cloud, e la diminuzione dell'occupazione di banda, in quanto sono trasmesse quantità di dati molto minori. I dispositivi di tipo Edge convenzionali sono caratterizzati da basse capacità computazionali ma sono in grado di effettuare un processamento dei dati locale (ad esempio, per effettuare una predizione o estrarre delle feature), o anche task di maggiore complessità se si equipaggia i dispositivi con modelli di *Machine Learning* [19]. Questo porta dei vantaggi che sono utili anche nel contesto dell'Early Warning, tra cui: sicurezza, latenza, minor consumo di banda, efficienza.

Lo scopo di questo lavoro di tesi è quello di mostrare la realizzazione di un dispositivo di tipo Edge per l'Early Warning sismico on-site, ossia che sia in grado di predire il verificarsi di un terremoto nei pressi del luogo in cui il dispositivo stesso è installato e poi di trasmettere un allarme usando il protocollo di comunicazione LoRaWAN. Essendo un dispositivo di tipo Edge, la predizione viene fatta dal dispositivo stesso senza la necessità di trasmettere dati ad altri nodi. Per fare questo verrà analizzato l'algoritmo di predizione detto *Filter Picker* che verrà implementato sul device, in particolare ponendo un accento sulla sua validazione tramite forme d'onda locali e su una procedura che ne consenta l'ottimizzazione prima dell'effettiva implementazione su dispositivo embedded. Inoltre, verranno effettuati dei test di copertura per valutare le caratteristiche della connessione LoRaWAN e poter scegliere opportunamente un set di parametri che lo renda adatto ad essere usato

come protocollo di comunicazione per questa applicazione.

Nel resto del presente Capitolo si porrà l'attenzione sull'architettura di un sistema IoT per l'Early Warning, sui principali protocolli di comunicazione usati in questo contesto, ed infine si revisioneranno degli articoli per tre diversi Use Case così da poter andare a definire una serie di vincoli che vogliamo che un sistema di EW rispetti.

## 1.2 Architettura di un Sistema IoT per l'Early Warning

Le funzioni di un sistema IoT possono essere descritte a partire dalla loro configurazione architetturale. Una generica applicazione basata sull'IoT può essere infatti suddivisa in "layer" o livelli che ne consentono la composizione in maniera modulare. È possibile utilizzare una architettura a 3 livelli per descrivere un sistema di Early Warning generico [20]: i) un *perception layer*, ii) un *network layer* e iii) un *application layer* [21]. Mentre architetture più specifiche possono introdurre o specificare nuovi layer o strati intermedi, come Edge/Fog layer, o Middleware e Interface layer, un'architettura IoT per EW semplice può essere considerata quella che raccoglie dati dall'ambiente, li processa e li trasmette ad un server o un servizio Cloud che utilizza i dati ricevuti e i dati precedentemente salvati per generare allarmi attraverso diversi metodi, dal signal processing a metodi statistici, fino a metodi di computer vision o intelligenza artificiale (*artificial intelligence AI*), come algoritmi di Machine Learning (ML) o Deep Learning (DL).

### 1.2.1 Perception Layer

Il Perception layer ha il compito di misurare e raccogliere dati dall'ambiente, di solito tramite l'utilizzo di sensori. Le reti di sensori wireless o Wireless Sensor Networks (WSNs) sono ampiamente utilizzate negli scenari di monitoraggio ambientale: si tratta di nodi equipaggiati con un'unità di sensing e di comunicazione che effettuano misure e trasmettono dati verso un nodo che funge da 'gateway' e si interfaccia e comunica con livelli superiori. Le WSN offrono diversi benefici, quali scalabilità, bassi consumi, reliability, riconfigurabilità, bassi costi e dimensioni contenute [20]. Alcuni aspetti del progetto di una rete di sensori sono particolarmente importanti per scenari di monitoraggio di disastri o scenari di Early Warning, come durata della batteria, copertura e *fault tolerance*, ovvero la capacità della rete di resistere alla perdita di nodi.

La scelta dell'unità di sensing giusta è essenziale nel fornire una risposta veloce ed accurata, e, poiché diversi parametri possono contribuire ad un particolare disastro, dovrebbero tutti essere monitorati. Il posizionamento di sensori in determinate zone o terreni può essere particolarmente complesso, e mentre certe applicazioni monitorano eventi localizzati (come, ad esempio, per le frane), altri potrebbero richiedere il dislocamento di sensori su aree più grandi (ad esempio, per il monitoraggio di grandi

bacini fluviali o delle ampie geografiche che possono essere colpite da terremoti), aumentando di conseguenza il costo della soluzione e richiedendo soluzioni ad hoc per coprire l'intera area da monitorare, ad esempio tramite una differenziazione tra nodi con capacità di copertura a lungo e corto raggio [22] o, per esempio, tramite una distribuzione intelligente e ottimizzata dei sensori a seconda di parametri quali, ad esempio, la probabilità di occorrenza [1]. Anche se le reti di sensori sono una soluzione fortemente utilizzata e con diversi vantaggi, è anche possibile l'implementazione di un sistema di Early Warning a singolo nodo, come nei sistemi di rilevamento on site, in cui si usa l'informazione di un nodo sensore per rilevare la presenza di un evento nel punto in cui il nodo stesso è installato. Infine, è anche possibile ricorrere a soluzioni di *crowd-sensing*, in cui si utilizza la grande mole di dati proveniente da dispositivi utenti come smartphone equipaggiati con sensori MEMS per realizzare una rete di monitoraggio distribuita che consenta l'allerta precoce.

### 1.2.2 Network Layer

Il network layer trasmette il dato misurato e processato dal Perception Layer verso un ser, un servizio cloud o più genericamente un'applicazione. Questo strato è responsabile del routing, della comunicazione tra reti eterogenee e dell'integrità dei dati. Ci sono diversi protocolli di comunicazione che possono essere usati per la trasmissione dei dati, sia a lungo raggio (come protocolli cellulari) che a corto raggio (come WiFi or Zigbee per quel che riguarda le comunicazioni wireless, o Ethernet per quelle cablate).

Di solito le comunicazioni wireless si sono dimostrate le più efficienti in situazioni di emergenza o disastri [23], anche se sia le comunicazioni cablate che non sono suscettibili ad errori. I disastri possono avere un forte impatto sulle infrastrutture di rete, ad esempio isolando le zone colpite in caso di rotture di antenne, collegamenti in fibra o linee aeree [24], e quindi è sempre bene implementare canali di comunicazione alternativi ed includere ridondanza nei metodi di comunicazione implementati nel progetto.

Ci sono diversi protocolli e tecnologie di comunicazione che possono essere ritenute adatte al contesto dell'Early Warning. Di seguito è riportata una breve rassegna dei protocolli che rispettano i principali vincoli richiesti da sistemi di allerta precoce, e che sono stati utilizzati più frequentemente nella letteratura consultata.

I protocolli di comunicazione per l'IoT possono essere divisi in due categorie: tecnologie a lungo raggio e tecnologie a corto raggio. I protocolli LPWAN (*Low Power Wide Area Network*) come LoRa, SigFox, NB-IoT e EC-GSM-IOT offrono grande copertura e possono essere suddivisi in protocolli *Licensed* e *Unlicensed*, a seconda delle bande e delle frequenze utilizzate. LoRaWAN and SigFox usano bande ISM (*Industrial, Scientific and Medical*) non licenziate, mentre NB-IoT and GSM usano le reti cellulari e lavorano quindi in bande licenziate. L'utilizzo di bande licenziate ed infrastrutture cellulari può fornire una migliore integrità e qualità del servizio,

nonché una maggiore copertura e pervasività, ma richiede solitamente il pagamento di un abbonamento, al contrario di protocolli che lavorano in bande ISM, che però sono solite implementare tecniche per regolare il *fair use* della risorsa condivisa, ad esempio imponendo dei duty cycle per la comunicazione. Inoltre, nonostante i numerosi vantaggi, le reti cellulari possono essere danneggiate da disastri ambientali o fornire peggiore qualità in certe condizioni [24] [25].

Tra le tecnologie a corto raggio, protocolli wireless come *Bluetooth Low Energy* (BLE) e *Zigbee* possono fornire soluzioni a basso costo e con consumi estremamente ridotti e supporto per architetture di tipo *mesh*. Il limite principale è la mancanza di supporto per comunicazioni a lunga distanza, a meno di usare ripetitori che però aumenterebbero i costi [26].

- **Zigbee:** Zigbee è un protocollo a basso costo, basso consumo e basse velocità basato sul protocollo IEEE 802.15.4 e sviluppato dalla ZigBee Alliance. Lavora nella banda 2.4 GHz e ha data rate dai 20 ai 250 kbps. Zigbee supporta topologie a stella, mesh e ad albero, tra cui la connessione mesh è la più flessibile ed affidabile [27], poiché può consentire alla WSN di sopravvivere a guasti o perdite di nodi. Ha uno stack protocollare semplice se paragonato a WiFi e Bluetooth e la vita della batteria può arrivare fino a 5 anni, ma ha un range relativamente limitato e velocità di trasmissione basse.
- **Bluetooth e BLE:** Bluetooth si basa sullo standard IEEE 802.15.1 standard. La versione a basso consumo e basso costo di questo standard è chiamata *Bluetooth Low Energy* (BLE). Sia Bluetooth che BLE operano nella banda ISM 2.4 GHz. Ha data rate fino a 1 Mbps, e usa la frammentazione per trasmettere pacchetti più lunghi [24]. In BLE c'è un trade-off tra consumo, latenza, dimensione della piconet e throughput, ma la scelta accurata dei suoi parametri caratteristici consente di ottimizzarlo per diverse applicazioni di IoT [28].
- **LoRaWAN:** LoRa è una tecnologia a livello Fisico che usa una tecnica a spettro espanso proprietaria. LoRaWAN è invece un protocollo a Livello Medium Access Control (MAC) di tipo open source standardizzato dalla LoRa Alliance che usa il livello fisico LoRa. Funziona nelle seguenti bande ISM: 868 MHz in Europa, 915 MHz in Nord America, e 433 MHz in Asia. La modulazione LoRa consente ottime performance contro le interferenze e fornisce diversi data rate, dai 300 bps to 50 kbps. LoRaWAN migliora l'affidabilità tramite ritrasmissioni ed offre ampia copertura (10–40 km in zone rurali e 1–5km aree urbane [29]), bassi costi e una lunga durata della batteria per gli end-device. Inoltre consente di usare 3 classi di dispositivi per diversi vincoli dell'applicazione, come latenza e consumi.
- **EC-GSM-IoT, GPRS:** EC-GSM-IoT riqualifica le portanti a banda stretta di 200 delle reti GSM e richiede solamente un update software di suddetta

rete, senza la necessità di ulteriore hardware. Come si vedrà nella sezione successiva, alcune soluzioni EW odierne usano ancora moduli GSM o GPRS, ma EC-GSM fornisce una miglior performance, da una migliore copertura indoor fino al supporto per un gran numero di dispositivi, ridotta complessità e miglior utilizzo della batteria rispetto a dispositivi GSM 'classici' [30].

- **NB-IoT**: NB-IoT è una tecnologia introdotta da 3GPP, insieme ad EC-GSM e LTE-M, che opera in spettro licenziato e riutilizza infrastrutture LTE esistenti. NB-IoT fornisce un'ampia copertura (20 dB più forte del GSM tradizionale) con un alto MCL (*Maximum Coupling Loss*) di 164 dB [30], che consente ai dispositivi NB-IoT di raggiungere zone interrate o seminterrate (ad esempio, per localizzare vittime [25]); ha un basso consumo e migliora i meccanismi di risparmio energetico di precedenti Release; procedure di rete, stack protocollare e schema di modulazione sono semplificati per ridurre la complessità e il costo dei dispositivi. Possono esserci diversi effetti di latenza per NB-IoT, ma questa deve essere mantenuta sotto i 10 secondi per applicazioni real time [26].
- **EnOcean**: EnOcean lavora in bande non licenziate, nelle bande 868 MHz in Europa e 315 MHz in America. EnOcean non è in grado di gestire topologie ad-hoc come altri protocolli e ha meno funzioni di altri protocolli, ma risponde all'esigenza principale di essere estremamente efficiente a livello energetico [31], perciò è adatto al monitoraggio di disastri ed eventi naturali, anche grazie alle sue capacità di energy harvesting [24].

Le reti 5G forniranno ulteriori soluzioni in termini di comunicazioni cellulari mobili. In particolare, i servizi di Ultra Reliable and Low Latency Communication (URLLC) mirano a fornire latenze sotto il millisecondo con 99% di affidabilità, con la possibilità quindi di dimostrarsi particolarmente utili per l'Early Warning.

Tabella 1.1: Confronto tra i Protocolli di Comunicazione più usati nell'EW.

Protocollo	Banda di Frequenza	Data Rate	Range	Latenza	Consumo Energetico
Zigbee	2.4 GHz	250 kbps	75-100m	10ms	Low
LoRaWAN	868, 915, 433 MHz	50 kbps	5-20 km	10ms	Low
BLE	2.4 GHz	1 Mbps	>100m	<10ms	Very Low
NB-IoT	700, 800, 900 MHz	150 kbps	1-10 km	<10s	Low
GSM	2.4 GHz	50bps-70kbps	15 km	10s	Very Low
EnOcean	315-868 MHz	125 kbps	300 m		Low

### 1.2.3 Edge/Fog Layer

Il Fog computing, o più generalmente l'Edge computing, è un paradigma che prevede l'elaborazione e il processamento di dati provenienti da livelli inferiori tra il Network

## 1.2 Architettura di un Sistema IoT per l'Early Warning

Layer e l'Application Layer, tramite l'avvicinamento delle capacità computazionali e di memoria della rete verso i suoi end node o gateway. L'implementazione di un Fog Layer consente di ottenere una risposta più rapida ed assicurare migliore qualità di soluzioni basate unicamente sul Cloud computing [20]. Infatti, mentre i servizi Cloud forniscono capacità di storage e processamento essenziali per diverse applicazioni, trasmettere una grande mole di dati da un gran numero di sensori o sorgenti di informazione può essere un'operazione costosa, e l'elaborazione di una grande quantità di dati su un server dedicato aggiungerà una latenza che può influenzare la performance complessiva del servizio di allerta precoce. In uno strato Fog/Edge, i dati dal Perception Layer sono elaborati all'estremità della rete (ad esempio nei gateway o sugli end-node stessi) prima di trasmetterlo verso livelli superiori, ad esempio verso un servizio Cloud, così che la latenza e la quantità di dati trasmessa sia ridotta. Questo può anche essere vantaggioso rispetto a situazioni di 'instabilità di banda' e problemi di comunicazione che possono insorgere durante un particolare evento [32], poiché il pre-processamento può ridurre il consumo di banda [19]. Nodi di questo tipo possono anche implementare algoritmi per effettuare predizioni basate sui dati collezionati dal Perception Layer [1], così da riuscire a generare allarmi direttamente a questo livello. È anche possibile integrare modelli di Machine Learning o Deep Learning su dispositivi Edge, ma la complessità e i requisiti di memoria potrebbero rendere una simile soluzione più complessa da realizzare in pratica [19].

### 1.2.4 Application Layer e Middleware Layer

Il livello applicazione è il livello più alto dell'architettura IoT. Utilizza i dati ricevuti dal livello di rete per fornire servizi od operazioni [20], possibilmente combinando quanto ricevuto dai livelli inferiori con dati storici, previsioni meteo, dati da satellite, dati da sistemi di monitoraggio strutturale etc. Questo livello implementa algoritmi per generare e propagare allarmi se un disastro è imminente; fornisce database per immagazzinare dati e salvare i nuovi dati che gli arrivano in tempo reale; può fare predizioni etc. Possono essere create delle interfacce utente sul livello applicazione e, nelle "architetture orientate al servizio" (*service-oriented-architectures* o SoA), livelli Middleware o di gestione dei servizi possono essere interposti fra il livello applicativo ed il livello di rete (ad esempio, tramite protocolli come il protocollo *Message query and telemetry transport* o MQTT), per fungere da collegamento tra i dispositivi e le applicazioni, ad esempio per reti con una forte eterogeneità di dispositivi, e per assicurare l'interoperabilità tra di essi [19].

Le piattaforme Cloud per l'IoT forniscono capacità di memoria e computazionali quasi illimitate. L'analisi dei dati può essere una parte essenziale dei sistemi IoT di Early Warning, i quali potrebbero avere a che fare con grandi quantità di informazioni da fonti, zone geografiche e istanti temporali diversi. Per questo, l'analisi dei dati può diventare il collo di bottiglia di un sistema di Early Warning [24], e i sistemi

moderni di allarmistica predittiva dovrebbero essere associati a sistemi Cloud per avere prestazioni ottimali [32]. Il Cloud computing presenta anch'esso dei limiti come latenze che aumentano quando la quantità di informazioni da processare è troppo grande, ma soluzioni di tipo Fog/Edge possono ridurre il carico sul livello applicazione, come già descritto nella sezione precedente.

## 1.3 Review dello Stato dell'Arte dei Sistemi di EW

Nella seguente sezione si farà una breve review di alcuni sistemi di Early Warning o proposte di sistemi di Early Warning per tre use case: alluvioni (*Flash Flood Early Warning* o FEW), terremoti (*Earthquake Early Warning* o EEW) tsunami (*Tsunami Early Warning* o TEW). La review occupa un intervallo temporale che va dal 2017 al 2021, con lo scopo di revisionare prevalentemente articoli più recenti. Lo scopo della review è quello di evidenziare e definire i requisiti di un sistema di EW basato sul paradigma dell'IoT, nonché di analizzare i protocolli di comunicazione più utilizzati. Pertanto, la sezione termina evidenziando i suddetti requisiti sia nel caso di un generico sistema di EW che per i tre use case revisionati.

### 1.3.1 Sistemi di EW per Flash Floods

Come per molti scenari di Ew, ci sono già molte soluzioni IoT e WSN realizzate per il flood EW, come la soluzione sviluppata da Basha et al. [22] ed installata con successo in Honduras e Massachusetts, la quale consiste di una rete eterogenea di sensori con diversi nodi che possono coprire grandi distanze e consentono di evitare punti di "fallimento" singoli. Anche se funzionante, Jayashree et al.[33] hanno indicato che questo genere di soluzione ha lo svantaggio di richiedere l'utilizzo di diverse tipologie di nodo di rete e dover gestire dati eterogenei, e perché può diventare costosa e richiedere elaborazioni dei dati più complesse. Propongono quindi un'architettura semplice e generica per risolvere questo ed altri problemi come il consumo di potenza, ritardi, assenza di copertura e costo computazionale. L'architettura consiste di sensori per il livello dell'acqua e sensori di flusso che mandano dati verso un server solo quando i livelli misurati superano una certa soglia. Un'app Android è poi impiegata su dispositivi mobili per allertare gli utenti, usando hardware Zigbee connesso ai dispositivi tramite modulo OTG (USB On-The-Go), così che gli allarmi possano essere distribuiti anche in assenza di copertura, ma questo approccio richiede che gli utenti interessati dispongano del suddetto sistema OTG.

Un'architettura di flood EW proposta da Sood et al. [1] è mostrata in Figura 1.1. Mentre il livello IoT è responsabile di collezionare ed aggregare dati dai sensori, il livello di Fog Computing è usato per diminuire la latenza pre-processando i dati dai sensori prima di mandarli a livelli superiori, così da assicurare previsioni tempestive e per diminuire la quantità di dati da trasmettere al cloud. Inoltre, sui singoli nodi è implementato un modello predittivo. Il framework proposto introduce inoltre

una divisione esagonale dell'area di monitorare così da poter comprendere grandi aree fornendo un sensing uniforme, costi ottimizzati e risparmio energetico basato sulla probabilità di alluvione in ogni area esagonale: definendo 5 etichette di probabilità di alluvione, gli esagoni sono caratterizzati così che solo gli esagoni con una probabilità alta abbiano i sensori attivi, mentre i dispositivi nelle aree con bassa probabilità possono essere messi in sleep mode, minimizzando il consumo energetico.

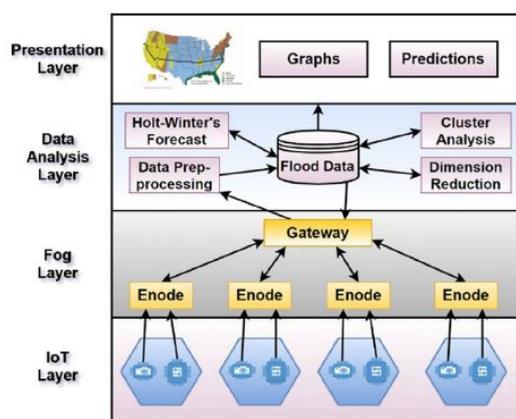


Figura 1.1: Architettura Generica di Early Warning per alluvioni proposta in [1]

Utilizzare infrastrutture di comunicazioni pre-esistenti è spesso una soluzione semplice ed efficiente, come mostrato dallo use case sviluppato da Ibarreche et al. in Colima, Messico [34], il quale utilizza la rete 3G ed il protocollo MQTT per trasmettere dati verso un server remoto per il processing. Il sistema inoltre utilizza dei "drifter" (nodi sensori mobili) che usano dei moduli LoRa per trasmettere dati verso i nodi connessi alla rete 3G. La grande quantità di dati da differenti fonti e la comunicazione efficace rendono la soluzione adatta a fornire allarmi velocemente, ma ne aumentano anche il costo.

Le reti cellulari di solito offrono una buona copertura per la maggior parte delle regioni geografiche e il futuro sviluppo del 5G migliorerà ulteriormente la performance. Questo però non è sempre vero, poiché certe zone potrebbero avere una cattiva copertura a causa di propagazione di tipo NLoS o perché non ci sono abbastanza stazioni base per coprire adeguatamente una determinata area, ad esempio per zone lontane da centri abitati. In tali situazioni, è meglio scegliere una soluzione ad hoc usando protocolli come LoRaWAN, come osservato da Nordin et al. in [35], che hanno valutato la performance radio di LoRaWAN e GSM ed hanno presentato un confronto basato su modelli empirici per una zona con bassa copertura. Un altro problema quando si utilizzano reti cellulari è che disastri ambientali o certe condizioni meteorologiche potrebbero comprometterli o impattare negativamente il canale radio, per cui la ridondanza per quel che riguarda la comunicazione è sempre da valutare.

Situazioni critiche come i livelli dell'acqua che salgono sopra una certa soglia potrebbero richiedere un numero maggiore di misurazione da parte dei nodi sen-

sore, con maggior consumo di potenza. Inoltre, per sensori equipaggiati con unità fotovoltaiche, condizioni meteo avverse potrebbero compromettere la ricarica delle batterie. D. Purkovic et al. [36] hanno preso in considerazione questi fattori per valutare l'energy budget della loro soluzione basata su EnOcean per ottimizzarla prima dell'installazione vera e propria. Hanno usato un sensore ultrasonico generico che ha due modalità operative: modalità normale e modalità critica. Quando il livello dell'acqua sale sopra una soglia, il sensore entra in modalità critica, effettuando più misurazioni e trasmettendole tutte. Lo studio ha dimostrato che la modalità critica e la modalità "oscura" (quando le celle solari non sono ricaricate) riducono la vita della batteria di più della metà rispetto alla modalità normale.

Anche Ragnoli et al. [37] hanno sviluppato una soluzione con un focus sul consumo energetico. Il loro sistema consiste di un'architettura LoRa standard, con un nodo sensore che comunica con un server tramite gateway. I sensori usati sono sensori resistivi di umidità, con costo molto basso ed con richiesto un processamento molto semplice, in quanto hanno un comportamento di tipo ON/OFF in presenza di acqua. Tre sensori sono posti a differenti altezze nell'area da monitorare, ottenendo tre livelli dell'acqua da 0 a 2 che vengono monitorati. Il basso consumo viene ottenuto usando un algoritmo di deep sleep dedicato. I test effettuati hanno mostrato che si ottengono buone performance (anche nella già citata modalità "oscura" di funzionamento). Anche se un sensore è stato testato, Ragnoli et al. non hanno ancora testato l'implementazione di una rete di sensori, e mentre il sistema è facilmente scalabile, questa richiederà un planning per quel che riguarda il management del traffico radio e potrebbe non scalare ottimamente con molti dispositivi.

Protocolli che supportano le reti mesh, come Zigbee, sono particolarmente utili per aree, come i bacini dei fiumi, in cui il nodo centrale potrebbe rompersi, e per questo implementazioni mesh sono in genere raccomandate ed hanno mostrato migliore affidabilità rispetto a topologie ad albero o a stella in termini di packet loss [38]. Dovrebbe esserci sempre un focus nello gestire la perdita di nodi nelle soluzioni sviluppate per far sì che la rete di sensori continui a funzionare anche in situazioni di funzionamento complesse. Furquim et al. [39] propongono un modello per EW che esibisce un'alta tolleranza alla perdita di nodi, specialmente per condizioni ambientali particolarmente difficili. L'architettura è divisa in tre livelli: i) nodi sensori, ii) Fog layer, iii) Cloud layer. La tolleranza ai guasti è ottenuta fornendo ai nodi sui primi due livelli algoritmi di "riorganizzazione" e modelli di predizione "light-weight" in termini di costo computazionale e di memoria, consentendogli di effettuare predizioni anche se la connessione al cloud viene meno, anche se con maggiore incertezza. Se i nodi sul secondo livello sono compromessi, quelli sul primo livello hanno il compito di aggregare dati da nodi sullo stesso layer. Inoltre sono equipaggiati con un sistema di allarme acustico per creare allarmi localizzati nel caso in cui le comunicazioni siano estremamente compromesse. Ovviamente raggiungere questo livello di ridondanza e resistenza ai guasti richiede un gran numero di nodi.

Ali et al. [40] hanno proposto una soluzione basata su Cloud ed una rete di

sensori e di attuatori (WSAN) per allertare in caso di alluvione. Il sistema proposto è pensato per essere installato non solo per il monitoraggio del livello dell'acqua, ma include anche attuatori che possono aprire "aperture vedetta" in grado di far defluire via l'acqua. Sensori di livello e di volume identificano lo stato dell'acqua e lo comparano con una soglia. Dei gateway sono usati per gestire la comunicazione dei sensori verso il Cloud. Un algoritmo gestisce sensori ed attuatori ed è stato valutato tramite tecniche di analisi dei modelli, e specifica diversi controlli per i livelli dell'acqua finché viene raggiunto il livello 'pericoloso' e viene quindi generato un messaggio di allarme. Il nodo attuatore vedetta viene invece attivato usando l'informazione proveniente dall'elaborazione dei dati su Cloud.

I dati relativi alle alluvioni sono spesso dinamici e non-lineari, e per questo utilizzare algoritmi di Machine Learning è una soluzione che viene spesso adottata, specialmente se c'è disponibilità di dati di alluvioni o meteo per quella zona. C'è una vasta letteratura su reti neurali usate per flash floods, e l'approccio si basa di solito sulla conoscenza di dati precedenti e mappe di allagamento per quella zona, insieme alla conoscenza di nuovi dati provenienti ad esempio da una rete di sensori, per cui è essenziale una comunicazione ottimale. Per esempio, Anbarasan et al. [41] propongono un generico diagramma a blocchi per predire alluvioni usando reti neurali convoluzionali deep (CDNN) che seguono il precedente approccio che consiste nel sensing, nella comunicazione dei dati misurati e nell'usa della rete neurale per effettuare predizioni (si veda Figura ??). L'algoritmo di data processing e la CDNN da loro sviluppata ha inoltre mostrato risultati migliori rispetto a sistemi che usano Artificial Neural Networks (ANN) or Deep Neural Networks (DNN) quando validati usando dati radar.

Al Qundus et. al. [42] hanno realizzato un FEWS che utilizza Machine Learning su dati ottenuti da una rete di sensori per creare allarmi. Hanno impiegato 24 sensori e 6 nodi coordinatori, dividendo l'area monitorata in 6 settori. I sensori utilizzati sono sensori di livello dell'acqua, di temperatura, umidità e velocità del vento, ed i nodi coordinatori utilizzano l'API Google Weather per ottenere dati aggiuntivi. Per aumentare il numero di dati a parità di sensori, è stata scelta un'alta frequenza di campionamento. I dati ottenuti nel corso di un anno sono stati usati per formare un training set che è stato usato per addestrare un modello di ML del tipo *Support Vector Machine* (SVM), poi implementato su ogni nodo coordinatore, e la validazione del modello ha mostrato un'accuratezza di fino il 98%. Le soglie per i sensori devono essere calcolate ogni volta che una nuova soluzione è installata in una certa zona (richiedendo quindi un periodo per collezionare i relativi dati), e c'è un'ambiguità per dati che vengono da zone che si intersecano, e sono ancora necessari maggiori sforzi per la separazione ed il clustering dei sensori provenienti dalle 6 location.

### 1.3.2 Sistemi di EW per Earthquake

Anche i sistemi di Early Warning per terremoti (Earthquake Early Warning System o EEW) possono beneficiare dall'IoT e dalle tecnologie ad esso connesse. Sensori ed unità di sensing possono monitorare vibrazioni e movimento del suolo per generare allarmi ed allertare la popolazione o enti preposti prima che abbia luogo l'effetto distruttivo del sisma. Come si approfondirà nel Capitolo successivo, quando si verifica un evento sismico onde Primarie (o onde P), onde Secondarie (o onde S) ed onde superficiali irradiano dall'epicentro. Le onde P viaggiano più velocemente e la loro rilevazione può quindi essere utilizzata per generare un allarme prima che le altre onde più pericolose raggiungano un centro abitato, un edificio etc. Un sistema IoT molto semplice e basato su questo approccio è quello proposto da Alphonsa et al. [43], che utilizza accelerometri connessi a microcontrollori per ottenere e processare misure di vibrazione del suolo, trasmesse poi tramite il protocollo Zigbee a un ricevitore connesso ad un Personal Computer che viene utilizzato per generare allarmi. Inoltre, moduli GSM connessi ai microcontrollori possono anche essere utilizzati per trasmettere allarmi ad una base transceiver station (BTS) che poi li disseminerà agli utenti.

Tariq et al. [40] hanno proposto e testato un algoritmo per la detection di eventi sismici chiamato SWEDA (*seismic wave event detection algorithm*) capace di rilevare eventi sismici in termini di millisecondi. Utilizza inclinometri per l'Industria 4.0 e il protocollo di comunicazione CANOpen per IIoT (Industrial IoT). Hanno prodotto e testato 2 tipologie di nodi: un inclinometro piatto che consiste di due accelerometri, ed un inclinometro cilindrico con al suo interno 7 sensori e un ADC sigma-delta a 24 bit e un amplificatore di guadagno per migliorare la risoluzione. Oltre le onde P, questo sistema può rilevare anche onde S e onde superficiali (onde Love e Rayleigh), ma in tal caso richiede topologie e distribuzioni dei sensori tali da avere un match con la direzione della relativa onda sismica. L'algoritmo di processamento è stato ottimizzato per evitare falsi allarmi e ridurre il costo computazionale di operazioni floating point e delle calibrazioni. La soluzione è stata testata tramite tre sistemi installati presso l'Università del Qatar. Il sistema è stato in grado di identificare eventi sismici e produrre trigger per la presenza di Onde P 11 secondi prima che un trigger per le onde S fosse generato, mostrando buone capacità di Early Warning.

Oltre che algoritmi di predizione, possono anche essere utilizzati modelli di Machine Learning per predire terremoti utilizzando dati da fonti multiple. Fauvel et al. [44] hanno proposto un approccio distribuito per EEW che utilizza data da stazioni GPS e da sismometri per fornire predizioni basate su un algoritmo che hanno validato utilizzando un dataset contenente eventi reali. L'approccio distribuito è basato sul paradigma di Edge/Fog computing, con lo scopo di ridurre la quantità di dati che viene trasmessa verso la rete. A tal scopo, è integrato un classificatore su ogni sensore che produce come output un classe che viene trasmessa ad un server centrale, il quale combina le classi predette per fornire una predizione finale. L'aggregazione fatta a

livello dei sensori insieme al fatto che il dato trasmesso non è correlato a misurazioni sismiche riduce lo sforzo trasmissivo, riducendo la latenza e rendendo più semplici le comunicazioni. L'articolo inoltre ha mostrato che le stazioni GPS e i sismometri hanno un effetto "complementare", e quindi l'approccio con più tipologie di sensori consente di ottenere una migliore accuratezza.

Khedo et al. [45] hanno simulato un modello di WSN "on-site" (ossia installato nella stessa location in cui si vuole predire l'evento sismico) nell'isola di Mauritius, ed hanno analizzato come la scelta di diversi parametri nella rete di sensori possa corrompere o migliorare le stime di velocità e posizione dell'epicentro ottenute dall'analisi delle onde P. L'articolo si concentra anche sul problema della sincronizzazione: è essenziale che tutti i nodi sensori e il nodo sink siano sincronizzati tra loro per avere una rilevazione precisa dell'evento sismico, altrimenti gli offset tra i clock dei diversi nodi diventano una componente del ritardo di propagazione dell'onda. L'architettura simulata utilizza il *Timing-sync protocol for sensor networks* (TPSN) per avere sincronizzazione tra i nodi. I risultati delle simulazioni mostrano che mentre le reti che utilizzano TPSN hanno una performance peggiore rispetto ad una rete ideale con perfetta sincronizzazione, reti che non utilizzano alcun protocollo di sincronizzazione hanno risultati molto peggiori in termini di stima della velocità e della posizione dell'epicentro. Il paper inoltre mostra che una maggiore densità di nodi sensore fornisce una migliore qualità dei dati e rendono le predizioni più resistenti in caso di fallimento di uno o più nodi. Infine, maggiore la frequenza di campionamento, migliore la performance della predizione.

Poiché l'efficienza del rilevamento è dipendente dalla densità della rete di sensori, questo potrebbe tradursi in un costo elevato nella soluzione di EEW. I sensori MEMS possono fornire un'alternativa a basso prezzo rispetto ai sismografi tradizionali o ai sismometri, e possono essere utilizzati per realizzare reti dense con un minore investimento. Fu et al. [46] hanno proposto una rete di accelerometri MEMS che è in grado di fornire le stesse performance di una rete EEW "classica". Ogni sismografo MEMS consiste di: un modulo accelerometrico MEMS, un modulo TCP/IP, un modulo *Power over Ethernet* (PoE), ed una memoria esterna opzionale. La sincronizzazione è ottenuta usando una versione semplificata del *Network Time Protocol* (NTP). L'unità TCP/IP gestisce tutta la comunicazione mediante porta Ethernet, così che l'unità di sensing possa essere controllata tramite una semplice unità MCU, poiché non deve gestire le routine di comunicazione. La soluzione è stata testata usando una rete con 10 sensori connessi a modem 3G/4G, mostrando che le registrazioni ottenute dalla rete erano comparabile con quelle ottenute dai sismografi standard, e che possono anche essere usati per ottenere dei "trigger" prima dell'evento sismico, consentendo di usarli come sistema di allerta precoce. Anche terremoti di bassa magnitudo ( $M_L$  3.1 fino a  $M_L$  3.6) sono stati identificati fino a 20 km dai sensori.

Peng et al. [47] hanno testato la performance di un rete per EEW di 170 sensori che hanno installato nella regione di Sichuan-Yunnan, in Cina. Il design dell'unità di sensing è specificato in [48] ed è costituito da un accelerometro MEMS a basso costo e

basso consumo, e l'unità mostra performance paragonabili a quelle di un accelerometro di Classe B (per confronto, i sensori MEMS standalone sono di solito sensori di Classe C, come quelli usati nella soluzione illustrata in precedenza [46]). Ogni unità inoltre è in grado di individuare onde P e trasmettere dati con bassa latenza. Per ogni stazione, un router 3G/4G è stato usato per trasmettere le misure di movimento del terreno verso un server centrale. I record sono stati poi confrontati con quelli di stazioni di monitoraggio classiche, mostrando una buona somiglianza, anche per sensori a più di 150 km dall'epicentro, dimostrando che una soluzione a basso costo di questo genere può essere usata anche per coprire regioni molto ampie. Il sistema è anche in grado di produrre "shake-map" quasi in tempo reale e con buona accuratezza, mentre di solito sono richiesti anche più di dieci minuti dall'evento.

I sensori MEMS sono installati nella maggior parte degli smartphone in commercio, per cui l'informazione proveniente da un gran numero di dispositivi mobili può essere utilizzata per creare una rete sismica ampiamente diffusa (paradigma di *mobile crowd-sensing*, o MCS). Un'architettura MCS proposta da Zambrano et. al. [49] è mostrata in Figura 1.2. Il primo Layer raccoglie dati dalla rete formata dagli smartphone, inclusa la posizione dell'utente usando l'unità GPS installata nel dispositivo, ed è in grado di rilevare picchi sismici. Poiché diversi dispositivi potrebbero avere diversi riferimenti temporali, la sincronizzazione è ottenuta tramite il protocollo NTP. Il secondo Layer determina se c'è stato effettivamente un evento sismico da notificare e assicura l'affidabilità globale del sistema. L'estrema eterogeneità della rete richiede una formata di standardizzazione dei dati raccolti, ottenuta tramite il framework chiamato Sensor Web Enablement (SWE) e nello specifico dalla sua componente chiamata Sensor Observation Service (SOS). Infine, il terzo ed ultimo Layer è il "centro di controllo", che comunica con i centri di gestione dell'emergenza.

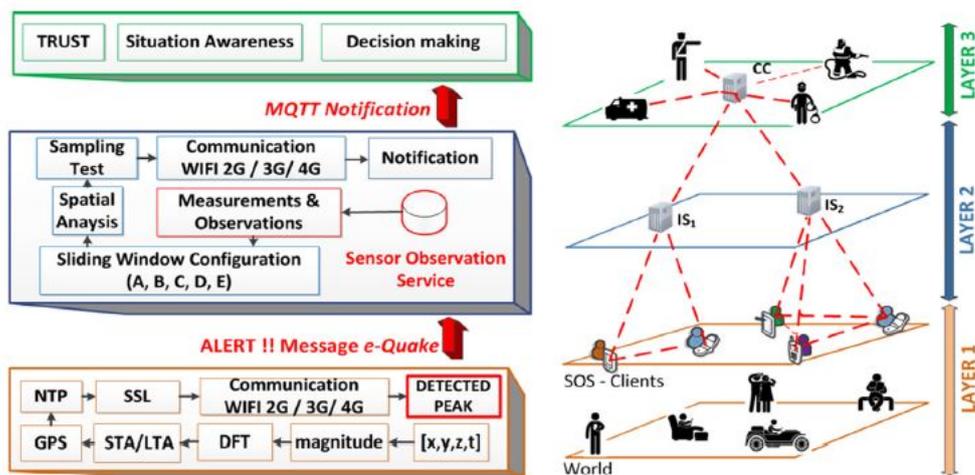


Figura 1.2: EEW basato sul paradigma di Mobile Crowd-Sensing proposto in [2]

La piattaforma Myshake è un framework già funzionante che fornisce servizi di EW e monitoraggio sismico, sviluppato da Allen et. al [50], disponibile come

applicazione per il telefono dal 2016. Si possono generare allarmi sia rilevando onde P tramite i sensori contenuti nel dispositivo in cui è installata l'applicazione, od ottenendo informazioni da reti sismiche regionali (ad esempio, attualmente riceve dati dal sistema ShakeAlert negli Stati Uniti). Mentre l'approccio di MCS rende la realizzazione di una rete sismica molto semplice, la caratterizzazione dei parametri del sisma (come epicentro, magnitudo e instante iniziale) dipende fortemente dal numero e dalla distribuzione geografica rispetto all'evento dei telefoni che utilizzano l'app. Una rete neurale artificiale (ANN) è integrata nell'applicazione ed è usata per distinguere movimenti dovuti ad un evento sismico rispetto a movimenti che sono dovuti ad attività quotidiane svolte dall'utente. Quando un evento è rilevato sul telefono, il server MyShake cerca di individuare dei cluster di dispositivi che hanno ottenuto un trigger nella stessa area per confermare che c'è un sisma in corso. Mentre l'applicazione è in grado di fornire indicazioni sulla posizione, magnitudo e shake map del terremoto, attualmente lo sforzo riguarda il miglioramento delle sue capacità come sistema di EW. È stata creata una piattaforma per testare tramite simulazioni la possibilità di usare MyShake per l'allerta precoce. Le simulazioni hanno mostrato che è possibile generare un'allerta 15.6 s prima che un'area densamente popolata sia investita dal sisma, ma ci sono 2.8 s di ritardo nella trasmissione dell'allerta, e non è ancora chiaro come scalerebbe questo ritardo con un gran numero di dispositivi coinvolti, come in uno scenario reale. Inoltre, la magnitudo predetta era inferiore rispetto a quella reale. La versione 2.0 dell'app MyShake mira a riuscire a generare allerte precoci e in generale migliorare le caratteristiche dell'applicazione.

Klapez et al. [51] hanno iniziato lo sviluppo di una soluzione low-cost e low-power basata su Cloud computing e chiamata Earthcloud, la quale utilizza geofoni invece che accelerometri MEMS, affermando che offrono minor rumore per misure a bassa frequenza. Quando vengono rilevati movimenti del terreno, i sensori inviano il dato relativo verso un server centrale che produce un primo allarme se il movimento è abbastanza forte. In particolare, il dato è incapsulato in un pacchetto MQTT e inviato alla piattaforma Amazon Web Services (AWS) IoT. Amazon Kinesis è usato sia per il preprocessing dei dati che per trasmetterli real-time verso i dispositivi connessi (in attesa di una seconda conferma al loro allarme, se sono stati loro a generarlo, o per rendergli noto un primo allarme se inizialmente non hanno generato loro il trigger) ed anche ad Amazon S3, un servizio di storage. La soluzione sviluppata è stata testata usando una rete di 3 sensori nella città di Modena in Italia, e solo uno dei tre sensori non è stato in grado di rilevare l'evento sismico.

D'Errico et al. [52] hanno proposto un'architettura di sistema di EW sismico che utilizza la rete 5G per ottenere bassa latenza nella trasmissione di dati da una WSN per *Structural Health Monitoring* (SHM) verso un server di data processing. Fornendo servizi con risorse dedicate (*network slicing*), la New Radio 5G è in grado di supportare servizi di URLLC (ultra reliability and low latency communication) che consentono la trasmissione di dati con latenze al di sotto del millisecondo e con alta affidabilità. *Software Defined Networking, Network Function Virtualization* e

*Mobile Edge Computing* migliorano ulteriormente le performance del sistema, essendo in grado di allocare risorse in maniera dinamica ad ogni servizio e "avvicinare" le risorse computazionali all'utente.

Anche Hung et al. hanno sviluppato una soluzione basata su SHM, ma con un focus sull'efficienza energetica, che rimane un vincolo chiave nell'EEW. Viene implementato un nodo sentinella costituito da un detector di onde P, un trasmettitore "wake-on-radio" (WoR), ed un nodo sink, integrati nel gateway di una WSN per SHM che utilizza lo standard IEEE 802.15.4 per la comunicazione. I dispositivi WoR monitorano il canale radio per segnali di wake-up, i quali sono usati per ridurre il consumo di energia nelle fasi di "rendezvous", ossia quando un ricevitore deve svegliarsi quando un nodo a livello superiore inizia una comunicazione. L'utilizzo dei trasmettitori WoR consente ai nodi di andare completamente in modalità di risparmio energetico quando non è richiesta nessuna trasmissione. L'architettura proposta consiste di: i) un gateway che utilizza 3G/WiFi, ii) un detector di onde P connesso al gateway, iii) nodi sensori con moduli WoR organizzati gerarchicamente. Il detector allerta la più accurata WSN per il monitoraggio strutturale quando rileva un'onda P utilizzando i moduli WoR. Il testing ha mostrato un ritardo di 229 ms al comando di wake-up, ma lo schema proposto ha un ritardo di wake-up inferiore rispetto allo schema di wake-up nativo di IEEE 802.15.4, ed è più energeticamente efficiente (consumo di  $350 \mu\text{A}$ , rispetto al consumo di IEEE 802.15.4 in beacon mode che va da 1.43 mA a 2.51 mA, a seconda dei parametri scelti).

### 1.3.3 Sistemi di EW per Tsunami

Gli tsunami sono grandi onde d'acqua generate da terremoti, eruzioni vulcaniche, frane etc. La predizione di uno tsunami include monitoraggio dei livelli del mare, misura delle onde e del movimento del fondale marino. Altri metodi che possono essere usati in TEW sono la misura di onde idro-acustiche, misura di pressione o metodi di computer vision. Per quel che riguarda i sistemi IoT di TEW, le Underwater Wireless Sensor Networks (UWSNs) possono essere utilizzate per collezionare dati da ampie zone geografiche e trasmetterli verso un hub centrale per il processamento. L'Internet of Underwater Things (IoUT) è un paradigma che consente il rilevamento e la predizione di eventi che potrebbero portare a disastri, ma l'uso di sensori sottomarini può comportare una serie di sfide. In questo ambito di solito il canale di comunicazione più usato è quello acustico, ma anche le comunicazioni ottiche sono adatte a questo genere di ambiente. Una rete subacquea di solito consiste di nodi fissi o statici equipaggiati con sensori e modem acustici [53] per comunicare con dei sink (boe, barche, veicoli sottomarini) che raccolgono i dati e li trasmettono verso un server remoto usando di solito comunicazioni basate su IP [54]. Inoltre, possono anche essere usate boe ed altri sensori per il monitoraggio delle onde e di anomalie della superficie marina, anche se di solito questo può comportare alti costi, anche se i sensori MEMS possono essere usati anche in questo contesto per ridurre i costi.

### 1.3 Review dello Stato dell'Arte dei Sistemi di EW

Freitag et al. [55] hanno sviluppato e testato le performance di un sistema sottomarino acustico per sviluppare un sistema TEWS nel bacino di Mentay, tra la Sumatra occidentale e l'Indonesia. In questa applicazione la distanza tra i sensori (sensori di pressione) è ampia (decine di chilometri), e la distanza dalla riva ancora maggiore, per cui è usata una bassa frequenza per la comunicazione. Il sistema è stato sviluppato vicino al fondale con differenze altezze e con diversi range del ricevitore. Il link di comunicazione ha dimostrata un'alta affidabilità con un data rate fino a 350 bps, dimostrando quindi di potere essere usato per sistemi di allerta per tsunami. Data rate maggiori possono essere raggiunti ma con una minore affidabilità del collegamento, che potrebbe richiedere ritrasmissioni. La latenza è di circa un 1 secondo, oltre la durata del pacchetto, che è un buon risultato per un sistema di questo tipo.

Meza et al. [56] hanno presentato un metodo per determinare una configurazione ottimale di un array di sensori per il rilevamento e l'allerta per tsunami lontano dalla costa. La metodologia sviluppata è stata poi testata nel Cile del Nord. Il problema principale per la rilevazione di tsunami di tipo "near-field" è che c'è un tempo breve fra la rilevazione dell'evento e l'arrivo dell'onda sulla costa, per cui può essere complesso determinare l'origine dello tsunami in tempo. Inoltre, il costo dei sensori potrebbe essere un problema per cui lo studio vuole anche determinare una configurazione che consenta di ottimizzare la performance e mantenere il numero minimo di sensori. Alcuni fattori che possono influenzare le predizioni sono: posizione dei sensori rispetto all'energia dell'onda (paralleli od ortogonali rispetto al fronte d'onda); tempo di arrivo dell'onda di pressione al sensore (sensori più vicini all'evento avranno un tempo di "osservazione" minore e quindi meno dati per effettuare una predizione); profondità del sensore (deve essere abbastanza elevato da poter ignorare effetti di non linearità); posizione relativa dei sensori (la granularità della rete è scelta in maniera tale che sia maggiore della lunghezza d'onda dell'onda di pressione). I risultati ottenuti hanno mostrato che una configurazione a 3 sensori può fornire stime accurati sui tempi di arrivo dello tsunami e sull'ampiezza della prima onda.

Sistemi pre-esistenti (ad esempio stazioni sottomarini o reti di sensori nazionali, sia cablate che wireless) possono essere usati per raccogliere dati (ad esempio, sul movimento dei fondali) e usarli per effettuare predizioni e generare allerte. S-net è una rete cablata sottomarina lungo la fossa oceanica Giapponese che consiste di 150 unità di osservazioni, comprendenti sia sismometri che sensori di pressione, connessi con cavi ad intervalli di 30 km. Tutti i dati arrivano a terra in tempo reale usando cavi in fibra ottica [57]. Inoeu et al. [58] hanno sviluppato un metodo per la previsione quasi-real time degli tsunami usando i dati provenienti da S-Net, osservando le forme d'onda ottenuti dai sensori di pressione. Il metodo è in grado di classificare i sensori della S-Net in termini di distanza dalla regione di sollevamento del fondale dovuta al terremoto, osservando la pressione rilevata dai sensori e, osservando quali sensori sono vicini, al di fuori o molto lontani dalla regione, è in grado di computarne l'area. Questa tecnica è usata per stimare sia il punto di origine dello tsunami che la

magnitudo del terremoto. Poiché, oltre alle onde di pressione, anche le onde sismiche e le onde idro-acustiche sono rilevate dalla rete, è necessario filtrare accuratamente i segnali della S-net. La finestra temporale richiesta per effettuare le stime è di circa 500 secondi dopo il sisma iniziale, ed è richiesto un minuto per computare l'area di origine dello tsunami. La tecnica è stata validata con successo usando i dati di due eventi passati nella regione di installazione.

Virmani et al. [59] hanno proposto un framework ed un algoritmo di predizione per TEW nell'oceano Pacifico che misura diversi parametri ambientali che sono influenzati dagli tsunami e monitorano il comportamento della fauna marina. Ad esempio, uno tsunami genera cambiamenti nei campi magnetici, per cui sensori di campo magnetico possono essere usati per rilevare questi cambiamenti come precursori di uno tsunami; l'onda stessa presenta dei pattern particolari che possono essere misurati da un *Tide Gauge* (un comportamento di tipo "shallow wave" ossia con un rapporto minore tra lunghezza d'onda e profondità rispetto al comportamento normale delle onde generate dal vento); contatori e sensori di movimento sono infine usati per monitorare la fauna locale che potrebbe essere influenzata dai cambiamenti nei campi magnetici dovuti allo tsunami. Queste informazioni sono rilevate da una WSN sul fondale marino e trasmessi ad una stazione base usando GSM, e poi processati su un server dedicato. Le predizioni sono ottenute usando un modello di Machine Learning (logistic model trees) ed è stato dimostrato che le feature più rilevanti per le predizioni sono cambiamenti nei livelli del mare e nei pattern di migrazione.

Gardner-Stephen et al. [60] hanno proposto un modello per TEW per migliorare alcuni aspetti di sistemi già esistenti, specialmente in termini di costo della soluzione (che può essere proibitivo per alcuni paesi) e di comunicazioni in situazioni compromesse. La soluzione da loro proposta include sia un sistema di EW (basato su veicoli subacquei), una piattaforma che funge da sistema di decisione in grado di funzionare anche offline (nel caso in cui un disastro danneggi i sistemi di comunicazione standard), ed un sistema di distribuzione dell'allerta. Il sistema di EW utilizza veicoli low-cost (prodotti da uno degli autori) che sono caratterizzata da bassa mobilità, una modalità operativa principalmente passiva con basso consumo (che consente fino ad una settimana di operatività) e la possibilità di rilevare tsunami a distanza usando idrofoni. I veicoli possono anche cooperare con sensori tradizionali fissi e sono equipaggiati con moduli GPS e moduli per comunicazioni satellitari. Queste ultimi sono anche usate nel sistema di distribuzione degli allarmi per contrastare la possibilità che comunicazioni standard basate su IP vengano meno durante un evento tsunagenico.

Reti di sensori subacquee non sono le uniche soluzioni per TEW. Adi et al. [61] hanno sviluppato un dispositivo IoT per il monitoraggio dell'altezza delle onde marine usando un sensore ultrasonico. Il dispositivo dovrebbe essere installato sulla costa, direttamente sopra l'acqua. Quando la distanza tra il sensore e l'acqua sale sopra una certa soglia, il modulo GSM installato sul dispositivo trasmette un SMS con un messaggio di allerta prima che il livello dell'acqua diventi troppo alto. Un indicatore

### 1.3 Review dello Stato dell'Arte dei Sistemi di EW

sonoro è installato sul dispositivo per allertare le persone vicine. Il prototipo è stato testato con successo ma osservando che il sensore ultrasonico non può leggere livelli al di sopra di 3.19 m dal mare, e per questo è necessario un planning accurato prima dell'installazione, ed un'indagine sulle condizioni dei livelli dell'acqua prima dell'installazione.

Kato et al. [62] hanno progettato un sistema di boe con connessione satellitare per il monitoraggio oceanico e per l'allerta precoce per gli tsunami. Il primo sistema installato consisteva di una boa equipaggiata con un'antenna GNSS in grado di trasmettere informazioni sulla costa prima che l'onda lo raggiungesse. I dati ricevuti (livello dell'acqua, posizione della boa) consentivano all'utente di visualizzare il movimento della superficie marina nei pressi della boa in tempo reale. Lo sviluppo di questa soluzione è iniziato nel 1997, con sviluppi successivi negli anni 2001–2003, 2004–2006 and 2011. Queste soluzioni hanno mostrato che boe poste a 20 km dalla costa non davano abbastanza tempo per fornire allarmi sulla costa ed evacuare. Inoltre hanno fornito i dati necessari per contrastare alcuni problemi nel design delle boe, con lo scopo di consentire ad ognuna di esse di sopravvivere per almeno 10 anni. Poiché le distanze considerate possono anche superare i 100 km, è proposto l'uso dell'algoritmo *Precise Point Positioning*(PPP) per conoscere la posizione delle boe, e le comunicazioni satellitari sono in grado di fornire una copertura migliore della soluzione GNSS usata in precedenza. Gli esperimenti sul campo hanno mostrato una precisione di  $\pm 3$  cm nel posizionamento della boa, anche se un esperimento ha mostrato gap nei dati ricevuti, forse a causa dell'inclinazione assunta dalla boa, che può degradare la comunicazione. L'articolo inoltre propone l'uso delle boe per fare misure sul movimento del fondale, fornendo una precisione minore rispetto a sistemi come S-Net ma una migliore risoluzione. L'esperimento in tal senso ha coinvolto l'uso di 3 transponder sul fondale marino che comunicano con una boa usando un canale audio, e stimando la posizione dei transponder si è in grado di ottenere una misura del movimento della crosta subacquea.

Darmawan et al. [63] hanno progettato un dispositivo IoT per generare allerte precoci quando vengono rilevate anomalie nel comportamento delle onde marine, usando un algoritmo di tipo *Fuzzy Logic* per ottenere predizioni accurate. Il sensore consiste di un giroscopio, un accelerometro ed un modulo LoRa per la trasmissione dei dati. Il sensore galleggia nel mare e rileva e processa dati per ottenere informazioni sull'altezza delle onde e la loro velocità. Questi due parametri sono poi utilizzati per classificare l'onda usando l'algoritmo di Fuzzy Logic integrato nell'unità di sensing. I dati misurati sono mandati ad un server usando LoRa. L'algoritmo di classificazione fornisce sia un'etichetta di "pericolo" (*safe, standby, dangerous*) ed un livello di pericolo da 0 a 100, ottenuto computando l'altezza e la velocità dell'onda. L'algoritmo di Fuzzy Logic ha un'accuratezza che va dal 98% al 100% in fase di testing. Inoltre, la comunicazione LoRa è stata valutata in termini di latenza ed error rate, con un ritardo di 4.6279 s ed error rate minimale.

### 1.3.4 Discussione

La revisione di articoli e progetti relativi all'Early Warning che utilizzano il paradigma dell'IoT ha avuto lo scopo di evidenziare quali sono i requisiti di cui tener conto quando si realizza un progetto in questo ambito. In particolare, si vogliono delineare vincoli ed eventuali criticità sia per un generico sistema che per i tre use case che sono stati approfonditi. Quanto evidenziato nella sezione seguente verrà poi utilizzato ed approfondito nei Capitoli successivi nella realizzazione di un nodo Edge per Earthquake Early Warning che rispetti i requisiti di un buon sistema di allerta precoce.

#### Definizione dei Requisiti di un Sistema di EW

Tutti i sistemi IoT hanno dei vincoli che devono essere presi in considerazione quando si sviluppa una soluzione IoT. I sistemi EW in particolare devono produrre allarmi tempestivi utilizzando dati che di solito si ottengono da un'unità di sensing o una WSN, che avranno a loro volta i propri requisiti come consumo, copertura, garanzia di alta o totale integrità dei dati, e ritardi limitati. Le trasmissioni di dati e il processamento su livelli superiori dovrebbero anche essi essere ottimizzati per non aggiungere latenza al sistema. I seguenti requisiti possono essere definiti quando si progetta una soluzione IoT per l'Early Warning, anche prendendo in considerazione quanto visto nella review:

- *durata della batteria*: le WSN utilizzano sensori che devono durare per un lungo tempo, specialmente quando sono installati in luoghi che non sono semplici da raggiungere o terreni difficili che renderebbero la sostituzione di una batteria un task costoso. La valutazione di un 'energy budget' è essenziale per ogni applicazione prima dell'effettiva installazione, e l'acquisizione di dati e le trasmissioni dovrebbero essere ottimizzati per limitare il consumo di potenza, anche in condizioni di lavoro critici (come, ad esempio, situazioni ambientali che richiedono un maggior numero di misurazioni, o funzionamento in assenza di luce solare per dispositivi equipaggiati con unità fotovoltaiche etc.). I protocolli di comunicazione dovrebbero essere energeticamente efficienti e minimizzare overhead e ritrasmissioni [64].
- *resistenza alle failure e affidabilità*: il sistema dovrebbe funzionare anche quando uno o più nodi non sono più disponibili o se la topologia della rete di sensori cambia. Molti fattori possono determinare una situazione di malfunzionamento, ad esempio valori bassi delle batterie, cattiva copertura, la distruzione o il danneggiamento di un nodo etc. Una strategia comunemente utilizzata per preservare l'energia di un nodo consiste nel farlo andare in una modalità di risparmio energetico o *sleep mode* quando non sono utilizzati, per cui quel nodo non sarà "disponibile" alla rete quando in una di queste modalità, ed inoltre la mobilità di nodi o gateway può complicare ulteriormente il routing

e cambiare lo stato della rete, ma esistono numerosi protocolli di routing che tengono conto di questo tipo di situazione [64]. Protocolli che supportano le topologie di tipo *mesh* (Zigbee, Bluetooth), sono particolarmente utili perchè forniscono flessibilità alla rete in caso in cui un nodo o più vengano a mancare. Algoritmi "self-organizing" e di predizione delle failure sono quindi essenziali per consentire ad una rete di sensori di continuare a produrre allarmi affidabili. [39]. Inoltre, la fabbricazione e la scelta del rivestimento/contenitore del sensore devono essere fatte in maniera tale che continui a funzionare anche in cattive condizioni meteorologiche, inondazioni o urti [24].

- *copertura*: le regioni geografiche che devono essere coperte per un sistema IoT di EW possono essere molto ampie, e pertanto il protocollo di comunicazione scelto deve essere in grado di consentire comunicazioni a lungo raggio tra nodi distanti e gateway, con rate, latenze, tasso di packet loss ed altri parametri determinati in fase di progetto. Alcune location potrebbero avere effetti di *blockage*, *shadowing* o altri problemi che potrebbero compromettere le comunicazioni radio, per cui è essenziale valutare il link budget per l'applicazione che si sta realizzando per capire se i link radio realizzati funzioneranno con i parametri scelti. Ad esempio, una caratterizzazione dell'RSSI (*Received Signal Strength Indicator*) può essere utile per riuscire a comprendere il range di copertura del protocollo e dell'hardware utilizzati.
- *latenza*: i sistemi di EW dovrebbero fornire allarmi tempestivi, per cui i sistemi dovrebbero essere in grado di trasmettere dati velocemente e l'elaborazione di questi ultimi non dovrebbe impiegare troppo tempo. A seconda dell'applicazione, saranno richiesti vincoli temporali differenti e, come già visto nella sezione precedente, ci sono diversi protocolli di comunicazione che possono fornire tempi di trasmissione brevi, dall'ordine dei secondi fino ai millisecondi o anche sotto il millisecondo usando le nuove tecnologie 5G. Edge/Fog computing limitano la quantità di dati da trasmettere a livelli superiori, riducendo la latenza introdotta nella pre-elaborazione, analisi e processamento dei dati a livello applicazione. Inoltre, la scelta del giusto algoritmo di predizione o signal processing può essere essenziale per ridurre la latenza.

#### **Requisiti specifici degli Use Case analizzati**

Gli specifici scenari analizzati richiedono ovviamente la definizione di requisiti più specifici ed introducono problemi particolari. Ad esempio, una rete acustica sottomarina usata per TEW deve essere sia affidabile che efficiente energeticamente, ed introduce una serie di limitazione rispetto al canale radio: attenuazione che dipende fortemente dalla frequenza, effetti di multipath e delay variabili. Per risolvere questo genere di problema, possono essere usati schemi di modulazione basati su OFDM [65], e protocolli di routing dedicati svolgono un ruolo essenziale

nelle comunicazioni acustiche, anche dal punto di vista di ottimizzare i consumi energetici, la cui diminuzione tramite un routing dedicato può però comportare un aumento dei ritardi [54]. Inoltre, nell'Internet of Underwater Things (IoUT) l'aggregazione dei dati è resa più complessa dal fatto che la sovrapposizione di diversi percorsi di routing aumentano le collisione, incrementando i consumi dovuti alle ritrasmissioni e diminuendo l'affidabilità [66]. Quando si usano nodi mobili, come veicoli subacquei, ulteriori sfide insorgono in termini di routing, localizzazione dei nodi, consumi e sincronizzazione. Soluzioni basate su boe hanno anch'esse problemi rispetto ad una rete terrestre, poichè la connettività tra nodi vicini può essere influenzata dall'altezza dell'onda, e per questo protocolli di routing e processamento dedicati sono necessari[67], e l'inclinamento delle boe può degradare la comunicazione.

Alcuni importanti requisiti che sono emersi per quel che invece riguarda i sistemi di Early Warning per le alluvioni sono: i) copertura in aree remote o difficili da raggiungere, che potrebbero rendere certe soluzioni IoT non realizzabili in pratica, e rendendo la valutazione del link budget molto importante per comprendere se un'area può essere raggiunta, ad esempio per aree con denso fogliame; ii) consumo energetico, specialmente per certe condizioni ambientali o meteorologiche (come le citate "modalità oscure"), o quando è necessario effettuare un gran numero di misurazioni, per cui la valutazione dei "worst case scenarios" è essenziale prima di installare la soluzione proposta iii) resistenza ai guasti ed abilità del sensore o della rete di sopravvivere alla perdita di nodi, ad esempio a causa di esaurimento della batteria o condizioni meteorologiche estreme, anche se spesso le soluzioni proposte in grado di offrire resilienza alla perdita di nodi richiedono un gran numero di sensori e dispositivi nella rete, aumentando pertanto i costi. Per quel che riguarda il processamento dei dati, soluzioni di tipo Fog/Edge sono raccomandate per migliorare l'efficienza complessiva del sistema, ed inoltre modelli di Machine Learning possono anche essi migliorare i risultati delle predizioni ed essere integrati nei dispositivi Edge, seppur con maggiori vincoli computazionali e di memoria.

Per quel che infine riguarda i terremoti, a casa della sismicità moderata in Europa, lo sviluppo di sistemi di EEW è stato per lo più focalizzato sulla velocità piuttosto che sulla caratterizzazione della sorgente del sisma, anche se ricerca su questo aspetto sta progredendo [15], e le basse latenze, in termini di processamento dei dati, algoritmo di predizione e trasmissione dell'allarme, dovrebbe essere sempre un focus per consentire di generare e disseminare allarmi in maniera tempestiva. Un sistema di Early Warning per i terremoti ha in genere limiti in termini di accuratezza della predizione, il che potrebbe portare sia a falsi allarmi che a mancate rivelazione (o "missed detection"). Delle soglie di allerte più basse, o anche soglie multiple, possono fornire più falsi allarmi, ma minimizzano la possibilità di danni ad utenti o edifici. [68] Altri aspetti che sono stati messi in luce dalla review sono: i) il ruolo essenziale della sincronizzazione e la necessità di implementare protocolli di sincronizzazione per reti sismiche; ii) la necessità di realizzare sensori e reti resistenti ai guasti e con risparmio energetico; iii) una correlazione tra la densità di nodi nella rete

e la performance, sia per la qualità delle predizioni che per la resilienza della rete, poiché WSN dense mostrano una minor corruzione dei dati in caso di alta densità dei nodi [45]. Per quest'ultimo motivo, c'è stato uno sforzo nella letteratura nel realizzare sensori a basso costo, ad esempio usando sensori MEMS, per consentire la realizzazione di reti dense e low cost. Inoltre, si dovrebbe sempre progettare una certa ridondanza sia per i sensori che per i canali di comunicazione [51], e tener conto del consumo energetico.

Anche per i sistemi di Early Warning per i terremoti si è evidenziata la possibilità di realizzare sistemi di tipo Edge/Fog, specialmente per ridurre la latenza nel generare e disseminare allerte. Infatti, la diminuzione dei dati da trasmettere verso un server e dei tempi computazionali richiesti al server stesso per effettuare la predizione può ridurre significativamente i ritardi del sistema. Per fare ciò, si possono integrare modelli di Machine Learning o algoritmi predittivi direttamente sui sensori o sui gateway così che la predizione possa essere fatta a basso livello, tenendo sempre conto dei limiti dei dispositivi utilizzati nella scelta del corretto algoritmo o modello di ML.



## Capitolo 2

# Early Warning per Earthquake Detection

Nel seguente Capitolo ci si concentrerà sul problema dell'Early Warning sismico o Earthquake Early Warning (EEW). Già nel capitolo precedente si sono evidenziati i requisiti principali di un sistema di allerta precoce per i terremoti, come la latenza, l'interesse nello sviluppare sistemi di tipo Edge, e l'importanza nella scelta dell'algoritmo di Early Warning. In questo Capitolo in particolare si analizzeranno le principali caratteristiche del fenomeno sismico che consentono non solo la sua caratterizzazione ma soprattutto la possibilità di anticipare i suoi effetti distruttivi. Si introdurranno poi i principali metodi ed algoritmi che consentono di ottenere un'allerta precoce, in particolare analizzando in dettaglio l'algoritmo *Filter Picker* ed ottimizzandolo usando forme d'onda locali affinché possa essere utilizzato per lo sviluppo di un dispositivo Edge per l'Early Warning sismico.

### 2.1 Terremoti: onde sismiche, Strong Motion Parameters e strumenti di misura.

Un terremoto avviene a causa di un movimento improvviso di massa rocciosa all'interno della superficie terrestre. Spesso i terremoti sono causati da una rottura in una faglia (ossia la zona tettonicamente attiva di una placca terrestre) che "viaggia" lungo la faglia stessa. Il movimento della faglia dovuto alla rottura genera delle onde sismiche che radiano dal punto di rottura (chiamato *ipocentro*) e causano il movimento del terreno e la distruzione tipicamente associati ad un evento sismico. La proiezione dell'ipocentro sulla superficie terrestre è chiamata *epicentro*. L'individuazione dell'epicentro è molto importante per la caratterizzazione del sisma.

Il range di frequenze di un'onda sismica è molto ampio, dai 20 Hz fino a periodi molto lunghi, anche fino a 54 minuti [4]. Anche il range di ampiezze può assumere valori in un intervallo ampio, e l'ampiezza dell'onda determina la magnitudo che caratterizza il terremoto. Esistono diverse scale per misurare la magnitudo di un terremoto, e in seguito si farà riferimento in genere a due tipologie di scale: la *scala di magnitudo del momento sismico* o MMS (*moment magnitude scale*) e la *scala Richter*. La magnitudo in scala MMS è proporzionale al logaritmo del momento sismico ed è indicata con  $M_S$ , mentre la magnitudo in scala Richter è indicata con  $M_L$  ed è pari

al logaritmo dell'ampiezza massima registrata su un sismometro standard a 100 km dall'epicentro.

Oltre l'indicazione sulla magnitudo, esistono altri parametri che consentono di determinare le caratteristiche di un evento sismico, noti come Parametri di Strong Motion o *Strong Motion Parameters* (SMP). Lo SMP più semplice è la *Peak Ground Acceleration* (PGA), definita come il valore assoluto del picco massimo dell'accelerazione che caratterizza il terremoto, la cui componente orizzontale è legata alle forze indotte dal sisma sulle strutture colpite. L'integrale della PGA fornisce la *Peak Ground Velocity*, utile per caratterizzare terremoti a media frequenza. Infine, altri Parametri di Strong Motion interessanti sono: i) l'*Arias Intensity*, che rappresenta la dissipazione di energia per unità di massa di un sistema elastico-plastico e consente di ottenere informazioni su ampiezza e frequenza del sisma, ii) la *maximum deformation energy*  $E_{e,max}$ , usato per valutare l'energia del sisma e la sua capacità di danneggiare edifici e strutture, e legato direttamente allo spettro della velocità, iii) la *Spectrum Intensity* (SIH), utile per determinare il danno arrecato ad edifici dopo il sisma [3].

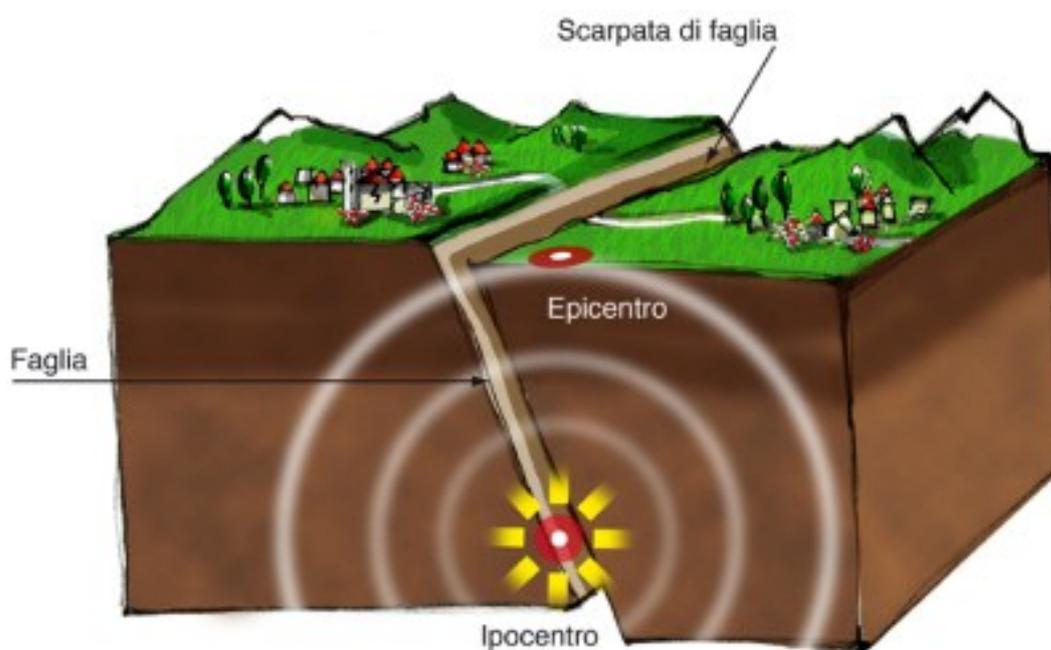


Figura 2.1: Rappresentazione di un terremoto con evidenziati l'ipocentro e l'epicentro. Dal sito dell'Istituto Nazionale di Geologia e Vulcanologia.

La misura delle onde sismiche è lo strumento principale per consentire la rilevazione di un terremoto. Possiamo distinguere tra due tipologie di onde: le *body waves* o onde di volume e le *surface waves* o onde di superficie. Mentre le *body waves* attraversano gli strati più interni della superficie terrestre, le onde di superficie viaggiano sulla crosta terrestre. Sono queste ultime le principali responsabili degli effetti distruttivi di un terremoto, con una gravità più o meno grande a seconda della loro profondità, e viaggiano più lentamente rispetto alle *body waves*. Inoltre, hanno frequenza minore

## 2.1 Terremoti: onde sismiche, Strong Motion Parameters e strumenti di misura.

e possono essere individuate più facilmente su un sismogramma. Invece, le onde di volume si propagano dall'ipocentro lungo tutte le direzioni di propagazione e attraverso il mezzo in cui è avvenuta la rottura che ha generato il sisma.

Il primo tipo di body wave è l'onda P o onda Primaria. Si tratta dell'onda più veloce e che raggiunge per prima lo strumento di misura. Le onde P possono muoversi attraverso la roccia solida o liquidi, e sono anche note come "onde di compressione", in quanto svolgono un'azione di spinta e tiraggio [3]. Le particelle attraversate da un'onda P si muovono nella direzione di propagazione dell'onda. Le onde S o onde secondarie invece sono body wave che muovono le particelle in direzione perpendicolare alla direzione di propagazione dell'onda, da un lato all'altro o dal basso all'alto. Sono più lente delle onde P e non possono propagarsi attraverso liquidi. Le onde S sono la seconda onda che viene percepita durante un terremoto.

In Figura 2.2 è riportato il grafico dell'accelerazione misurata all'inizio di un evento sismico in  $cm/s^2$ . Sono evidenziate tre finestre: i) una (in nero) contenente solamente rumore di fondo che precede l'evento sismico, ii) una (in rosso) che contiene l'onda P, iii) una finestra (in verde) che contiene l'onda S. Il contributo evidenziato dalle finestre rossa e verde è essenziale per individuare e classificare il terremoto.

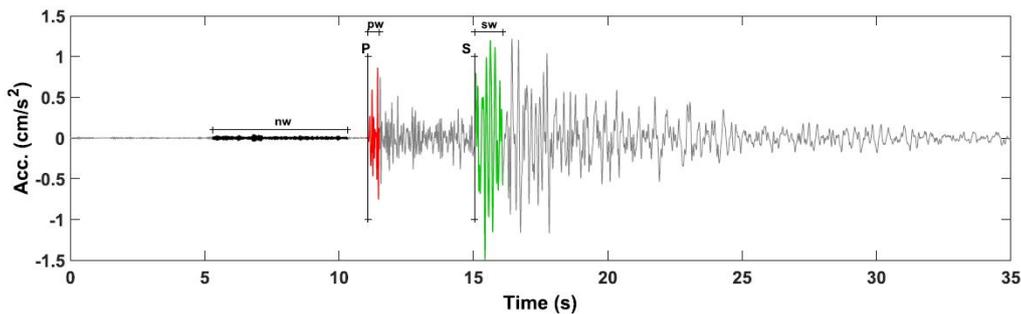


Figura 2.2: Accelerazione misurata durante un evento sismico con evidenziate le finestre temporali relative all'Onda P e all'Onda S. (da *Performance Evaluation of a Low-Cost Sensing Unit for Seismic Applications: Field Testing During Seismic Events of 2016-2017 in Central Italy* [3])

Il rumore di fondo individua invece la soglia al di sotto della quale lo strumento non è in grado di misurare il segnale utile in modo distinguibile dal rumore. Esistono diversi strumenti che consentono di misurare le onde sismiche, come ad esempio i sismometri. I sismometri moderni misurano il movimento del terreno e contengono elementi che consentono la misura lungo 3 direzioni ortogonali (updown, north-south e east-west). La rete dell'Istituto Nazionale di Geofisica e Vulcanologia (INGV) utilizza velocimetri ed accelerometri. I velocimetri sono di solito utilizzati per monitorare movimenti deboli, mentre le unità accelerometriche sono usate per la rilevazione di movimenti forti. Pertanto, nella rete INGV i velocimetri sono usati per migliorare la risoluzione della rete e consentire di misurare terremoti di piccola scala, e gli accelerometri per misurare sismi di maggiore entità ed ottenere le relative forme

d'onda senza che il sensore vada in saturazione.

Come evidenziato dalla review riportata nel capitolo precedente, l'utilizzo di sensori MEMS (Micro Electro-Mechanical System) può essere di interesse nella realizzazione di sistemi e dispositivi di Early Warning sismico. I sensori MEMS sono infatti dispositivi a basso costo e di piccole dimensioni che possono essere utilizzati per realizzare reti sismiche dense e diffuse, o consentire la realizzazione di dispositivi con costi contenuti. Come dimostrato da Pierleoni et al. [3], l'utilizzo di sensori MEMS all'interno di stazioni sismiche può considerarsi adatto al monitoraggio sismico ed alla sua caratterizzazione, ad esempio integrandoli in una rete sismica esistente, seppur con limiti relativi all'entità della magnitudo rilevabile imposti dal rumore di fondo e dalla sensibilità del sensore. Ad esempio, sotto i 2  $M_L$  l'unità di test sviluppata nell'articolo citato non è in grado di rilevare fasi P, che risulta indistinguibile dal rumore, mentre al di sopra di 2.5  $M_L$  si ottengono buone rilevazioni per distanze relativamente brevi ( $\leq 40km$ ) e con un buon rapporto segnale rumore. Anche in alcuni articoli revisionati nel capitolo precedente (come [48] e [46]) sono stati sviluppati sistemi basati su unità MEMS, testando il loro corretto funzionamento e la possibilità di utilizzarli al posto di unità sismiche 'classiche', evidenziando in particolare l'interesse verso i sensori MEMS per realizzare reti dense e low-cost. Quindi si può dire che è possibile riuscire ad utilizzare questa tipologia di sensore per la rilevazione di onde P e quindi anche per scopi relativi all'allerta precoce, in accordo con i limiti inferiori imposti dalla risoluzione e dal rumore di fondo dell'unità di sensing scelta.

## 2.2 Algoritmi per Earthquake Early Warning

Diversi "indizi" relativi al verificarsi di un terremoto sono da tempo noti ai geologi, come ad esempio quelli riportati in Figura 2.3, anche se non tutti sono supportati da misurazioni sperimentali [4]. Come già detto nella sezione precedente, la misura delle onde sismiche (tramite misura di accelerazione, velocità o spostamento del terreno) può essere un metodo efficace per la caratterizzazione di un terremoto ed è implementabile anche tramite sensori low-cost, ed inoltre le caratteristiche delle onde sono ben note e facilmente individuabili. Un semplice sistema di EEW può pertanto funzionare seguendo il principio per cui le onde P arrivano al sensore più velocemente delle onde S e delle onde di superficie, per cui possono innescare un sistema di allarme prima che l'effetto distruttivo dell'onda raggiunga la zona da monitorare. Diverse implementazioni di sistema di allerta precoce sismica usano questo principio, come visto anche nella review riportata nel capitolo precedente, che spesso sfruttavano la rilevazione delle onde P come metodo di allerta precoce. Nella Figura 2.4 è riportato il funzionamento del sistema di Early Warning *ShakeAlert*, installato in California e funzionante dal 2012 [5]. L'arrivo di un'onda P viene rilevato dai sensori, che trasmettono informazioni ad un centro di allerta, il quale elabora i dati ricevuti per generare e disseminare un allarme. Poiché l'informazione può essere trasmessa più

## 2.2 Algoritmi per Earthquake Early Warning

velocemente rispetto alla velocità di propagazione delle onde sistemiche, un sistema di questo tipo può essere utilizzato come sistema di allerta precoce. La rilevazione veloce ed affidabile dell'onda P tramite opportuni strumenti ed algoritmi diventa quindi un requisito essenziale del sistema.

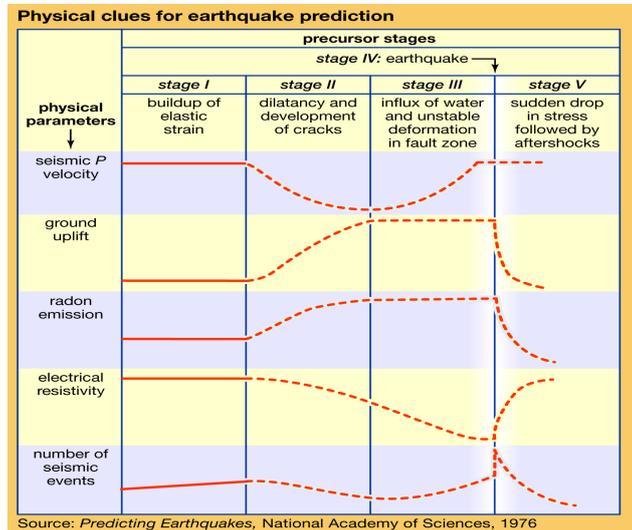


Figura 2.3: Diverse variazioni fisiche che precedono il verificarsi di un terremoto (da [4]).

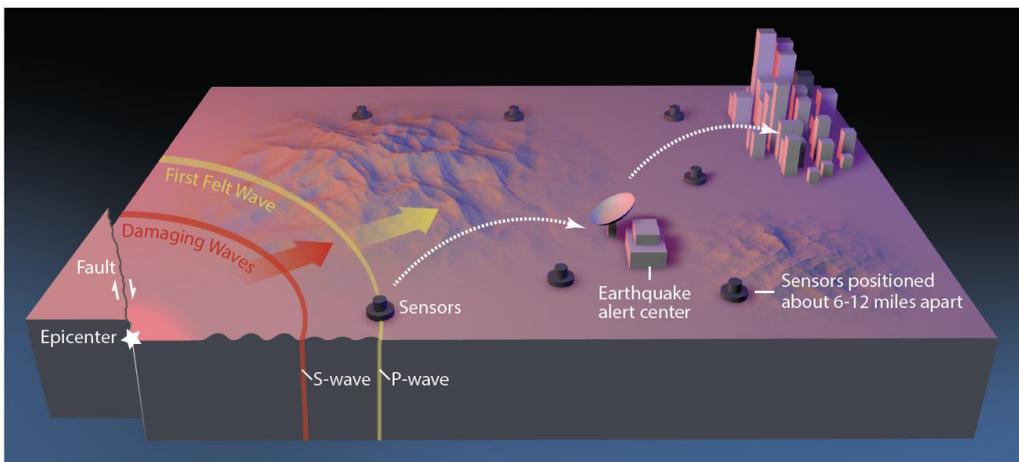


Figura 2.4: Schema del Sistema di Early Warning *ShakeAlert*, basato sul rilevamento di onde P (da [5]).

I sistemi di Early Warning per monitoraggio sismico sono già presenti in diversi paesi, ma ci sono anche sforzi per ottimizzare le reti sismiche, fornire una comunicazione affidabile e con basse latenze, e diminuire la latenza dovuta al processing dei dati. Quando un messaggio di Early Warning è generato da una rete sismica, sensore o centro di processamento, ogni millisecondo di ritardo nel trasmettere il messaggio corrisponde ad un incremento di 8 metri del raggio dell'area colpita dal

terremoto [52]. Ridurre la latenza è quindi un requisito essenziale nell'Early Warning sismico e devono essere considerati precisi requisiti temporali. I tempi di allerta e la raccolta di dati geologici dovrebbe quindi rientrare nel range dei secondi fino a qualche minuto. Poiché la maggior parte dei danni dovuti ad un terremoto sono di solito localizzati in un'area entro un certo perimetro attorno all'epicentro, l'obiettivo più critico di un sistema di EEW è di fornire allarmi entro secondi verso questa area, mentre allerte più accurate possono essere disseminate in decine di secondi dopo l'evento [15]. Con algoritmi di data processing ottimizzati per ridurre la latenza fino al secondo, è possibile ottenere performance di picco di fino 3-6s per generare e disseminare l'allarme [69]. La tabella 2.1 elenca alcune tecniche ed algoritmi generici per Earthquake EW con un focus sulla performance relativa a latenza e complessità computazionale.

Algoritmo	Metodo	Vantaggi	Svantaggi
<b>Point source algorithms</b>	Usano dati relativi alle onde P da 2 fino a 6 stazioni attorno all'epicentro per caratterizzare il sisma.	Molto veloce.	Satura per terremoti di alta intensità.
<b>Finite fault algorithms</b>	Stimano l'estensione della rottura della faglia che generano il sisma.	Possono rilevare terremoti su grandi aree e con migliore accuratezza di altri metodi.	Più lenti.
<b>Modelli di Ground Motion</b>	Non caratterizzano la sorgente del terremoto, ma usano una stima del movimento del terreno attuale per prevedere il movimento futuro.	Non suscettibile all'incertezza nella determinazione della magnitudo e dell'ipocentro e nel gestire sorgenti sismiche multiple.	L'accuratezza della predizione decresce come una funzione del tempo di allerta.
<b>Metodi Onsite</b>	Si utilizzano sensori nello stesso sito dell'evento da monitorare, sia implementando semplici soglie o algoritmi per la rilevazione di onde P, generando un allarme se l'evento misurato soddisfa determinati criteri.	Semplici e veloci.	Rileva l'evento solamente nel punto in cui il dispositivo è installato.

Tabella 2.1: Algoritmi e metodi per Earthquake Early Warning (tabella ricavata da [15]).

In seguito andremo a porre l'attenzione sugli algoritmi di Early Warning *on site*, ossia algoritmi che vengono utilizzati per la rilevazione di un evento sismico nel luogo in cui l'unità di sensing (ad esempio, un sensore accelerometrico connesso ad un dispositivo embedded con capacità di comunicazione) è installata.

Gli algoritmi per Earthquake Early Warning on-site di solito utilizzano la rilevazione delle onde P per individuare la presenza di un terremoto. Gli algoritmi che vengono utilizzati per individuare o effettuare il "picking" di tale onda devono essere veloci,

precisi e possibilmente poco complessi, specialmente se l'implementazione deve essere fatta su dispositivi con limiti di memoria e capacità computazionale.

Il "picking" dell'onda sismica può essere definito come "l'inviduazione dell'istante temporale in cui l'energia della fase dell'onda P arriva al sismometro" [70], individuato da un cambiamento in energia, ampiezza, contenuto in frequenza o polarizzazione rispetto al rumore di fondo. L'inviduazione di un'onda S è invece un task più complesso, in quanto spesso la sua forma d'onda è contaminata dalla coda dell'onda P che l'ha preceduta. Inoltre, mentre la rilevazione di entrambe le onde è importante per la localizzazione e precisa caratterizzazione di un terremoto, nel contesto dell'allerta precoce risulta essere più interessante la rilevazione della fase dell'onda P, in quanto consente effettivamente di "predire" l'evento distruttivo che segue l'arrivo di tale onda.

Il metodo più semplice di rilevazione di un'onda P è quello di osservare l'uscita di un sismometro ed individuare un picco nella forma d'onda rilevata. Questo può essere fatto abbastanza facilmente anche da un operatore ad occhio nudo, ma ovviamente un metodo automatico è molto più veloce e costante nella sua performance rispetto alla rivelazione "manuale", ed in generale tecniche di questo tipo superano le prestazioni di un operatore umano.

Esistono diverse tipologie di algoritmi per rilevare la fase di un'onda P, sia nel dominio del tempo che nella frequenza. I metodi dovrebbero fornire informazioni sull'istante di rilevazione, incertezza di tale dato e ampiezza del terremoto, preferibilmente in tempi dell'ordine del secondo rispetto all'istante di rilevazione. Nel caso di implementazione su sistema integrato, si deve anche tener conto che l'algoritmo scelto non deve avere un "peso" computazionale troppo elevato.

Tra i metodi più semplici ed utilizzati ci sono i metodi STA/LTA o *energy methods*, basati sul metodo introdotto da Allen nel 1978. Questo metodo consiste nell'andare a confrontare le medie su brevi istanti (*short time averages* o STA) di una funzione caratteristica (*characteristic function* o CF) e le medie su istanti lunghi (*long time averages* o LTA) della medesima funzione caratteristica. Questa può essere anche il semplice valore assoluto o valore al quadrato del segnale utile (solitamente pre-filtrato) o avere una rappresentazione più complessa in termini delle caratteristiche statistiche del segnale, e la rilevazione avviene quando il rapporto tra le due medie di questa funzione (il rapporto STA/LTA) supera una determinata soglia.

Un'altra categoria di algoritmi è quella dei metodi autoregressivi, i quali vengono utilizzati per rilevare un picco dopo che l'inizio di un'onda P è già stato rilevato. Seguono il principio secondo il quale una serie di campioni che contiene solamente rumore ha caratteristiche statistiche diverse rispetto ad una serie che contiene un evento sismico [71]. L'utilizzo di metodi autoregressivi, seppur dia risultati estremamente precisi, non è solitamente adatto alla rilevazione in tempo reale di eventi sismici, ed inoltre richiede una buona stima a priori del tempo di arrivo dell'onda, altrimenti le sue prestazioni degradano o richiedono un eccessivo sforzo computazionale [6].

Metodi multibanda come il metodo AMPA (Adaptive Multiband Picking Algorithm) sono particolarmente utili per il processamento di segnali sismici con forti contributi di rumore. Come principio utilizzano il comportamento osservato su uno spettrogramma: infatti guardando lo spettrogramma di un evento sismico si nota come all'istante di arrivo dell'onda P si osservano picchi di energia su diverse bande, come mostrato nella Figura 2.5. Pertanto i metodi multibanda dividono il segnale in tanti segnali passabanda con diverse frequenze centrali, per ognuno dei quali si ricavano degli involuipi o altre caratteristiche tramite cui, eventualmente dopo ulteriori operazioni di filtraggio, si costruisce anche qui una funzione caratteristica.

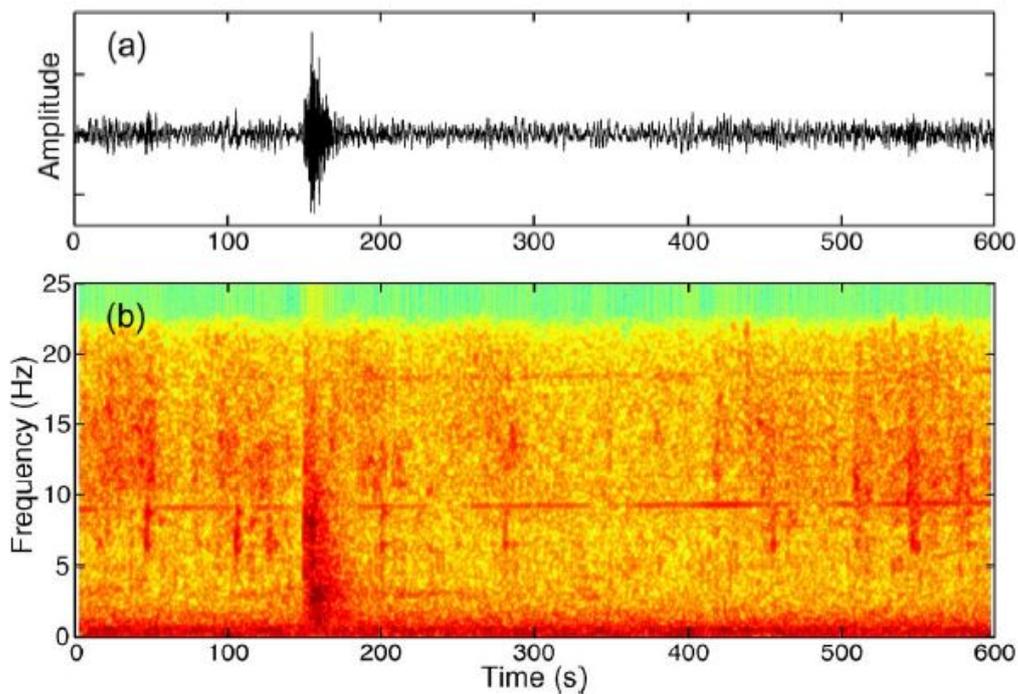


Figura 2.5: Accelerazione e spettrogramma di un evento sismico (da *An Automatic P-Phase Picking Algorithm Based on Adaptive Multiband Processing* [6]). Si nota come attorno all'istante  $t=150$  s si ha un picco di energia su diverse bande, in corrispondenza dell'arrivo dell'onda P.

Esistono poi metodi basati su reti neurali; metodi nel dominio della frequenza; metodi basati sulla polarizzazione; metodi basati su *wavelet analysis* etc. Le tendenze attuali consistono nell'andare a combinare i diversi metodi elencati, per sfruttarne al meglio i punti di forza [70]. Ad esempio, il metodo AMPA può essere migliorato in termini di accuratezza del picking utilizzandolo in tandem con un metodo autoregressivo, come mostrato nell'implementazione in [6], anche se una simile implementazione però non è auspicabile nel contesto di lavorare con vincoli di memoria o capacità computazionali. Anche l'algoritmo Filter Picker, derivato dall'algoritmo di Allen,

combina gli energy methods con l'approccio di tipo multibanda, come si approfondirà nella sezione successiva.

## 2.3 Algoritmo Filter Picker

L'algoritmo Filter Picker rientra tra gli "energy methods" ed è basato sull'algoritmo di Allen. Si tratta di un algoritmo molto semplice ed adatto ad applicazioni con forti vincoli computazionali, e quindi una buona scelta per applicazioni embedded o di reti di sensori. Infatti, utilizza solamente operazioni logiche ed aritmetiche semplici, come le radici quadrate, e senza dover ricorrere ad esponenziali, logaritmi o trasformate. Inoltre, come già accennato, utilizza il principio dell'analisi multibanda di cui si è parlato in merito dell'algoritmo AMPA nella sezione precedente, andando ossia a generare una funzione caratteristica lavorando su diverse bande di frequenza, a partire da una serie di segnali passabanda ottenuti dal segnale di partenza, in principio a banda larga. Unisce quindi la semplicità dei metodi STA/LTA con l'efficacia rispetto a segnali affetti da rumore dei metodi multibanda.

Filter Picker utilizza una serie temporale di campioni di accelerazione o velocità come segnale utile. Su questi può anche essere applicata una qualche operazione di filtraggio, ma l'algoritmo in principio funziona anche utilizzando un segnale broadband qualsiasi, andando ad utilizzare ad esempio dati accelerometrici non processati.

Come già detto, i metodi di tipo STA/LTA come l'algoritmo di Allen utilizzano una funzione caratteristica che rappresenta una qualche statistica del segnale. In questo caso la funzione viene calcolata a partire da un segnale differenziale  $y'(i)$ , ottenuto da un segnale utile tempo-discreto  $y(i)$  con tempo di campionamento  $\Delta T$ . Lo scopo di questa differenziazione iniziale è quello di accrescere le alte frequenze del segnale utile. Si ottiene quindi

$$y'(i) = y(i) - y(i - 1) \quad (2.1)$$

con  $i$  indice dell' $i$ -esimo campione e  $y(0)$  inizializzato al valor medio di  $y(i)$  lungo un intervallo lungo  $T_{long}$ .

A partire da  $y'(i)$  si ottengono una serie di segnali applicando un doppio filtraggio passa-alto ed un filtraggio passa-basso, seguendo i passi seguenti, con  $n$  da 0 a  $N_{band} - 1$

$$Y_n^{HP1}(i) = C_n^{HP} [Y_n^{HP1}(i - 1) + y'(i) - y'(i - 1)] \quad (2.2)$$

$$Y_n^{HP2}(i) = C_n^{HP} [Y_n^{HP2}(i - 1) + Y_n^{HP1}(i) - Y_n^{HP1}(i - 1)] \quad (2.3)$$

$$Y_n^{LP}(i) = C_n^{LP} [Y_n^{LP}(i - 1) + Y_n^{HP2}(i) - Y_n^{HP2}(i - 1)] \quad (2.4)$$

I parametri di filtraggio sono riportati nella tabella seguente. Si noti come vengano utilizzati solamente dei semplici filtri a un polo per limitare la complessità di questa operazione.

Costante del filtro passa-alto	$C_n^{HP}$
Costante del filtro passa-basso	$C_n^{LP}$
Costante di Tempo	$w_n = T_n/2\pi$
Frequenza di taglio	$f_n = \frac{1}{T_n} = \frac{1}{2^n \Delta T}$
$n$	da 0 a $N_{band} - 1$

Tabella 2.2: Parametri di Filtraggio.

La scelta dei parametri precedenti è essenziale sia per consentire la buon riuscita dell'algoritmo di picking che per evitare che il costo computazionale delle operazioni di filtraggio e successive sia eccessivo.  $N_{band}$  viene scelto in maniera tale che  $T_{N_{band}} = 2^{N_{band}} \Delta T$  sia maggiore del periodo dominante delle fasi da analizzare. Allo stesso tempo, il tempo di pre-processamento aumenta proporzionalmente con  $N_{band}$ , che quindi deve essere limitato per applicazioni che richiedono basse latenze, come quelle di allerta precoce. Lomax et al. suggeriscono di utilizzare  $N_{band} = ceiling[\log_2(T_{max}/\Delta T) + 1]$  con  $T_{max}$  lunghezza del periodo dominante del segnale sismico.

Al termine di questo pre-processamento, si ottengono delle sequenze filtrate  $Y_n^{LP}(i) = Y_n(i)$  con differenti periodi al di sotto del rispettivo  $T_n$ . Per  $n$  grandi, si ottengono sequenze di periodo maggiore. Per  $n=0$  si ottiene invece la semplice sequenza iniziale doppiamente differenziata.

La funzione caratteristica è ottenute andando a definire un involuppo per ogni serie filtrata  $Y_n(i)$ .

$$E_n(i) = Y_n^2(i) \quad (2.5)$$

La funzione caratteristica  $F_n$  è poi ottenuta come

$$F_n(i) = \frac{E_n(i) - \langle E_n \rangle (i-1)}{\langle \sigma(E_n) \rangle (i-1)} \quad (2.6)$$

in cui  $\langle E_n \rangle (i-1)$  è il valor medio fino all'istante  $i-1$  dell'involuppo, mentre  $\langle \sigma(E_n) \rangle (i-1)$  la sua deviazione standard, entrambi calcolati lungo il periodo  $T_{long}$ .  $F_n$  quindi rappresenta la variazione dell'involuppo rispetto al livello di fondo (rappresentato dal suo valore medio) e scalata di  $\sigma_n$ . Poiché si ottengono in questo modo  $n$  funzioni caratteristiche, una per ogni sottobanda definita dal suo  $T_n$ , infine si calcola una singola funzione "riassuntiva"  $F_C$

$$F_C(i) = max[F_n(i)] \quad con \ n = [0, N_{BAND} - 1] \quad (2.7)$$

Ossia si prende il valore massimo per ogni campione tra tutte le  $n$  funzioni caratteristiche. L'utilizzo del valore massimo risulta utile quando il rumore di fondo

risulta essere alto o di ampiezza simile alla fase dell'onda P. Un altro metodo di formare la funzione caratteristica finale può essere quello di sommare le singole  $F_n$ , ma questo andrebbe a sommare anche i contributi di rumore su tutte le funzioni caratteristiche, rendendo meno probabile una rivelazione corretta della fase.

Definita la funzione caratteristica che rappresenti le caratteristiche statistiche del segnale e vari in tempo reale con esso, come la funzione  $F_C(i)$ , si deve scegliere un metodo per individuare un "trigger" e dichiarare un pick, ossia definire l'effettivo algoritmo per rilevare un picco nella fase dell'onda P. A questo scopo si definiscono due soglie  $S_1$  ed  $S_2$  che vengono utilizzate rispettivamente per generare un trigger e per dichiarare un picking della fase.

Un trigger viene generato quando  $F_C(i) \geq S_1$ . L'istante di trigger  $t_{trig}$  viene salvato assieme alla più alta banda di frequenza per cui  $F_n(i) \geq S_1$ , indicata con  $k$ . Definito un tempo  $T_{UP}$ , allora si dichiara un pick avvenuto se nella finestra tra  $t_{trig}$  e  $t_{trig} + T_{UP}$  l'integrale di  $F_C(i)$  è maggiore di  $S_2$ . Per evitare un numero di dichiarazioni eccessivamente lunghe, ad esempio per grandi eventi, dopo che un pick viene dichiarato non è più possibile dichiarare altri pick prima che  $F_C(i)$  diventi minore di  $F_C(i) = 2$ . Algoritmi come il BK87 (Baer and Kradolfer) usano una combinazione di valor medio e deviazione standard della funzione caratteristica per variare dinamicamente il valore delle soglie utilizzate, mentre algoritmi come il Filter Picker (basati sull'algoritmo di Allen) utilizzano soglie fisse.

L'algoritmo è riassunto in seguito.

Algoritmo 2.1: Algoritmo di triggering e picking.

---

```

1  input:  $S_1, S_2, F_C(i), F_n(i), T_{UP}$ 
2  output:  $t_{trig}, k$ 
3
4  if  $F_C(i) \geq S_1$ 
5      memorizza  $t_{trig}, k$ 
6
7  if  $\sum_{t_{trig}}^{t_{trig}+T_{UP}} F_C(i)\Delta T \geq S_2$ 
8      dichiara il pick nella banda  $k$ 

```

---

Rilevato un evento, è interessante anche andare a ottenere un'indicazione sull'istante in cui è avvenuto, l'incertezza nella sua rilevazione e la sua ampiezza.

Ogni qualvolta che  $F_n(i)$  assume un valore maggiore del valore medio  $\langle F_n \rangle$  per quella banda, si salva un potenziale istante di picking  $t_n^{pick}$ . Se c'è stato un trigger per la banda  $n = k$ , si calcola la differenza  $t_k^{pick} - t_{trig}$  per quella banda, e se tale differenza è minore di  $T_k/40$ , allora si ritarda  $t_{trig}$  fino a soddisfare questa condizione (ossia fin quando  $t_k^{pick}$  precede il tempo di trigger). L'istante di picking  $t_{pick}$  è pari a  $t_k^{pick}$ , e avviene prima di  $t_{trig}$ . L'incertezza temporale è posta pari alla differenza  $\sigma_{pick} = t_k^{pick} - t_{trig}$ , dando complessivamente un'informazione sull'istante di picking pari a  $t_{pick} \pm \sigma_{pick}$ . Il valore  $F_C(t_{trig})$  è preso come intensità  $I$  dell'evento.

Algoritmo 2.2: Determinazione di istante di picking, sua incertezza e intensità dell'evento.

---

```

1  input:   $F_C(i)$ ,  $F_n(i)$ ,  $\langle F_n \rangle$ ,  $k$ ,  $t_{trig}$ 
2  output:  $t_{pick}$ ,  $I$ 
3
4  for n=0:N-1
5      if ( $F_n(i) \geq \langle F_n \rangle$ )
6          memorizza il possibile istante di picking  $t_n^{pick}$ 
7  if rilevato un trigger nella banda k
8      if  $t_k^{pick} - t_{trig} \leq T_k/40$ 
9          ritarda  $t_{trig}$ 
10      $\sigma_{pick} = t_k^{pick} - t_{trig}$ 
11      $t_{pick} = t_k^{pick} \pm \sigma_{pick}$ 
12      $I = F_C(t_{trig})$ 

```

---

Infine, la polarità del pick si trova sommando i valori e i valori assoluti del segnale differenziale filtrato  $Y_n(i)$  per  $n = k$ , tra l'istante di trigger e quello di pick. Se il valore della somma è circa 2/3 maggiore rispetto al valore della somma dei valori assoluti, allora la polarità è data dal segno della somma, altrimenti è indicata una polarità "incerta".

Algoritmo 2.3: Determinazione della polarità del pick.

```

1  input:   $Y_n(i)$ ,  $k$ ,  $t_{trig}$ 
2  output: polarità
3
4   $sum1 = \sum_{t_{trig}}^{t_{pick}} Y_n(i)$ 
5   $sum2 = \sum_{t_{trig}}^{t_{pick}} |Y_n(i)|$ 
6
7  if  $sum1 \geq \frac{2}{3} sum2$ 
8      la polarità è pari al segno di sum1
9
10 else la polarità è incerta

```

---

## 2.4 Testing e Ottimizzazione dell'Algoritmo Filter Picker

Prima di implementare l'algoritmo Filter Picker sul sistema embedded scelto, lo si è testato utilizzando Matlab. Il software Matlab del Filter Picker è disponibile a <https://it.mathworks.com/matlabcentral/fileexchange/69211-filterpicker-a-robust-broadband-phase-detector-and-picker>.

In Figura 2.6 è riportata l'uscita dell'algoritmo Filter Picker usando la forma d'onda disponibile nel pacchetto Matlab. Sono impostati i parametri standard riportati in Tabella 2.3. Come si vede dall'immagine, l'algoritmo individua un istante di picking (in rosso nella prima traccia della figura) ed un istante di triggering (in blu), che avviene dopo il picking o è ritardato in maniera tale che venga soddisfatta questa condizione. La prima traccia inoltre riporta l'ingresso non processato (in nero), mentre le tracce sottostanti rappresentano i segnali passabanda che vengono utilizzati per formare la funzione CF "riassuntiva" (non riportata per migliorare la leggibilità).

Notiamo come con questa particolare forma d'onda si ha un numero di pick  $N_p = 1$ . In generale, è bene che si eviti un eccessivo numero di picking [71][72], specialmente se utilizziamo la rilevazione di un pick come condizione per la trasmissione di un

## 2.4 Testing e Ottimizzazione dell'Algoritmo Filter Picker

Parametro	Valore Standard	Significato
$T_{long}$	$500\Delta T$	Finestra su cui si calcolano le medie temporali.
$T_{filter}$	$300\Delta T$	Periodo massimo dei segnali passabanda ottenuti dal segnale differenziale.
$T_{UP}$	$20\Delta T$	Massima distanza temporale tra picking e triggering.
$S1$	10	Soglia per il triggering.
$S2$	10	Soglia per il picking.

Tabella 2.3: Parametri dell'Algoritmo e loro valori standard.

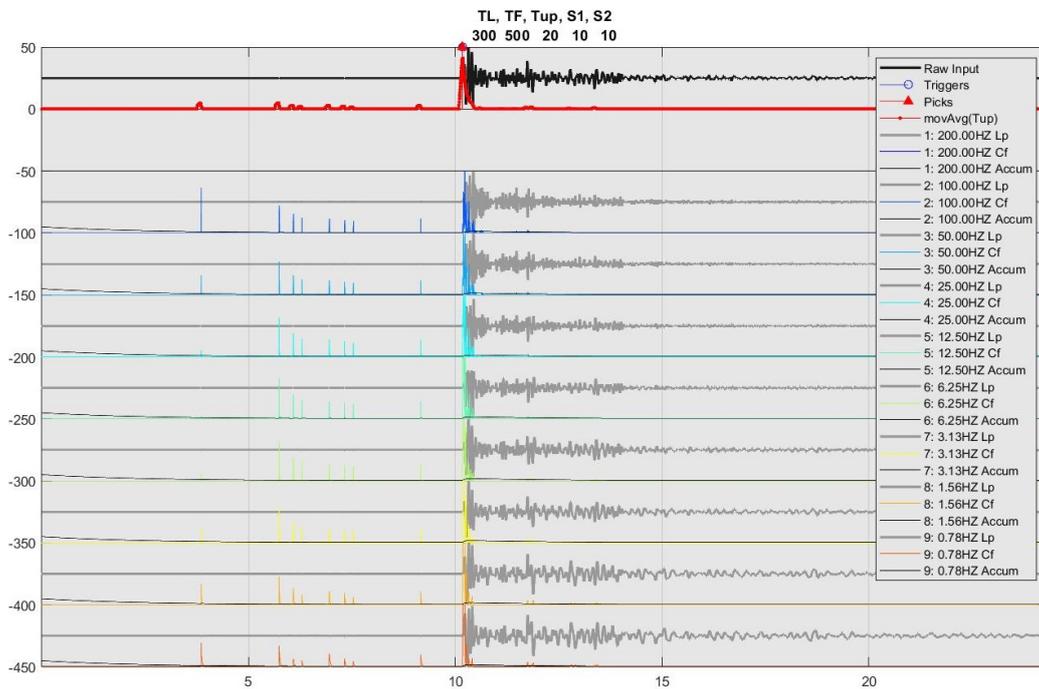


Figura 2.6: Uscita del Filter Picker utilizzando la forma d'onda disponibile nel relativo pacchetto Matlab.

allarme. In questo caso, infatti, si avrebbero tante trasmissioni successive che, specie su un nodo Edge, comporterebbero un eccessivo consumo di potenza.

Ripetiamo il test con altre due forme d'onda ed osserviamo l'uscita dell'algoritmo. Le forme d'onda sono state prese dall'Engineering Strong Motion Database [7] e sono relative alla stazione Gualdo di Macerata o stazione GUMA installata nelle Marche, in Italia. I segnali sono segnali accelerometrici non processati che corrispondono a sismi di magnitudo rispettivamente 4.1 e 5  $M_L$ .

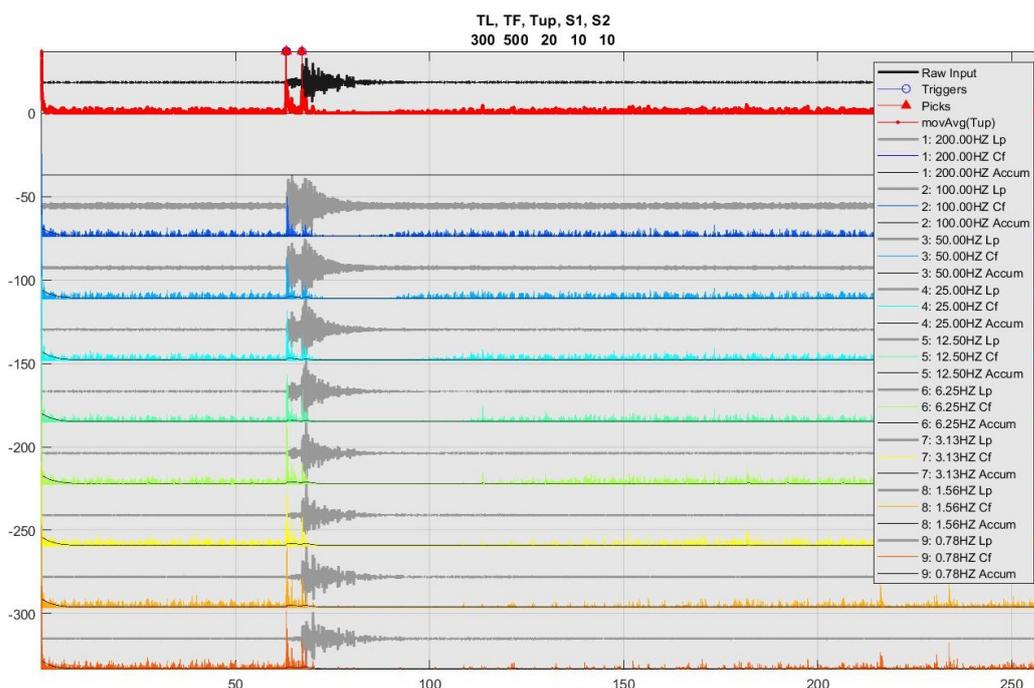


Figura 2.7: Uscita del Filter Picker utilizzando un sisma di magnitudo 4.1 $M_L$  [7]

Mentre per la prima forma d'onda abbiamo un'immediata rilevazione del sisma, seppur con un numero maggiore di pick ( $N_P = 3$ ), per la seconda forma d'onda non si è in grado di rilevare l'evento sismico. Poiché il primo sisma è di intensità minore del secondo, ne deduciamo che sono le caratteristiche delle forme d'onda che influenzano la capacità di rivelazione del terremoto, e non semplicemente l'entità dell'evento sismico, e i valori standard utilizzati dall'algoritmo non sono adatti alle particolari forme d'onda scelte.

L'algoritmo dovrebbe quindi essere preferibilmente ottimizzato utilizzando forme d'onda "locali" al luogo o alla regione di installazione. Per questo motivo, si è eseguita una procedura di ottimizzazione seguendo quanto fatto da Vassalo et al. [72], seppur con una leggera differenza di implementazione.

Vogliamo che l'algoritmo ottimizzato consenta di ottenere le seguenti prestazioni:

- Rilevare terremoti per diverse forme d'onda e intensità;

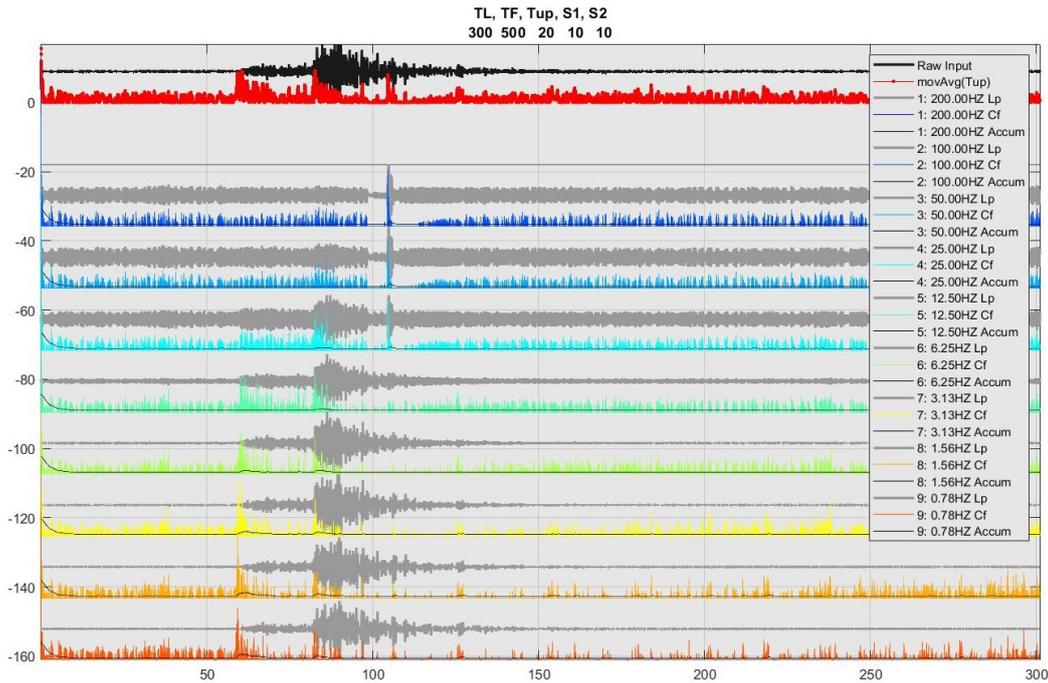


Figura 2.8: Uscita del Filter Picker utilizzando un sisma di magnitudo  $5M_L$  [7]

- Evitare che vengano ottenuti troppi pick dell'onda P, specialmente per eventi lunghi, dopo il primo pick;
- Evitare che si ottengano falsi allarmi dovuti a pick non relativi ad un evento sismico, ma ad esempio al rumore di fondo.

Come in ogni procedura di ottimizzazione, si deve implementare una "funzione di fitness", ossia una qualche funzione che consenta di valutare le prestazioni dell'algoritmo man mano che questo viene applicato ad una serie di eventi sismici. In particolare, dovrà essere una funzione che, per ogni evento su cui si ottimizza l'algoritmo:

- è tanto più alta quanto più è accurata la rilevazione dell'evento;
- aumenta se il numero di pick è al di sotto di una soglia prestabilita, mentre diminuisce se c'è un picking eccessivo;
- è nulla se l'evento non viene rilevato;

La funzione di fitness ha valori compresi fra 0 e 1, ossia nel caso peggiore non rileverà alcun evento, mentre nel caso migliore rileverà  $N_P$  eventi, con  $N_P$  il numero massimo di picking consentiti. Per ogni evento, utilizziamo come riferimento temporale il primo istante di trigger (indicato con  $t_{trig}^{best}$ ), non avendo a disposizione l'informazione relativa al picking "manuale", che invece veniva usata nell'implementazione di Vassallo et al.

Possiamo definire la funzione in questo modo:

$$G = \frac{1}{M} \sum g(i) \quad (2.8)$$

in cui  $M$  è un fattore di normalizzazione pari al numero di eventi su cui si applica l'algoritmo, mentre  $g(i)$  è un valore relativo all' $i$ -esimo evento sismico. Distinguiamo 3 valori possibili per  $g(i)$ , a seconda del numero di pick per quell'evento, indicato con  $N_P$ .

$$\begin{aligned} N_P &\leq P \\ g(i) &= \exp(-(t_i^{best} - t_{trig}^{best})) \end{aligned} \quad (2.9)$$

nel caso in cui il numero di pick sia pari a  $N_P \leq P$ , con  $P$  soglia relativa al numero massimo di pick ammissibili e  $t_i^{best}$  l'istante di picking più vicino al primo trigger.

$$\begin{aligned} N_P &> P \\ g(i) &= \frac{1}{N_P} \sum \exp(-(t_i^{best} - t_{trig}^{best})) \end{aligned} \quad (2.10)$$

Nel caso invece in cui si abbia un numero eccessivo di pick, si introduce una penalizzazione proporzionale a  $N_P$ , riducendo il valore di  $g(i)$ .

$$\begin{aligned} N_P &= 0 \\ g(i) &= 0 \end{aligned} \quad (2.11)$$

Infine, quando non si rileva nessun evento sismico, si impone che il valore  $i$ -esimo della funzione di fitness sia nullo. In questa maniera si è ottenuta una funzione che fornisce una stima di quanto "bene" l'algoritmo si comporti con un determinato evento, tenendo conto non solo della mancata rilevazione ma anche dell'eccesso nel segnalare picking.

### 2.4.1 Algoritmo 'Genetico' per l'ottimizzazione

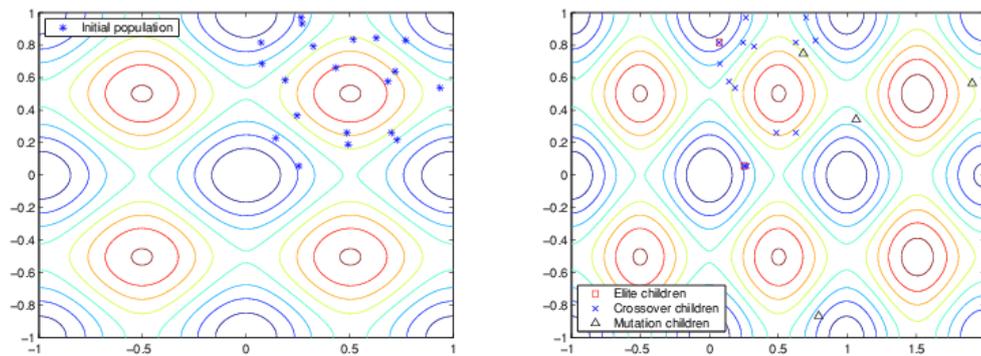
Come algoritmo di ottimizzazione si è utilizzato l'Algoritmo Genetico, come fatto da Vassallo et al. [72] nel loro articolo in cui è stato ottimizzato il Filter Picker.

L'Algoritmo Genetico è un algoritmo di ottimizzazione basato sulla teoria dell'evoluzione di Darwin. Utilizza la funzione di Fitness definita nella sezione precedente per determinare se una soluzione (indicata con il nome di "individuo") è adatta ("fit") a "sopravvivere" e a procedere al passo successivo dell'algoritmo, fino a raggiungere la soluzione finale che ottimizza il problema.

Il processo di ottimizzazione inizia andando a definire una "popolazione", ossia una serie di individui caratterizzati da parametri (detti "geni") che insieme formano un "cromosoma", ossia una possibile soluzione del problema. Usando la funzione di fitness sulla popolazione iniziale, si scelgono gli individui che contribuiranno alla

creazione della popolazione successiva tra quelli che hanno un *fitness score* maggiore. Da questi si formano coppie di "genitori" che formeranno una nuova generazione.

La procedura di creazione di un nuovo individuo a partire dai "genitori" può produrre tre tipi di individuo "figlio": *elite*, *crossover*, *mutato*. Solo il genitore con il miglior fitness score può produrre un figlio di tipo elite, il quale eredita il medesimo cromosoma del genitore. La procedura di *Crossover* invece sceglie un punto random nel cromosoma di ogni genitore, e poi scambia i geni dei genitori a partire dal primo fino al punto casuale. Quando si raggiunge questo punto, si sono ottenuti i due individui figli di tipo crossover che contribuiranno alla seconda popolazione. Infine, è possibile aggiungere un certo livello di "mutazione" per ogni popolazione, che produce un individuo mutato ottenuto tramite la variazione random del cromosoma di un individuo precedente, per evitare che si raggiungano punti di stallo nell'algoritmo.



(a) Esempio di Popolazione Iniziale.

(b) Esempio di Prima Generazione ottenuta dalla Popolazione Iniziale.

Figura 2.9: Esempi di Popolazione (da <https://it.mathworks.com/help/gads/how-the-genetic-algorithm-works.html>).

A questo punto, si ripete il procedimento fino a quando si raggiunge una condizione di stop, ad esempio un punto di convergenza, in cui la nuova popolazione non è sostanzialmente differente dalla precedente, oppure dopo aver settato un numero prestabilito di generazioni dopo il quale fermare l'algoritmo.

L'Algoritmo Genetico è disponibile in Matlab nel *Global Optimization Toolbox*, che fornisce diverse funzioni per la soluzione di problemi di ottimizzazione e ricerca di massimi o minimi globali. La funzione Matlab *ga* del Toolbox consente di trovare il set di variabili che minimizza una generica funzione utilizzando il Genetic Algorithm. Poiché la funzione *ga* trova un minimo globale, andremo a passarle non la funzione di fitness definita nella sezione precedente, bensì la differenza tra il valore desiderato (che si ottiene quando  $G \approx 1$ , ossia per ogni evento sismico si rileva un unico picco e con buona accuratezza) e il valore reale della funzione di fitness. In questa maniera l'uscita di *ga* sarà il set di variabili che ci fa avvicinare il più possibile al comportamento

desiderato. È possibile utilizzare la funzione *gaoptimset* per impostare i valori dei seguenti parametri dell'algoritmo:

- *Crossover Fraction*: frazione della popolazione che viene usata per creare figli di tipo 'crossover', esclusi i figli elite. È stata impostata a 0.5;
- *Population Size*: grandezza della popolazione ad ogni iterazione dell'algoritmo, impostata a 100;
- *StallGen*: numero di generazioni oltre la quale l'algoritmo si ferma se non è rilevato un incremento sostanziale nella funzione di fitness rispetto alla generazione precedente. Poiché Vassallo et al. [72] hanno mostrato che l'algoritmo converge abbastanza velocemente, è stato impostato a 50.
- *Generations*: numero di generazioni dopo il quale l'algoritmo si ferma. È stato posto a 150.
- *PlotFcns*: opzioni di plotting dell'algoritmo. Usando l'opzione @gaplotbestf il programma visualizza il valore migliore della funzione di fitness per una data generazione ed il corrispettivo valore medio. Ovviamente, per la buona riuscita dell'algoritmo è bene che i valori convergano dopo un certo numero di generazioni.

La funzione *ga* prende in ingresso la funzione di fitness definita nella sezione precedente, il numero di parametri da usare nell'ottimizzazione (riportati in Tabella 3.4) e che costituiranno il "cromosoma" dei membri della popolazione, i vettori dei limiti superiori e dei limiti inferiori di questi ultimi, e consente anche di specificare se e quali parametri possono assumere solo valori interi. La funzione di fitness è stata implementata in Matlab per lavorare con numeri interi (corrispondenti agli indici dei vettori dei valori possibili dei parametri del Filter Picker), per cui si impone che tutti i parametri assumano valore intero. Infine, si deve fornire un dataset su cui calcolare la funzione di fitness.

## 2.4.2 Dataset Utilizzato

Per procedure con l'ottimizzazione si sono utilizzate dati accelerometrici non processati ottenuti dalle stazioni della Rete Sismica Nazionale (RSN) italiana, utilizzando diverse forme d'onda per eventi che hanno interessato la regione Marche nel periodo 2016-2018. I dati sono stati presi dall'Engineering Strong Motion Database (ESM) [7], il quale mette a disposizione dati accelerometrici sia processati che non processati. Nel nostro caso, sono stati utilizzati dati non pre-processati (a voler simulare il funzionamento del dispositivo Edge) su cui si è applicato l'algoritmo Filter Picker (che funziona anche su segnali broadband non pre-filtrati).

ESMI consente di filtrare le forme d'onda nel suo database mediante parametri quali magnitudo, location, ID della Stazione che ha rilevato l'evento etc. Nel formare

il dataset, si sono considerati solamente eventi che sono avvenuti nella Regione Marche, restringendo la ricerca ad un certo numero di stazioni, e considerando eventi sismici di entità rilevante. In seguito sono riportati i dati per ognuna delle stazioni scelte, indicando i range per alcuni parametri che caratterizzano le forme d'onda rilevate da ognuna stazione. In totale sono state utilizzate 70 forme d'onda.

ID Stazione	Range Magnitudo [ $M_L$ ]	Range Distanze dall'Epicentro [ $km$ ]	Range PGA [ $cm/s^2$ ]
GUMA	3.8-5.9	19.6-7.1	182.4-24.7
FEMA	3.8-5.4	0.6-50.2	1.895-239
MDAR	3.8-5.9	15.8-52.3	0.813- 124.854
SEF1	3.9-5.4	11.8-38.7	2.703-44.279

Tabella 2.4: Stazioni utilizzate nel formare il dataset per l'ottimizzazione.

### 2.4.3 Risultati dell'Ottimizzazione

Lo scopo dell'ottimizzazione è quello di far scendere il numero di "missed detection" a 0, così da rendere il dispositivo capace di rilevare tutti gli eventi sismici del dataset, e allo stesso tempo mantenere un basso numero di picking per evento. Ovviamente avere una situazione che massimizza le rivelazioni corrette porterà ad un abbassamento delle soglie di triggering and di picking, che di conseguenza aumenterà il picking rispetto ad un algoritmo con soglie alte, e viceversa. Poiché siamo interessati a realizzare un dispositivo che sia in grado di rilevare eventi sismici il meglio possibile, si è disposti ad accettare un aumento del picking per assicurare la corretta rivelazione.

Le soglie che l'algoritmo di ottimizzazione ha testato sono quindi nell'intervallo tra 5 e 8.7, avendo scelto la soglia massima come quella che consente di rilevare l'evento in Figura 2.8, per cui si era già vista l'impossibilità dell'algoritmo standard di rilevare il pick. Le soglie per gli altri parametri sono state scelte tra il valore standard e il valore massimo che è stato ottenuto per  $T_{long}$  in [72], ossia 12 s. L'algoritmo converge dopo poco più di 50 generazioni (la velocità di convergenza per il Filter Picker è in linea con quanto trovato da Vassalo et al.) e si sono ottenuti i seguenti valori ottimizzati per i parametri dell'algoritmo.

Parametro	Valore Standard	Valore Ottimizzato
$T_{long}$	$500\Delta T$	2.5 s
$T_{filter}$	$300\Delta T$	4.2 s
$T_{UP}$	$20\Delta T$	0.1 s
$S1$	10	7.9
$S2$	10	8

Tabella 2.5: Parametri dell'Algoritmo ottimizzato e loro valori standard.

Inoltre si è visto che facendo scendere  $N_{band}$ , ossia il numero di segnali passabanda che sono usati per costruire CF, fino a 7 non comporta variazioni dei risultati dell'algoritmo e quindi si è scelto questo valore, in quanto la velocità di esecuzione dell'algoritmo cresce linearmente con  $N_{band}$ , che quindi è da limitare.

In Figura 3.21 sono riportate delle statistiche per le uscite del Filter Picker rispetto allo stesso dataset, ma con parametri ottimizzati o meno. Si vede come in questo caso si rilevano tutte le forme d'onda, ma il picking incrementa rispetto all'algoritmo con parametri standard, come potevamo aspettarci, anche se in numero non eccessivo rispetto all'algoritmo standard. In Figura 2.11 è riportato la percentuale di eventi che presentano un certo valore  $N_P$  di pick, fissando la soglia di pick massimo a  $N_P = 4$  come in [72]. Si vede come in generale l'algoritmo standard presenti un minor numero di pick anche quando supera la soglia  $N_P$ .

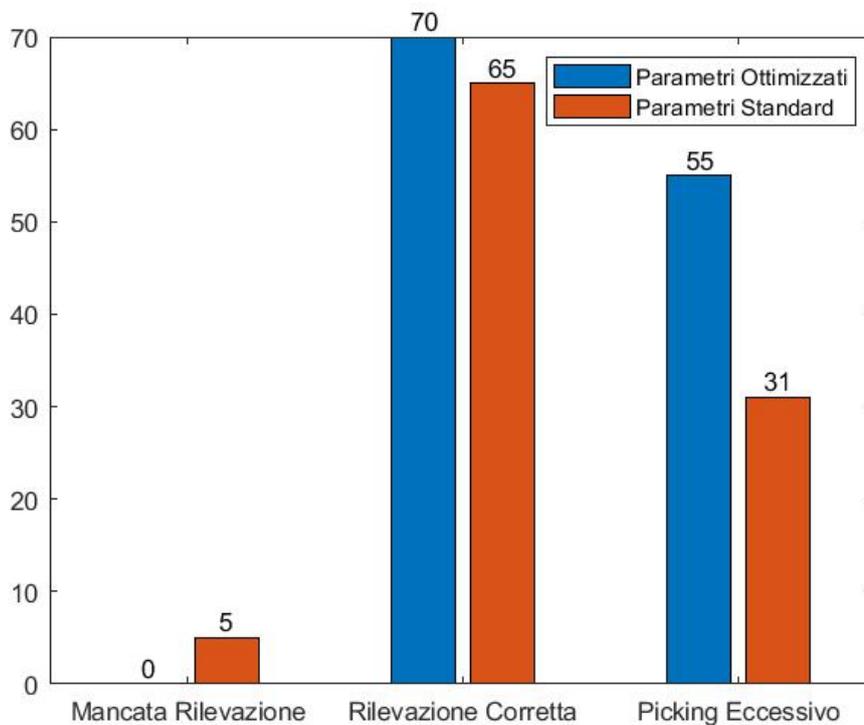


Figura 2.10: Statistiche dell'algoritmo dopo l'ottimizzazione e usando invece i parametri standard.

Si è quindi trovato che l'algoritmo Filter Picker, seppur sia in grado di rilevare eventi sismici con buona precisione, presenta le seguenti criticità:

- Richiede una procedura di ottimizzazione globale basata su una funzione di fitness adeguatamente definita ed un dataset che raccoglie forme d'onda locali al luogo di installazione o, più in generale, della regione nella quale si vogliono rilevare gli eventi sismici;

## 2.4 Testing e Ottimizzazione dell'Algoritmo Filter Picker

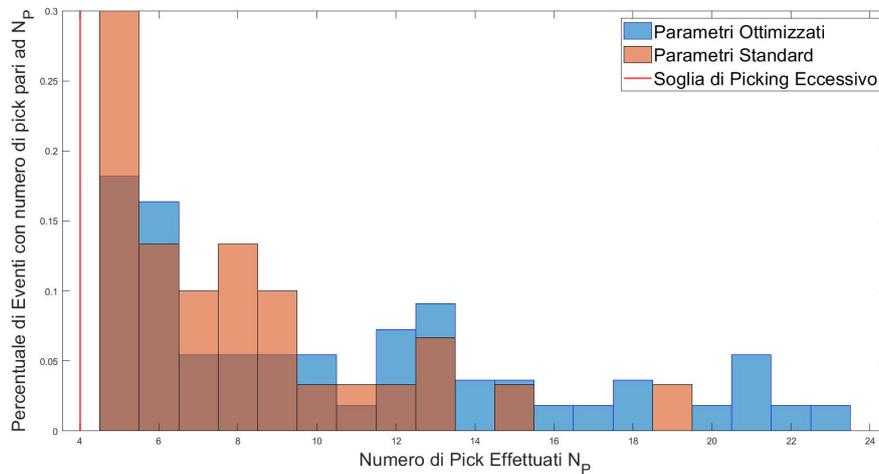


Figura 2.11: Numero di Pick  $N_P$  per eventi che superano la soglia massima ( $N_P = 4$ ) e relative percentuali degli eventi che presentano quel valore di  $N_P$ . Si nota come l'algoritmo ottimizzato, seppur non abbia molti più eventi con picking eccessivo rispetto a quello standard, presenta alcuni eventi con un numero molto alto di pick.

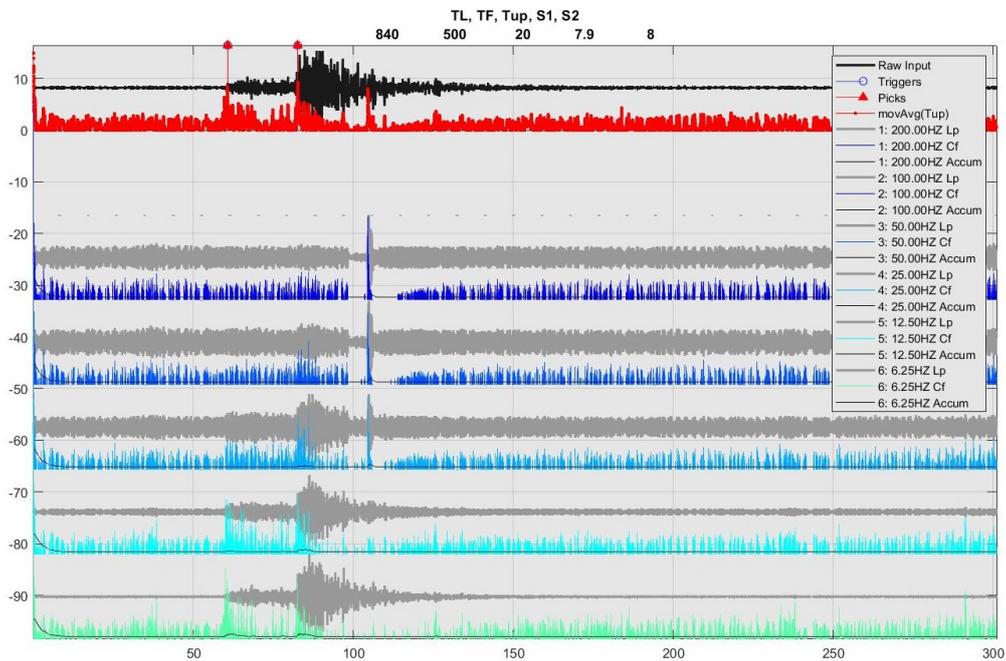


Figura 2.12: Uscita dell'Algoritmo Filter Picker dopo l'ottimizzazione usando in ingresso la stessa forma d'onda riportata in Figura 2.8. In questo caso il picking dell'onda P avviene correttamente.

- C'è un evidente tradeoff tra i valori scelti per le soglie di triggering e picking e il numero di picking effettuato. Un picking eccessivo è in generale da evitare [72]. Nella prospettiva di implementare un sistema su dispositivo embedded che trasmetta allarmi quando l'algoritmo effettua un picking, limitare il picking durante lunghi eventi è essenziale per evitare eccessive trasmissioni. Ulteriori strategie possono essere implementate per impedire che il dispositivo trasmetta troppi allarmi durante il singolo evento, ad esempio mandando il dispositivo in *sleep mode* dopo il primo allarme. Ovviamente è più importante essere in grado di rilevare il verificarsi dell'evento sismico piuttosto che limitare completamente il picking, per cui possiamo considerare i risultati ottenuti buoni nel proposito di implementare un sistema di allerta precoce.

In conclusione, l'algoritmo scelto si dimostra in grado di effettuare un picking dell'onda P una volta ottimizzato e con basse incertezze (dall'1% al 4% secondo la definizione di incertezza data dall'Algoritmo e con le forme d'onda di test usate). Seppur l'ottimizzazione comporti un aumento del numero di pick per forma d'onda, i risultati sono comunque accettabili e, anche grazie alla semplicità dell'algoritmo, risultano essere adeguati all'implementazione su un dispositivo embedded che funga da nodo Edge di un sistema di Early Warning on-site.

## Capitolo 3

# Nodo Edge per Earthquake Early Warning

### 3.1 Introduzione a LoRa e LoRaWAN

LoRa™ (*Long Range*) è uno schema di modulazione radio a spettro espanso sviluppato da Semtech, anche se con il termine LoRa si fa riferimento spesso anche all'insieme di dispositivi che vengono utilizzati per realizzare la modulazione, nonché all'intera rete LoRa per applicazioni di IoT (*Internet of Things*). LoRaWAN™ (*LoRa Wide Area Network*) è invece un protocollo di livello MAC (*Media Access Control*) costruito sul livello fisico costituito dalla modulazione LoRa. Si tratta di una soluzione *Low Power Wide Area Network* (LPWAN), adatta ad applicazioni di IoT e reti di sensori con ampia diffusione sul territorio e con dispositivi che richiedono un basso consumo di alimentazione.

LoRa opera in bande ISM (*Industrial, Scientific and Medical*) non licenziate. In particolare tra 433MHz e 868Mhz in Europa, mentre negli Stati Uniti opera nelle bande a 900 Mhz. In Europa, lo *European Telecommunication Standards Institute* (ETSI) impone le linee guida per l'utilizzo delle bande di comunicazione, per consentire un accesso garantito a tutti gli utenti senza che la risorsa venga occupata per troppo tempo da ogni dispositivo. In particolare, un parametro importante nel caso di protocolli che operano entro bande non licenziate è il duty-cycle, che indica in termini percentuali il periodo per cui un dispositivo LoRaWAN può utilizzare la banda assegnata. In Europa questi valori vanno dallo 0.1% al 10% a seconda della sottobanda utilizzata [73]. Un vantaggio nell'utilizzo delle bande ISM sta nel fatto che sono libere e non necessitano il pagamento di licenze per il loro utilizzo, ma allo stesso tempo presentano più interferenze e hanno data rate limitati [11].

#### 3.1.1 La Modulazione LoRa

LoRa (*Long Range*) è un formato di modulazione basato su *Chirp Spread Spectrum Modulation* (CSSM). Il chirp (*Compressed High Intensity Radar Pulse*) è un segnale la cui frequenza aumenta (up-chirp) o diminuisce (down-chirp) con il tempo e in maniera monotona. Analiticamente, il chirp è espresso come

$$s = \exp(2\pi f_c t + 2\pi \frac{\beta}{2} t^2)$$

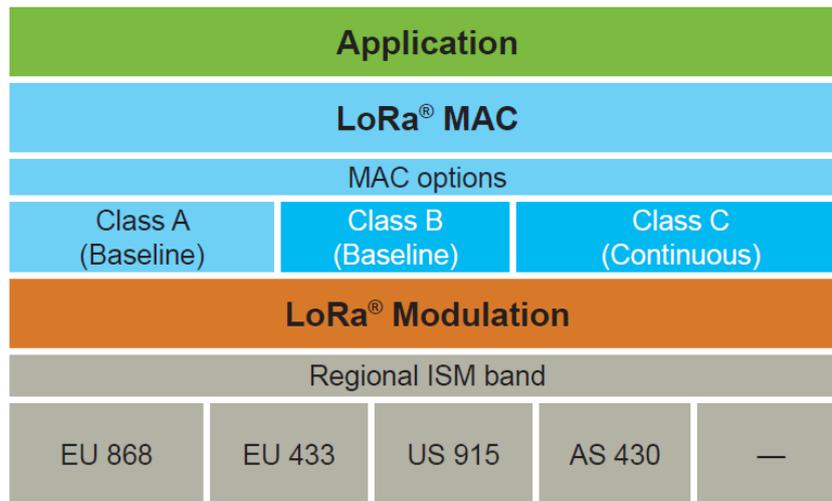


Figura 3.1: Stack Protocollore LoRaWAN [8]. LoRaWAN si trova al secondo livello della pila ISO/OSI. Si basa a livello fisico sullo schema di modulazione LoRA.

in cui  $\beta$  è pari a

$$\beta = BW/T_{symb}$$

con  $T_{symb}$  tempo di simbolo (durata di una sequenza di uno o più bit) e  $BW$  larghezza di banda.  $\beta$  rappresenta quindi la pendenza della caratteristica tempo/frequenza, e indica la velocità di variazione della frequenza del chirp (nel caso di chirp lineare, anche se possono esserci anche chirp la cui frequenza ha andamento diverso).

Inoltre, legati al tempo di simbolo sono lo Spreading Factor  $SF$  e il data rate<sup>1</sup>  $CR$  che, come vedremo in seguito, assumono un ruolo importante nella modulazione CSSM. Vale infatti

$$T_{symb} = \frac{2^{SF}}{BW} CR$$

Una modulazione a spettro espanso è un tipo di modulazione in cui la banda di ogni simbolo trasmesso è allargata, con lo scopo principale di limitare l'effetto di rumori e interferenze, nonché di consentire a più utenti o canali di utilizzare la stessa banda contemporaneamente, mantenendo una comunicazione affidabile. L'espansione di banda può essere realizzata in diversi modi, come ad esempio con *Frequency-Hopping Spread Spectrum* (FHSS), *Direct-Sequence Spread Spectrum* (DSSS), *Time-Hopping Spread Spectrum* (THSS). Le prime due tecniche sono realizzate tramite sequenze pseudo-random e possono essere utilizzate per implementare tecniche di accesso multiplo a divisione di codice (CDMA o *Code Division Multiple Access*). Nel caso di LoRa, viene invece utilizzata una modulazione CSSM, per cui ogni simbolo viene modulato attraverso un chirp e non è richiesto l'utilizzo di generatori pseudorandom. Questa tecnica consente infatti di generare simboli fissando lo slope del chirp (dato

<sup>1</sup>Fattore che tiene conto dell'informazione ridondante trasmessa in ogni simbolo. Se  $CR = \frac{k}{n}$  allora, su  $n$  simboli totali,  $n - k$  sono ridondanti e  $k$  informativi.

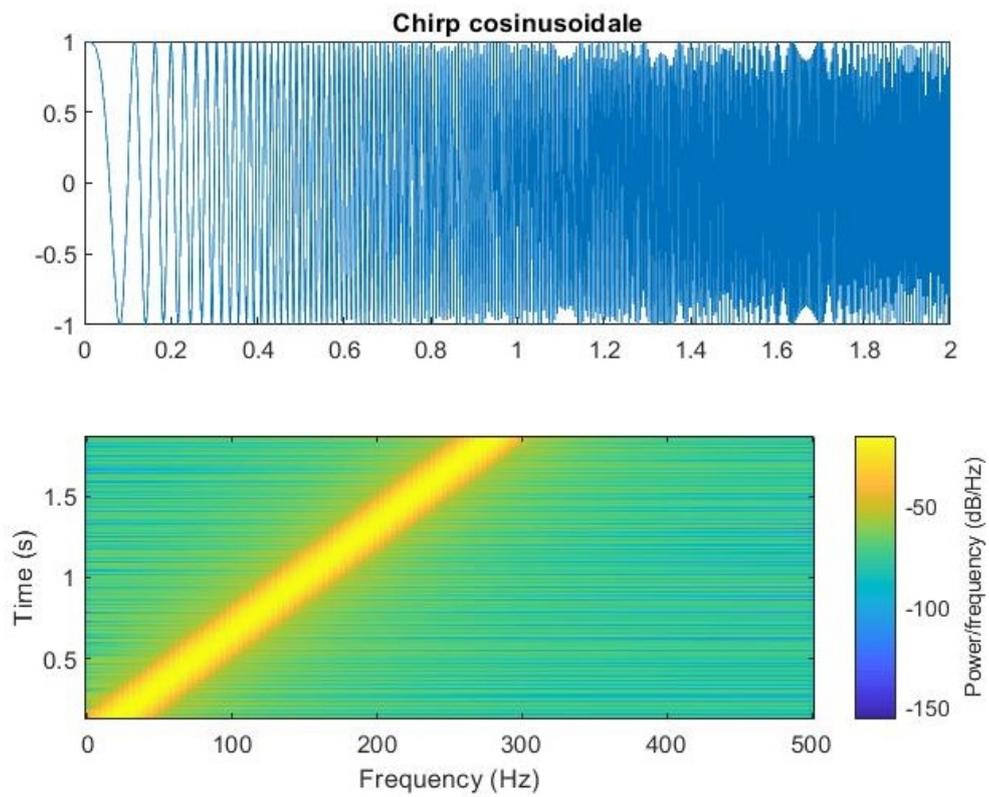


Figura 3.2: Andamento nel tempo (sopra) e spettrogramma (sotto) di un up-chirp cosinusoidale con frequenza che aumenta linearmente. Tramite lo spettrogramma, possiamo vederlo come una rampa che va da  $f_{low}$  a  $f_{high}$ , o viceversa per i down-chirp, in un tempo  $t_{chirp}$ .

dal valore di  $\beta$ ), lo  $SF$  e la banda a disposizione, la quale viene completamente occupata da ogni simbolo trasmesso. Per distinguere un simbolo dall'altro viene applicato uno shift circolare ai singoli chirp che vengono inviati sul canale, così che ognuno possa identificare univocamente un valore nel range a disposizione. Lo shift  $s$  da applicare a un chirp rappresentante un simbolo di valore  $v$  è pari a

$$s = \frac{v}{2SF} T_{symp}$$

Lo  $SF$  consente di identificare effettivamente il range di valori numerici che è possibile codificare, in quanto ogni simbolo può contenere un numero di bit pari proprio a  $SF$  [74]. Di conseguenza, maggiore lo  $SF$ , maggiore è il numero di bit che possiamo trasmettere con un singolo simbolo LoRa. Ad esempio, con uno  $SF$  pari a 7, è possibile trasmettere 7 bit in un  $T_{symp}$  e codificare quindi 128 valori, fissata la larghezza di banda  $BW$ . Come già detto, per distinguere i valori è necessario shiftare gli up-chirp l'uno rispetto all'altro. Per aumentare il range di simboli che è possibile trasmettere, possono anche essere utilizzati dei down-chirp insieme agli up-chirp, nonché una loro combinazione. In questa maniera, ad esempio, con  $SF=7$  possiamo trasmettere 128 valori tramite up-chirp e 128 valori tramite down-chirp.

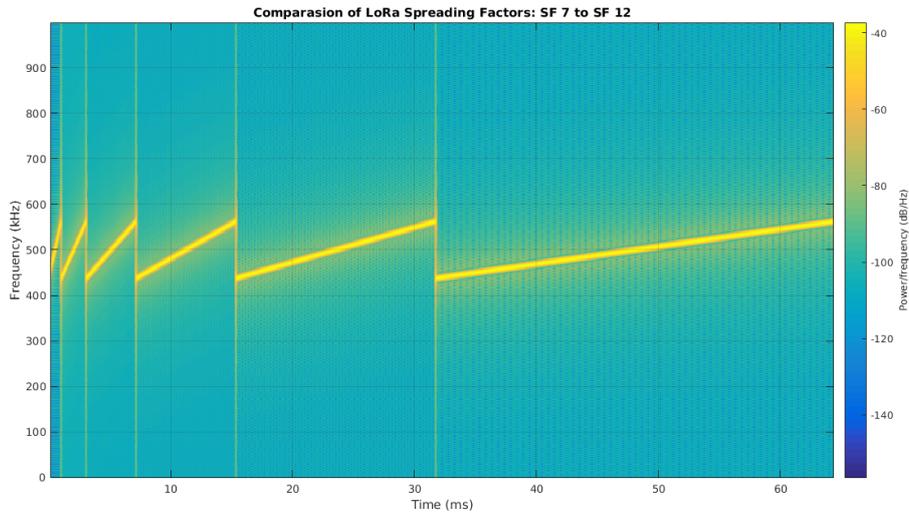


Figura 3.3: Confronto di chirp con differenti Spreading Factor [9]. All'aumentare di  $SF$ , aumenta la durata del singolo chirp e quindi il numero di bit che possono essere trasmessi in un tempo di simbolo. Al contempo, il tempo di simbolo risulta maggiore (rate diminuiscono).

Dalle considerazioni precedenti risulta inoltre che al variare di  $SF$  cambia il numero di bit trasmessi al secondo  $Rb$  che, trascurando il Code Rate, risulta essere pari a

$$Rb = R_s SF = SF \frac{1}{T_{symp}} = SF \frac{BW}{2SF} \quad (3.1)$$

Un aumento unitario dello SF dimezza approssimativamente la velocità di trasmissione, e quindi fa sì che aumenti il tempo di trasmissione (o *Time on Air* TOA) dell'intero pacchetto informativo. Di solito un nodo LoRa utilizzerà uno SF più alto quando il segnale è debole o ci sono molte interferenze o anche quando la distanza da coprire è elevata (aumenta il TOA e con esso il segnale utile che arriva al ricevitore), a discapito però della velocità di trasmissione (fissata ovviamente la larghezza di banda). Di seguito una rassegna dei valori di SF, BW e CR che possono essere utilizzati in LoRa [11].

BW	125, 250 o 500 kHz
SF	7-12
CR	1-4

La possibilità di utilizzare diversi SF e la modulazione a spettro espanso consentono a LoRa di raggiungere lunghe distanze con un alto link budget, permettendo di coprire anche zone molto ampie. Inoltre, segnali con diversi SF possono essere trasmessi sulla stessa banda contemporaneamente poiché risultano ortogonali fra loro (e quindi non interferenti). Nella successiva tabella vengono indicati i valori del TOA e della sensibilità richiesta dal lato ricevente al variare dello SF. Da notare come SF più alti, oltre ad aumentare il Time on Air, consentono di avere una sensibilità minore dal lato del ricevitore [10]. Di contro, c'è una diminuzione della velocità di trasmissione all'aumentare dello Spreading Factor.

Spreading factor	Receiver sensitivity for bandwidth fixed at 125 kHz
SF7	-123 dBm
SF8	-126 dBm
SF9	-129 dBm
SF10	-132 dBm
SF11	-134.5 dBm
SF12	-137 dBm

Figura 3.4: Sensibilità ottenuta al ricevitore con diversi SF fissata una banda fissa di 125 kHz [10].

Le considerazioni precedenti sono state fatte fissando il valore della larghezza di banda. Tenendo SF costante, invece è possibile vedere come siano richieste larghezze di banda sempre più elevate per raggiungere alte velocità.

Una trama LoRa a livello fisico è costituita da un preambolo, un header (opzionale), un payload e un codice a ridondanza ciclica per la correzione di errori (opzionale), come riportato nella Figura seguente. Lo SF determina la lunghezza della trama ed

è possibile applicare della ridondanza per la rivelazione di errori, determinata dal valore di CR scelto.

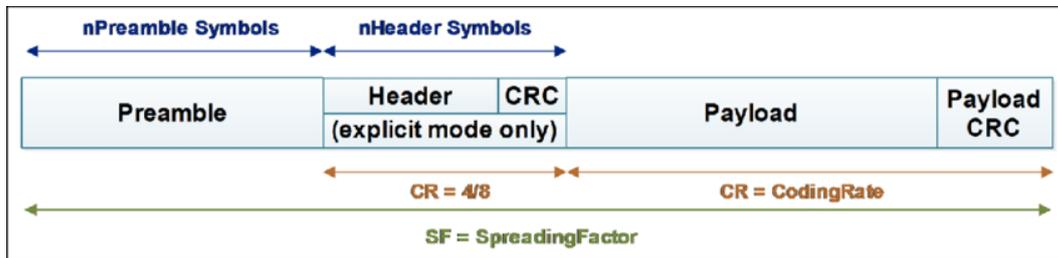


Figura 3.5: Trama LoRa a Livello Fisico [8].

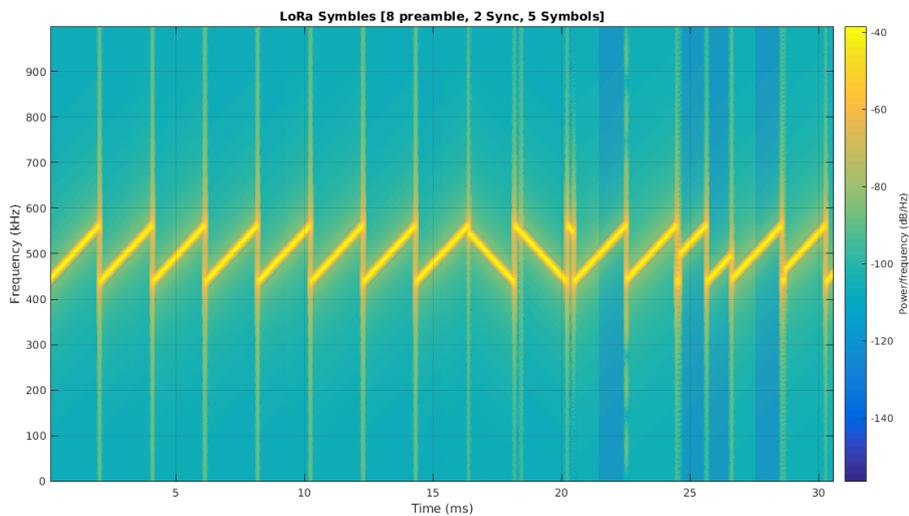


Figura 3.6: Sequenza di simboli LoRa [9]. Si individua chiaramente l'inizio di una trama per il susseguirsi di 6 up-chirp e 2 down-chirp (preambolo).

Il preambolo contiene 6 simboli trasmessi usando come modulante un up-chirp, e 2 simboli trasmessi con un down-chirp. Oltre a funzioni di sincronizzazione, il preambolo determina l'SF con cui viene trasmesso l'intero pacchetto. L'header è opzionale e, se presente, è trasmesso con un CR pari a 4/8 (quattro simboli di ridondanza ogni quattro simboli informativi), e contiene informazioni sulla lunghezza totale del pacchetto in byte, il CR utilizzato per il payload e la presenza o meno alla fine della trama del CRC (*Cyclic Redundancy Check*) a 16 bit, che consente di rilevare errori sul payload. Anche l'header può presentare un controllo con simboli di ridondanza.

### 3.1.2 Topologia di una rete LoRaWAN

Come già detto, mentre LoRa rappresenta uno schema di modulazione a spettro espanso, LoRaWAN è invece il protocollo a livello MAC e Applicativo che viene

costruito su quello schema di modulazione e che determina l'architettura di rete del sistema.

Molte reti esistenti utilizzano una topologia di tipo "mesh". In una rete mesh, ogni nodo di rete è connesso a quanti più nodi possibili in maniera dinamica e non gerarchica. In questa maniera tutti i nodi contribuiscono al trasferimento dell'informazione e la rete può adattare dinamicamente le proprie strategie di routing. I nodi terminali di una rete mesh inoltrano i messaggi trasmessi da tutti i nodi a cui sono connessi. Poiché più nodi contribuiscono al trasferimento dell'informazione, viene aumentato il range di comunicazione e inoltre si possiede una migliore *fault tolerance* in caso di fail di uno o più nodi. Di contro, vengono aumentate di molto la complessità della rete e il carico su di essa, e inoltre la durata della batteria dei singoli nodi è molto diminuita poiché ognuno si fa carico di trasmettere anche l'informazione dei nodi a cui è connesso.

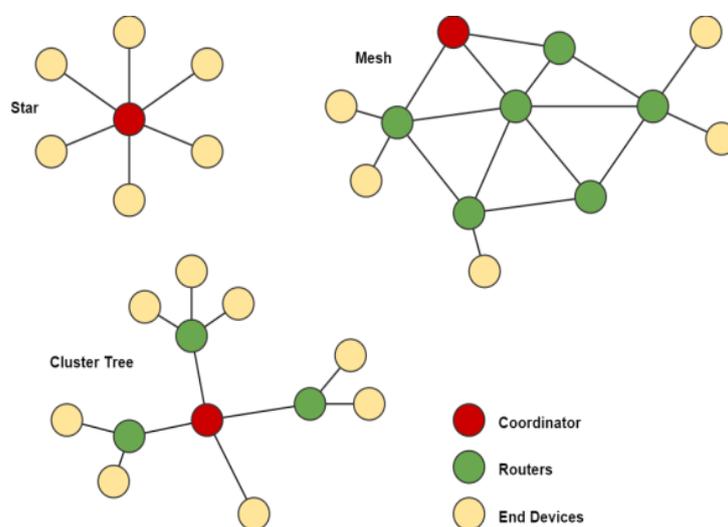


Figura 3.7: Esempi di topologia di rete (da <https://datarespons.com/industrial-or-commercial-wireless-mesh-technologies>). Una rete LoRa può essere vista come una rete 'a stella di stelle' in cui i coordinatori sono i gateway, che comunicano verso gli End Device e il Network Server.

In una rete LoRaWAN i nodi non sono connessi ad un singolo gateway, ma in genere più gateway si fanno carico dell'informazione trasmessa da un nodo terminale. Ogni gateway trasferisce poi l'informazione tramite una rete di backhaul (che può essere una rete cellulare, Wi-Fi, Ethernet etc ed è in genere basata su IP.) verso il server di rete, dove sono localizzate tutta l'intelligenza e la complessità del sistema, piuttosto che sulla rete stessa come in una topologia mesh. Il server di rete si occupa quindi di sicurezza, invio di *acknowledgments* verso i nodi trasmettenti, gestione dinamica dei data rate e così via. Possiamo vedere la rete LoRa quindi come una rete 'a stella di stelle' *long range*, in cui ogni gateway riceve connessioni da più nodi come in una rete a stella, ma lo stesso nodo è connesso a più gateway.

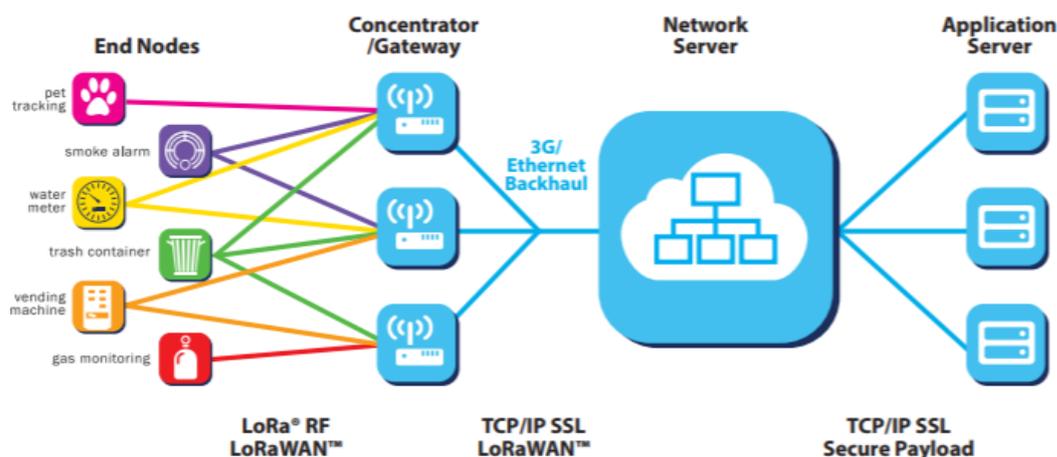


Figura 3.8: Topologia di una rete LoRaWAN [8]. Ogni gateway riceve connessioni da più nodi come in una rete a stella, ma uno stesso nodo è connesso a più gateway.

I nodi in una rete LoRaWAN sono asincroni e comunicano con i gateway solamente quando hanno dell'informazione da trasmettere (metodo *Aloha*<sup>2</sup>), mentre in una rete mesh i nodi devono sincronizzarsi periodicamente con la rete e controllare i messaggi ricevuti. L'approccio Aloha consente ai nodi LoRa di diminuire di molto il consumo di batteria, in quanto eliminano l'esigenza di effettuare frequenti risincronizzazioni, consentendo di ottenere performance da 3 fino a 5 volte migliori rispetto ad altre soluzioni IoT per quel che riguarda il consumo energetico. Il problema dell'utilizzo dello schema Aloha può essere però quello delle *collisioni*, che possono degradare fortemente la qualità del canale [76].

Poiché i gateway sono dispositivi che devono gestire diversi nodi, esattamente come un hub o switch in una rete a stella, è necessario che abbiano un'ampia capacità. Per questo motivo i gateway usano dei transceiver multicanale e multi-modem che consentono a messaggi multipli su diversi canali di essere ricevuti simultaneamente. I gateway in commercio possono supportare dagli 8 ai 64 canali di comunicazione e inoltre sono in grado di gestire diversi data rate sui diversi canali contemporaneamente. Ciò consente di aumentare la capacità della rete e di ridistribuire le risorse in maniera intelligente, nonché di aumentare la durata della batteria di un nodo consentendogli di variare dinamicamente le proprie velocità di trasmissione. Il gateway riesce a ricevere segnali su differenti canali e con differenti data rate sfruttando il fatto che la modulazione LoRa è una modulazione a spettro espanso, in cui quindi segnali con differenti SF sono effettivamente ortogonali fra loro. Come già spiegato, cambiare

<sup>2</sup>Semplice schema di comunicazione sviluppato dalla *University of Hawaii* per gestire comunicazioni satellitari e adattato a comunicazioni cablate e wireless in genere. Si tratta di uno schema piuttosto semplice in cui ogni trasmettitore della rete invia dati solo quando ne ha necessità. In questo schema, maggiore il numero di trasmettitori maggiore è il rischio di collisioni, per cui si sono sviluppati metodi come lo *Slotted Aloha* per evitare situazioni di collisione. L'uso di protocolli più sofisticati come il *Carrier Sense Multiple Access with Collision Detection* (CSMA/CD) consente di migliorare ulteriormente la performance [75]

lo SF corrisponde a cambiare la velocità effettiva di trasmissione del segnale e di mantenere i messaggi trasmessi dai diversi nodi separati tra loro. Inoltre, a seconda della vicinanza del nodo al gateway o del livello di interferenze sul canale, è possibile per un nodo modulare di conseguenza il proprio SF, consentendo un adattamento dinamico alle condizioni della rete e del mezzo trasmissivo.

### 3.1.3 Tipologia di Nodo LoRaWAN

In una rete LoRaWAN non tutti i nodi sono trattati allo stesso modo. Il protocollo specifica 3 categorie di nodi, che vengono divisi nelle seguenti classi:

- **Classe A** o *Battery Powered Sensors*, supportata da tutti i dispositivi e la più energeticamente efficiente. La ricezione in downlink è possibile solo dopo una trasmissione in uplink;

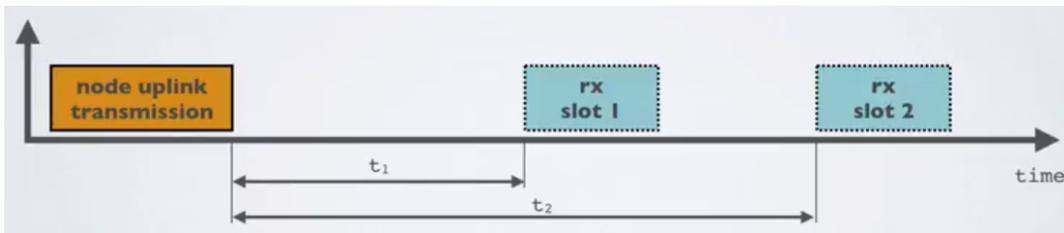


Figura 3.9: Funzionamento di un Device di Classe A [11].

- **Classe B** o *Battery Powered Actuators*, con una buona performance energetica e comunicazione verso il dispositivo che viene sincronizzata periodicamente tramite beacon;

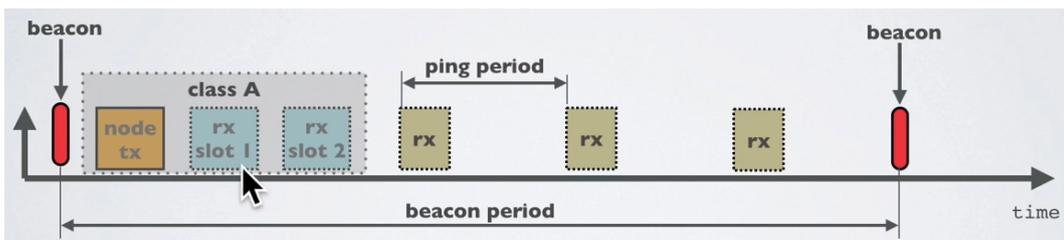


Figura 3.10: Funzionamento di un Device di Classe B [11].

- **Classe C** o *Main Powered Actuators*, possono permettersi un funzionamento continuo e non hanno bisogno di periodi di ritardo per la separazione degli slot in downlink.

La Classe A consente una trasmissione bidirezionale ed è interamente basata su Aloha, per cui ogni trasmissione è pianificata direttamente dal dispositivo. Ciò consente di aumentare la durata della batteria, anche se trasmissioni dalla rete o da un gateway verso il nodo sono vincolate in quanto possibili solo dopo che un

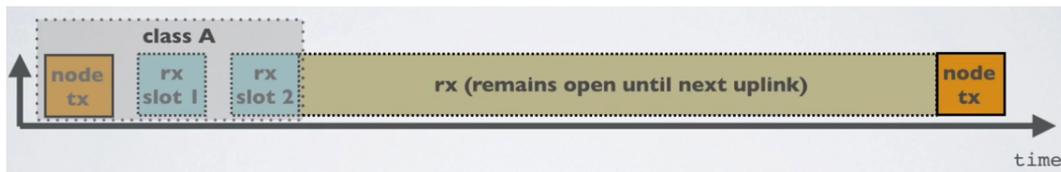


Figura 3.11: Funzionamento di un Device di Classe C [11].

nodo ha trasmesso. In particolare, dopo ogni trasmissione il dispositivo apre due finestre di ricezione entro le quali il server di rete può inviargli informazioni. Se non utilizza nessuna delle due finestre, la rete dovrà attendere che il dispositivo trasmetta nuovamente per comunicarci.

I dispositivi di Classe B sono bidirezionali e hanno una finestra di ricezione fissa che viene attivata tramite un segnale di beacon inviato dal gateway, così che il server di rete possa sapere quando il dispositivo lo sta effettivamente ascoltando.

La Classe C invece consente una trasmissione bidirezionale continua, in cui il dispositivo è sempre in ascolto della rete. Si tratta quindi di dispositivi che non hanno particolari esigenze dal punto di vista del risparmio di batteria.

Impostare un dispositivo come un End Node di una delle tre classi precedenti è molto importante, in quanto determina la durata di vita della batteria. Di solito LoRaWAN è pensato per lo più per le comunicazioni in uplink verso il Network Server, e le comunicazioni in downlink sono di solito limitate anche da eventuali *Fair Policy* del Network Server a cui ci si appoggia [77].

## 3.2 Valutazione della Connessione LoRa

Nella sezione seguente si valuteranno sperimentalmente alcune caratteristiche della connessione LoRa, quali la sua copertura, e si farà una stima del throughput e del consumo che è possibile ottenere da un device connesso alla rete LoRa, anche in base alle limitazioni imposte dal protocollo stesso e dalla dimensione dei pacchetti trasmessi. Le prove sono state effettuate in ambiente indoor, con un gateway posizionato all'interno di un appartamento nel centro di Ancona e valutando il livello di segnale dopo l'installazione del dispositivo in diversi punti dell'edificio. Lo scopo della prova è quello di valutare le potenzialità del dispositivo utilizzato e poter scegliere i parametri adatti per la connessione LoRa.

### 3.2.1 Hardware e Servizio Cloud Utilizzati

L'hardware utilizzato nelle prove che verranno descritte nelle sezioni successive è il seguente:

- B-L072Z-LRWAN1 LoRa®Discovery di STM (board di sviluppo con antenna LoRa)

- Sentrus RG1xx LoRa-Enabled Gateway

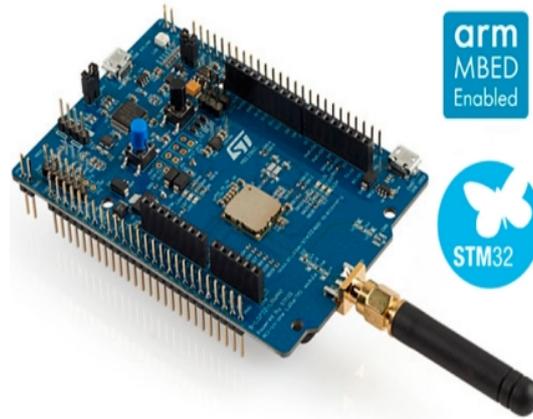


Figura 3.12: B-L072Z-LRWAN1 LoRa®Discovery Kit. [12]



Figura 3.13: Sentrus RG1xx LoRa-Enabled Gateway. [13]

Per il test della connessione è stato utilizzato il programma *End Node* per il discovery kit. Si tratta di un programma che trasmette dati di sensori fittizi utilizzando il formato di trasmissione Cayenne LPP. I programmi del Discovery Kit utilizzano la versione dello Stack LoRa di STM Microelectronics, ottimizzata per lavorare con i loro dispositivi. Lo stack è disponibile per tutta la serie di dispositivi con MCU *STM32Lxx*.

Come spiegato precedentemente, un'applicazione LoRa si appoggia su un Network Server che gestisce gateway, dispositivi e rende disponibili i dati provenienti dagli End Node a livello Applicazione. Per poter analizzare o gestire i dati provenienti da un End Node è necessario quindi appoggiarsi ad un servizio che funga da Network Server. Nel nostro caso, si è utilizzato The Things Network (TTN). Lo stack di The Things Network è organizzato come nella figura seguente.

Tramite la propria Console, ogni user è in grado di creare delle Applicazioni, ossia contenitori di End Device che possono essere usate per monitorare il flusso di dati,

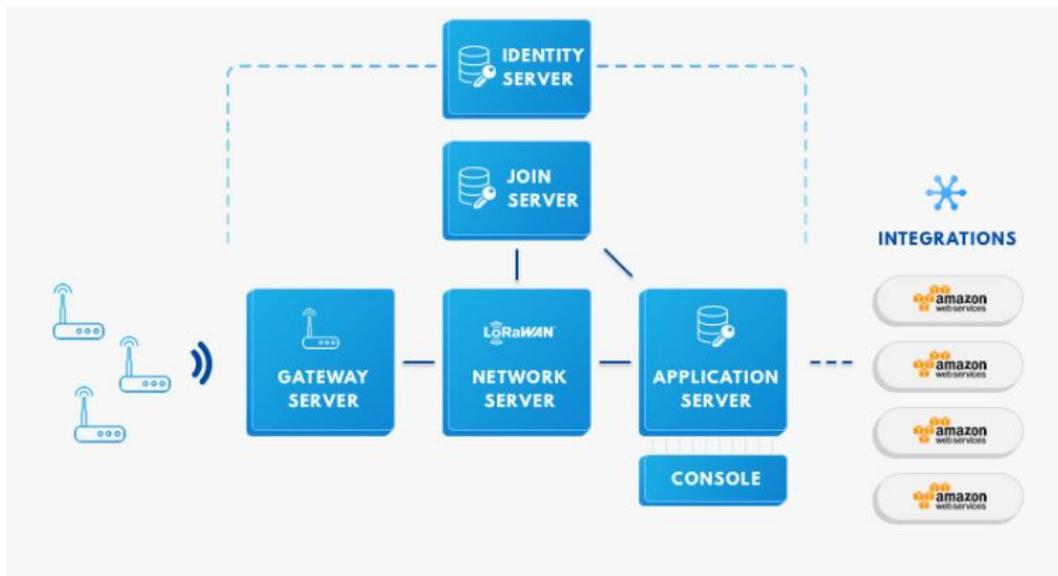


Figura 3.14: Stack di TTN [14]

decodificare i payload dei pacchetti ricevuti o, assieme alle Integrazioni fornite da TTN, per eseguire diverse operazioni e integrare i dati ricevuti con vari servizi come AWS (Amazon Web Services). Tramite TTN è anche possibile monitorare i gateway registrati e vedere i pacchetti ricevuti e inviati verso i singoli End Device.

Un End Device può essere creato all'interno di un'Applicazione, che fornisce i seguenti identificativi per il dispositivo:

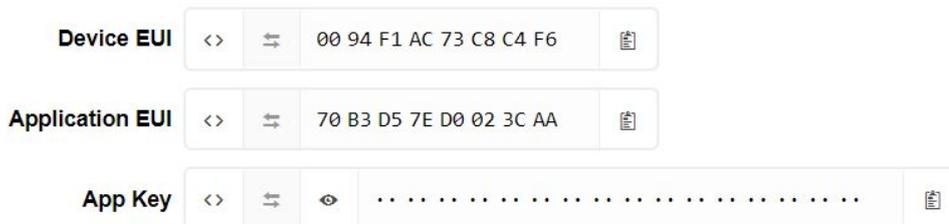


Figura 3.15: Identificativi di un Device LoRaWAN sulla Console TTN.

Per poter far sì che il nostro dispositivo risulti registrato su TTN, è necessario che DEVICE EUI, Application EUI e App Key che sono generati da TTN siano replicati nel codice dell'applicazione che carichiamo sulla scheda o sul tracker. Una volta che gli identificativi del dispositivo e quelli su TTN sono gli stessi, è possibile visualizzare i pacchetti che arrivano tramite Console e vederne il payload, nonché applicare le diverse integrazioni fornite da TTN ai dati ricevuti.

Oltre agli identificativi, si possono settare duty cycle dell'applicazione End Node, i data rate ed accendere o meno l'ADR. Si faccia riferimento alla tabella in Figura 3.16 per i valori di Data Rate e i corrispondenti SF.

DataRate	Modulation	SF	BW	bit/s
0	LoRa	12	125	250
1	LoRa	11	125	440
2	LoRa	10	125	980
3	LoRa	9	125	1'760
4	LoRa	8	125	3'125
5	LoRa	7	125	5'470
6	LoRa	7	250	11'000
7	FSK 50 kbps			50'000

Figura 3.16: Valori dei Data Rate e corrispondenti SF.

### 3.2.2 Test di Copertura

Il test di copertura indoor è stato realizzato utilizzando il dispositivo e The Things Network per valutare l'RSSI (*Received Signal Strength Indicator*) del segnale ricevuto dal gateway. L'RSSI è un indicatore del livello di potenza ricevuto dal dispositivo dopo le perdite relative all'antenna e al cavo, e quindi ci fornisce un'indicazione della qualità relativa del segnale utile. L'RSSI viene misurato in dBm ed è tanto più buono quanto più è vicino a 0, mentre si considera che la connessione viene a mancare quando si va sotto a -120 dBm [11]. LoRa è in grado di demodulare anche con livello di segnale molto al di sotto del noise floor (fino a 6 db), quindi livelli di RSSI al di sopra dei -100 dBm possono essere considerati come un buon livello di segnale.

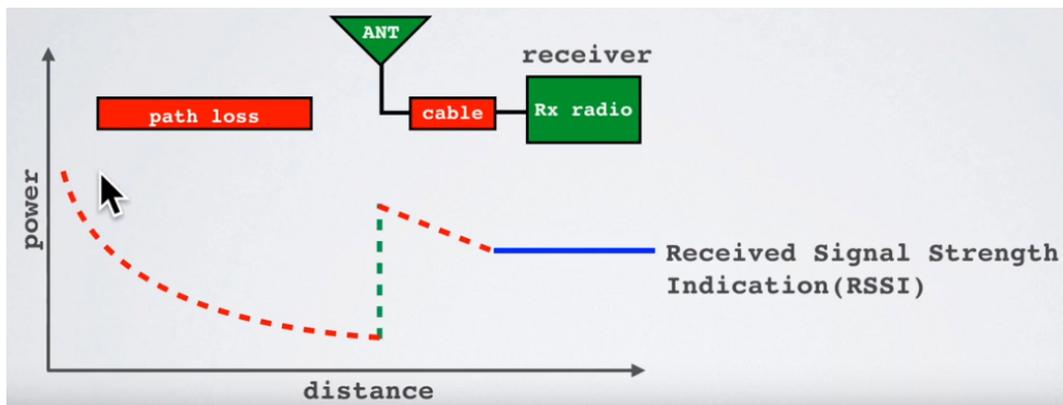


Figura 3.17: Rappresentazione grafica dell'RSSI [11].

Per l'esperimento, il dispositivo è stato posto in 4 posizioni diverse per valutare la copertura sotto diverse condizioni di propagazione, sia in un ambiente indoor che in ambiente seminterrato, dove la scheda potrebbe essere posta per certe applicazioni pratiche come smart metering, monitoraggio strutturale (a cui può appoggiarsi un sistema di allerta precoce) o, appunto Early Warning. Le posizioni scelte sono state le seguenti:

### Capitolo 3 Nodo Edge per Earthquake Early Warning

- Sensore posto accanto al gateway (per valutare il livello di segnale massimo che è possibile ricevere);
- Sensore posto a 4 m dal gateway e in linea di vista (*line of sight* o LOS) dal gateway;
- Sensore posto a 10 m dal gateway e non in linea di vista (*non-line of sight* o NLSO), con pareti e porte che lo separano dal gateway;
- Sensore posizionato in un locale seminterrato 4 piani al di sotto della posizione del gateway.

Come già detto, il dispositivo trasmetterà dati di sensore fittizi e l'applicazione consente di cambiare lo Spreading Factor, scegliendo un valore di data rate dalla tabella di conversione riportata in Figura 3.16. I due Spreading Factor che sono stati utilizzati sono SF=8 e SF=12, quindi rispettivamente uno SF più veloce ed uno più lento, per valutare come questi influenzano la comunicazione. La potenza in trasmissione è di 14 dBm (25 mW), e si lavora nella banda EU 868 e con canali da 125 kHz (implementazione standard del protocollo in Europa).

Sono stati osservati più pacchetti in ogni posizione per valutare come il segnale oscilla nel tempo, cosa da aspettarsi in un ambiente indoor a causa di effetti di multipath o eventuale shadowing, e che può portare a momenti di cattiva ricezione. Dai risultati ottenuti si vede che la differenza tra i due SF scelti in ambienti indoor non è molto grande, ma si inizia a notare la differenza quando ci si sposta rispetto al gateway in ambiente seminterrato. In media con SF=12 si mantiene un valore di segnale maggiore, come potevamo aspettarci.

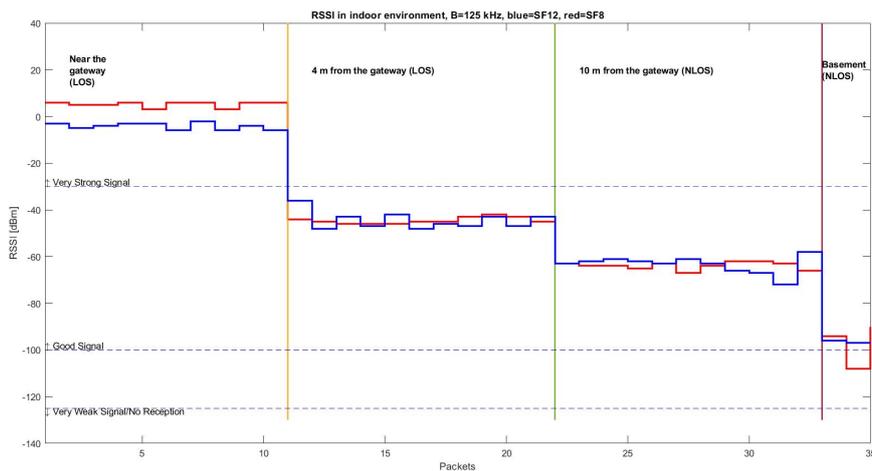


Figura 3.18: Risultati della Prova di Copertura. L'andamento del segnale è stato osservato in 4 punti di interesse, e sono stati misurati più valori di RSSI per ogni zona per valutare l'oscillazione del segnale.

Scenario	SF	$RSSI_{average}$	Qualità del Segnale
Vicino al Gateway	8	5.2 dBm	Segnale molto forte.
	12	-4 dBm	Segnale molto forte.
4 m dal Gateway (LOS)	8	-49 dBm	Buona qualità.
	12	-41 dBm.	Buona Qualità.
10 m dal Gateway (NLOS)	8	-63.9 dBm	Buona qualità.
	12	-63.45 dBm	Buona qualità.
Seminterrato	8	-96.6 dBm	Sporadicamente sotto la soglia di buona qualità.
	12	-97.3 dBm	Sopra la soglia di buona qualità.

Tabella 3.1: Risultati del test per i diversi scenari. La soglia di disconnessione di  $-120dBm$  non viene mai raggiunta per entrambi gli SF.

Dai grafici possiamo infatti vedere come SF=8 dia prestazioni migliori rispetto allo SF più lento (SF=12) quando molto vicino al gateway. Infatti, SF bassi mappano più valori di RSSI rispetto a quelli con SF bassi [78], dando un maggior valore di segnale in buone condizioni di propagazione, anche se entrambi sono largamente sufficienti in questo caso. Inoltre, la probabilità di scendere al di sotto di un certo valore di RSSI cresce con lo SF (maggior vulnerabilità allo shadowing e al multipath), e infatti vediamo come con SF=12 il livello di segnale ricevuto oscilla maggiormente rispetto a SF=8 (le oscillazioni sono dovute alle riflessioni multiple tipiche degli ambienti indoor), anche a causa del numero minore di valori di RSSI a disposizione. Allontanandosi dal gateway, si inizia a notare il vantaggio a livello di copertura per gli Spreading Factor più lenti. Infatti, per SF che tendono a valori più alti il range di copertura aumenta notevolmente, anche grazie all'aumento del time on air, quindi al costo di una velocità di trasmissione minore. In entrambi i casi si rimane con buoni livelli di segnale: SF scende momentaneamente sotto la soglia di -100 dBm per SF=8 in ambiente seminterrato, ma mai al di sotto della soglia di -120 dBm (soglia al di sotto della quale c'è disconnessione). Possiamo quindi dire che la modulazione LoRa è in grado di fornire una qualità di segnale in grado di coprire un intero ambiente indoor e con minime variazioni dell'RSSI. Anche quando ci si sposta in posizioni meno 'privilegiate' come un locale seminterrato, si riesce a mantenere la connessione al gateway. Possiamo quindi concludere che nelle condizioni di propagazione indoor in cui si lavora è perfettamente lecito lavorare con SF più veloci senza che la qualità del segnale degradi in maniera visibile, mentre SF alti sono più convenienti su grandi distanze, ma risultano molto più lenti e consumano più potenza, anche se in media danno una qualità del segnale migliore.

### 3.3 Dispositivo LoRaWAN per la trasmissione di allarmi.

In seguito si utilizzerà quanto ottenuto nelle sezioni precedenti per determinare le caratteristiche e scegliere i giusti parametri di trasmissione di un dispositivo che utilizza il protocollo di comunicazione LoRaWAN per la trasmissione di allarmi dell'arrivo di un'onda P, rilevata tramite Filter Picker.

La scelta dei parametri LoRaWAN è essenziale quando si sta progettando un'applicazione che utilizza il protocollo. La scelta dipende da diversi fattori, innanzitutto il contesto applicativo. Nel caso dell'Early Warning vogliamo che l'applicazione sia veloce ed affidabile, così che gli allarmi siano tempestivi e non vengano persi durante la trasmissione. I parametri vanno quindi scelti di conseguenza per assicurare queste caratteristiche e trovare compromessi adeguati per eventuali trade-off (come può essere quello tra range di copertura e velocità di trasmissione).

Alcuni dei parametri che possiamo scegliere per LoRaWAN sono Spreading Factor, frequenza, larghezza di banda e potenza in trasmissione  $P_{TX}$ , ed inoltre il payload scelto per l'applicazione è essenziale nel determinare caratteristiche come il Time on Air e di conseguenza latenza e consumi. Questi ultimi dipendono anche dal modo di operazione del nodo (numero di trasmissioni/interrogazioni giornaliere, link bidirezionali tra gateway ed End node o meno, modalità *Periodic Receiver* o trasmissioni iniziate dall'End node etc.).

Lo Spreading Factor rappresenta lo "sweep rate" del chirp LoRa, ossia la velocità con cui il chirp va da un'estremità all'altra della banda utile, rappresentando quindi l'effettiva durata del singolo chirp e determinando la quantità di informazione che esso contiene. Lo Spreading Factor influenza diversi parametri della comunicazione LoRaWAN, per cui la sua scelta è essenziale per determinare il comportamento e le performance dell'applicazione che stiamo realizzando:

- **Data Rate:** al variare di SF cambia il numero di bit trasmessi al secondo. Dall'Equazione 3.1 si può vedere come un aumento di un'unità di SF comporta approssimativamente un dimezzamento della velocità di trasmissione.
- **Time on Air:** il ToA è il tempo necessario affinché un messaggio arrivi dal trasmettitore al ricevitore, ed aumenta con lo Spreading Factor. SF alti implicano quindi trasmissioni più lunghe;
- **Range di copertura:** il range di copertura dell'applicazione, come visto anche nel test sperimentale fatto nella sezione precedente, aumenta con SF, fornendo coperture a lungo raggio per SF alti;
- **Sensibilità del Ricevitore:** aumenta anch'essa con lo SF, per cui è necessario usare SF alti quando il segnale al ricevitore è debole, ad esempio per ambienti seminterrati;

### 3.3 Dispositivo LoRaWAN per la trasmissione di allarmi.

- **Consumi:** poiché la durata della trasmissione varia con SF, i consumi risultano anch'essi influenzati dalla scelta dello Spreading Factor, e sono maggiori per SF alti.

Il Data Rate può anche essere imposto dalla larghezza di banda scelta. In Europa, è possibile utilizzare una larghezza di bandi pari a 125 kHz per gli SF da 8 a 12, ed una banda di 125 kHz o 250 kHz per uno SF pari a 7 (i corrispondenti valori di Data Rate sono riportati in 3.16). A parità di SF, una banda più larga incrementa il bit rate. Ad esempio, il bit rate nominale per SF=7 e BW=125 kHz è 5.5 kbps, mentre per SF=7 e BW=250 kHz diventa di 10.9 kbps circa.

Nella sezione precedente si è mostrato come con SF=8 e una banda di 125 kHz si riesce ad ottenere una buona prestazione in termini di copertura, senza mai scendere sotto la soglia di disconnessione, ma solo sporadicamente sotto la soglia di buona qualità per posizioni sfavorevoli. Pertanto possiamo utilizzare quanto visto nella prova sperimentale per scegliere la migliore configurazione dei parametri a radiofrequenza. Per la stima del Data Rate si è utilizzato il Tool di Semtech *LoRa Calculator*, che consente di calcolare una buona stima delle caratteristiche della comunicazione LoRa effettuata con i chip della serie SX1276/77/78/79 (la board B-L072Z-LRWAN1 usa il modulo SX1276). Nella Tabella seguente sono riportate le prestazioni per SF=8 ed SF=12.

SF	Data Rate Massimo [kbps]	BW [kHz]
12	0.292	125 kHz
8	3.125	125 kHz

Tabella 3.2: Confronto dei diversi Data Rate che possono essere ottenuti con SF=8 ed SF=12. SF=8 fornisce il data rate migliore.

La latenza è fortemente influenzata dal ToA, che diminuisce usando SF=8 rispetto ad SF=12. Il ToA dipende anche dalla dimensione del payload del messaggio di allarme che stiamo trasmettendo, il quale influenza anche i consumi della board (trasmissioni più lunghe implicano un maggiore spreco di potenza). Anche queste caratteristiche possono essere stimate usando il Tool Semtech impostando adeguatamente i parametri della connessione, come riportato in seguito. Risulta evidente da quanto detto che nel volere trasmettere allarmi tempestivamente è conveniente utilizzare uno SF più basso per limitare la durata della trasmissione.

SF	Time on Air [ms]	Consumo in Trasmissione [uC]
12	761.9	68567
8	57.6	5207.7

Tabella 3.3: Confronto dei diversi ToA e consumi ottenibili con SF=12 ed SF=8. Anche in questo caso SF=8 risulta essere più conveniente. La dimensione del payload è fissata a 8 byte.

Riassumendo, la comunicazione sarà caratterizzata dalla seguente scelta di parametri: SF=8, BW=125 kHz, banda EU868,  $P_{TX}$ =14 dBm, pacchetti da 8 byte.

L'algoritmo è stato implementato sulla board in linguaggio C. Al momento della rilevazione del primo pick dell'onda P, viene salvato un timestamp che viene inserito nel pacchetto da 8 byte che deve poi essere trasmesso tramite LoRaWAN. I parametri LoRaWAN sono facilmente programmabili all'interno dello script End Node fornito insieme alla board, andando a modificare i valori di Data Rate corrispondenti alla scelta dello Spreading Factor, secondo quanto riportato in Tabella 3.16. Si tiene conto nel diagramma di flusso di quanto visto nel Capitolo 2 per quel che riguarda il picking eccessivo, per cui dopo la prima trasmissione il dispositivo viene mandato in Sleep Mode per un tempo tale da far terminare l'evento sismico.

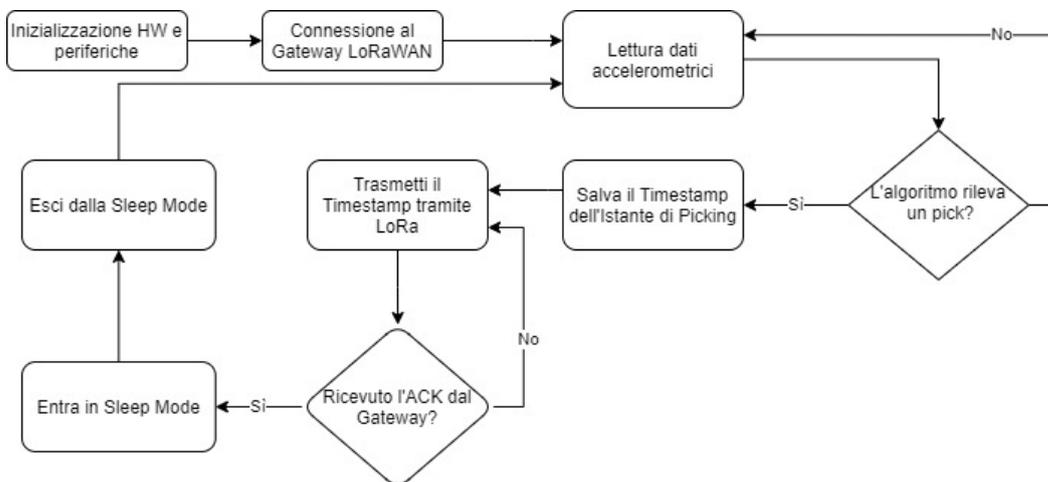


Figura 3.19: Diagramma di Flusso dell'Applicazione per la Trasmissione di Allarmi.

# Risultati

Il lavoro di tesi ha consentito di ottenere risultati riguardanti l'ottimizzazione di un algoritmo di Earthquake Early Warning basato sul "picking" della fase dell'onda P, nonché risultati riguardanti i test della performance del protocollo di comunicazione LoRaWAN, usati poi per la scelta accurata dei parametri da utilizzare per la comunicazione LoRa con dispositivo B-L072Z-LRWAN1 in maniera tale da ottenere risultati che siano conformi alle necessità di un sistema di Early Warning, innanzitutto bassi tempi di trasmissione, bassi consumi e data rate sufficienti. Questi requisiti sono stati definiti a partire da un'iniziale review dello stato dell'arte dei sistemi di EW relativi a tre use cases (*earthquakes, flash floods, tsunamis*). In seguito vengono riassunti in breve i risultati.

## Risultati dell'Ottimizzazione dell'Algoritmo Filter Picker

Come algoritmo di EEW si è scelto l'algoritmo *Filter Picker*, il quale unisce la semplicità dei metodi "energy based" basati sull'algoritmo di Allen e la capacità di lavorare con segnali broadband rumorosi dei metodi multibanda. Prima di eseguire l'algoritmo su dispositivo embedded lo si è però testato su un computer e se ne sono riscontrate le principali criticità:

- impossibilità di rilevare l'onda P per certe forme d'onda;
- eccesso di "picking" effettuati su altre, specie per eventi lunghi.

Ciò ha richiesto di implementare una procedura di ottimizzazione basata sull'Algoritmo "Genetico" per consentire il "fine tuning" dei parametri dell'algoritmo.

L'Algoritmo Genetico è stato implementato su Matlab utilizzando il *Global Optimization Toolbox* ed è stato usato un dataset relativo a 70 forme d'onda di eventi sismici nella regione Marche, in un range che va da 3.8 a 5.2  $M_L$ . Lo scopo di questa procedura è infatti di ottenere dei parametri che consentano di ottenere le migliori prestazioni per la regione di installazione del dispositivo [72], rispetto ad utilizzare la semplice implementazione standard. Nella Tabella 3.4 sono riportati i valori dei parametri dell'algoritmo che sono stati ottenuti tramite la procedura di ottimizzazione.

I risultati dell'ottimizzazione possono essere valutati sia in termini di numero di "missed detection" che in termini di picking eccessivo. Infatti, come evidenziato da [71], sia l'assenza di rilevazione del sisma che l'eccesso di segnalazioni sono condizioni da evitare, così da assicurare sempre sia la generazione di un allarme ma anche per

Parametro	Valore Standard	Valore Ottimizzato
$T_{long}$	$500\Delta T$	2.5 s
$T_{filter}$	$300\Delta T$	4.2 s
$T_{UP}$	$20\Delta T$	0.1 s
$S1$	10	8.6
$S2$	10	8.7

Tabella 3.4: Parametri dell'Algoritmo ottimizzato e loro valori standard.  $\Delta T$  indica il tempo di campionamento.

non avere un gran numero di allarmi durante eventi lunghi. Su questi due eventi, corrispondenti alle due criticità già citate prima, è basata anche la funzione di fitness che consente l'ottimizzazione.

Rispetto al caso non ottimizzato, ora è possibile rilevare tutti gli eventi del dataset considerato, come mostrato in Figura 3.20 per lo stesso evento sismico, che prima non era invece rilevabile. Una procedura di questo tipo richiede però che vengano modificate le soglie di rilevazione del sisma (in questo caso si hanno due soglie, una per il *triggering* dell'algoritmo ed una per il *picking* dell'onda P), e ciò ovviamente andrà ad influenzare il numero di pick che vengono effettuati. I risultati hanno quindi evidenziato un trade-off tra la capacità dell'algoritmo di rilevare il sisma ed il riuscire a limitare le segnalazioni per eventi lunghi, come evidenziato in Figura 2.11, in cui sono confrontati il numero di eventi con mancate rivelazioni, il numero di eventi con rivelazioni avvenute ed il numero di eventi con picking eccessivo per il caso standard e per quello ottimizzato. Il trade-off diventa particolarmente forte per certi eventi, in cui si supera di molto la soglia di picking eccessivo, fissata ad  $N_P = 4$ .

L'algoritmo ottenuto comunque può ritenersi adatto per scopi di rilevazione di eventi sismici, essendo in grado di funzionare con tutte le forme d'onda considerate con basse incertezze (dal 1% al 4%), per cui questi risultati possono essere considerati perfettamente accettabili in questo contesto. L'eccesso di picking può essere comunque facilmente gestito implementando determinate strategie nel momento dell'implementazione dell'algoritmo su dispositivo embedded.

### Risultati delle Prove di Copertura e dell'Ottimizzazione dei Parametri LoRaWAN

Il protocollo di comunicazione che è stato scelto per la trasmissione dell'allarme generato dal dispositivo di Early Warning tramite l'algoritmo Filter Picker è il protocollo LoRaWAN. Questo protocollo è particolarmente adatto alla trasmissione di piccole quantità di dati ed in maniera sporadica, ed è quindi utile per la trasmissione di allarmi, a patto però che i suoi parametri a radiofrequenza come lo Spreading Factor vengano scelti opportunamente.

### 3.3 Dispositivo LoRaWAN per la trasmissione di allarmi.

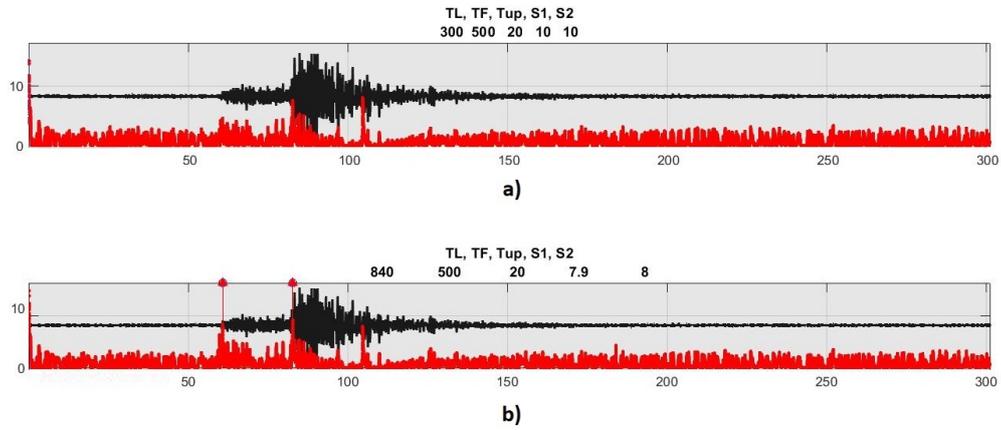


Figura 3.20: Confronto delle uscite del Filter Picker per la stessa forma d'onda di un sisma di magnitudo  $5.1 M_L$  rilevata dalla stazione GUMA, con i parametri standard (a) e dopo l'ottimizzazione (b) .

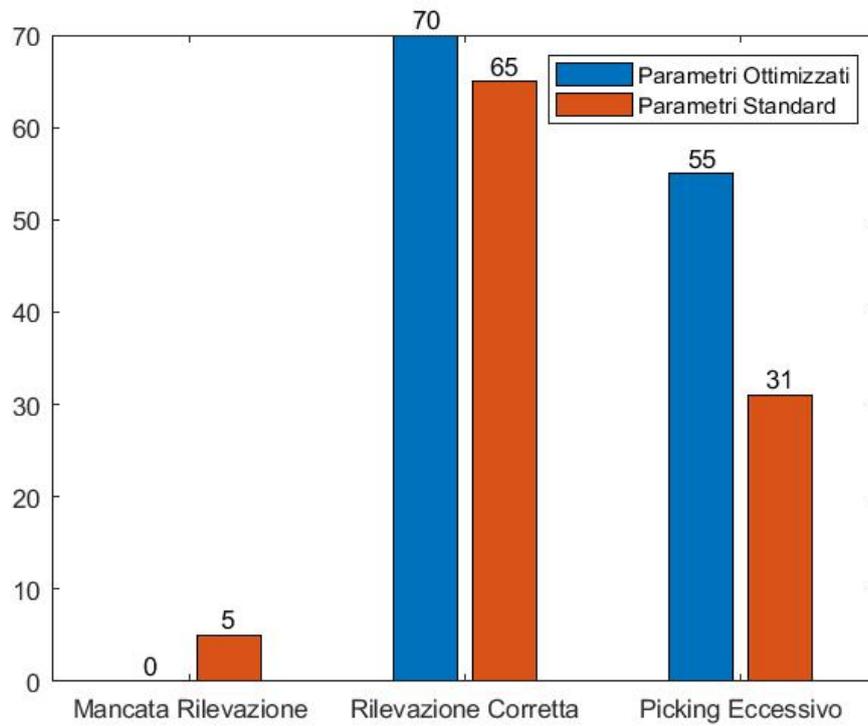


Figura 3.21: Statistiche dell'algoritmo dopo l'ottimizzazione e usando invece i parametri standard.

### Capitolo 3 Nodo Edge per Earthquake Early Warning

Per fare ciò è stata eseguita una prova di copertura dopo aver installato un gateway LoRaWAN in un ambiente indoor. Si è spostato il dispositivo in diversi punti di interesse (nell'ipotesi di installare il dispositivo di EW all'interno di un edificio), e se ne è valutato l'RSSI per due diversi Spreading Factor (SF=8 ed SF=12) e con potenza in trasmissione di 14 dBm, banda EU 868 e larghezza di banda di 125 kHz. I risultati ottenuti sono riportati in seguito. Sono state considerate le seguenti soglie per la valutazione della copertura:

- -30 dBm = soglia di segnale molto forte;
- -100 dBm = soglia di buona qualità;
- - 120 dBm = soglia di disconnessione;

Scenario	SF	$RSSI_{average}$	Qualità del Segnale
Vicino al Gateway	8	5.2 dBm	Segnale molto forte.
	12	-4 dBm	Segnale molto forte.
4 m dal Gateway (LOS)	8	-49 dBm	Buona qualità.
	12	-41 dBm.	Buona Qualità.
10 m dal Gateway (NLOS)	8	-63.9 dBm	Buona qualità.
	12	-63.45 dBm	Buona qualità.
Seminterrato	8	-96.6 dBm	Sporadicamente sotto la soglia di buona qualità.
	12	-97.3 dBm	Sopra la soglia di buona qualità.

Tabella 3.5: Risultati del test per i diversi scenari. La soglia di disconnessione di  $-120dBm$  non viene mai raggiunta per entrambi gli SF.

Oltre che il range di copertura, è importante andare a valutare come la scelta dello Spreading Factor influenza data rate, consumi e latenze. Poiché stiamo realizzando un sistema di EW, si desidera che la durata di una singola trasmissione LoRa (o *Time on Air*) sia il più breve possibile, anche con lo scopo di limitare i consumi. Si ottengono le seguenti stime dei parametri di interesse per i due SF presi in considerazione, considerando di trasmettere un payload di 8 byte contenente il timestamp dell'istante di rilevazione.

SF	Data Rate Massimo [kbps]	Time on Air[ms]	Consumo in Trasmissione [uC]
12	0.292	761.9	68567
8	3.125	57.6	5207.7

Tabella 3.6: Confronto dei diversi Data Rate, ToA e consumi che possono essere ottenuti con SF=8 ed SF=12.

Dai risultati si evince che, nonostante SF=12 dia risultati migliori dal punto di vista della copertura, la scelta di SF=8 può ritenersi comunque soddisfacente anche

### 3.3 Dispositivo LoRaWAN per la trasmissione di allarmi.

in tal senso, per cui risulta complessivamente la scelta migliore. Riassumendo, si ottengono le seguenti prestazioni della comunicazione LoRa.

Duty Cycle	5.7 s
Time on Air	57.6 ms
Consumo in Trasmissione	5207.7 $\mu C$
Data Rate	3.125 kbps

Tabella 3.7: Caratteristiche della comunicazione LoRa ottenuti con SF=8, BW=125 kHz, banda EU868,  $P_{TX}$ =14 dBm, pacchetti da 8 byte.



## Conclusioni

Lo sviluppo di sistemi di *Internet of Things* e di *Edge Computing* rappresenta un'importante opportunità per realizzare e migliorare sistemi di Early Warning. I sistemi IoT odierni sono in grado di effettuare misure ambientali, aggregare dati e comunicare in maniera affidabile e veloce, consentendo di soddisfare i requisiti di sistemi di allerta precoce per un'ampia gamma di disastri naturali, dai terremoti alle alluvioni. In particolare, in questo lavoro di tesi si è mostrato come progettare, testare ed ottimizzare un sistema di Early Warning per terremoti che sia in grado di trasmettere tempestivamente un allarme in prossimità di un evento sismico, ai fini della prevenzione di danni alle persone.

A partire da una review dello stato dell'arte dei sistemi di Early Warning per disastri naturali, si sono evidenziati i requisiti che un sistema di allerta precoce basato sul paradigma dell'IoT deve rispettare, come bassa latenza, affidabilità e bassi consumi di potenza. Spostando l'attenzione poi sui sistemi di Early Warning per eventi sismici si sono delineate le specifiche necessità di questo use case, analizzando le principali tecniche che vengono usate per rilevare o predire un sisma, e sottolineando l'importanza di ridurre la latenza per poter ottenere allarmi tempestivi. La possibilità di utilizzare sistemi di tipo Edge per la realizzazione di un sistema di Early Warning sismico si prospetta interessante proprio per riuscire a soddisfare quest'ultimo vincolo: infatti, un sistema Edge "avvicina" memoria e capacità di calcolo all'estremità della rete, permettendo di compiere pre-processamento dei dati o anche di effettuare predizioni direttamente sugli End-Node, senza dover trasmettere dati verso un servizio Cloud. Ciò consente di ridurre sensibilmente la latenza del sistema, in quanto l'allarme può essere direttamente generato dal dispositivo, che avrà poi il compito di trasmetterlo verso un server, servizio Cloud o infrastruttura di competenza. Inoltre, l'utilizzo di sistemi Edge diminuisce anche la quantità di dati che il dispositivo deve trasmettere, assicurando un minor consumo dovuto alle trasmissioni, un minore carico su Cloud e server, e maggior resistenza ad eventuali instabilità di banda che si possono riscontrare durante un evento disastroso. Rispetto a sistemi esistenti come ISNet, la rete sismica installata nella regione dell'Irpinia in Italia, un sistema di questo tipo può evitare problemi quali i ritardi di pacchettizzazione che vengono inevitabilmente introdotti quando si trasmettono dei dati accelerometrici verso una stazione remota per l'elaborazione, nonché i ritardi di processamento che possono insorgere al server, ad esempio se questo viene rallentato o saturato da un'ampia mole di dati da processare.

Le prestazioni di un sistema Edge sono però limitate dai vincoli di memoria e

capacità computazionali che tipicamente possiede un sistema embedded. Pertanto la scelta del corretto algoritmo di predizione diventa essenziale in questo senso: deve essere leggero ed utilizzare poche e semplici elaborazioni dei dati. Nel nostro caso si sono analizzati algoritmi di tipo *on site*, ossia che rilevano l'arrivo di un'onda sismica nel luogo di installazione. Studiando i principali algoritmi in uso, si è visto come la rilevazione o "picking" dell'onda P sia la strategia più utilizzata e, nel contesto dei sistemi con vincoli di memoria, i metodi multibanda e i metodi "energy based" risultano essere la scelta più adeguata. Si è però anche mostrato come la semplice scelta dell'algoritmo di predizione (nel nostro caso l'algoritmo Filter Picker) non sia sufficiente ad assicurare il buon funzionamento del sistema: l'ottimizzazione dell'algoritmo rispetto a parametri locali è essenziale per riuscire ad effettuare il picking dell'onda P correttamente. Mentre l'algoritmo Filter Picker è ben consolidato seppur presenti le già citate criticità (è attualmente in uso anche nella rete sismica ISNet già menzionata), sviluppi futuri, evidenziati anche dalla letteratura revisionata, potrebbero riguardare l'integrazione di modelli di Machine Learning all'interno di dispositivi embedded per ottenere predizioni, sempre considerati però i vincoli di memoria e calcolo che interessano questo genere di sistema, e per i quali algoritmi come Filter Picker risultano invece estremamente semplici da implementare.

La realizzazione di una soluzione di questo tipo richiede la scelta accurata anche del protocollo di comunicazione per la trasmissione di allarmi. Poiché vogliamo trasmettere solamente un allarme contenente un timestamp dell'istante di picking, si è scelto come protocollo di comunicazione il protocollo LoRaWAN. LoRaWAN rientra tra i protocolli più usati anche nella letteratura nel contesto dell'Early Warning, essendo adatto per la trasmissione sporadica di messaggi brevi e poiché consente basse latenze, ottima copertura e bassi consumi, purché si tenga conto di alcune sue caratteristiche e si scelgano correttamente i parametri a radiofrequenza per ottenere queste prestazioni. In tal senso, si sono effettuate delle prove di copertura che consentano di scegliere lo *Spreading Factor* adatto ai requisiti del sistema di EW, mostrando come uno Spreading Factor pari ad 8 consenta di ottenere prestazioni adeguate per l'Early Warning. Ulteriori prove future potrebbero essere fatte testando lo SF=7, che consente di usare diverse configurazioni di larghezza di banda, seppur presenti prestazioni più degradate in termini di copertura rispetto a quelli testati, e per questo ad esso è stato preferito un fattore maggiore. Inoltre, tra gli sviluppi futuri per questo genere di caso d'uso, è da considerare l'integrazione nel sistema di Early Warning del 5G ed in particolare del servizio URLLC (*Ultra Reliability Low Latency Communication*), che si prospetta di offrire grandi vantaggi in termini di latenza e di affidabilità, due vincoli che sono essenziali per l'allerta precoce sismica.

In conclusione, la scelta dell'algoritmo e la sua ottimizzazione effettuata in questo lavoro di tesi, nonché la selezione del protocollo di comunicazione LoRaWAN e dei suoi parametri, hanno dimostrato di essere adatti ai requisiti di un sistema di EW per il monitoraggio sismico. Inoltre, si ribadisce che i vincoli imposti da questo tipo di sistema suggeriscono di migrare verso un paradigma di Edge Computing, che si

### *3.3 Dispositivo LoRaWAN per la trasmissione di allarmi.*

è dimostrato essere in grado di generare allarmi con bassa incertezza e con minor complessità rispetto ad un sistema basato esclusivamente sul Cloud Computing o architetture simili.



## Bibliografia

- [1] S. K. Sood, R. Sandhu, K. Singla, and V. Chang, "IoT, big data and HPC based smart flood management framework," *Sustainable Computing: Informatics and Systems*, vol. 20, pp. 102–117, 2018.
- [2] A. M. Zambrano, X. Calderón, S. Jaramillo, O. M. Zambrano, M. Esteve, and C. Palau, "3 - community early warning systems," in *Wireless Public Safety Networks 3* (D. Câmara and N. Nikaein, eds.), pp. 39 – 66, Elsevier, 2017.
- [3] P. Pierleoni, S. Marzorati, C. Ladina, S. Raggiunto, A. Belli, L. Palma, M. Cattaneo, and S. Valenti, "Performance Evaluation of a Low-Cost Sensing Unit for Seismic Applications: Field Testing during Seismic Events of 2016-2017 in Central Italy," *IEEE Sensors Journal*, vol. 18, no. 16, pp. 6644–6659, 2018.
- [4] B. A. Bolt, "Earthquake". Encyclopedia Britannica," 2021. <https://www.britannica.com/science/earthquake-geology>.
- [5] E. R. Burkett, D. D. Given, L. M. Jones, Erin R. Burkett, D. D. Given, and L. M. Jones, "ShakeAlert-An Earthquake Early Warning System for the United States West Coast," *U.S. Geological Survey Fact Sheet*, no. February, 2014.
- [6] I. Alvarez, L. Garcia, S. Mota, G. Cortes, C. Benitez, and A. De La Torre, "An automatic p-phase picking algorithm based on adaptive multiband processing," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 6, pp. 1488–1492, 2013.
- [7] L. Luzi, G. Lanzano, C. Felicetta, M. D'Amico, E. Russo, S. Sgobba, F. Pacor, and O. W. G. . (2020), "Engineering Strong Motion Database (ESM) (Version 2.0). Istituto Nazionale di Geofisica e Vulcanologia (INGV)," 2020. <https://doi.org/10.13127/ESM.2>.
- [8] "Overview LoRaWAN di LoRa Alliance," <https://lorawan-alliance.org/wp-content/uploads/2020/11/what-is-lorawan.pdf>.
- [9] D. H. Kim, E. K. Lee, and J. Kim, "Experiencing LoRa network establishment on a smart energy campus testbed," *Sustainability (Switzerland)*, vol. 11, no. 7, 2019.
- [10] "Documentazione TTN sugli Spreading Factor.," <https://www.thethingsnetwork.org/docs/lorawan/spreading-factors/>.

## Bibliografia

- [11] “Documentazione LoRa,” <https://lorareadthedocs.io/en/latest/#>.
- [12] “Documentazione B-L072Z-LRWAN1 LoRaR©Discovery Kit,” <https://www.st.com/en/evaluation-tools/b-l072z-lrwan1.html>.
- [13] “Sentrius Gateway Installation Guide,” <https://www.thethingsnetwork.org/docs/gateways/laird/12>.
- [14] “Documentazione sulla tecnologia TTN.,” <https://www.thethingsnetwork.org/tech-stack#section1>.
- [15] R. M. Allen and D. Melgar, “Earthquake early warning: Advances, scientific challenges, and societal needs,” *Annual Review of Earth and Planetary Sciences*, vol. 47, pp. 361–388, 2019.
- [16] J. Mulero Chaves and T. De Cola, “1 - public warning applications: Requirements and examples,” in *Wireless Public Safety Networks 3* (D. Câmara and N. Nikaein, eds.), pp. 1 – 18, Elsevier, 2017.
- [17] B. Dorsemaine, J. P. Gaulier, J. P. Wary, N. Kheir, and P. Urien, “Internet of Things: A Definition and Taxonomy,” *Proceedings - NGMAST 2015: The 9th International Conference on Next Generation Mobile Applications, Services and Technologies*, no. September, pp. 72–77, 2016.
- [18] J. A. Manrique, J. S. Rueda-Rueda, and J. M. T. Portocarrero, “Contrasting Internet of Things and Wireless Sensor Network from a Conceptual Overview,” *Proceedings - 2016 IEEE International Conference on Internet of Things; IEEE Green Computing and Communications; IEEE Cyber, Physical, and Social Computing; IEEE Smart Data, iThings-GreenCom-CPSCoM-Smart Data 2016*, pp. 252–257, 2017.
- [19] S. Hamdan, M. Ayyash, and S. Almajali, “Edge-computing architectures for internet of things applications: A survey,” *Sensors (Switzerland)*, vol. 20, no. 22, pp. 1–52, 2020.
- [20] K. Singh and D. D. Singh Tomar, “Architecture, enabling technologies, security and privacy, and applications of internet of things: A survey,” *Proceedings of the International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2018*, vol. 4, no. 5, pp. 642–646, 2019.
- [21] M. Acosta-Coll, F. Ballester-Merelo, M. Martinez-Peiró, and E. De la Hoz-Franco, “Real-time early warning system design for pluvial flash floods—a review,” *Sensors (Switzerland)*, vol. 18, no. 7, 2018.
- [22] E. A. Basha, S. Ravela, and D. Rus, “Model-based monitoring for early warning flood detection,” *SenSys’08 - Proceedings of the 6th ACM Conference on Embedded Networked Sensor Systems*, no. January, pp. 295–308, 2008.

- [23] S. A. Shah, D. Z. Seker, S. Hameed, and D. Draheim, "The rising role of big data analytics and IoT in disaster management: Recent advances, taxonomy and prospects," *IEEE Access*, vol. 7, pp. 54595–54614, 2019.
- [24] P. P. Ray, M. Mukherjee, and L. Shu, "Internet of Things for Disaster Management: State-of-the-Art and Prospects," *IEEE Access*, vol. 5, no. i, pp. 18818–18835, 2017.
- [25] Y. Ran, "Considerations and suggestions on improvement of communication network disaster countermeasures after the wenchuan earthquake," *IEEE Communications Magazine*, vol. 49, no. 1, pp. 44–47, 2011.
- [26] S. Popli, R. K. Jha, and S. Jain, "A Survey on Energy Efficient Narrowband Internet of Things (NB-IoT): Architecture, Application and Challenges," *IEEE Access*, vol. 7, pp. 16739–16776, 2019.
- [27] O. G., "A Survey of ZigBee Wireless Sensor Network Technology: Topology, Applications and Challenges," *International Journal of Computer Applications*, vol. 130, no. 9, pp. 47–55, 2015.
- [28] C. Gomez, J. Oller, and J. Paradells, "Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology," *Sensors (Switzerland)*, vol. 12, no. 9, pp. 11734–11753, 2012.
- [29] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," *ICT Express*, vol. 5, no. 1, pp. 1–7, 2019.
- [30] S. H. Hwang and S. Z. Liu, "Survey on 3GPP low power wide area technologies and its application," *Proceedings - 2019 IEEE VTS Asia Pacific Wireless Communications Symposium, APWCS 2019*, 2019.
- [31] J. Ploennigs, U. Ryssel, and K. Kabitzsch, "Performance analysis of the EnOcean wireless sensor network protocol," *Proceedings of the 15th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2010*, no. October 2010, 2010.
- [32] G. Mei, N. Xu, J. Qin, B. Wang, and P. Qi, "A Survey of Internet of Things (IoT) for Geohazard Prevention: Applications, Technologies, and Challenges," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4371–4386, 2020.
- [33] S. Jayashree, S. Sarika, A. L. Solai, and S. Prathibha, "A novel approach for early flood warning using android and IoT," *Proceedings of the 2017 2nd International Conference on Computing and Communications Technologies, ICCCT 2017*, pp. 339–343, 2017.

## Bibliografia

- [34] J. Ibarreche, R. Aquino, R. M. Edwards, V. Rangel, I. Pérez, M. Martínez, E. Castellanos, E. Álvarez, S. Jimenez, R. Rentería, A. Edwards, and O. Álvarez, “Flash flood early warning system in colima, mexico,” *Sensors (Switzerland)*, vol. 20, no. 18, pp. 1–26, 2020.
- [35] Q. Udpol, P. P. Lvklgx, Q. Jr, M. S. I. Nrmlpd, Q. Jr, M. S. Pd-kdprg, X. N. P. Hgx, and P. P. Xnp, “7Kh : Ruog ) Luvw ’ Hsor \$\\$PhqwRi1Duurzedqg, R7Iru,” vol. 81, no. November, pp. 28 – –30, 201708.
- [36] D. Purkovic, L. Coates, M. Honsch, D. Lumbeck, and F. Schmidt, “Smart river monitoring and early flood detection system in Japan developed with the EnOcean long range sensor technology,” *2019 2nd International Colloquium on Smart Grid Metrology, SMAGRIMET 2019 - Proceedings*, pp. 2–7, 2019.
- [37] M. Ragnoli, G. Barile, A. Leoni, and G. Ferri, “An Autonomous Low-Power LoRa-Based Flood-Monitoring System,” 2020.
- [38] P. Mitra, R. Ray, R. Chatterjee, R. Basu, P. Saha, S. Raha, R. Barman, S. Patra, S. S. Biswas, and S. Saha, “Flood forecasting using Internet of things and artificial neural networks,” *7th IEEE Annual Information Technology, Electronics and Mobile Communication Conference, IEEE IEMCON 2016*, pp. 1–5, 2016.
- [39] G. Furquim, G. P. R. Filho, R. Jalali, G. Pessin, R. W. Pazzi, and J. Ueyama, “How to improve fault tolerance in disaster predictions: A case study about flash floods using IoT, ML and real data,” *Sensors (Switzerland)*, vol. 18, no. 3, pp. 1–20, 2018.
- [40] H. Tariq, F. Touati, M. A. E. Al-Hitmi, D. Crescini, and A. B. Mnaouer, “A real-time early warning seismic event detection algorithm using smart geo-spatial bi-axial inclinometer nodes for Industry 4.0 applications,” *Applied Sciences (Switzerland)*, vol. 9, no. 18, pp. 1–25, 2019.
- [41] M. Anbarasan, B. A. Muthu, C. B. Sivaparthipan, R. Sundarasekar, S. Kadry, S. Krishnamoorthy, D. J. Samuel, and A. A. Dasel, “Detection of flood disaster system based on IoT, big data and convolutional deep neural network,” *Computer Communications*, vol. 150, no. November 2019, pp. 150–157, 2020.
- [42] J. Al Qundus, K. Dabbour, S. Gupta, R. Meissonier, and A. Paschke, “Wireless sensor network for AI-based flood disaster detection,” *Annals of Operations Research*, no. 0123456789, 2020.
- [43] A. Alphonsa and G. Ravi, “Earthquake early warning system by IOT using Wireless sensor networks,” *Proceedings of the 2016 IEEE International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2016*, pp. 1201–1205, 2016.

- [44] K. Fauvel, D. Balouek-Thomert, D. Melgar, P. Silva, A. Simonet, G. Antoniu, A. Costan, V. Masson, M. Parashar, I. Rodero, and A. Termier, “A Distributed Multi-Sensor Machine Learning Approach to Earthquake Early Warning,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, pp. 403–411, 2020.
- [45] K. K. Khedo, Y. Bissessur, and D. S. Goolaub, “An inland Wireless Sensor Network system for monitoring seismic activity,” *Future Generation Computer Systems*, vol. 105, pp. 520–532, 2020.
- [46] J. Fu, Z. Li, H. Meng, J. Wang, and X. Shan, “for Dense Earthquake Early Warning : 2018 – 2019 Field,” pp. 2018–2019, 2019.
- [47] C. Peng, P. Jiang, Q. Chen, Q. Ma, and J. Yang, “Performance evaluation of a dense MEMS-based seismic sensor array deployed in the Sichuan-Yunnan border region for earthquake early warning,” *Micromachines*, vol. 10, no. 11, 2019.
- [48] C. Peng, Y. Chen, Q. Chen, J. Yang, H. Wang, X. Zhu, Z. Xu, and Y. Zheng, “A new type of tri-axial accelerometers with high dynamic range MEMS for earthquake early warning,” *Computers and Geosciences*, vol. 100, no. January, pp. 179–187, 2017.
- [49] A. M. Zambrano, I. Perez, C. Palau, and M. Esteve, “Technologies of Internet of Things applied to an Earthquake Early Warning System,” *Future Generation Computer Systems*, vol. 75, no. 2017, pp. 206–215, 2017.
- [50] R. M. Allen, Q. Kong, and R. Martin-Short, “The MyShake Platform: A Global Vision for Earthquake Early Warning,” *Pure and Applied Geophysics*, vol. 177, no. 4, pp. 1699–1712, 2020.
- [51] M. Klapez, C. A. Grazia, S. Zennaro, M. Cozzani, and M. Casoni, “First Experiences with Earthcloud, a Low-Cost, Cloud-Based IoT Seismic Alert System,” *International Conference on Wireless and Mobile Computing, Networking and Communications*, vol. 2018-Octob, pp. 262–269, 2018.
- [52] L. D’Errico, F. Franchi, F. Graziosi, A. Marotta, C. Rinaldi, M. Boschi, and A. Colarieti, “Structural health monitoring and earthquake early warning on 5g urllc network,” *IEEE 5th World Forum on Internet of Things, WF-IoT 2019 - Conference Proceedings*, pp. 783–786, 2019.
- [53] C. C. Kao, Y. S. Lin, G. D. Wu, and C. J. Huang, “A comprehensive study on the internet of underwater things: Applications, challenges, and channel models,” *Sensors (Switzerland)*, vol. 17, no. 7, 2017.
- [54] S. Fattah, A. Gani, I. Ahmady, M. Y. I. Idris, and I. A. T. Hashem, “A survey on underwater wireless sensor networks: Requirements, taxonomy, recent advances,

## Bibliografia

- and open research challenges,” *Sensors (Switzerland)*, vol. 20, no. 18, pp. 1–30, 2020.
- [55] L. Freitag, K. Ball, P. Koski, J. Partan, S. Singh, and D. Giaya, “Acoustic Communications for Bottom-toBottom Ocean Sensor Networks,” *OCEANS 2018 MTS/IEEE Charleston, OCEAN 2018*, pp. 8–12, 2019.
- [56] J. Meza, P. A. Catalán, and H. Tsushima, “A Methodology For Optimal Designing Of Monitoring Sensor Networks For Tsunami Inversion,” *Natural Hazards and Earth System Sciences Discussions*, no. October, pp. 1–25, 2018.
- [57] M. Mochizuki, K. Uehira, T. Kanazawa, T. Kunugi, K. Shiomi, S. Aoi, T. Matsumoto, N. Takahashi, N. Chikasada, T. Nakamura, S. Sekiguchi, M. Shinohara, and T. Yamada, “S-net project: Performance of a large-scale seafloor observation network for preventing and reducing seismic and tsunami disasters,” *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans, OCEANS - Kobe 2018*, pp. 1–4, 2018.
- [58] M. Inoue, Y. Tanioka, and Y. Yamanaka, “Method for near-real time estimation of Tsunami sources using ocean bottom pressure sensor network (S-net),” *Geosciences (Switzerland)*, vol. 9, no. 7, 2019.
- [59] N. Jain and D. Virmani, “Feature Classification for Underwater Seismic Prediction Using Wireless Sensor Nodes.,” *Lecture Notes in Engineering and Computer Science*, vol. 2227, pp. 151–156, 2017.
- [60] P. Gardner-Stephen, A. Wallace, K. Hawtin, G. Al-Nuaimi, A. Tran, T. Le Mozo, and M. Lloyd, “Reducing cost while increasing the resilience effectiveness of tsunami early warning systems,” *2019 IEEE Global Humanitarian Technology Conference, GHTC 2019*, pp. 1–8, 2019.
- [61] P. Adi, D. Prasetya, A. Setiawan, N. Nachrowie, and R. Arifuddin, “Design Of Tsunami Detector Based Sort Message Service Using Arduino and SIM900A to GSM/GPRS Module,” no. January, 2019.
- [62] T. Kato, Y. Terada, K. Tadokoro, N. Kinugasa, A. Futamura, M. Toyoshima, S. I. Yamamoto, M. Ishii, T. Tsugawa, M. Nishioka, K. Takizawa, Y. Shoji, and H. Seko, “Development of GNSS buoy for a synthetic geohazard monitoring system,” *Journal of Disaster Research*, vol. 13, no. 3, pp. 460–471, 2018.
- [63] S. Darmawan, B. Irawan, C. Setianingsih, and M. A. Murty, “Design of detection device for sea water waves with fuzzy algorithm based on internet of things,” *Proceedings - 2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology, IAICT 2020*, pp. 75–80, 2020.

- [64] P. R. Pereira, A. Grilo, F. Rocha, M. S. Nunes, A. Casaca, C. Chaudet, P. Almström, and M. Johansson, “End-To-End Reliability in Wireless Sensor Networks: Survey and Research Challenges,” *EuroFGI Workshop on IP QoS and Traffic Control*, vol. 54, no. June 2014, pp. 67–74, 2007.
- [65] P. Kumar, P. Kumar, P. Priyadarshini, and Srija, “Underwater acoustic sensor network for early warning generation,” *OCEANS 2012 MTS/IEEE: Harnessing the Power of the Ocean*, pp. 1–6, 2012.
- [66] R. W. L. Coutinho, A. Boukerche, L. F. M. Vieira, and A. A. F. Loureiro, “Underwater Sensor Networks for Smart Disaster Management,” *IEEE Consumer Electronics Magazine*, vol. 9, no. 2, pp. 107–114, 2020.
- [67] M. Marin-Perianu, S. Chatterjea, R. Marin-Perianu, S. Bosch, S. Dulman, S. Kininmonth, and P. Havinga, “Wave monitoring with wireless sensor networks,” *ISSNIP 2008 - Proceedings of the 2008 International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pp. 611–616, 2008.
- [68] S. E. Minson, A. S. Baltay, E. S. Cochran, T. C. Hanks, M. T. Page, S. K. McBride, K. R. Milner, and M. A. Meier, “The Limits of Earthquake Early Warning Accuracy and Best Alerting Strategy,” *Scientific Reports*, vol. 9, no. 1, pp. 1–13, 2019.
- [69] J. Clinton, A. Zollo, A. Marmureanu, C. Zulfikar, and S. Parolai, “State-of-the art and future of earthquake early warning in the European region,” *Bulletin of Earthquake Engineering*, vol. 14, no. 9, pp. 2441–2458, 2016.
- [70] L. García, I. Álvarez, C. Benítez, M. Titos, Á. Bueno, S. Mota, Á. de la Torre, J. C. Segura, G. Alguacil, A. Díaz-Moreno, J. Prudencio, A. García-Yeguas, J. M. Ibáñez, L. Zuccarello, O. Cocina, and D. Patanè, “Advances on the automatic estimation of the P-wave onset time,” *Annals of Geophysics*, vol. 59, no. 4, 2016.
- [71] A. Lomax, C. Satriano, and M. Vassallo, “Automatic picker developments and optimization: Filterpicker-A robust, broadband picker for real-time seismic monitoring and earthquake early warning,” *Seismological Research Letters*, vol. 83, no. 3, pp. 531–540, 2012.
- [72] M. Vassallo, C. Satriano, and A. Lomax, “Automatic picker developments and optimization: A strategy for improving the performances of automatic phase pickers,” *Seismological Research Letters*, vol. 83, no. 3, pp. 541–554, 2012.
- [73] “Documentazione TTN sui Duty Cycle.” <https://www.thethingsnetwork.org/docs/lorawan/duty-cycle/#:~:text=Duty%20Cycle%20indicates%20the%20fraction,get%20a%20bit%20more%20complicated>.

## Bibliografia

- [74] “Approfondimento sulla relazione tra chirp rate, bit rate e spreading factor in LoRa.,” <https://electronics.stackexchange.com/questions/278192/understanding-the-relationship-between-lora-chips-chirps-symbols-and-bits>.
- [75] “Schema ALOHAnet,” <https://en.wikipedia.org/wiki/ALOHAnet>.
- [76] N. Chinchilla-Romero, J. Navarro-Ortiz, P. Muñoz, and P. Ameigeiras, “Collision avoidance resource allocation for LoRaWAN,” *Sensors (Switzerland)*, vol. 21, no. 4, pp. 1–19, 2021.
- [77] “Documentazione TTN sulla Fair Policy.,” <https://www.thethingsnetwork.org/forum/t/fair-use-policy-explained/130057>.
- [78] T. Attia, M. Heusse, B. Tourancheau, and A. Duda, “Experimental characterization of LoRaWAN link quality,” *2019 IEEE Global Communications Conference, GLOBECOM 2019 - Proceedings*, no. Section II, pp. 6–11, 2019.