



SAPIENZA
UNIVERSITÀ DI ROMA

Relazione bando BS-J 11/20

Dipartimento di Ingegneria dell'Informazione, Elettronica e Telecomunica-
zioni

Corso di Laurea in

Candidato

Emanuele Giacomini

Matricola 1743995

Relatore

Francesca Cuomo

Anno Accademico 2021

Relazione bando BS-J 11/20

Tesi di Laurea. Sapienza – Università di Roma

© 2021 Emanuele Giacomini. Tutti i diritti riservati

Questa tesi è stata composta con L^AT_EX e la classe Sapthesis.

Email dell'autore: emanuele.giacomo@gmail.com

Indice

1	Introduzione	1
1.1	LoRaWAN	1
2	Contesto Applicativo	2
3	Localizzazione	3
3.1	RSSI Fingerprint	4
3.1.1	Density Estimation	4
3.1.2	NonLinear Regression	5
3.2	Simulated Radio Waves via Raycasting techniques	5

Capitolo 1

Introduzione

Durante l'intercorso di questa borsa, sono stati analizzati molteplici soluzioni alla localizzazione di un dispositivo equipaggiato con antenna LoRaWAN, all'interno di un contesto urbano. Come intuibile, il contesto urbano risulta particolarmente complesso a causa di un innumerevole quantità di fattori che possono alterare le condizioni del canale radio utilizzato dai messaggi LoRa. Nei seguenti capitoli vengono analizzati nel dettaglio gli argomenti chiave di questo lavoro, quale un introduzione al funzionamento di LoRa/LoRaWAN e al corrispettivo campo di applicazione, la descrizione del problema di localizzazione ed infine un introduzione alle possibili tecniche risolutive tenute in considerazione (prima in forma diretta e poi attraverso l'uso di filtri ad-hoc unimodali e multi-modali).

Al momento della realizzazione di questo documento, non sono ancora stati effettuati test pratici che possano validare lo studio di seguito mostrato.

1.1 LoRaWAN

L'architettura LoRaWAN (Long Range Wide Area Network) si occupa di connettere dispositivi principalmente alimentati a batteria, fra loro o ad Internet. La topologia di LoRaWAN tipicamente è descritta con uno schema *star-of-stars* in cui dei nodi fissi detti *Gateways* si occupano di ritrasmettere i messaggi fra i dispositivi finali ed il server di rete centrale. La connessione fra server centrale e gateways avviene tramite connessione IP e, questi ultimi si occupano di convertire qualsiasi pacchetto radio in pacchetto IP e viceversa. La comunicazione wireless fra i dispositivi ed i gateways, sfrutta la caratteristica di lungo raggio del livello fisico LoRA per consentire la trasmissione single-gateway o multi-gateway a singolo salto.

Capitolo 2

Contesto Applicativo

Il costo contenuto dei moduli di trasmissione e le caratteristiche low-power, rendono LoRaWAN adatto ad applicazioni relative all'Internet Of Things (IoT). In diversi contesti, può essere interessante discutere del problema della localizzazione riferita a questi ultimi. Sebbene il contesto applicativo scelto di ambiente urbano, abbia la disponibilità di strumenti come il GPS per poter stimare la sua posizione, risulta altrettanto interessante riuscire ad ottenere una stima verosimile attraverso l'interazione che il dispositivo effettua nei confronti della rete LoRaWAN prestabilita.

Lo studio in questione si occupa di valutare la fattibilità della localizzazione partendo proprio dalle misurazioni ricevute dai gateway presenti nella città di Roma, in prossimità della Basilica di San Pietro in Vincoli. In seguito ad una campagna di dati effettuata in precedenza, si è notata una presenza relativamente scarsa di segnale, ma forse sufficiente a produrre un risultato soddisfacente. Infatti dallo studio è emerso che nella zona considerata, vi è una disponibilità media di 1.3 gateway, una quantità troppo bassa per poter utilizzare tecniche convenzionali di triangolazione attraverso Time Of Arrival (ToA) o Time Difference Of Arrival (TDoA). D'altra parte lo studio ha rilevato circa 12 gateway diversi, dunque potrebbe essere possibile approssimare la localizzazione attraverso un approccio probabilistico che non basa la sua stima solamente sulle informazioni appena ottenute dalla rete LoRa, ma bensì anche sulla base di informazioni precedentemente ottenute. Sfruttando per esempio una composizione di mappe fingerprint e una sequenza di misurazioni, è possibile localizzare il trasmettitore, trovando la successione di posizioni che meglio descrive le misurazioni ottenute (problema di Maximum Likelihood Estimation o MLE). Sebbene questo approccio possa funzionare, è necessario realizzare le mappe di fingerprint RSSI (Received Signal Strength Indication) per ogni gateway disponibile nella zona e questo chiaramente significa effettuare una campagna di raccolta dati. In oltre la costruzione della mappa può risultare assolutamente non banale in quanto l'ambiente urbano è soggetto a interferenze radio e alla presenza di ostacoli che possono degradare il segnale LoRa, dunque è necessaria una fase di simulazione per verificare che il sistema funzioni anche in un ambiente occluso e in generale fortemente disturbato. Nella prossima sezione verrà descritto formalmente il problema della localizzazione per poi studiare le possibili soluzioni.

Capitolo 3

Localizzazione

Modelliamo la sonda come un sistema dinamico stocastico il cui stato non è direttamente osservabile, la cui funzione di osservazione è affetta da un errore e, per semplicità, di non avere accesso ai controlli che modificano lo stato della sonda. Assumendo che lo spazio topologico in cui risiede la sonda, sia una varietà di ordine 2, allora la sonda si muove in uno spazio euclideo bidimensionale e dunque la sua posizione al tempo i può essere identificata con $x_i = (x_i^l, y_i^l)^t$. Nel piano sono anche presenti dei gateway che compongono la rete LoRaWAN. Il gateway i -esimo è situato in $G_i = (x_i^g, y_i^g)$. Definiamo lo stato del problema come posizione corrente della sonda $x_i \in \mathbb{R}^2$. Non conoscendo i controlli che determinano il movimento della sonda sul piano, dobbiamo assumere che la funzione di transizioni sia statica:

$$x_t = f(x_{t-1}) = x_{t-1} \quad (3.1)$$

Riguardo alla funzione di osservazione $h(z_t|x_t)$, è possibile costruire una struttura a blocchi della forma:

$$C_t = \begin{pmatrix} C_t^0 \\ C_t^1 \\ \vdots \\ C_t^m \end{pmatrix} \quad (3.2)$$

Dove C_t^k rappresenta le osservazioni ottenute all'istante t dal k -esimo gateway:

$$C_t^k = \begin{pmatrix} \text{ToA}_t^k \\ \text{RSSI}_t^k \end{pmatrix} \quad (3.3)$$

Queste due misure sono ottenute direttamente come risposta dal network server della rete e sono correlate con la distanza tra il gateway e la sonda. In condizioni ideali, possiamo stimare il valore di ToA nella seguente maniera:

$$\text{ToA}_t^k = \frac{|x_t - G_i|}{c} \quad (3.4)$$

dove c rappresenta la velocità di propagazione del segnale, in questo caso la velocità della luce. Nel secondo caso, non è nota una forma analitica di stima RSSI basata sulla distanza, in quanto la path loss del canale di propagazione non è facilmente modellabile in un contesto caotico come quello urbano. Proprio per questo motivo, è possibile raggiungere una predizione usando un approccio inverso, ovvero campionando l'ambiente operativo e producendo una mappa di fingerprint, che associa ad

ogni possibile posizione (x, y) un valore stimato di RSSI.

Avendo formalizzato il sistema dinamico della sonda, possiamo incominciare a parlare del vero e proprio sistema di localizzazione. È stato scelto un filtro di particelle (PF) in quanto rilassa il vincolo posto da EKF o filtri simili di modellare la probabilità dello stato $p(x_t)$ come una distribuzione **unimodale** gaussiana, dunque ottenendo una distribuzione **multimodale**. Questo consente all'algoritmo di localizzazione di considerare in una prima fase transitoria, più località probabili in parallelo, e di aggiornare la loro massa probabilistica in base alle misurazioni future. La fase transitoria può considerarsi terminata quando tutte le località stimate, collassano in una singola zona considerata di probabilità maggiore. Un ulteriore vantaggio dei PF è quello di non sfruttare informazioni provenienti dalle derivate, che potrebbero porre un problema durante la fase di predizione delle misure, in quanto parte di queste ultime vengono generate da una rete neurale. Nella prossima sezione verrà descritto il processo con cui viene generata la mappa di fingerprint RSSI.

3.1 RSSI Fingerprint

In precedenza è stata valutata l'idea di sfruttare sequenze di misure RSSI di diversi gateway, accoppiate con le rispettive posizioni di trasmissione per stimare la diretta relazione tra di loro.

3.1.1 Density Estimation

Sebbene recentemente, approcci del genere che si basano su Recurrent Neural Networks (RNN) o Convolutional Neural Networks (CNN), sono apparsi diversi problemi di accoppiamento di questi ultimi con il PF precedentemente esposto. Infatti una funzione di predizione per un filtro, può dipendere unicamente dai caratteri dello stato (dunque la posizione corrente stimata della sonda) e restituire in uscita, una predizione della misura associata, mentre l'approccio appena proposto, presenta una forma inversa. Si è pensato dunque di approcciare il problema in forma inversa, dunque risolvendo un problema di **Density Estimation** (DE). Come spiegato in precedenza, riuscire a stimare l'RSSI studiando direttamente il canale di trasmissione, risulta particolarmente complesso in quanto l'ambiente urbano presenta molte forme di attenuazione e disturbo non trivialmente modellabili. L'approccio pensato dunque, è quello di campionare l'ambiente attraverso una tecnica deterministica (a griglia) o stocastica (random sampling) per ottenere letture stimate di RSSI in quelle specifiche posizioni. Da queste, si cerca poi di modellare la densità come un modello di **Gaussian Mixture Model** (GMM) della forma qui mostrata:

$$p(y) = \sum_{i=1}^N \phi_i \mathcal{N}(\mu_i, \Sigma_i) \quad (3.5)$$

dove N rappresenta il numero di componenti gaussiane prese in considerazione, ϕ_i il peso della componente i -esima e μ_i, Σ_i come rispettivamente media e covarianza della componente i -esima. È possibile dimostrare che data questa struttura di distribuzione, anche la probabilità condizionata $p(y|x)$ è una GMM, i cui parametri possono essere stimati tramite un algoritmo di **Expectation Maximization** (EM). Una tecnica nota per riuscire a produrre $p(y|x)$ passa per un modello di rete neurale chiamata **Mixture Density Network** (MDN) progettata per restituire in output, i parametri ϕ, μ e Σ delle componenti considerate.

3.1.2 NonLinear Regression

Un altro approccio attualmente in validazione, tenta di ottenere la mappa di fingerprint risolvendo un problema di regressione non lineare. Dai campioni descritti nella sezione precedente, potrebbe essere possibile ricostruire il comportamento di RSSI percettivo per uno o più gateway.

Formalmente, definiamo la funzione $f_{RSSI}^k(x, y, \mathbb{O}) \in \mathbb{R}$ che associa ogni punto (x, y) nel piano, ad un valore di RSSI misurato dall'antenna k -esima. Assumiamo in oltre che il valore di f_{RSSI}^k sia anche influenzato dalla presenza di ostacoli statici presenti nell'ambiente (i.e. edifici) contenuti nell'insieme \mathbb{O} .

Definiamo un approssimatore di funzione $g_{RSSI}^k(x, y)$ che associa ogni punto (x, y) nel piano, ad un valore di RSSI *stimato* dall'antenna k -esima. Si noti come g_{RSSI}^k non dipenda dal gruppo di ostacoli \mathbb{O} . Tramite i campioni precedentemente ottenuti, si vuole ottenere un'approssimazione sufficientemente valida, ovvero

$$g_{RSSI}^k(x, y) \simeq f_{RSSI}^k(x, y, \mathbb{O}) \quad (3.6)$$

L'approssimatore di funzione può essere realizzato come una Deep Neural Network (DNN) con una dimensione di ingresso pari a 2 features e una dimensione di output pari ad 1 feature. D'altra parte, gli iperparametri (numero di hidden layer, numero di unità per layer, funzioni di attivazione etc) della rete sono da ricercarsi empiricamente. Questo modello è stato realizzato e sperimentato nell'environment descritto nella prossima sezione.

3.2 Simulated Radio Waves via Raycasting techniques

Per validare questi modelli, è stato sviluppato un progetto su Python3.8 atto a simulare il comportamento di onde radio all'interno di un ambiente ostruito da ostacoli. Questi ultimi, nel caso scelto, sono rappresentati da segmenti di lunghezza parametrica sparsi nell'ambiente. Il simulatore sfrutta la tecnologia di *Ray Casting* alla sua base, e la sfrutta per simulare l'effetto di path loss, e per simulare l'eventuale collisione con gli ostacoli presenti. Preso un gateway la cui posizione è nota nel piano, è possibile effettuare una query per sapere la sua intensità percepita in un qualsiasi altro punto del piano. Nel caso non ci siano ostacoli presenti nel percorso che congiunge il gateway al punto richiesto, il raggio utilizza come fattore di attenuazione d_k la seguente formula:

$$d_k(x, y) = \frac{1}{||p_k - (x, y)||^2 * \gamma} \quad (3.7)$$

dove γ rappresenta il fattore di attenuazione del mezzo di propagazione. Quindi secondo la formula, l'intensità del segnale scende come il quadrato del raggio. Nel caso più complesso, può essere presente uno o più ostacoli nel percorso, e dunque è necessario gestire le eventuali collisioni. Per ogni collisione si tengono in considerazione due fattori principali: l'angolo di incidenza della collisione θ e il fattore di attenuazione dell'ostacolo $\beta_i \in \{0, 1\}$:

$$d_{coll}(x, y, i) = \beta_i \sin(\theta) \quad (3.8)$$

Per ottenere l'attenuazione complessiva dal gateway al primo ostacolo, si utilizza il prodotto tra $d_k(x_c, y_c)$ e $d_{coll}(x, y, i)$. Successivamente il raggio viene fatto ripartire dal punto di collisione verso il punto richiesto e, nel caso siano presenti altri ostacoli, il procedimento appena descritto viene reiterato. Per poter validare l'approccio

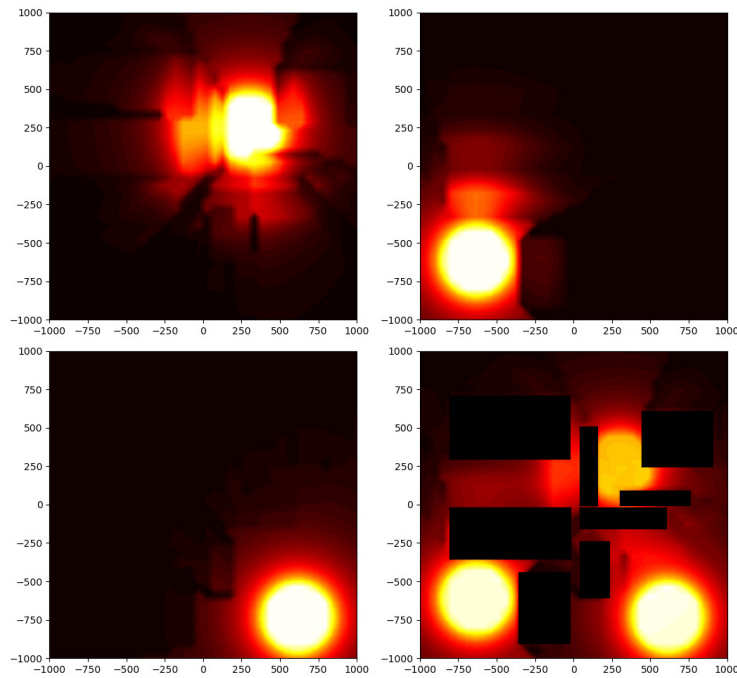


Figura 3.1. Un esempio di propagazione di 3 gateway posti all'interno di un ambiente urbano simulato. Ogni palazzo è composto da 4 ostacoli.

regressivo, è stato sviluppato un ambiente urbano in cui ogni ostacolo ha un fattore di attenuazione randomico distribuito uniformemente ($\beta_i \sim U(0, 1)$).

In base a precedenti esperimenti reali, è stato generato un dataset contenente

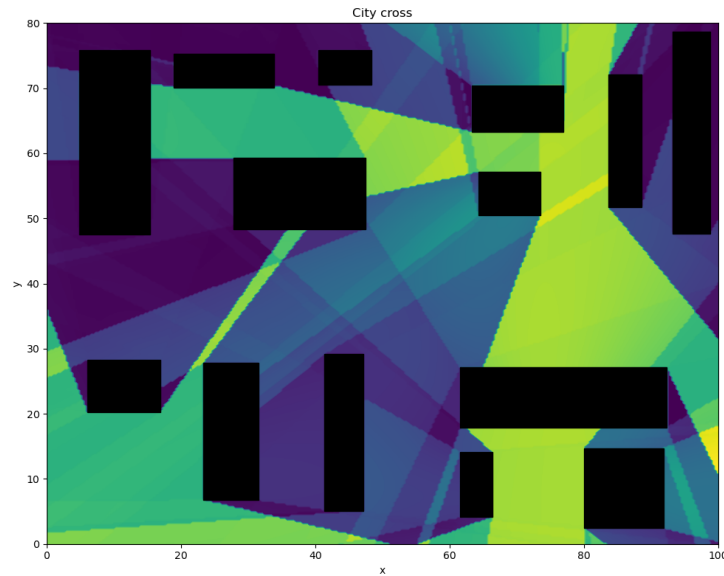


Figura 3.2. Simulazione nell'ambiente *cross00*. Il colore associato ad ogni punto rappresenta la somma delle intensità percepite dai 3 gateway.

all'incirca 150 campioni estratti manualmente, in cui ogni campione q_i è composto dal

vettore $(x_i, y_i, RSSI_1, RSSI_2, RSSI_3)$ dove $RSSI_k$ rappresenta l'intensità percepita dal k -esimo gateway. Attraverso questi campioni, sono state trainate tre reti neurali

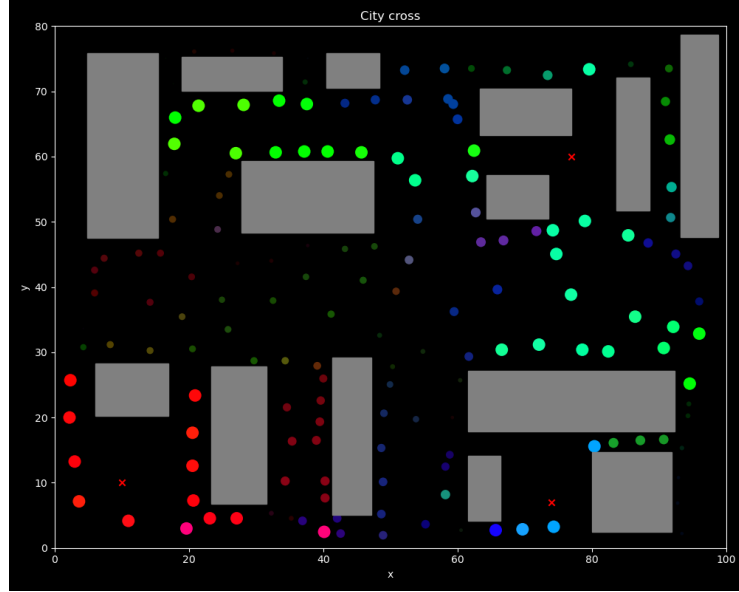


Figura 3.3. I campioni raccolti dall'ambiente *cross00*. Ad ogni campione è associato un colore RGB e una dimensione. Ogni canale è associato all'intensità di una antenna (Rosso per l'antenna in basso a sinistra, Verde per l'antenna in alto a destra, Blu per l'antenna in basso a destra) mentre la dimensione è proporzionale all'intensità complessiva misurata.

composte dal medesimo modello progettato con il framework *PyTorch*. Il modello è composto da una sequenza di layer lineari (Fully Connected) la cui dimensione per layer è riportata nel seguente vettore:

$$FCNN_{dim} = (2, 20, 20, 1) \quad (3.9)$$

I layer utilizzano funzioni di attivazioni diverse per poter ottenere in output, una distribuzione non lineare, infatti il primo ed il penultimo layer utilizzano una *ReLU* (REctified Linear Unit), il layer intermedio utilizza una Sigmoidale mentre il layer di uscita utilizza una funzione di attivazione lineare con l'obiettivo di non deformare l'output prodotto dalla rete. utilizzando diverse funzioni di attivazione è possibile ottenere predizioni molto diverse tra loro, ma la configurazione appena descritta pare restituire dei risultati promettenti. Il ciclo di training viene eseguito su 30000 epoche, in cui l'intero set di dati viene trasferito alla rete. Viene utilizzata come funzione di loss la *Mean Squared Error* (MSE) riportata nella seguente equazione:

$$MSE(x, y) = \frac{1}{N} \sum_{i=1}^N (y_i - g_{RSSI}^k(x_i))^2 \quad (3.10)$$

È stata scelta questa funzione di loss, in quanto si presta particolarmente per problemi di regressione multidimensionali. È stato scelto l'ottimizzatore *Adam* con un learning rate $\gamma = 2e^{-3}$. Al termine del training, tramite un campionamento a griglia sullo spazio preso in considerazione (una griglia di 80m×100m) sono stati testati gli approssimatori per i 3 gateway, ottenendo il risultato mostrato in figura 3.4.

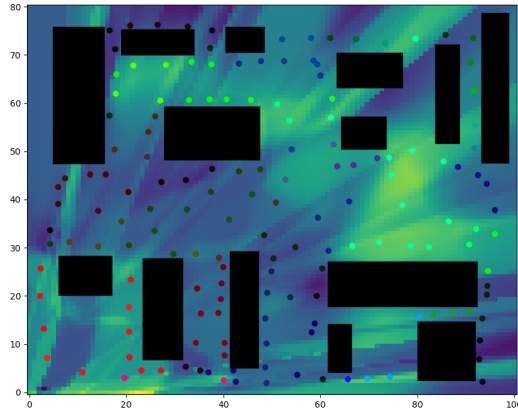


Figura 3.4. Predizione delle reti nell'ambiente *cross00*

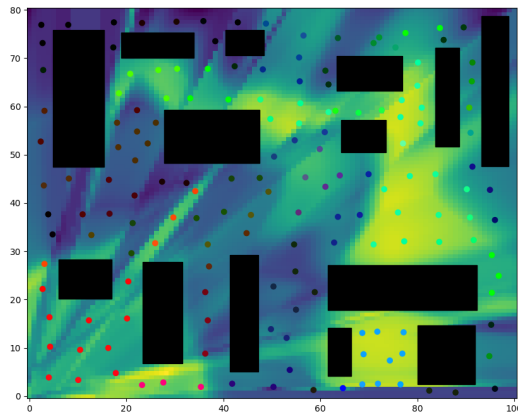


Figura 3.5. Predizione delle reti nell'ambiente *cross00* con un campionamento più uniforme

Notiamo dalla figura che la distribuzione ottenuta dalla rete, rispecchia i segnali percepiti durante il campionamento. Per validare ulteriormente questo risultato, sarà necessario effettuare un esperimento con la stack di localizzazione attiva, nel quale il dispositivo campionante, sarà mosso su una traiettoria predefinita, mentre per le predizioni verrà utilizzata la mappa appena generata. È importante notare che la qualità delle mappe, è fortemente caratterizzata dalla qualità dei campioni, infatti in figura 3.5 viene mostrata la predizione a fronte di un campionamento più grande di circa 200 samples, che sono distribuiti più uniformemente nello spazio libero dell'ambiente. Dunque per una futura campagna di sperimentazione, sarà necessario definire una buona metrica con cui valutare il posizionamento dei campioni.