

MONITORAGGIO SELETTIVO DI STAZIONI E TRATTE
FERROVIARIE IN BASE ALLA LORO ESPOSIZIONE ALLA
PERICOLOSITÀ DI FRANA: UN APPROCCIO GIS



CARUSO FEDERICA
DI SABATINO FABIO
GIANCOLA DAVIDE

INDICE

1	INTRODUZIONE	1
2	MATERIALI E METODI	4
2.1	Notazioni	4
2.2	Un metodo per classificare le stazioni ferroviarie	16
2.2.1	Metodo di calcolo originario	16
2.2.2	Metodo di calcolo ex novo	18
2.2.3	L'algoritmo	21
2.3	Un metodo per classificare le linee ferroviarie	28
2.3.1	Metodo di calcolo	28
3	UN CASO DI STUDIO: L'ABRUZZO	31
3.1	Territorio	31
3.2	Rete Ferroviaria	32
3.2.1	Stazioni Ferroviarie	32
3.2.2	Linee Ferroviarie	33
3.3	Zone	34
3.4	Base di dati spaziale	35
3.4.1	Progettazione concettuale	35
3.4.2	Progettazione logica	39
4	RISULTATI E LORO DISCUSSIONE	40
4.1	Ranking e Valutazione delle stazioni ferroviarie	40
4.1.1	Analisi critica del Nuovo Metodo Proposto	41
4.1.2	Discussione dei risultati ottenuti	47
4.2	Ranking e Valutazione delle Linee Ferroviarie	49
4.2.1	Linea Bologna - Bari	51
4.2.2	Linea Ortona - Crocetta	53
4.2.3	Linea Marina di San Vito - Castel di Sangro	54
4.2.4	Linea Pescara - Roma	56
4.2.5	Linea Roccasecca - Avezzano	57
4.2.6	Linea Archi Stazione - Atessa	58
4.2.7	Linea Sulmona - Carpinone	60
4.2.8	Linea Sulmona - L'Aquila - Rieti	62
4.2.9	Linea Giulianova - Teramo	64
4.2.10	Discussione dei risultati ottenuti	65
5	CONSIDERAZIONI FINALI E PROPOSTE PER LAVORI FUTURI	67
5.1	Proposte per lavori futuri sul metodo di ranking delle stazioni ferroviarie	67
5.2	Proposte per lavori futuri sul metodo di ranking delle tratte ferroviarie	67
5.3	Considerazioni sui dataset	67
5.4	Considerazioni finali	68

I APPENDICI	70
A APPENDICE A	71
A.1 Dataset delle zone sviluppato dal gruppo	71
A.1.1 Processamento dei poligoni interni	72
A.1.2 Processamento dei poligoni lungo il bordo	73
A.1.3 Conclusioni	75
A.2 QuantumGIS	75
B RISULTATI DI LABORATORIO	77
B.1 Classificazione completa delle stazioni tramite il NMC	77
B.2 Classificazione completa delle linee ferroviarie tramite il NMCE	82
B.2.1 Bologna-Bari	82
B.2.2 Ortona-Crocetta	83
B.2.3 Marina di San Vito - Castel di Sangro	83
B.2.4 Roma-Pescara	84
B.2.5 Avezzano-Roccasecca	85
B.2.6 Archi stazione-Atessa	85
B.2.7 Sulmona-Carpinone	86
B.2.8 Teramo-Giulianova	86
B.2.9 Rieti- L'Aquila-Sulmona	87
B.3 Classificazione ufficiale delle stazioni utilizzata per la validazione	88
C UDF	94
C.1 UDF del NMC	94
C.1.1 esposizione	94
C.1.2 quotaMediaZoneVicine	96
C.1.3 stimaQuotaStazione	97
C.1.4 dislivelloPiuVicino	99
C.2 UDF del NMCE	101
C.2.1 esposizione_linee	101
C.2.2 semplificaLinee	102
C.2.3 campiona	102
C.2.4 dividiInSegmenti	103
C.2.5 expSegmenti	104

ELENCO DELLE FIGURE

Figura 1	La nostra <i>GeoArea</i> : La regione Abruzzo	4
Figura 2	Generico elemento z_k dell'insieme Z	5
Figura 3	Generico elemento b_i dell'insieme B	6
Figura 4	(a) Generico elemento e_j dell'insieme; (b)Esempio di curve di livello per un certo territorio	7
Figura 5	Esempio di buffer_i costruito su una generica b_i	7
Figura 6	$V=\{v_1, v_2, v_3, v_4, v_5\}$	8
Figura 7	$P=\{p_2, p_3\}$	8
Figura 8	$T=\{t_1, t_2, t_3, t_4, t_5\}$	9
Figura 9	Esempio di una v_γ con tre curve di livello	9
Figura 13	Esempio di angolo azimuth per un certo punto	12
Figura 14	Esempio di vettore direzionale per un certo edificio b_i	13
Figura 15	$Q = \{q_1, q_2, q_3\}$	13
Figura 16	Esempio di un <i>trend=salita</i> nel caso in cui la stazione si trova ad una quota minore rispetto la curva di livello q_1	14
Figura 17	Esempio di un <i>trend = salita</i> nel caso in cui la stazione si trova alla stessa quota della curva di livello q_1	14
Figura 18	Esempio di un <i>trend=discesa</i> nel caso in cui la stazione si trova a quota maggiore rispetto la della curva di livello q_1	15
Figura 19	Esempio di un <i>trend=discesa</i> nel caso in cui la stazione si trova a quota maggiore rispetto la della curva di livello q_1	15
Figura 20	dove	16
Figura 21	Esempio di un edificio b_i all'interno di una zona z_k	17
Figura 22	Esempio di un edificio b_i esterno alla zona z_k	18
Figura 23	Esempio di un edificio all'interno di una zona v_γ	19
Figura 24	Esempio di una v_γ con $h_{\text{media}}_{v_\gamma}$ elevata la cui intersezione con il buffer_i risulta essere un'area piccola	20
Figura 25	Esempio di rappresentazione geometrica di una linea ferroviaria ove b_i e b_{i+1} sono due stazioni	28
Figura 26	Esempio di campionamento di una linea ferroviaria	29
Figura 27	dove	29
Figura 28	dove	30
Figura 29	La regione Abruzzo	32
Figura 30	Stazioni ferroviarie presenti in Abruzzo	33
Figura 31	Linee ferroviarie presenti in Abruzzo	34
Figura 32	Partizionamento completo della regione Abruzzo	34
Figura 33	Diagramma E-R della base di dati	36
Figura 34	Schema logico	39

Figura 37	Visione d'insieme della classificazione in fasce di rischio dei segmenti della rete ferroviaria abruzzese	50
Figura 38	Visione d'insieme della classificazione in fasce di rischio dei segmenti della linea Bologna - Bari	51
Figura 39	Visione d'insieme della classificazione in fasce di rischio dei segmenti della linea Ortona - Crocetta	53
Figura 40	Visione d'insieme della classificazione in fasce di rischio dei segmenti della linea Marina di San Vito - Castel di Sangro	55
Figura 41	Visione d'insieme della classificazione in fasce di rischio dei segmenti della linea Pescara - Roma	56
Figura 42	Visione d'insieme della classificazione in fasce di rischio dei segmenti della linea Roccasecca - Avezzano	58
Figura 43	Visione d'insieme della classificazione in fasce di rischio dei segmenti della linea Archi Stazione - Atessa	59
Figura 44	Visione d'insieme della classificazione in fasce di rischio dei segmenti della linea Sulmona - Carpinone	61
Figura 45	Visione d'insieme della classificazione in fasce di rischio dei segmenti della linea Sulmona - L'Aquila - Rieti	62
Figura 46	Visione d'insieme della classificazione in fasce di rischio dei segmenti della linea Giulianova - Teramo	64
Figura 47	Vista 2D (a) e 3D (b) della stazione di Aversa e della linea ferroviaria che l'attraversa	65
Figura 48	Vista 2D (a) e 3D (b) della tratta Pescara - Roma dal km 154 al 157	66
Figura 49	Istantanea di GoogleMaps della galleria <i>Colli di Monte Bove</i> . . .	66
Figura 50	Stazione di Archi	68
Figura 51	Stazione di Perano	68
Figura 52	Stazione di Pettorano	68
Figura 53	Stazione di Civitaluparella	68
Figura 54	Visione d'insieme delle 24142 zone generate	71
Figura 55	Particolare delle zone in prossimità del <i>boundary</i>	72

ABSTRACT

Short summary of the contents...a great guide by Kent Beck how to write good abstracts can be found here:

<https://plg.uwaterloo.ca/~migod/research/beck00PSLA.html>

INTRODUZIONE

La mobilità delle persone e delle merci sono una componente essenziale del mercato interno dell'Unione Europea (UE) ed è di fondamentale importanza garantire la sua fattibilità al fine di salvaguardare la crescita economica. La rete ferroviaria ha un ruolo strategico in questo contesto almeno sotto due punti di vista:

1. Offre un impulso rilevante all'integrazione e all'efficienza dell'economia dell'UE.
2. Facilita il libero movimento delle persone e delle merci garantendo una modalità di trasporto efficiente e sostenibile dal punto di vista ambientale. Secondo l'Agenzia europea dell'ambiente, le emissioni di CO₂ provenienti dal trasporto ferroviario sono 3,5 volte inferiori, per tonnellata-chilometro, a quelle prodotte dal trasporto su strada. Di conseguenza, promuovere modalità di trasporto come quello su rotaia, piuttosto che su strada, permetterebbe all'UE di essere meno dipendente dall'importazione di petrolio e di ridurre l'inquinamento.

Alla luce di queste considerazioni preliminari non sorprende che la sicurezza assuma un rilievo crescente in questo contesto; Una testimonianza di ciò è il regolamento su un metodo comune di sicurezza (CSM) per la valutazione del rischio e l'accertamento, rilasciato negli ultimi anni dall'agenzia ferroviaria europea (ERA) e dalla Commissione Europea [Regulation EU, No.402/2013, 2015]. Un recente regolamento [GE/GN8642, 2014] pubblicato dalla Rail Safety and Standards Board Ltd (<http://www.rssb.co.uk/>) definisce "*un pericolo, come una condizione che potrebbe portare a un incidente*". A sua volta, la direttiva sulla sicurezza [2004/49/CE] del Parlamento europeo definisce un incidente come: "*per 'incidente' si intende un evento improvviso indesiderato e non intenzionale o specifica catena di siffatti eventi aventi conseguenze dannose. Gli incidenti sono suddivisi nelle seguenti categorie: collisioni, deragliamenti, incidenti ai passaggi a livello, incidenti a persone causati da materiale rotolante in movimento, incendi ed altri.*"

Nel tempo sono stati sviluppati molti metodi per predire, prevenire e ridurre il numero di incidenti e per aumentare la sicurezza nel contesto ferroviario. Inoltre è fondamentale sottolineare il fatto che le tratte e le relative stazioni insistono su un territorio esposto, tra gli altri, al rischio idrogeologico. Di riflesso anche la rete ferroviaria e le relative stazioni che su di esso insistono lo sono. I deragliamenti dei treni sono il rischio principale di incidenti. [Lloyd et al., 2001], [Manning et al., 2008], [He et al., 2011] sono esempi di studi che hanno affrontato la questione. In tali studi si fa uso di metodi matematici classici. Ad esempio, [He et al. (2011)] fanno ricorso alla

Principal Component Analysis per identificare un numero limitato di fattori che devono essere presi in considerazione per studiare la stabilità dei pendii rocciosi. Studi teorici come quelli appena citati sono molto interessanti perché aiutano a comprendere la complessità del problema, fornendo delle informazioni globali circa le variabili in gioco e come esse sono correlate, ma hanno il limite di non essere direttamente traducibili in piani di lavoro attuabili da parte di coloro che hanno responsabilità di garantire la sicurezza delle tratte ferroviarie e le relative stazioni.

Nel nostro documento si adotta un approccio differente, basato su metodi proposti nell'ambito della Geographical Information Science e riferiti allo studio di relazioni topologiche tra caratteristiche geografiche. Come vedremo, facendo ricorso a tali metodi, possibile identificare quali sono le potenziali pericolosità delle tratte ferroviarie e delle relative stazioni determinate dalle caratteristiche morfologiche e litologiche del terreno dove sono situate, dalla sua pendenza nonché dall'uso dello stesso (prevalentemente abitativo, prevalentemente ad uso coltivazione, ecc).

Le frane costituiscono uno dei più importanti rischi naturali in molti paesi in tutto il mondo, [Brabb e Harrod, 1989]. Questo è il caso dell'Italia, dove le frane sono diffuse e provocano notevoli danni e decessi ogni anno, [Guzzetti, Stark, e Salvati, 2005], [Trigila et al., 2010]. Secondo il risultato di un recente studio effettuato da [Jaedicke et al. (2014)], “*l’Italia ha il più alto numero di persone esposte al pericolo di frana tra i paesi europei.*”

Mantenendosi ad un alto livello di astrazione quello che si può anticipare del metodo che ci si accinge a presentare è che esso consente di fare delle valutazioni circa le potenziali pericolosità delle tratte ferroviarie e delle relative stazioni che insistono sul territorio oggetto di studio. Ciò consente di restituire agli utenti finali informazioni dettagliate per la localizzazione delle porzioni di tratte ferroviarie e delle stazioni ove prioritariamente è opportuno fare dei controlli periodici. La strategia proposta quindi supera i limiti posti dai controlli periodici sull'intero percorso ferroviario e su tutte le stazioni situate sul territorio, i quali risultano essere lunghi e, di conseguenza, costosi e non sempre utili come sostenuto da [Sadler et al., (2016)]. Utilizzando tale metodo sarà possibile individuare quali sono le porzioni di tratte ferroviarie e le stazioni le cui misure di controllo possono essere ridotte in modo sicuro, permettendo al settore ferroviario significativi risparmi di costi a favore del potenziamento dei controlli di porzioni di tratte ferroviarie e stazioni che hanno una pericolosità potenziale maggiore.

In sintesi, obiettivo dello studio metodologico-sperimentale descritto in questo documento è sviluppare una metodologia scientificamente robusta per l'identificazione dei segmenti ferroviari ad alto rischio e che sia di supporto nella definizione di una strategia di monitoraggio selettiva dello stato di sicurezza per una categoria di beni strategica per un Paese, ovvero la sua rete ferroviaria, rispetto all'esposizione della

stessa ai movimenti franosi.

Le tratte ferroviarie e le relative stazioni sono dunque i due target da proteggere e data la loro diversa connotazione spaziale che le descrive vanno trattate ed analizzate separatamente. Infatti, le tratte sono modellabili con la geometria *linea*, mentre le stazioni sono modellabili con la geometria *punto*.

Lo strumento software da noi sviluppato è basato sull'uso di un Geo-DB arricchito con un ampio numero di User Defined Functions (UDF), tramite le quali è possibile effettuare delle elaborazioni complesse scrivendo delle queries SQL semplici alla portata della maggior parte dei tecnici che lavorano nel settore della sicurezza ferroviaria. In dettaglio, il sistema informatico è in grado di fornire quanto segue:

1. Circa le stazioni, una classifica delle stazioni ferroviarie costruita in relazione al loro valore di esposizione al rischio di frane e una relativa classificazione in fasce di rischio delle stesse;
2. Circa le linee ferroviarie che sono presenti sul territorio, una classifica delle tratte ferroviarie costruita in relazione al loro valore di esposizione al rischio di frane e una relativa classificazione in fasce di rischio delle stesse.

Il documento è organizzato come segue: nel Capitolo 2 verrà mostrata una panoramica degli strumenti e gli algoritmi utilizzati per valutare l'esposizione al rischio di frana delle stazioni ferroviarie dislocate sul territorio. Nel Capitolo 3 verrà discussa la situazione attuale del caso di studio, ove il metodo proposto sarà sperimentato con riferimento alla rete ferroviaria italiana dislocata all'interno del territorio della regione Abruzzo e quindi, verranno presentati i datasets di input. Nel Capitolo 4 verranno presentati e discussi i risultati ottenuti dall'applicazione dei metodi discussi nel Capitolo 2, in particolare verranno presentate anche i punti di forza e le criticità di tali applicazioni. Il Capitolo 5 invece presenterà quelle che sono le considerazioni finali circa il lavoro realizzato e presenterà alcuni possibili spunti per eventuali lavori futuri.

Come già preannunciato, il metodo proposto sarà sperimentato con riferimento alla rete ferroviaria Italiana dislocata all'interno del territorio della regione Abruzzo (Italia Centrale).

MATERIALI E METODI

2.1 NOTAZIONI

Da qui in poi useremo le seguenti notazioni:

1. *GeoArea*: Porzione di territorio di interesse per lo studio. La *GeoArea* può coincidere con un comune, una provincia, una regione, una nazione o un intero continente. Nel nostro caso di studio si tratta della regione Abruzzo (Fig. 1).



Figura 1: La nostra *GeoArea*: La regione Abruzzo

La *GeoArea* viene descritta dalla seguente tupla: <gid, geom>.

- *gid*: Identificativo numerico univoco del territorio.
 - *geom*: Rappresentazione geometrica del territorio (*MultiPolygon*).
2. $Z = \{z_k, k=1,2,\dots\}$ dove z_k denota una zona k -esima della *GeoArea*, ognuna caratterizzata da un insieme di informazioni territoriali. L'insieme Z è una partizione della *GeoArea*, ovvero è una collezione di elementi z_k , non nulli, tali da avere intersezione vuota a due a due e tali che la loro unione coincida con l'intera *GeoArea*, per maggiore chiarezza si rimanda alla Fig. 2.



Figura 2: Generico elemento z_k dell'insieme Z

Il generico elemento di Z (i.e. z_k) è definito mediante la tupla: $\langle \text{gid}, S_{z_k}, \text{geom}, \text{avgElevation} \rangle$.

- gid : Identificativo numerico univoco dell'elemento z_k .
- S_{z_k} : Valore numerico che quantifica la pericolosità di frana dell'elemento z_k .
- geom : Rappresentazione geometrica del *boundary* dell'elemento z_k .
- avgElevation : Altitudine media dell'elemento z_k , ottenuta mediante una media ponderata delle altitudini delle curve di livello che intersecano, almeno in parte, l'elemento z_k . Il modo in cui verrà calcolato questo valore verrà descritto nel dettaglio nella notazione successiva di Δh poiché bisogna prima introdurre ulteriori concetti.

Si noti che z_k è una notazione sovraccaricata dato che rappresenta sia il *gid* della zona che la sua geometria.

3. $B = \{ b_i, i=1,2,\dots \}$ dove b_i indica l'edificio della stazione (B sta per *Station Building*) contenuta nel *boundary* della *GeoArea*. Di questo edificio non teniamo conto della sua conformazione architettonica, della sua altezza o del numero di piani della struttura poiché spesso sono delle informazioni che difficilmente sono reperibili ma bensì consideriamo la sua posizione geografica, espressa da una coppia di coordinate, informazione spesso nota. Un esempio di un generico elemento b_i è rappresentato in Fig. 3.

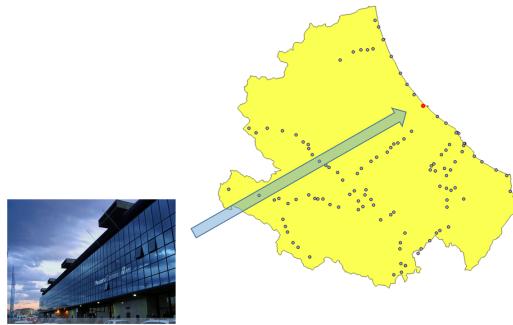


Figura 3: Generico elemento b_i dell'insieme B

Il generico elemento di B (i.e. b_i) è definito mediante la tupla: $\langle gid, name, geom, Exp_b_i, elevation \rangle$.

- *gid*: Identificativo numerico univoco dell'elemento b_i .
- *name*: Nome testuale dell'i-esimo elemento dell'insieme B , espresso tramite una stringa.
- *geom*: Rappresentazione geometrica (*Point*) dell'elemento b_i .
- *Exp_b_i*: Grado di esposizione al pericolo di frana del generico elemento b_i , determinato da tutte le z_k in Z .
- *elevation*: Altitudine espressa in metri a cui si trova il generico elemento b_i .

Si noti che b_i è una notazione sovraccaricata dato che rappresenta sia il *gid* della stazione ferroviaria che la sua geometria.

4. $E = \{ e_j, j=1,2,\dots \}$ dove e_j denota una curva di livello contenuta almeno in parte nel *boundary* della *GeoArea*. In geografia, con particolare riguardo alla cartografia, la curva di livello è quella curva che unisce punti con uguale quota (Fig. 4b), ovvero uguale distanza verticale dal piano di riferimento al quale è stato attribuito quota zero; se sono sopra il livello del mare si chiameranno *isoipse* (dal greco í̄sos = "uguale" e hýpsos = "altezza") mentre al contrario *isobate* (dal greco í̄sos = "uguale" e báthos = "profondità"). In Fig. 4a è possibile vedere un esempio di e_j . Ogni elemento dell'insieme E (i.e. e_j) è definito mediante la tupla: $\langle gid, elevation, geom \rangle$.

- *gid*: Identificativo numerico univoco dell'elemento e_j .
- *elevation*: Valore numerico che rappresenta la quota espressa in metri dell'elemento e_j .
- *geom*: Rappresentazione geometrica (*MultiLineString*) dell'elemento e_j .

Si noti che e_j è una notazione sovraccaricata dato che rappresenta sia il *gid* della curva di livello che la sua geometria.

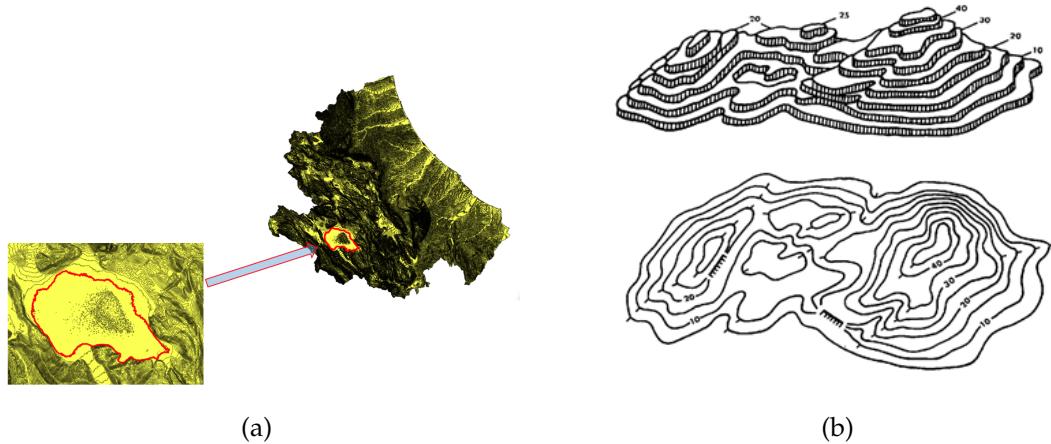


Figura 4: (a) Generico elemento e_j dell'insieme; (b)Esempio di curve di livello per un certo territorio

5. $buffer_i = buffer(b_i, r)$ rappresenta una geometria costituita da tutti i punti la cui distanza dal punto b_i risulta essere minore o uguale del raggio r . Nella figura 5 è possibile vedere un esempio di $buffer_i$.

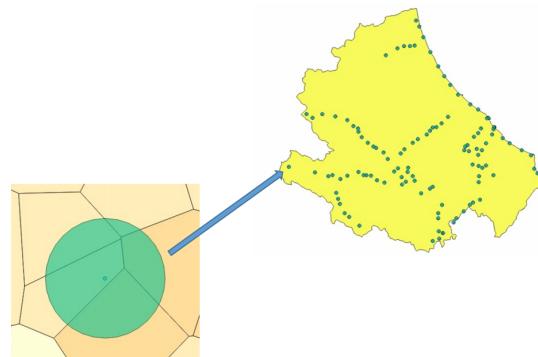


Figura 5: Esempio di $buffer_i$ costruito su una generica b_i

6. $V = \{v_\gamma \ (\gamma=1,2,\dots) := z_k \in Z \mid \text{buffer}_i \cap z_k \neq \emptyset \}$ è l'insieme di tutte le z_k contenute nella *GeoArea* che intersecano almeno in parte buffer_i . La cardinalità di tale insieme è pari a N . Un esempio di insieme V è osservabile nella fig. 6.



Figura 6: $V = \{v_1, v_2, v_3, v_4, v_5\}$

7. $P = \{p_m \ (m=1,2,\dots) := \text{buffer}_i \cap e_j \neq \emptyset, \forall e_j \in E\}$ è l'insieme dei segmenti delle curve di livello, determinati dall'intersezione tra le curve di livello contenute nell'insieme E e il buffer_i . In Fig. 7 è riportato un esempio di insieme P .

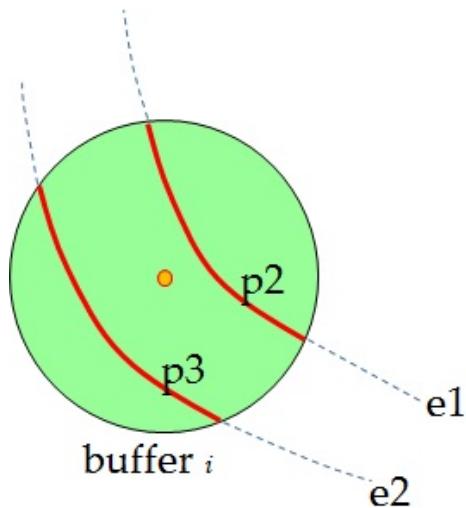


Figura 7: $P = \{p_2, p_3\}$

8. $T = \{t_f \mid f=1,2,\dots\} := v_\gamma \cap e_j \neq \emptyset, \forall e_j \in E, \forall v_\gamma \in V\}$ è l'insieme dei segmenti delle curve di livello, determinati dall'intersezione tra le curve di livello contenute nell'insieme E e le zone v_γ contenute nell'insieme V . In Fig. 8 è possibile osservare un esempio di insieme T .

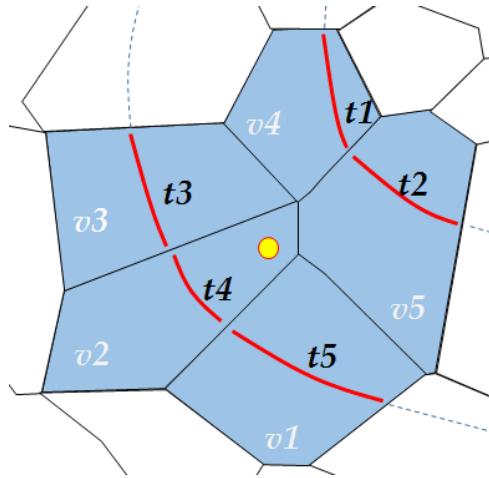


Figura 8: $T = \{t_1, t_2, t_3, t_4, t_5\}$

9. Δh è così definito:

$$\Delta h = |h_{stazione} - h_{media}_{v_\gamma}| \quad (1)$$

Dove:

- $h_{stazione}$ indica l'*elevation* dell'edificio b_i
- $h_{media}_{v_\gamma}$ indica l'*avgElevation* media di v_γ

Il metodo di calcolo dell'*elevation* per gli edifici e le zone è diverso. Infatti qui di seguito descriveremo le diverse modalità di calcolo:

- a) $h_{media}_{v_\gamma}$: l'*elevation* delle v_γ viene calcolata effettuando la media aritmetica dell'*elevation* delle curve di livello che intersecano v_γ .

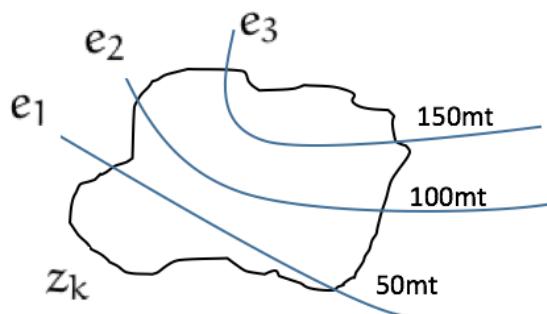


Figura 9: Esempio di una v_γ con tre curve di livello

Ad esempio, in Fig. 9 la quota media di v_Y risulta essere:
 $\frac{50 + 100 + 150}{3} = 100\text{mt}$

- b) *h_stazione*: il calcolo dell'*elevation* per gli edifici è più sofisticato ma permette di ottenere valori molto accurati. Le possibili situazioni che si possono presentare sono tre e verranno analizzate qui di seguito:
- edificio racchiuso da un'unica curva di livello (Fig. 10) oppure in prossimità del confine della *GeoArea* (Fig. 11). Si definisce un edificio di questo tipo come edificio in posizione "atipica".

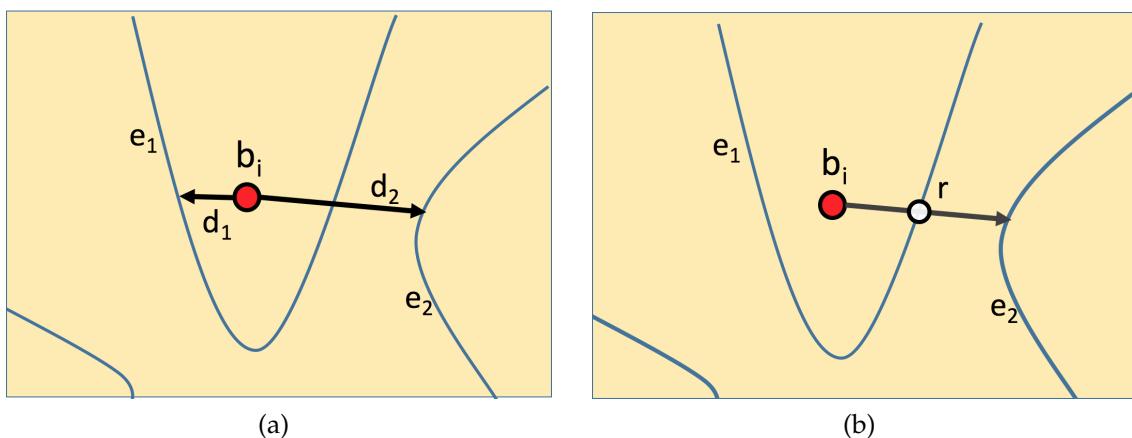


Figura 10: Edificio racchiuso da un'unica curva di livello.

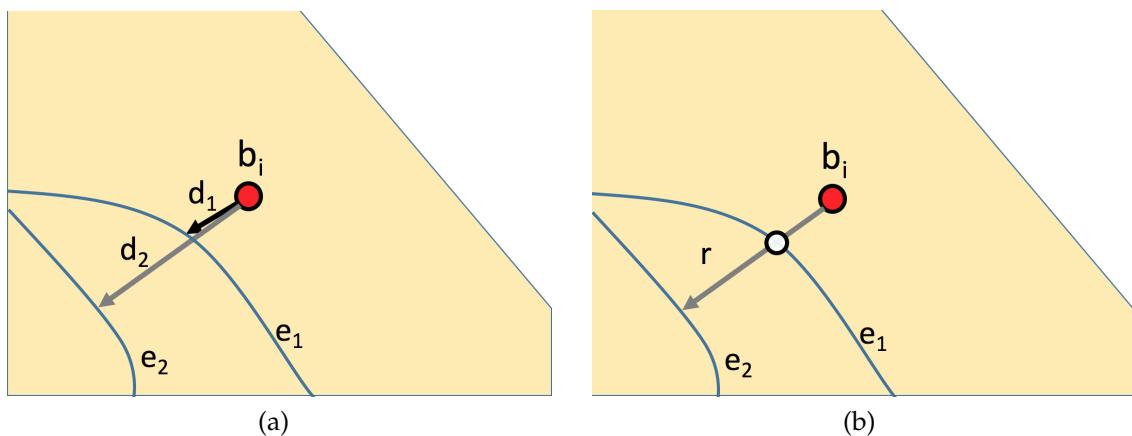


Figura 11: Edificio in prossimità del confine della *GeoArea*.

Entrambi i casi vengono rilevati nel medesimo modo, ovvero tracciando la semiretta avente come origine l'edificio b_i (pallino rosso) e passante per la curva e_2 . Se la semiretta interseca anche e_1 , allora ci troviamo in una delle situazioni esplicate in Fig. 10 (b) e in Fig. 11 (b). Quindi *h_stazione* equivale all'*elevation* della curva di livello e_1 , supponendo che $d_1 < d_2$.

- Edificio circondato da due curve di livello come in Fig.12. Si definisce un edificio di questo tipo come edificio in posizione "tipica".

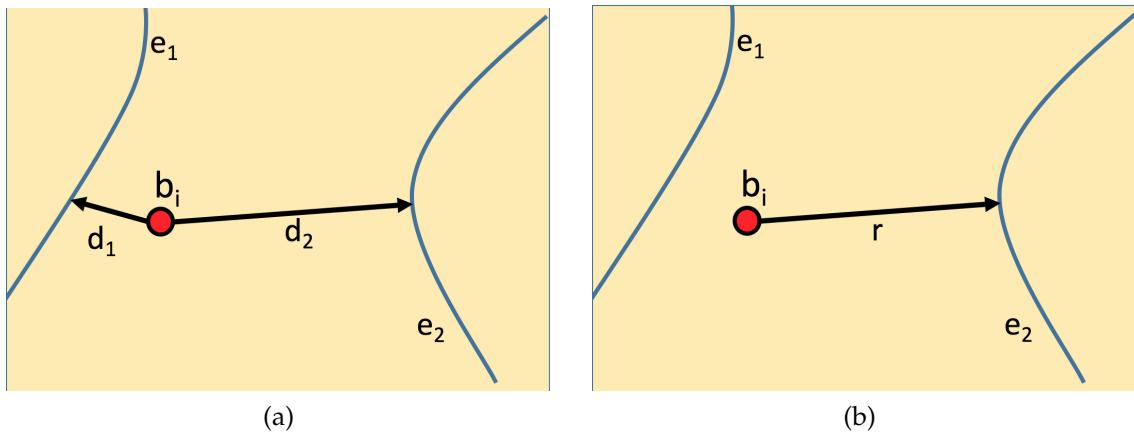


Figura 12: Edificio in prossimità del confine della *GeoArea*.

Dalla Fig. 12 (b) si evince che, a differenza delle situazioni esplicate dalle Fig. 10 (b) e Fig. 11 (b), la semiretta avente origine in b_i e passante per e_2 non interseca la curva di livello e_1 . Dunque h_{stazione} è così definito:

$$h_{stazione} = \frac{(h_{e1} \times d_2) \times (h_{e2} \times d_1)}{d_1 + d_2} \quad (2)$$

Dove:

- h_{-e_j} indica l'*elevation* della curva di livello e_j , con $j = 1, 2$
 - d_j indica la distanza tra l'edificio b_i e la curva di livello e_j con $j = 1, 2$

L'Eq.2 è una media ponderata delle quote delle due curve di livello più vicine all'edificio b_i . Al numeratore le distanze vengono utilizzate come pesi della media, affinché la quota della curva e_1 , più vicina all'edificio b_i , risulti più determinante nel calcolo della $h_{stazione}$ rispetto alla quota della curva e_2 .

10. *Azimuth*: L'azimuth è la coordinata orizzontale angolare espressa dall'arco d'ortodromia della sfera celeste che si forma partendo convenzionalmente dal punto cardinale nord fino all'oggetto di osservazione, muovendosi in senso orario verso est, quindi a sud e a ovest, fino a tornare al punto di inizio a nord (cioè un angolo giro, 360° sessagesimali); la coordinata azimutale quindi, verrà espressa in gradi angolari sessagesimali/minuti/secondi, oppure in radianti, e avrà sempre un valore numerico positivo. Un punto nel cielo che si trovi esattamente a nord (il polo celeste dell'emisfero boreale) avrà una coordinata azimutale di 0° (o 360°), se invece si trova esattamente a est sarà di 90° , esattamente a sud di 180° , esattamente a ovest di 270° , ecc. Si propone un esempio in Fig.13.



Figura 13: Esempio di angolo azimuth per un certo punto

11. *vettoreDirezionale*: il vettore direzionale viene così costruito:

- a) trova l'*Azimuth* tra b_i e il punto più vicino della p_m più vicina.
- b) Crea il punto A , le cui coordinate (x_a, y_a) sono le coordinate di b_i traslate di Δx e Δy .

Dove:

- $\Delta x = \cos(\text{Azimuth}) \times (r + 1)$
- $\Delta y = \sin(\text{Azimuth}) \times (r + 1)$

con r pari al raggio del $buffer_i$.

- c) crea infine il *vettoreDirezionale*, ovvero il vettore avente come origine b_i e come punto finale il punto A sopra definito.

Si propone un esempio esplicativo in Fig. 14.



Figura 14: Esempio di vettore direzionale per un certo edificio b_i

12. $Q = \{ q_w \ (w=1,2,\dots,\beta) \} := p_m \in P \mid (p_m \cap \text{vettoreDirezionale}) \neq \emptyset \mid d_1 < d_2 < \dots < d_\beta \}$.

Dove:

- d_w è la distanza tra b_i e il punto $(p_m \cap \text{vettoreDirezionale})$

Per costruzione i pedici stabiliscono un ordinamento crescente degli elementi dell'insieme Q rispetto la distanza appena definita.

La Fig. 15 propone un esempio esplicativo basato sulla Fig.14. Come si vede sono state escluse dall'insieme Q le $p_m \mid (p_m \cap \text{vettoreDirezionale} = \emptyset)$

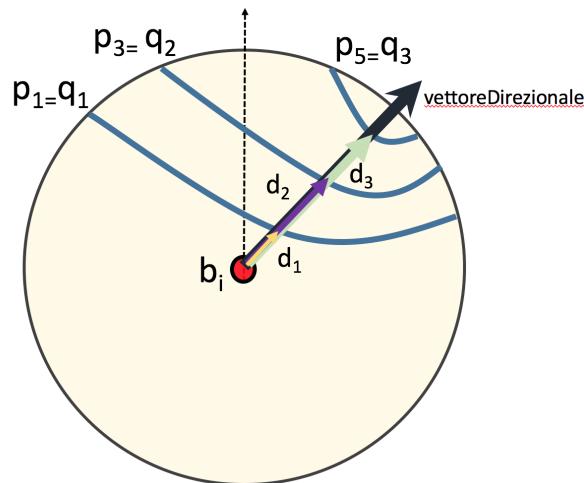


Figura 15: $Q = \{ q_1, q_2, q_3 \}$

13. *trend*: Il trend rappresenta l'andamento del territorio nelle immediate vicinanze di b_i in direzione del vettore *Direzionale* e può assumere uno dei seguenti valori {*salita*, *discesa*}, in base alle seguenti condizioni:

- $trend = \text{salita}$ se $\text{elevation}_{q_1} > h_{\text{stazione}}$.

Esempio in Fig.16



Figura 16: Esempio di un *trend=salita* nel caso in cui la stazione si trova ad una quota minore rispetto la curva di livello q_1

- $trend = \text{salita}$ se $\text{elevation}_{q_1} = h_{\text{stazione}} \wedge \text{elevation}_{q_2} > \text{elevation}_{q_1}$.
Esempio in Fig.17.



Figura 17: Esempio di un *trend = salita* nel caso in cui la stazione si trova alla stessa quota della curva di livello q_1

- $trend = \text{discesa}$ se $\text{elevation}_{q_1} < h_{\text{stazione}}$.
Esempio in Fig. 18

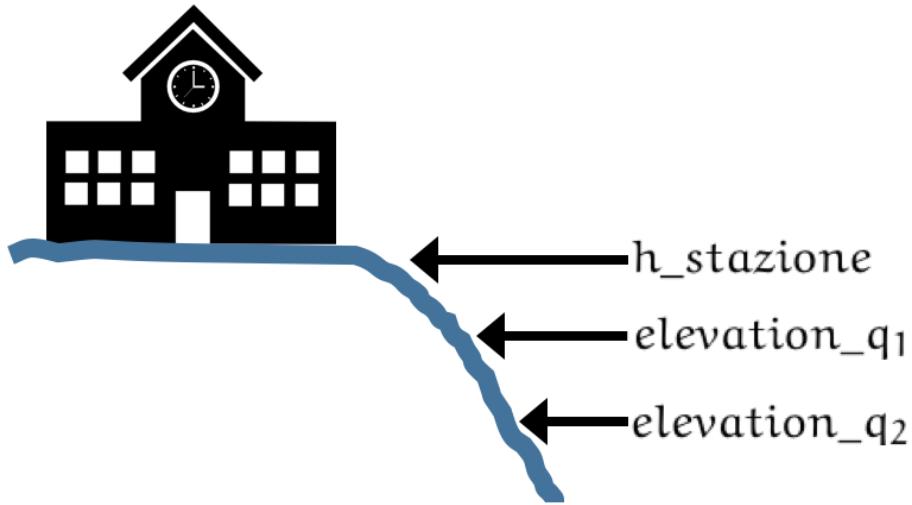


Figura 18: Esempio di un $trend=\text{discesa}$ nel caso in cui la stazione si trova a quota maggiore rispetto la della curva di livello q_1

- $trend = \text{discesa}$ se $\text{elevation}_{q_1} = h_{\text{stazione}} \wedge \text{elevation}_{q_2} < \text{elevation}_{q_1}$.
Esempio in Fig. 19



Figura 19: Esempio di un $trend=\text{discesa}$ nel caso in cui la stazione si trova a quota maggiore rispetto la della curva di livello q_1

14. *situazione Monotonica*: una situazione monotonica è garantita fino ad un certo $q_{\bar{w}}$ se:

- è garantita $\forall w < \bar{w}$
- la differenza di elevation tra $q_{\bar{w}}$ e $q_{\bar{w}-1}$ conferma l'andamento del $trend$.
Ovvero:

$$\begin{cases} \text{elevation}_{q_{\bar{w}}} < \text{elevation}_{q_{\bar{w}-1}}, & \text{se } trend = \text{discesa} \\ \text{elevation}_{q_{\bar{w}}} > \text{elevation}_{q_{\bar{w}-1}}, & \text{se } trend = \text{salita} \end{cases} \quad (3)$$

15. *pendenza* è una stima dell'andamento del territorio vicino l'edificio b_i lungo il suo vettore *Direzionale* associata a q_w . Dal punto di vista geometrico la *pendenza* rappresenta il coefficiente angolare tra due punti. In un caso tra una stazione b_i e il punto ottenuto dall'intersezione tra il *vettoreDirezionale* e la curva di livello q_1 . Nell'altro caso tra i punti ottenuti dall'intersezione tra il *vettoreDirezionale* e le curve q_w e q_{w+1} . In entrambi i casi il coefficiente viene calcolato rispetto al sistema di riferimento della *GeoArea*. Come mostrato in Fig. 20



Figura 20: Esempio esplicativo di *pendenza*. Δy corrisponde al numeratore e Δx al denominatore dell'Eq.4, per $w = 1$ subFig.20 (a) e per $w \geq 2$ subFig.20 (b)

Dal punto di vista matematico l'equazione per determinare la *pendenza* è la seguente:

$$\begin{cases} \frac{|h_{\text{stazione}} - \text{elevation}_{q_w}|}{d(b_i, (\text{vettoreDirezionale} \cap q_w))} & \text{se } w=1 \\ \frac{|\text{elevation}_{q_w} - \text{elevation}_{q_{w-1}}|}{d((\text{vettoreDirezionale} \cap q_w), (\text{vettoreDirezionale} \cap q_{w-1}))} & \text{se } w \geq 2 \end{cases} \quad (4)$$

16. slope_i : E' la somma aritmetica delle *pendenze* di tutti gli elementi di Q che garantiscono una *situazione Monotonica*.

2.2 UN METODO PER CLASSIFICARE LE STAZIONI FERROVIARIE

2.2.1 *Metodo di calcolo originario*

Prima di procedere con l'ideazione di un metodo di calcolo, si è pensato di effettuare un'analisi critica e dettagliata del metodo originario, frutto del lavoro di altri studenti negli anni passati. Supponiamo di avere quindi una *GeoArea* partizionata in zone diverse e indicate con z_k , ognuna delle quali ha associato un valore s_{zk} rappresentante il pericolo di frana. Si ricorda che tali notazioni sono già state introdotte in 2.1. Tale

metodo assegna un valore di *exposure* (Exp_{bi}) ad un generico edificio (b_i) all'interno della *GeoArea* nel seguente modo:

1. L'*Exposure* per l'edificio b_i è ottenuto come somma dei contributi di tutte le zone nella *Geoarea* per tale edificio.

$$\text{Exp}_{bi} = \sum_{k=1}^N \text{Exp}_{b_{i,k}} \quad (5)$$

2. Il contributo della generica zona z_k all'*exposure* dell'edificio b_i viene calcolato utilizzando la seguente equazione:

$$\text{Exp}_{b_{i,k}} = \overline{\text{Size}} \times S_{z_k} \times \begin{cases} 1, & \text{se } b_i \text{ è contenuta in } z_k \\ \frac{1}{d^3}, & \text{altrimenti} \end{cases} \quad (6)$$

Dove:

- $\overline{\text{Size}}$ è il rapporto tra l'area di z_k e l'area media di tutte le zone che costituiscono Z
- d denota la distanza minima tra b_i e il *boundary* della zona z_k

Al prodotto di questi due parametri viene poi moltiplicato un certo valore in base alla posizione relativa tra la zona z_k e l'edificio b_i . Tale valore corrisponderà ad 1 se vi è una situazione analoga alla figura seguente:

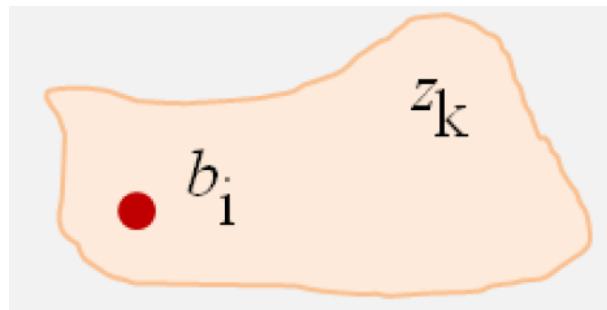


Figura 21: Esempio di un edificio b_i all'interno di una zona z_k

Se invece l'edificio b_i si trova esternamente rispetto alla zona z_k , il valore sarà $\frac{1}{d^3}$.

Detto ciò possiamo concludere questa breve panoramica sul metodo di calcolo originario con alcune critiche e considerazioni, che hanno rappresentato la base per la realizzazione di un nuovo metodo:

- Il metodo non tiene conto delle curve di livello che cadono nella *GeoArea*;
- Il valore del parametro *Size* è vicino allo zero per zone molto piccole, mentre è maggiore di 1 per tutte le zone che hanno area superiore al valore dell'area media delle zone della *GeoArea*. Inoltre *Size* assume valori molto alti per zone con area molto estesa.



Figura 22: Esempio di un edificio b_i esterno alla zona z_k

2.2.2 *Metodo di calcolo ex novo*

Il metodo proposto vuole sostituirsi a quello precedente, vedi 2.2.1, con lo scopo di migliorare il calcolo dell'*exposure*. Ovvero si vogliono assegnare valori che rappresentino con maggiore fedeltà il pericolo reale di frana per gli edifici. Il punto di forza del nuovo metodo, d'ora in avanti indicato con *NMC*, consiste nello sfruttare il "potenziale informativo" aggiuntivo fornito con l'introduzione delle **curve di livello**.

Viene da sé che un uso "intelligente" di queste curve, permette di ricavare importanti informazioni ai fini del nostro obiettivo, tra le quali:

- La quota dell'edificio
- La morfologia del territorio

Supponiamo per un attimo di essere riusciti a ricavare queste informazioni e di averle utilizzate in modo corretto, il *NMC* avrebbe così superato la prima debolezza del metodo originario, vedi 2.2.1. Tuttavia rimarrebbero due questioni in sospeso, ovvero:

1. Quali zone z_k della *GeoArea* influiscono realmente sul calcolo dell'*exposure* del generico edificio b_i ?
2. Supposto di aver trovato risposta al punto precedente, in che proporzione tali zone influiscono? tutte allo stesso modo?

Il metodo originario, rispondeva in parte a queste domande (vedi equazione 6). Utilizzando il parametro $(\frac{1}{d^3})$ si riusciva a rendere nullo il contributo di zone z_k abbastanza lontane, in quanto la funzione al cubo inversa smorza rapidamente, rispondendo così alla prima domanda.

Per quanto riguarda il secondo punto, si utilizzava il parametro *Size*, questo però porta alla generazione di falsi positivi/negativi nel calcolo dell'*exposure* dell'edificio, come già sottolineato nel paragrafo precedente.

Dunque vediamo come il *NMC* ricava le informazioni dalle curve di livello e come le utilizza nel calcolo delle *exposure* degli edifici b_i .

L'equazione del *NMC* che determina il valore finale di *exposure* per la generica b_i è la seguente:

$$\text{Exp}_{b_i} = \left(\sum_{\gamma=1}^N \text{Exp}_{b_i, \gamma} \right) + (\text{slope}_i \times \alpha_{\text{slope}}) \quad (7)$$

Tale valore dipende dalla somma di due addendi.

Il primo addendo rappresenta la sommatoria dei contributi delle zone $v_\gamma \in V$, ovvero la sommatoria degli $Exp_b_{i,\gamma}$. Si ricorda che il limite superiore N della sommatoria nell'Eq.7 si riferisce al numero di zone $v_\gamma \in V$, ovvero alla cardinalità dell'insieme V .

La formula per calcolare il contributo della zona v_γ alla determinazione del pericolo di frana dell'edificio b_i è la seguente:

$$Exp_b_{i,\gamma} = Size \times S_{v_\gamma} \times \Delta h \times \begin{cases} 1, & \text{se } b_i \text{ è contenuta in } v_\gamma \\ \frac{1}{d}, & \text{altrimenti} \end{cases} \quad (8)$$

Si notino le differenze rispetto l'equazione del metodo originario (vedi Eq.5).

Nella versione proposta dal NMC il parametro $(\frac{1}{d})$ non è elevato al cubo. Questo per evitare di smorzare troppo il contributo delle v_γ vicine. Il parametro *Size* è così definito:

$$Size = \frac{\text{area}(\text{buffer}_i \cap v_\gamma)}{\text{area}(v_\gamma)} \quad (9)$$

Ovvero l'area della geometria restituita dall'intersezione tra il poligono (v_γ) e il buffer_i che come abbiamo definito nelle notazioni ha raggio r ed è costruito attorno all'edificio b_i . Per un dato valore del raggio r , il valore del numeratore dell'Eq. 8 cresce se la posizione di b_i si avvicina al centroide della zona che lo contiene (ovvero v_γ), di conseguenza, in base all'Eq.7, anche il corrispondente valore dell'exposure sarà più alto.

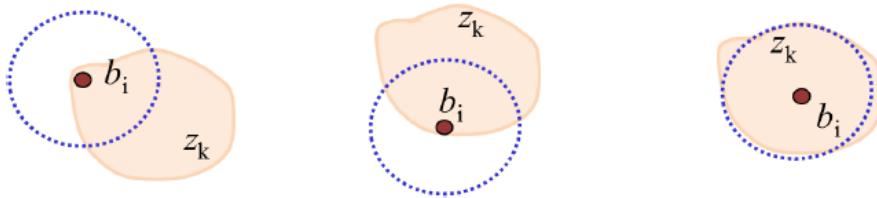


Figura 23: Esempio di un edificio all'interno di una zona v_γ

Il parametro *Size* risponde alle domande che ci siamo posti precedentemente (vedi 2.2.2), ovvero:

- Quali zone z_k della *GeoArea* influiscono realmente sul calcolo dell'*exposure* di b_i ? È facile ora rispondere a questa domanda, tutte e sole le zone v_γ , ovvero come abbiamo già detto le zone z_k per le quali vale:

$$\text{area}(\text{buffer}(b_i, r) \cap z_k) \neq \emptyset \quad (10)$$

Bisogna stabilire quanto deve essere il valore del raggio, infatti se tale valore è sufficientemente grande da garantire che eventi franosi a distanza maggiore da

b_i non interessino quest'ultimo, allora le zone z_k che non intersecano il Buffer non devono contribuire al calcolo dell'*exposure*.

- In che proporzione tali zone v_γ influiscono? Tutte allo stesso modo? Ovviamente no. Grazie all'espressione 9, il parametro *Size* tiene conto anche dell'area effettivamente occupata da v_γ all'interno del buffer. In questo modo si riduce notevolmente la probabilità che il *NMC* generi falsi positivi/negativi.

Tale parametro rappresenta, in modo quantitativo, un ulteriore legame tra l'Eq.8 e la morfologia del territorio. Vediamo un esempio concreto della sua utilità.

La Fig. 24 è stata estratta mediante il software di visualizzazione di dati territoriali QGIS e mostra una particolare situazione.

La sfera rossa rappresenta un edificio mentre la circonferenza in blu il buffer, di cui abbiamo ampiamente parlato, di raggio r pari a 500mt.

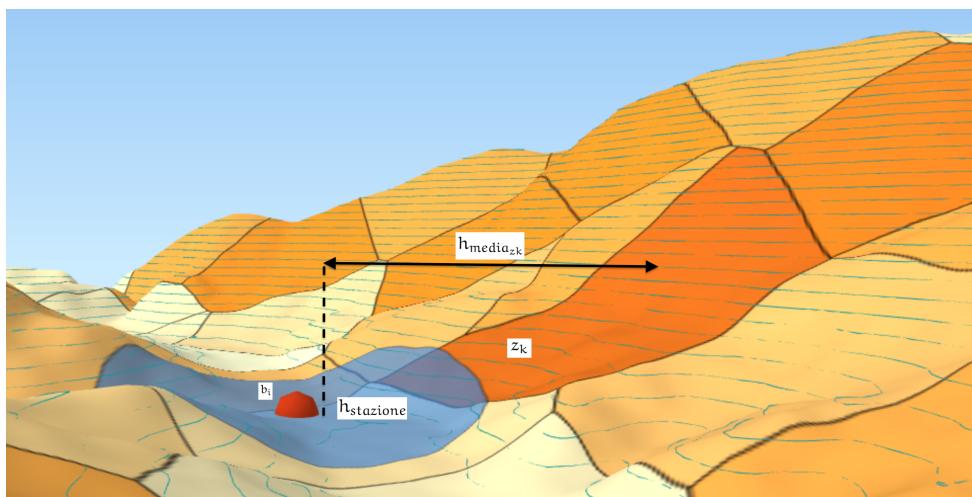


Figura 24: Esempio di una v_γ con $h_{\text{media}} v_\gamma$ elevata la cui intersezione con il buffer_i risulta essere un'area piccola

Notiamo come la v_γ in questione occupi una parte molto piccola del buffer_i . Questo sicuramente porterà ad avere un valore di *Size* molto basso e di conseguenza anche il contributo di v_γ all'*exposure*, generando un falso negativo. Il fattore Δh interviene in soccorso di *Size*, permettendo all' $\text{Exp}_{b_i, \gamma}$ di assumere in valore non eccessivamente basso, tenendo conto dell'elevata differenza tra l'altezza della stazione b_i e l'altezza media di v_γ .

Il secondo addendo dell'Eq.7 va a rappresentare un fattore di rischio dovuto al dislivello che si trova più vicino a b_i e che quindi va considerato con maggior peso.

α_{slope} è un fattore moltiplicativo intero utilizzato per incrementare o diminuire il peso di slope_i nel calcolo di Exp_{-b_i} rispetto al primo addendo.

2.2.3 L'algoritmo

Lo scopo di questo paragrafo è formalizzare gli algoritmi che implementa NMC, descritto nella sezione [2.2.2](#).

Algoritmo 1

Il primo algoritmo denominato averageElevationNearZones ha come obiettivo quello di calcolare il valore di avgElevation associato ai v_γ in V . Prende come parametri di input gli insiemi Z , E e l'elemento $buffer_i$ mentre restituisce come output l'insieme V .

Algorithm 1: averageElevationNearZones

Input : Z , E , $buffer_i$
Output: V
Result: Calcola la quota media delle zone che intersecano il buffer

```
1   $V = \emptyset;$ 
2  avgElevation = 0;
3  contatore = 0;
4  for each  $z_k \in Z$  do
5      if  $z_k \cap buffer_i$  then
6          for each  $e_j \in E$  do
7              if  $z_k \cap e_j$  then
8                  avgElevation = avgElevation + (elevation di  $e_j$ );
9                  contatore = contatore + 1;
10             end
11         end
12         if contatore > 0 then
13             avgElevation = avgElevation / contatore;
14         end
15         aggiorna il valore avgElevation di  $z_k$ ;
16          $v_\gamma = z_k;$ 
17          $V = V \cup v_\gamma;$ 
18         avgElevation = 0;
19         contatore = 0;
20     end
21 end
22 return  $V$ ;
```

Commenti:

Dalla riga 1 alla 3 sono presenti alcune inizializzazioni: l'insieme V inizialmente è vuoto, mentre sono fissati a 0 i valori di avgElevation e contatore. Dalla riga 4 alle

21 è presente un ciclo che scandisce gli elementi z_k di Z . Immediatamente all'interno del ciclo (righe 5-20) viene effettuato un controllo che verifica se la zona in esame ha un'intersezione con il buffer.

In caso di esito negativo si passa subito al successivo elemento di Z . In caso di esito positivo, per ogni elemento e_j in E , si verifica se interseca con la zona z_k in esame. Se anche questo controllo viene superato si incrementa il valore di avgElevation con l'elevation di e_j e il valore di contatore di 1.

Una volta completato il ciclo sugli elementi in E , si controlla (riga 12) se contatore ha un valore maggiore di zero, quindi se almeno per una linea e_j è verificata l'intersezione con la z_k esaminata. In questo caso il valore di avgElevation viene sovrascritto a avgElevation/contatore.

Si passa quindi (riga 15) ad aggiornare il valore avgElevation nella tupla z_k con il valore appena calcolato.

Alla riga 16 la tupla z_k viene copiata nella tupla v_γ che viene inserita nell'insieme V . Questa operazione viene eseguita per mantenere nell'insieme V solo gli elementi z_k di interesse, ossia quelli per cui è verificata l'intersezione con $buffer_i$.

Alle righe 18 e 19 le variabili avgElevation e contatore sono resettate a 0, per poter essere appropriatamente riutilizzate nella successiva iterazione del ciclo.

L'algoritmo infine (riga 22) restituisce l'insieme V .

Riassumendo l'algoritmo prende tutte e sole le z_k che intersecano il buffer e per ognuna calcola la quota media. Inserisce infine queste zone nell'insieme V che restituisce come risultato.

La complessità computazionale dell'algoritmo è data dal ciclo in riga 4 e da quello in riga 6. È quindi pari a $\mathcal{O}(\text{card}(Z) \times \text{card}(E))$ anche se mediamente è molto inferiore in quanto è molto improbabile che ogni elemento in Z intersechi con $buffer_i$ e con tutte le linee in E .

Algoritmo 2

L'algoritmo denominato elevationBuild ha come obiettivo quello di stimare l'elevation di b_i . Prende come parametri di input gli insiemi V (l'insieme delle zone che intersecano $buffer_i$), E (l'insieme delle curve di livello) e l'elemento b_i . In output restituisce lo stesso elemento b_i di input il cui campo elevation è stato aggiornato con la stima calcolata.

Algorithm 2: elevationBuild

Input : V, E, b_i
Output: b_i
Result: Calcola una stima dell'elevation di b_i

```
1  $T = \emptyset;$ 
2 for each  $v_\gamma \in V$  do
3   for each  $e_j \in E$  do
4     if  $v_\gamma \cap e_j$  then
5       geom di  $t_f$  = st_intersection (  $v_\gamma, e_j$  );
6       elevation di  $t_f$  = elevation di  $e_j$ ;
7        $T = T \cup t_f$ ;
8     end
9   end
10 end
11 for each  $t_f \in T$  do
12   dist = st_distance (  $b_i, t_f$  );
13   Trova  $t_{f\min}$  t.c. dist è minima;
14   Trova  $t_{f\min2}$  t.c. dist è la successiva al minimo;
15 end
16 if  $b_i$  in posizione atipica then
17    $t_{f\min2} = t_{f\min}$ ;
18 end
19 Calcola  $h_{stazione}$ 
20 Aggiorna elevation di  $b_i$  con  $h_{stazione}$ ;
21 return  $b_i$ ;
```

Commenti:

Nella prima riga dell'algoritmo l'insieme T è inizializzato come insieme vuoto.

Nelle righe 2-10 è presente un ciclo annidato. Il ciclo più esterno itera sugli elementi v_γ in V (l'insieme delle zone che intersecano il buffer), mentre quello più interno sugli elementi e_j in E (l'insieme delle linee di livello). Se l'intersezione tra gli elementi v_γ e e_j esiste, allora questa intersezione viene assegnata come geometria di t_f mentre come sua elevation viene assegnata l'elevation di e_j . L'ennupla t_f così definita è inserita nell'insieme T (riga 7).

All'interno del ciclo in riga 11 si vanno a scandire i t_f in T . All'interno del ciclo si va a calcolare la distanza dist tra l'elemento t_f e b_i . Si vanno quindi a individuare (nelle righe 13-14) $t_{f\min}$ e $t_{f\min2}$.

$t_{f\min}$ è la $t_f \in T$ tale che la distanza dist tra t_f e b_i è la distanza minima tra tutte le distanze calcolate tra ogni altra $t_f \in T$ e b_i .

In modo analogo $t_{f\min2}$ è la $t_f \in T$ la cui distanza dist tra t_f e b_i ha un valore immediatamente successivo alla distanza minima.

Alla riga 16 si effettua un controllo per verificare se b_i si trova in una posizione "atipi-

ca" come definito nella Sezione 2.1. Nel caso di esito positivo si copia l'ennupla $t_{f\min}$ in $t_{f\min2}$.

Si calcola quindi alla riga 19 il valore di h_{stazione} come descritto precedentemente (in Sezione 2.1), dove $t_{f\min}$ rappresenta da e_1 , mentre $t_{f\min2}$ rappresenta e_2 .

Nella riga 20 si aggiorna l'elevation di b_i con il valore appena calcolato h_{stazione} . Viene infine restituiti b_i .

Riassumendo l'algoritmo individua i segmenti di linea frutto dall'intersezione tra tutte le linee in E e le zone in V e li inserisce nell'insieme T . Individua poi i 2 elementi in T che sono più vicini a b_i . Sfruttando questi 2 elementi calcola h_{stazione} cioè una stima dell'elevation di b_i .

La complessità computazionale è data dal ciclo annidato in riga 2 ed è pari a $\mathcal{O}(\text{card}(V) \times \text{card}(E))$ anche se mediamente è molto inferiore in quanto è molto improbabile che ogni elemento in V intersechi tutte le linee in E .

Algoritmo 3

L'algoritmo denominato slopeFactor ha come obiettivo quello di calcolare lo slope associato a b_i . Prende come parametri di input l'insieme E delle curve livello e l'elemento $buffer_i$, mentre restituisce $slope_i$ ossia lo slope associato a b_i .

Algorithm 3: slopeFactor

Input : E , $buffer_i$
Output: $slope_i$
Result: Calcola lo slope associato a b_i

```
1  $P = \emptyset;$ 
2  $Q = \emptyset;$ 
3  $slope_i = 0;$ 
4 for each  $e_j \in E$  do
5   if  $buffer_i \cap e_j$  then
6     geom di  $p_m = st\_intersection ( buffer_i, e_j );$ 
7     elevation di  $p_m = elevation di e_j;$ 
8      $P = P \cup p_m;$ 
9   end
10 end
11 Calcola il vettoreDirezionale;
12 for each  $p_m \in P$  do
13   if  $p_m \cap vettoreDirezionale$  then
14     Calcola la distanza tra  $b_i$  e  $(p_m \cap vettoreDirezionale);$ 
15     Aggiorna distance di  $p_m$  con la distanza calcolata;
16      $q_w = p_m;$ 
17      $Q = Q \cup q_w ;$ 
18   end
19 end
20 for each  $q_w \in Q$  ordinato rispetto a distance do
21    $slope_i = slope_i + (pendenza associata a q_w);$ 
22 end
23 return  $slope_i;$ 
```

Commenti:

Nelle prime righe (1-3) vengono inizializzati gli insiemi P , Q come insiemi vuoti e la variabile $slope_i$ a zero.

Dalla riga 4 inizia un ciclo che scandisce gli elementi e_j in E e per ognuno controlla se interseca con $buffer_i$. Se questa intersezione esiste, questa viene assegnata come geometria di p_m mentre come sua elevation viene assegnata l'elevation di e_j . La tupla p_m così definita, viene inserita nell'insieme P . Queste operazioni permettono di lavorare su un insieme che è molto più piccolo di E e che contiene solo le porzioni di linee di interesse.

Alla riga 11 viene calcolato il vettoreDirezionale come definito nella Sezione 2.1. Nella righe 12-19 è presente un ciclo che scandisce gli elementi p_m in P e per ognuno verifica se interseca con il vettoreDirezionale. Se questa intersezione esiste viene calcolata la distanza tra b_i e l'intersezione. Il valore di distanza calcolato va ad aggiornare il valore distance in p_m . La tupla p_m così costruita viene quindi copiata in q_w . Alla riga 17, q_w viene inserito nell'insieme Q . In questo modo in Q si vanno a inserire tutti e soli gli elementi p_m che intersecano il vettoreDirezionale.

Nella riga 20 inizia un ciclo che scandisce gli elementi q_w in Q ordinati rispetto al campo distance. All'interno del ciclo viene calcolata la pendenza associata a q_w come definita nella Sezione 2.1. Questo valore va a incrementare slope_i.

Viene infine restituito slope_i.

Riassumendo l'algoritmo individua le porzioni di linee di livello che si trovano tutte in una stessa direzione rispetto a b_i e per ognuna di esse calcola la pendenza, ossia la ripidità tra la linea e la linea precedente, dove la linea precedente è più vicina a b_i . slope_i è la somma di queste pendenze.

La complessità computazionale è data dal ciclo in riga 4 ed è quindi pari a $\mathcal{O}(\text{card}(E))$.

Algoritmo 4

L'algoritmo denominato `computeExposure` è l'algoritmo che effettivamente calcola l'exposure associato a b_i . All'interno dell'algoritmo vengono richiamati tutti e tre gli algoritmi precedentemente definiti.

Algorithm 4: `computeExposure`

Input : B , r , α_{slope} , E , Z
Output: B
Result: Calcola l'exposure di tutte le b_i

```
1 for each  $b_i \in B$  do
2      $buffer_i = st\_buffer(b_i, r);$ 
3      $V = averageElevationNearZones(Z, E, buffer_i);$ 
4      $b_i = elevationBuild(V, E, b_i);$ 
5      $slope_i = slopeFactor(E, b_i, buffer_i);$ 
6     for each  $v_\gamma \in V$  do
7         Calcola Size;
8         Calcola  $Exp_{b_i, \gamma}$ ;
9     end
10    Calcola  $Exp_{b_i}$ ;
11    Aggiorna il valore di  $Exp_{b_i}$  in  $b_i$ ;
12 end
13 return  $B$ ;
```

Commenti:

Nelle righe 1-11 è presente un ciclo che scandisce tutte le b_i in B . Per ogni b_i viene calcolato un buffer circolare di raggio r . Viene successivamente costruito un insieme V che contiene tutte le zone che intersecano il buffer, dove a ogni zona viene associata una quota media (richiamando l'algoritmo 1).

Nella riga 4 la tupla b_i viene aggiornata inserendo un valore che rappresenta una stima della sua quota (richiamando l'algoritmo 2).

Nella riga 5 viene quindi calcolato $slope_i$ (richiamando l'algoritmo 3).

Dalla riga 6 inizia un ciclo che scandisce gli elementi in V e per ognuno calcola Size e $Exp_{b_i, \gamma}$.

In riga 10 si calcola Exp_{b_i} (tenendo conto anche del fattore α_{slope} come mostrato nell'Eq.7) che va ad aggiornare il valore nella tupla b_i .

L'algoritmo ritorna come risultato l'insieme B dove ogni tupla b_i viene arricchita con un stima della quota e un valore di Exposure.

La complessità computazionale è data dai due cicli presenti alle righe 1 e 10 ed è quindi pari a $\mathcal{O}(\text{card}(V) \times \text{card}(B))$.

2.3 UN METODO PER CLASSIFICARE LE LINEE FERROVIARIE

Il metodo proposto per classificare le stazioni (vedi Sez. 2.2.2) può essere utilizzato anche per classificare le linee ferroviarie. Occorre tuttavia estendere il NMC a causa della diversa rappresentazione geometrica dei due elementi.

2.3.1 *Metodo di calcolo*

Poiché il NMC è stato progettato per classificare stazioni, rappresentate attraverso dei punti geometrici, non può essere utilizzato così com'è per classificare le linee ferroviarie rappresentate invece attraverso linee geometriche (Fig. 25).



Figura 25: Esempio di rappresentazione geometrica di una linea ferroviaria ove b_i e b_{i+1} sono due stazioni

L'idea di fondo è campionare con un certo passo, le linee ferroviarie. Fatto ciò si avrà una sequenza di punti sui quali poter applicare il NMC.

Nonostante l'idea sembri essere molto semplice, ci sono tre questioni sulle quali bisogna fare delle riflessioni e proporre delle soluzioni:

1. che passo di campionamento adottare? La scelta di un passo piccolo giova alla correttezza del calcolo finale di *exposure* della tratta ma aumenta significativamente i tempi di esecuzione del calcolo stesso.
2. le linee sono semplici? Le linee ferroviarie potrebbero avere delle traiettorie complesse, il metodo deve essere in grado di gestirle.
3. come aggregare i punti campionati? Dopo aver campionato una certa tratta e aver determinato i valori di *exposure* di ogni punto della sequenza che la compongono, bisognerà aggregare in qualche modo tali risultati per poter aver un

risultato utile e consultabile. Il campionamento potrebbe generare un numero considerevole di punti che presi singolarmente non forniscono un contributo significato al problema in questione (vedi Cap. 1)

L'estensione del NMC (d'ora in avanti indicato con NMCE, dove E sta per esteso) per le linee ferroviarie propone una soluzione per ognuna di queste problematiche. Nell'ordine in cui sono state precedentemente presentate:

1. Il passo di campionamento scelto è di 300 mt, ma può essere facilmente modificato essendo un parametro del metodo(Fig.26)

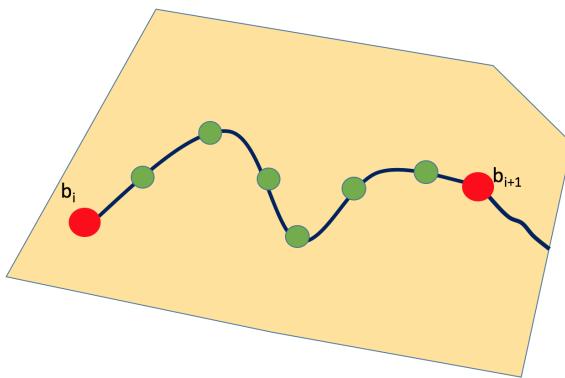


Figura 26: Esempio di campionamento di una linea ferroviaria

2. Nel caso in cui una tratta abbia una traiettoria complessa, il NMCE rileva tale situazione e provvede a dividere la linea in due o più linee semplici. In Fig. 27 un esempio esplicativo.

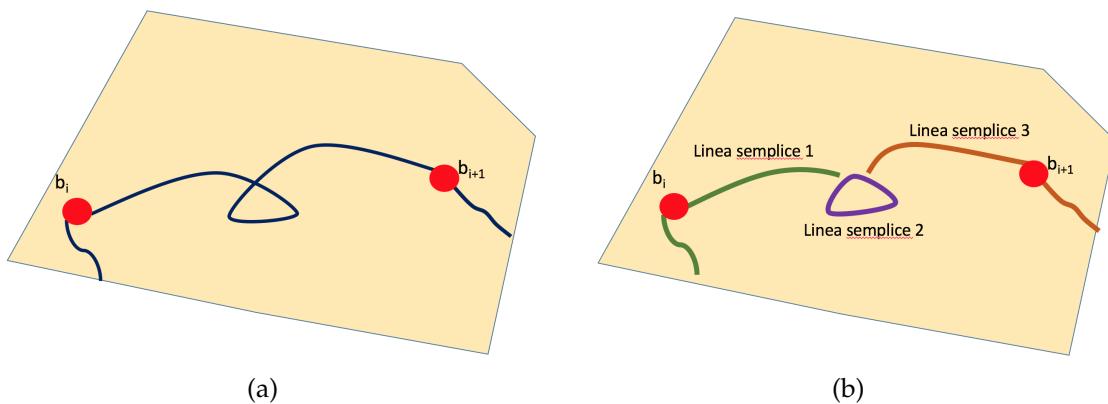


Figura 27: Esempio di una linea complessa (a) e della sua divisione in tre linee semplici (b)

3. Una volta campionato la tratta semplice, si utilizza il NMC per determinare il valore di *exposure* di ogni punto che la compone.
La linea ferroviaria viene quindi divisa in segmenti di lunghezza arbitraria, nel

nostro caso 1 km ciascuno (fatta eccezione per l'ultimo segmento che potrebbe essere di dimensione più inferiore) come in Fig. 28a. Il valore di *exposure* del generico segmento è pari alla media aritmetica dei punti contenuti al suo interno.

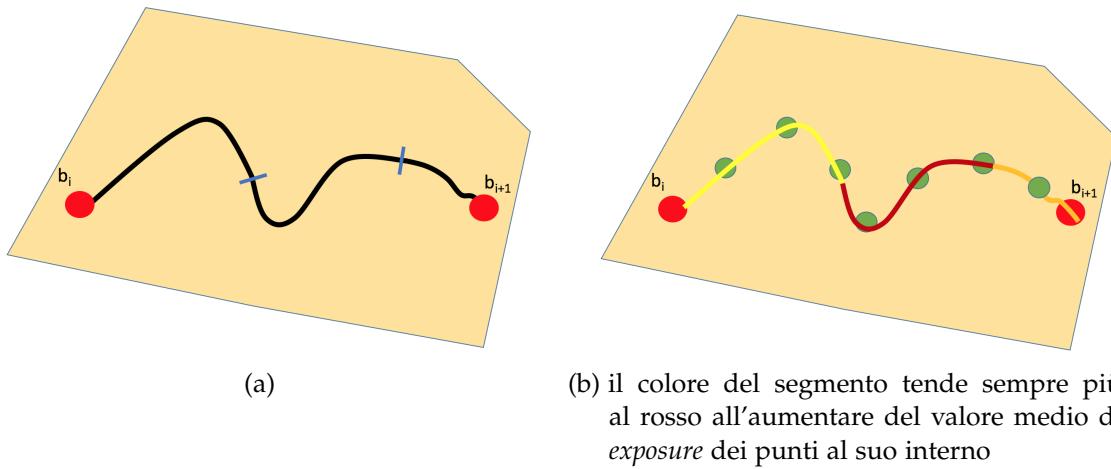


Figura 28: Esempio di divisione di una linea ferroviaria (a) e assegnazione dell'*exposure* ad ogni suo segmento (b)

Per ogni segmento si tiene traccia anche del valore massimo e della somma dell'*exposure* dei punti al suo interno. Queste informazioni potrebbero in alcuni casi essere utili. Ad esempio se ci fosse un punto il cui valore di *exposure* risulti essere molto alto rispetto gli altri punti, la media aritmetica non sarebbe sufficiente a rilevare tale situazione.

UN CASO DI STUDIO: L'ABRUZZO

Il focus del presente studio è posto, come è stato già preannunciato in precedenza, unicamente sulla regione Abruzzo, la cui rete ferroviaria, nonostante non abbia una estensione considerevole, è immersa in un territorio eterogeneo ed esposto a diversi livelli di rischio di natura idrogeologica. La ridotta estensione dei dati in esame, ha consentito di focalizzarsi su una analisi puntuale e non dispersiva dei metodi proposti, al fine di valutarne accuratamente la validità ed i limiti. Il presente *report* è stato prodotto a fronte di un lavoro di laboratorio volto alla costruzione dei seguenti dati:

- Una classifica delle stazioni, in base al potenziale rischio idrogeologico a cui esse sono esposte;
- Una classifica dei tratti di ferrovia maggiormente esposti a rischio di smottamenti del terreno su cui sono posti i binari.

La produzione di tali classifiche ha l'obiettivo di consentire una fruibilità facile ed immediata dei dati per quegli enti preposti al monitoraggio e alla manutenzione delle tratte e stazioni ferroviarie.

Di seguito si mostrerà una breve istantanea della attuale situazione osservata nel territorio abruzzese, con particolare attenzione ai dati selezionati, tra quelli disponibili, e alla loro provenienza.

3.1 TERRITORIO

L'Abruzzo (rappresentata in Fig. 29) è una regione a statuto ordinario dell'Italia peninsulare, compresa tra il mare Adriatico e l'Appennino centrale, con capoluogo L'Aquila. La regione è geograficamente ed economicamente parte dell'Italia centrale, mentre dal punto di vista storico e linguistico è inserita all'interno dell'Italia meridionale.

Occupava una superficie di 10.831 km² e ha una popolazione di 1.322.349 abitanti. È diviso in quattro province: L'Aquila, Chieti, Pescara e Teramo, e in 305 comuni. Confina a nord con le Marche, ad est con il mare Adriatico, ad ovest con il Lazio e a sud con il Molise. Si divide principalmente in una parte costiera nel versante orientale con le spiagge dell'Adriatico, e in una parte montuosa dal lato occidentale con il Gran Sasso d'Italia (2.914 m s.l.m.), la Majella (2.793 m s.l.m.) e il Sirente-Velino (2.487 m s.l.m.) che costituiscono i tre massicci montuosi più alti dell'intera catena appenninica.



Figura 29: La regione Abruzzo

3.2 RETE FERROVIARIA

Per quanto riguarda il trasporto ferroviario, vi è in Abruzzo una forte disparità tra quello moderno sulla costa (anche se con numero e qualità delle corse e del servizio imparagonabili rispetto all'asse ferroviario "tirrenico" e non è prevista alcuna linea di TAV) e quello delle zone interne, molto carente in termini di modernità e qualità del servizio e in attesa da decenni di interventi di potenziamento e ammodernamento (vedi in particolare la linea Pescara-Avezzano-Roma). Complessivamente comunque la rete ferroviaria abruzzese in attivo è abbastanza sviluppata e si estende per 524 km, mentre l'estensione totale complessiva della rete (attiva o dismessa) è pari a 648 km.

3.2.1 *Stazioni Ferroviarie*

I dati iniziali a nostra disposizione circa i nodi ferroviari ci forniscono le informazioni relative alle 114 stazioni distribuite sul territorio abruzzese mostrate in Fig.30.



Figura 30: Stazioni ferroviarie presenti in Abruzzo

3.2.2 *Linee Ferroviarie*

Sul territorio abruzzese sono presenti nove tratte ferroviarie:

- Bologna - Bari
- Ortona - Crocetta
- Marina di San Vito - Castel di Sangro
- Roma - Pescara
- Avezzano - Roccasecca
- Archi stazione - Atessa
- Sulmona - Carpinone
- Rieti - L'Aquila - Sulmona
- Teramo - Giulianova

Di queste tratte solo alcune sono completamente interne alla regione, altre invece la attraversano (avendo origine e/o fine al di fuori dei confini); riguardo queste ultime si è scelto di considerare unicamente le porzioni che cadono nella regione Abruzzo mostrate in Fig. 31.



Figura 31: Linee ferroviarie presenti in Abruzzo

3.3 ZONE

L'intero territorio abruzzese è stato completamente suddiviso in 22.000 zone più piccole la cui area media è pari a $0,5 \text{ km}^2$. Il dataset relativo a queste zone ,che ci è stato fornito, è stato costruito mediante un algoritmo che segue un approccio basato sul *diagramma di Voronoi*, tale da costruire una partizione completa del territorio determinata dalle distanze rispetto ad un determinato insieme discreto di elementi dello spazio (nel nostro caso un insieme finito di punti). Il risultato di tale partizionamento dello spazio è visibile nella Fig. 32.



Figura 32: Partizionamento completo della regione Abruzzo

Ad ogni z_k è associato un valore numerico compreso tra 0 e 1, il quale quantifica la pericolosità determinata dal rischio idrogeologico della zona alla quale si riferisce.

3.4 BASE DI DATI SPAZIALE

Si descrive in seguito la progettazione e conseguente realizzazione della base di dati geografica realizzata per un efficace implementazione degli algoritmi precedentemente descritti. Una base di dati spaziale accompagnata da un opportuno SGBD permette infatti di operare in modo agevole e naturale con dati geometrici, offrendo una gamma di operatori spaziali ottimizzati per la loro elaborazione.

I dati da memorizzare sono:

- Le stazioni e le linee ferroviarie abruzzesi con informazioni relative al nome e alla geometria;
- La geometria dei confini dell'area geografica;
- La geometria delle zone che vanno a partizionare in modo completo l'area con relative informazioni di rischio idrogeologico;
- La geometria delle linee di livello presenti nel territorio di riferimento con la quota associata;
- I risultati ottenuti dalle elaborazioni.

3.4.1 *Progettazione concettuale*

Per ogni entità verranno mostrate:

- Gli attributi
- L'attributo identificante
- I vincoli di partecipazione nelle relazioni, con la notazione (min, max)

In Figura 33 si riporta il modello E-R della base di dati spaziale, mentre in Tabella 1 sono elencate le entità con i relativi attributi, descrizione e attributi identificanti, ed infine in Tabella 2 l'elenco delle associazioni.

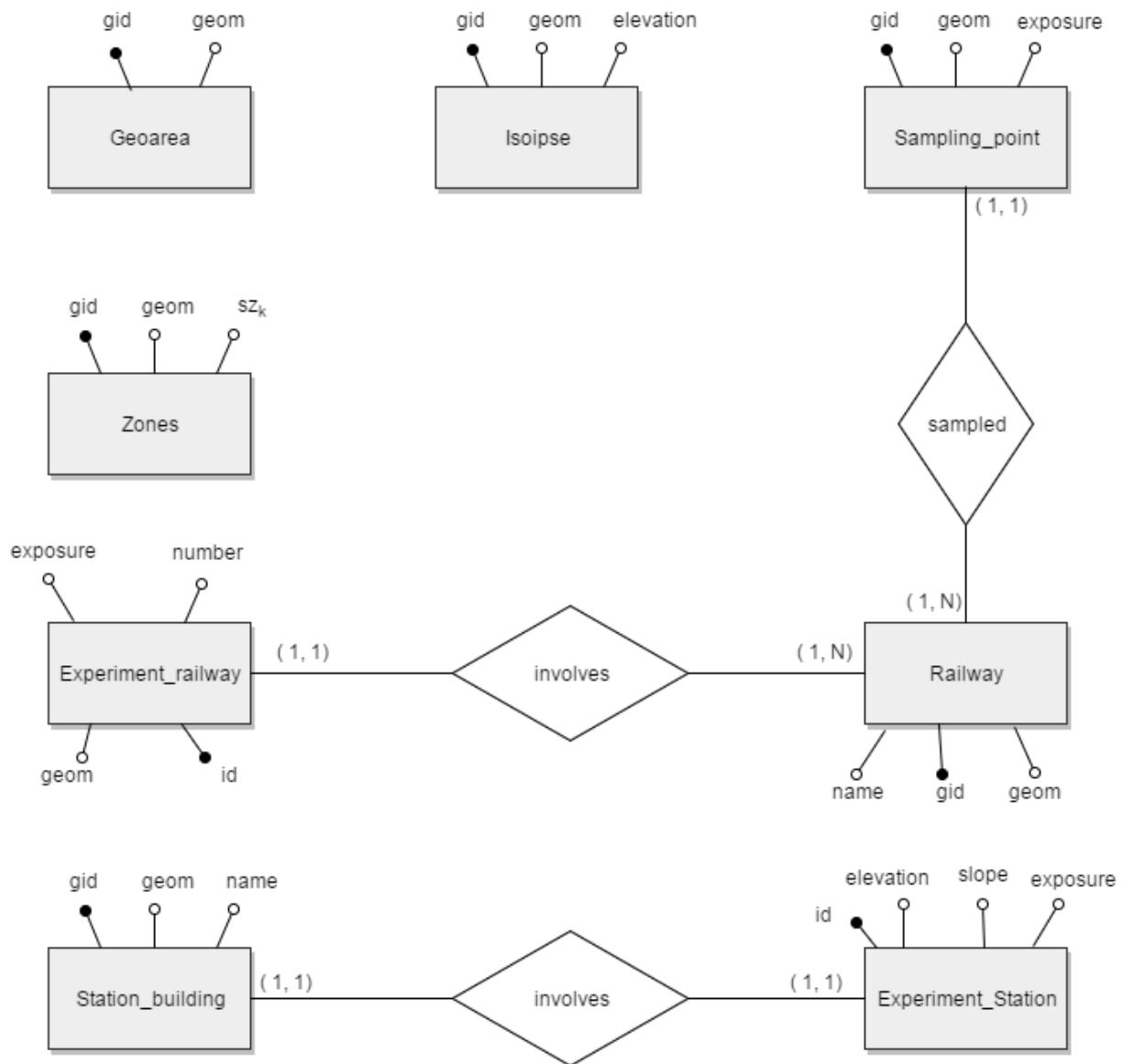


Figura 33: Diagramma E-R della base di dati

Note:

L'attributo `number` dell'entità `Experiment_railway` rappresenta il numero del segmento relativo alla linea ferroviaria di appartenenza. Questo valore non è identificativo tra tutti i segmenti, ma solo tra i segmenti della linea al quale appartiene, quindi ad esempio su ogni linea sarà presente il segmento con `number` pari ad uno.

Entità	Descrizione	Attributi	ID
Geoarea	Entità relativa al territorio di interesse. Nel nostro caso è la regione Abruzzo.	<i>gid</i> <i>geom</i>	<i>gid</i>
Zones	Entità relativa al territorio di interesse. Nel nostro caso è un partizionamento completo della regione Abruzzo.	<i>gid</i> <i>sz_k</i> <i>geom</i>	<i>gid</i>
Station_building	Entità relativa alle stazioni presenti nel territorio di riferimento.	<i>gid</i> <i>name</i> <i>geom</i>	<i>gid</i>
Isoipse	Entità relativa alle isoipse presenti nel territorio di riferimento.	<i>gid</i> <i>elevation</i> <i>geom</i>	<i>gid</i>
Railway	Entità relativa alle linee ferroviarie presenti nel territorio di riferimento.	<i>gid</i> <i>name</i> <i>geom</i>	<i>gid</i>
Sampling_point	Entità relativa ai punti individuati come campionamento delle linee ferroviarie presenti nel territorio di riferimento.	<i>id</i> <i>geom</i> <i>exposure</i>	<i>id</i>
Experiment_station	Entità relativa agli esperimenti effettuati sulle stazioni presenti nel territorio di riferimento.	<i>id</i> <i>elevation</i> <i>slope</i> <i>exposure</i>	<i>id</i>
Experiment_railway	Entità relativa agli esperimenti effettuati sulle linee ferroviarie presenti nel territorio di riferimento.	<i>id</i> <i>geom</i> <i>number</i> <i>exposure</i>	<i>id</i>

Tabella 1: Tabella delle entità

Associazione	Descrizione	Entità coinvolte	Attributi
involves	Relazione che evidenzia il coinvolgimento di una sola stazione in ogni esperimento.	<i>station_building</i> <i>experiment_station</i>	<i>nessuna</i>
involves	Relazione che evidenzia il coinvolgimento di almeno una linea ferroviaria in uno o più esperimenti.	<i>railway</i> <i>experiment_railway</i>	<i>nessuna</i>
sampled	Relazione che lega ogni linea ferroviaria con uno o più punti di campionamento.	<i>railway</i> <i>sampling_point</i>	<i>nessuna</i>

Tabella 2: Tabella delle associazioni

3.4.2 Progettazione logica

Il diagramma E-R in figura 33 viene tradotto nello schema logico rappresentato graficamente in figura 34

Visual Paradigm Standard (Università degli Studi dell'Aquila)

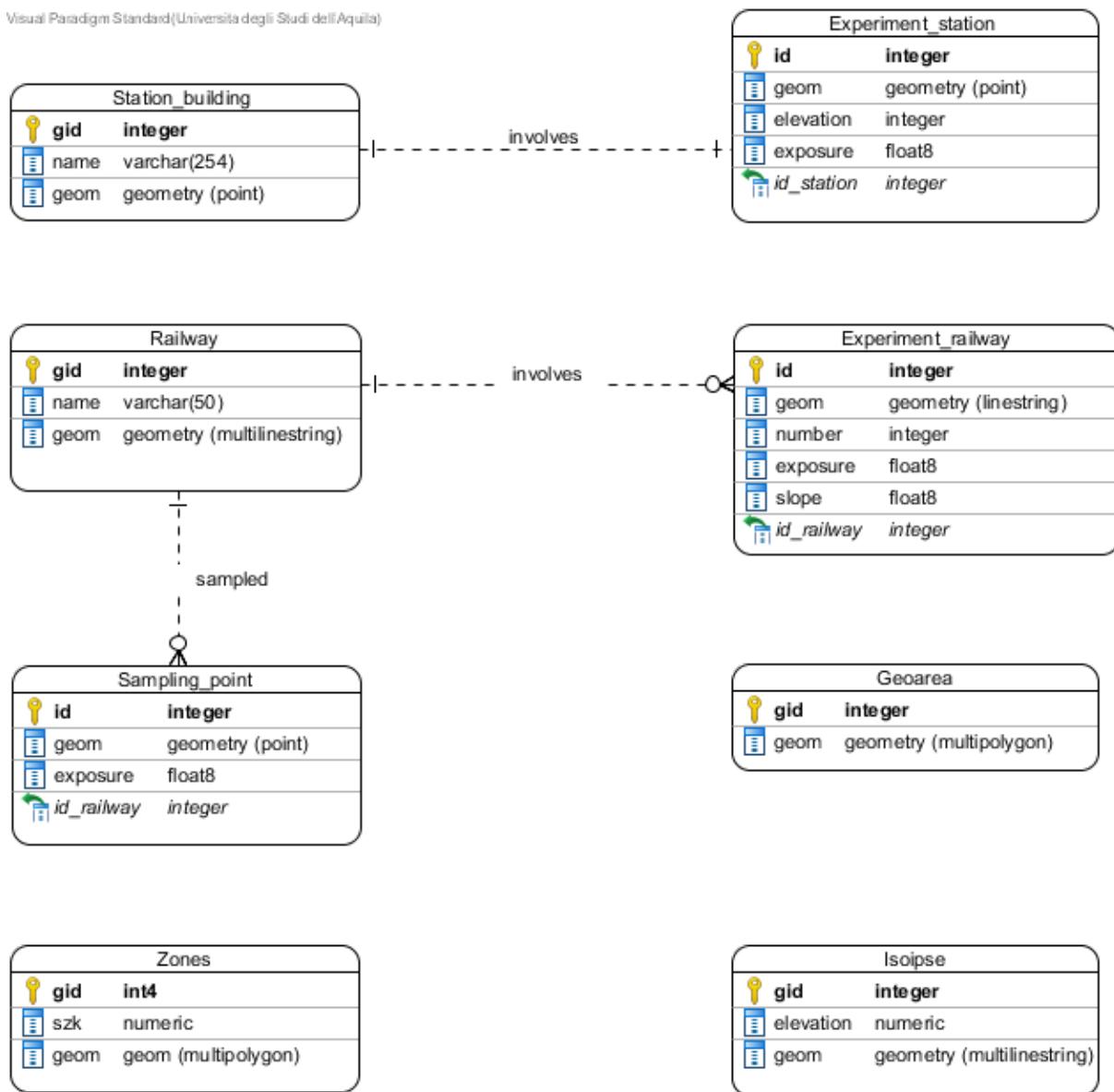


Figura 34: Schema logico

RISULTATI E LORO DISCUSSIONE

In questo capitolo verranno illustrati i risultati più significativi ottenuti dall'applicazione dei metodi esposti (vedi Sez. 2) per il calcolo del ranking, rispettivamente delle stazioni (Sez. 4.1) e delle tratte ferroviarie (Sez. 4.2). Per i risultati completi si rimanda alla lettura dell'appendice B.

4.1 RANKING E VALUTAZIONE DELLE STAZIONI FERROVIARIE

Applicando il metodo discusso nella Sez. 2.2.2 abbiamo ottenuto i risultati che andremo qui di seguito ad illustrare. Tali risultati sono stati utilizzati per realizzare una classifica delle stazioni in base al loro grado di esposizione al pericolo idrogeologico.

Il valore di *exposure* ottenuto per le stazioni in analisi varia da un valore minimo di 0 ad un massimo di 50. A fronte di tali risultati si è deciso di definire tre diverse fasce di esposizione al pericolo in base alle quali aggregare le stazioni all'interno della classifica:

- *High*
- *Moderate*
- *Low*

La Tabella 3 mostra la suddetta suddivisione in fasce di rischio, gli intervalli in cui le stazioni sono state divise secondo il valore di *exposure*, il numero e la percentuale sul totale delle stazioni appartenenti ad una determinata fascia. Inoltre ad ogni fascia è stato associato un colore identificativo.

Fascia	Intervallo valori Exp_b_i	# b_i	%
High	$20 < \text{Exp_b}_i < 50$	6	5,26%
Moderate	$1 < \text{Exp_b}_i < 20$	62	54,39%
Low	$0 < \text{Exp_b}_i < 1$	46	40,35%

Tabella 3: Aggregazione delle stazioni b_i in base al valore di *exposure*

Come si evince dalla Tabella 3 tale suddivisione in fasce non è lineare ma bensì frutto di un'analisi critica (vedi Sez. 4.1.1) del metodo discusso (vedi Sez. 2.2.2) e dei risultati ottenuti. Inoltre è possibile notare come la maggioranza delle stazioni presenta un livello di esposizione al rischio *Moderate* mentre solo il 5,26% presenta valori di *exposure* tali da far emergere il bisogno di controlli più frequenti (fascia *High*).

Nella Tabella 4 si riportano i primi dieci record, appartenenti alla classifica delle stazioni ordinati in modo decrescente rispetto il valore di *exposure*. In riferimento alla singola stazione, per ogni record vengono quindi mostrati la posizione all'interno della classifica, l'identificativo, il nome e il suo valore di *exposure*.

n.	id Stazione	Nome Stazione Ferroviaria	Valore di <i>exposure</i>
1	114	Acciano	49,85
2	54	Fontecchio	42,68
3	44	Sant'Ilario	31,73
4	38	Pettorano Sul Gizio	31,14
5	94	Isca d'Archi	22,71
6	66	Prezza	20,54
7	33	Bussi	19,98
8	81	Capistrello	18,57
9	91	Civiluparella	16,32
10	63	Sella di Corno	14,24

Tabella 4: Classifica delle prime dieci stazioni ordinata in base al valore di *exposure*

4.1.1 *Analisi critica del Nuovo Metodo Proposto*

Osservando i risultati ottenuti dall'applicazione del metodo di cui abbiamo discusso nella Sez. 2.2.2 abbiamo potuto osservare come i valori di *exposure* delle stazioni ferroviarie non presentano una distribuzione

"omogenea/lineare" lungo l'asse dei numeri reali bensì risultano "aggregati" in gruppi di valori i quali ci hanno guidato nella definizione dei range di valori per ogni fascia che abbiamo presentato nella Tabella 3. Il NMC restituisce valori in , quindi utilizzare un'equazione lineare per aggregare le stazioni in 3 fasce in base al loro valore di *exposure* risulterebbe errato. Al fine di poter valutare le performance del metodo proposto nella Sez. 2.2.2 abbiamo confrontato la classificazione in fasce delle stazioni ottenuta in base ai valori di *exposure* con la classificazione in fasce delle stazioni ferroviarie presenti sul territorio realizzata dai membri dei gruppi partecipanti al Laboratorio di Basi di Dati II nell'A.A. 2016/2017 tramite un'indagine visiva del territorio ove tali edifici sono situati. L'ipotesi di lavoro implicita è che tale classificazione "umana", indicata d'ora in avanti come classificazione ufficiale, sia corretta per definizione. Pertanto i risultati ottenuti dal metodo discusso nella Sez. 2.2.2 saranno tanto più soddisfacenti quanto più questi si avvicineranno ai dati ufficiali.

La tabella che riporta la classificazione in fasce ufficiale delle stazioni ferroviarie nella sua completezza è presente in appendice B.3, mentre qui di seguito, in tabella

[5](#), sono stati riportati i record delle stazioni che sono stati classificati ufficialmente in fascia *High*.

id Stazione	Nome Stazione Ferroviaria
44	Sant'Ilario
65	Aversa
54	Fontecchio
38	Pettorano sul Gizio
114	Acciano
62	Vigliano d'Abruzzo

Tabella 5: Elenco delle stazioni ferroviarie ufficialmente in fascia *High*

Lo studio delle performance del metodo è stato realizzato mediante la *tabella di contingenza* (detta anche *matrice di errore* o *di confusione*). Ogni colonna di tale tabella rappresenta le istanze in una fascia "prevista" (*Predicted Classes*) mentre le righe rappresentano le istanze in una fascia reale (*Actual Classes*). La corrispondente *tabella di contingenza* (Tabella 6) riassume i risultati ottenuti con il metodo.

		Predicted Classes			Totale
		High	Moderate	Low	
Actual Classes	High	4	2	0	6
	Moderate	2	53	6	61
	Low	0	7	40	47
Totale		6	62	46	114

Tabella 6: *Tabella di Contingenza* del metodo NMC discusso nella Sez. [2.2.2](#)

Osservando i dati contenuti nella Tabella 6 è possibile ricavare le seguenti deduzioni:

- Delle 6 stazioni classificate ufficialmente in fascia *High* il metodo ha previsto che 2 di queste sono in fascia *Moderate*;
- Delle 61 stazioni classificate ufficialmente in fascia *Moderate* il metodo ha previsto che 2 di queste sono in fascia *High* mentre 6 sono in fascia *Low*;
- Delle 47 stazioni classificate ufficialmente in fascia *Low* il metodo ha previsto che 7 sono in fascia *Moderate*;
- Le stazioni correttamente classificate dal nostro metodo sono localizzate lungo la diagonale della tabella.

Dall'esame della letteratura del settore del *Machine Learning* emerge che la maniera più semplice di effettuare misurazioni circa le performance della predizione restituita dal metodo adottato consiste nel ricondurre il problema della classificazione dei risultati ottenuti al caso di *tabella di contingenza binaria*, ossia al caso nel quale sia coinvolta una sola classe alla volta. Il problema può essere quindi formulato come segue:

Dati n valori (v_1, v_2, \dots, v_n) e una classe (C), costruire la tabella di contingenza binaria che riassume come detti valori sono classificati sia nel mondo reale che secondo quanto stimato dall'algoritmo del quale si vuole misurare la efficacia. Evidentemente il valore v_1 potrà cadere in C oppure no, idem per v_2, \dots, v_n .

Dalla Tabella 6 è possibile ricavare le seguenti *tabelle di contingenza* riferite relativamente alla fascia *High* (Tabella 7), alla fascia *Moderate* (Tabella 8) e alla fascia *Low* (Tabella 9).

		Predicted Classes		Totale
		High	Not High	
Actual Classes	High	4	2	6
	Not High	2	106	108
Totale		6	108	114

Tabella 7: *Tabella di contingenza binaria* riferita alla fascia *High*

Segue un commento alla Tabella 7

Prima Riga:

- Il numero 4 denota le stazioni che il metodo ha classificato correttamente in fascia *High* (da questo momento chiameremo questi casi *Veri Positivi*);
- Il numero 2 denota le stazioni che il metodo **non** ha classificato **correttamente** in fascia *High* ma in un'altra fascia (da questo momento chiameremo questi casi *Falsi Negativi*).

Seconda Riga:

- Il numero 2 denota i casi nei quali l'algoritmo ha classificato **erroneamente** delle stazioni in classe *High* (da questo momento chiameremo questi casi *Falsi Positivi*);
- Il numero 106 denota le stazioni che l'algoritmo ha classificato correttamente come **non** di classe *High* (da questo momento chiameremo questi casi *Veri Negativi*).

		Predicted Classes		Totale
		Moderate	Not Moderate	
Actual Classes	Moderate	53	8	61
	Not Moderate	9	44	53
Totale		62	52	114

Tabella 8: *Tabella di contingenza binaria riferita alla fascia Moderate*

Segue un commento alla Tabella 8

Prima Riga:

- Il numero 53 denota le stazioni che il metodo ha classificato correttamente in fascia *Moderate*;
- Il numero 8 denota le stazioni che il metodo **non** ha classificato **correttamente** in fascia *Moderate* ma in un'altra fascia.

Seconda Riga:

- Il numero 9 denota i casi nei quali l'algoritmo ha classificato **erroneamente** delle stazioni in classe *Moderate*;
- Il numero 44 denota le stazioni che l'algoritmo ha classificato correttamente come **non** di classe *Moderate*.

		Predicted Classes		Totale
		Low	Not Low	
Actual Classes	Low	40	7	47
	Not Low	6	61	67
Totale		46	68	114

Tabella 9: *Tabella di contingenza binaria riferita alla fascia Low*

Segue un commento alla Tabella 9

Prima Riga:

- Il numero 40 denota le stazioni che il metodo ha classificato correttamente in fascia *Low*;
- Il numero 7 denota le stazioni che il metodo **non** ha classificato **correttamente** in fascia *Low* ma in un'altra fascia;

Seconda Riga:

- Il numero 6 denota i casi nei quali l'algoritmo ha classificato **erroneamente** delle stazioni in classe *Low*;

- Il numero 61 denota le stazioni che l'algoritmo ha classificato correttamente come **non** di classe *Low*.

Prima di elencare le metriche che abbiamo adottato per giudicare le performance del metodo è utile introdurre la Tabella 10, nella quale andremo ad evidenziare i sei parametri che descrivono la *tabella di contingenza binaria*, ovvero: P, N, VP, FP, FN, VN, a ciascuno dei quali corrisponde un valore intero.

In letteratura sono state proposte molte metriche per giudicare le performance di un algoritmo che restituisca la realtà osservata, le più diffuse verranno elencate di seguito. Si segnala inoltre che il valore di ciascuna di esse esprime una probabilità, quindi oscilla tra 0 e 1.

		Predicted Classes		Totale
		Predizione Affermativa	Predizione Falsa	
Actual Classes	Veri Positivi (VP)	Falsi Negativi (FN)	Valori \in alla classe $P=VP + FN$	
	Falsi Positivi (FP)	Veri Negativi (VN)	Totali altri valori $N=FP+VN$	
Totale	VP + FP	FN + VN	$P + N =$ $VP + FP + FN + VN$	

Tabella 10: *Tabella di contingenza binaria "neutra"*

La **Percentuale dei Veri Positivi** (True Positive Rate) è definita come la percentuale dei casi positivi riconosciuti (dal metodo di classificazione adottato) correttamente come tali. E' auspicabile che tale valore sia prossimo a 1. In formule:

$$PVP = \frac{VP}{P} = \frac{VP}{(VP + FN)} \quad (11)$$

Dalle *tabelle di contingenza binaria* relative relativamente alla fascia *High*, alla fascia *Moderate* e alla fascia *Low* otteniamo i seguenti valori di PVP:

- PVP fascia *High* = 0,67
- PVP fascia *Moderate* = 0,87
- PVP fascia *Low* = 0,85

La **Percentuale dei Falsi Negativi** (False Negative Rate) o Percentuale dei Mancati Allarmi è definita in formule come segue:

$$PFN = \frac{FN}{P} = \frac{FN}{(VP + FN)} = 1 - PVP \quad (12)$$

Dalla formula si evince che esso è il complementare a PVP, ovvero che è auspicabile che il suo valore tenda a 0.

Dalle *tabelle di contingenza binaria* riferite relativamente alla fascia *High*, alla fascia *Moderate* e alla fascia *Low* otteniamo i seguenti valori di PFN:

- PFN fascia *High* = 0,33
- PFN fascia *Moderate* = 0,13
- PFN fascia *Low* = 0,15

La **Percentuale dei Veri Negativi** (True Negative Rate) indica la percentuale dei casi che non destano allarme. In formule:

$$PVN = \frac{VN}{N} = \frac{VN}{(FP + VN)} \quad (13)$$

Dalle *tabelle di contingenza binaria* riferite relativamente alla fascia *High*, alla fascia *Moderate* e alla fascia *Low* otteniamo i seguenti valori di PVN:

- PVN fascia *High* = 0,98
- PVN fascia *Moderate* = 0,83
- PVN fascia *Low* = 0,91

La **Percentuale dei Falsi Positivi** (False Positive Rate) indica la percentuale di falsi allarmi. In formule:

$$PFP = FP/N = FP/(FP + VN) = 1 - PVN \quad (14)$$

Dalle *tabelle di contingenza binaria* riferite relativamente alla fascia *High*, alla fascia *Moderate* e alla fascia *Low* otteniamo i seguenti valori di PFP:

- PFP fascia *High* = 0,02
- PFP fascia *Moderate* = 0,17
- PFP fascia *Low* = 0,09

La **Precisione** è definita dalla seguente formula:

$$P = \frac{VP}{(VP + FP)} \quad (15)$$

Dalle *tabelle di contingenza binaria* riferite relativamente alla fascia *High*, alla fascia *Moderate* e alla fascia *Low* otteniamo i seguenti valori di Precisione:

- Precisione fascia *High* = 0,67
- Precisione fascia *Moderate* = 0,85
- Precisione fascia *Low* = 0,87

L' **Accuratezza** è definita dalla seguente formula:

$$ACC = \frac{(VP + VN)}{(P + N)} = \frac{(VP + VN)}{((VP + FN) + (VN + FP))} \quad (16)$$

Dalle *tabelle di contingenza binaria* riferite relativamente alla fascia *High*, alla fascia *Moderate* e alla fascia *Low* otteniamo i seguenti valori di Accuratezza:

- Accuratezza fascia *High* = 0,89
- Accuratezza fascia *Moderate* = 0,85
- Accuratezza fascia *Low* = 0,89

4.1.2 *Discussione dei risultati ottenuti*

Alla luce dei risultati ottenuti dal confronto tra la classificazione in fasce ottenuta dall'elaborazione dell'output del metodo illustrato nella Sez. 2.2.2 e la classificazione ufficiale e dai valori delle metriche utilizzate per valutare le performance del metodo proposto è stato possibile concludere che l'algoritmo presenta sì alcune limitazioni che andremo a discutere qui di seguito ma che ha buone performance e riesce generalmente (nell'87% dei casi) a classificare correttamente le stazioni ferroviarie presenti sul territorio abruzzese.

Le limitazioni del metodo proposto derivano dal fatto che l'algoritmo, nel calcolare il *vettoreDirezionale*, prende in considerazione la curva di livello più vicina alla stazione ferroviaria a cui si riferisce e non sempre coincide con la direzione lungo la quale si propaga la pendenza più rilevante presente nelle vicinanze della stazione. Per maggiore chiarezza, qui di seguito, portiamo in esame due casi critici nei quali questa limitazione è ben visibile:

- **Aversa**

La stazione ferroviaria di Aversa viene classificata dal nostro metodo in fascia *Moderate* mentre *ufficialmente* viene classificata in fascia *High*. In figura 35 è visibile come il *vettoreDirezionale* viene costruito considerando la curva di livello più vicina (a 68m) al punto che indica la stazione di Aversa mentre il pendio con una pendenza maggiore si riferisce alla curva di livello che dista 93m dal punto. Ragion per cui, il valore di *exposure* ottenuto applicando il metodo proposto nella Sez. 2.2.2 per questa stazione ferroviaria è tale da classificare erroneamente tale edificio in fascia *Moderate*.

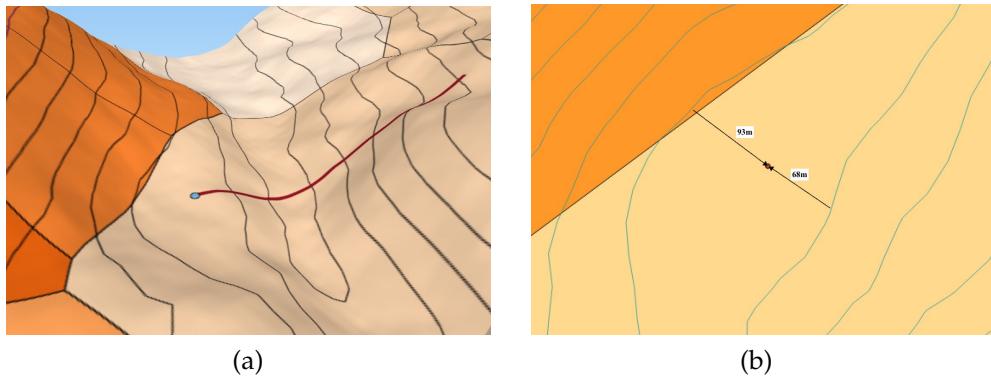


Figura 35: Stazione Ferroviaria di Aversa

- **Vigliano d'Abruzzo**

La stazione ferroviaria di Vigliano d'Abruzzo viene classificata dal nostro metodo in fascia *Moderate* mentre *ufficialmente* viene classificata in fascia *High*. In figura 36 è visibile come il *vettoreDirezionale* viene costruito considerando la curva di livello più vicina (a 38m) al punto che indica la stazione di Aversa mentre il pendio con una pendenza maggiore si riferisce alla curva di livello che dista 56m dal punto. Ragion per cui, il valore di *exposure* ottenuto applicando il metodo proposto nella Sez. 2.2.2 per questa stazione ferroviaria è tale da classificare erroneamente tale edificio in fascia *Moderate*.

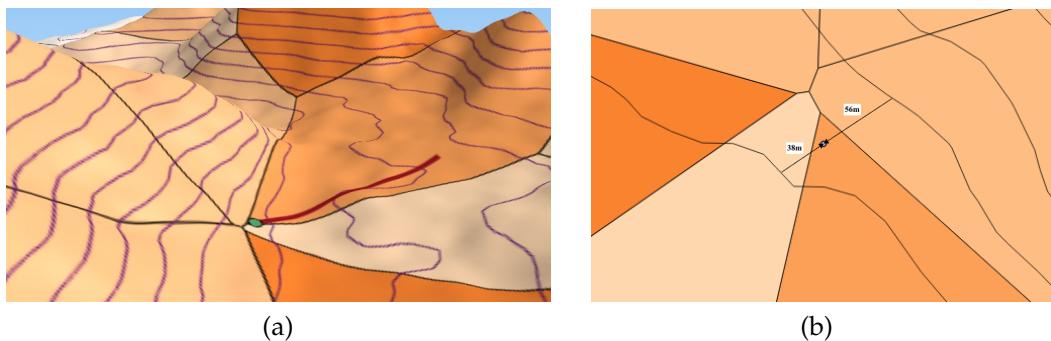


Figura 36: Stazione Ferroviaria di Vigliano d'Abruzzo

4.2 RANKING E VALUTAZIONE DELLE LINEE FERROVIARIE

In questa sezione andremo a discutere i risultati ottenuti applicando il metodo NMCE presentato in Sez. 2.3.1 per la determinazione della classifica delle linee ferroviarie in relazione al grado di esposizione al rischio idrogeologico.

Il metodo ha restituito 690 segmenti in quanto abbiamo definito la lunghezza di ognuno pari a 1 km ed il valore di *exposure* di ognuno varia da un valore massimo di 130 ad un valore minimo di 0. A fronte di tali risultati abbiamo ritenuto opportuno, per analogia al caso delle stazioni ferroviarie, utilizzare il medesimo meccanismo di classificazione in tre fasce di rischio:

- High
- Moderate
- Low

La Tabella 11 mostra la suddetta divisione in fasce di rischio, gli intervalli in cui i segmenti sono stati divisi secondo il valore di *exposure*, il numero e la percentuale sul totale dei segmenti appartenenti ad una determinata fascia. Come nel caso delle stazioni ferroviarie ad ogni fascia abbiamo associato un colore identificativo.

Fascia	Intervallo valori Exp_segmento	#segmenti	%
High	$20 < \text{Exp_}b_i$	78	11,31%
Moderate	$1 < \text{Exp_}b_i < 20$	386	55,94%
Low	$0 < \text{Exp_}b_i < 1$	226	32,75%

Tabella 11: Aggregazione dei segmenti in base al valore di *exposure*

Come è possibile notare chiaramente dai dati riportati in Tabella 11 solo una bassa percentuale de segmenti presenta un livello di esposizione al rischio di frane tale da far emergere il bisogno di controlli più frequenti (livello di esposizione *High*). Da tali risultati si ha quindi la conferma dell'utilità del lavoro esposto, in particolare per gli addetti al controllo della rete ferroviaria: viene ridotto al 11,31% il numero di segmenti di rete ferroviaria da manutenere più frequentemente; ciò comporta senza ombra di dubbio una riduzione, in termini di costi e di tempi, per gli addetti ai lavori.

In Figura 37 è possibile avere una visione d'insieme dei risultati ottenuti dall'applicazione del metodo NMCE sui segmenti delle linee che compongono la rete ferroviaria abruzzese, si noti come ogni segmento in figura ha lo stesso colore della fascia di rischio a cui appartiene.

La classifica riportata in Tabella 12 mostra i primi venti (di 690) segmenti ordinati in maniera decrescente rispetto il relativo valore di *exposure*. In riferimento al singolo segmento, per ognuno di essi vengono mostrati la linea di appartenenza, il numero del segmento e il relativo valore di *exposure*.



Figura 37: Visione d'insieme della classificazione in fasce di rischio dei segmenti della rete ferroviaria abruzzese

n.	Linea Ferroviaria	# Segmento	Valore di Exp. del Segmento
1	Sulmona - L'Aquila - Rieti	15	130, 14
2	Sulmona - L'Aquila - Rieti	14	107, 79
3	Sulmona - L'Aquila - Rieti	17	103, 77
4	Sulmona - L'Aquila - Rieti	16	93, 95
5	Sulmona - Carpinone	57	82, 19
6	Pescara - Roma	47	76, 54
7	Pescara - Roma	46	74, 56
8	Sulmona - L'Aquila - Rieti	13	71, 89
9	Sulmona - Carpinone	10	64, 49
10	Pescara - Roma	158	62, 02
11	Sulmona - Carpinone	58	61, 57
12	Pescara - Roma	78	59, 00
13	Sulmona - Carpinone	56	55, 16
14	Sulmona - L'Aquila - Rieti	37	53, 91
15	Pescara - Roma	50	51, 42
16	Roccasecca - Avezzano	35	50, 76
17	Sulmona - L'Aquila - Rieti	23	49, 65
18	Sulmona - Carpinone	62	48, 71
19	Sulmona - Carpinone	36	48, 04
20	Sulmona - Carpinone	41	47, 93

Tabella 12: Classifica generale dei primi venti (di 690) segmenti con relativo valore di *exposure* e linea di appartenenza

Si espone qui di seguito un'analisi puntuale dei risultati per ogni linea ferroviaria.

4.2.1 Linea Bologna - Bari

Si riportano in Tabella 13 i primi dieci (di 125) segmenti di tale tratta ferroviaria ordinati in maniera decrescente rispetto il valore di *exposure*.

n.	# Segmento	Valore di Exp. del Segmento
1	41	8,59
2	90	6,43
3	40	6,07
4	53	5,05
5	42	4,84
6	91	4,82
7	116	4,78
8	84	4,70
9	85	4,57
10	86	4,27

Tabella 13: Classifica dei primi dieci (di 125) segmenti appartenenti alla linea Bologna - Bari

La Figura 38 mostra invece una visione d'insieme della classificazione in fasce di rischio dei segmenti che compongono la linea.

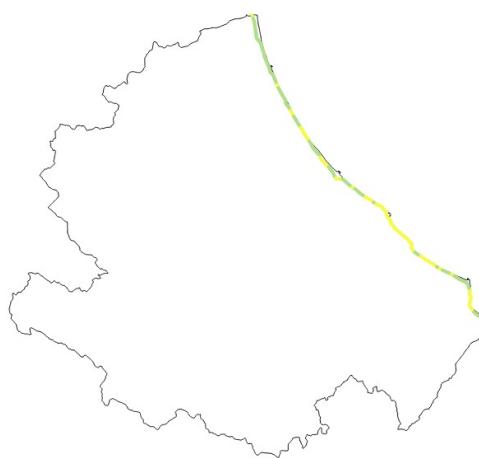


Figura 38: Visione d'insieme della classificazione in fasce di rischio dei segmenti della linea Bologna - Bari

Analizzando i risultati esposti in Tabella 14 si evince che in questa linea non ci sono segmenti che destano preoccupazione circa il rischio di smottamenti. Per una visione

completa della classificazione dei segmenti della linea Bologna - Bari si rimanda alla lettura dell'appendice [B.2.1](#).

Fascia	# Segmenti	%
High	0	0
Moderate	54	43,20
Low	71	56,80

Tabella 14: Analisi statistica della classificazione in fasce di rischio dei segmenti della linea Bologna - Bari

4.2.2 Linea Ortona - Crocetta

Si riportano in Tabella 15 i primi dieci (di 36) segmenti di tale tratta ferroviaria ordinati in maniera decrescente rispetto il valore di *exposure*.

n.	# Segmento	Valore di Exp. del Segmento
1	23	10,22
2	22	9,38
3	32	5,38
4	25	5,31
5	24	5,18
6	31	5,10
7	33	5,01
8	36	4,39
9	35	4,13
10	27	3,92

Tabella 15: Classifica dei primi dieci (di 36) segmenti appartenenti alla linea Ortona - Crocetta

La Figura 39 mostra invece una visione d'insieme della classificazione in fasce di rischio dei segmenti che compongono la linea.

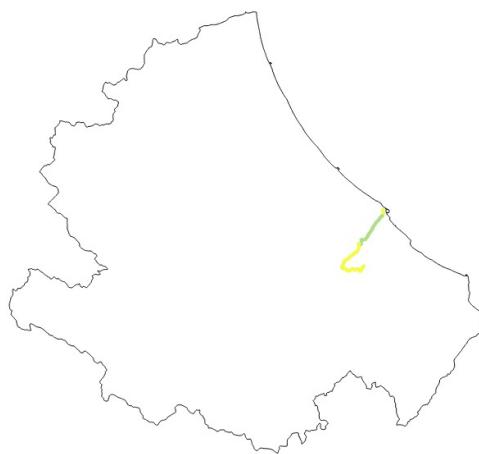


Figura 39: Visione d'insieme della classificazione in fasce di rischio dei segmenti della linea Ortona - Crocetta

Analizzando i risultati esposti in Tabella 16 si evince che in questa linea non ci sono segmenti che destano preoccupazione circa il rischio di smottamenti. Per una visione completa della classificazione dei segmenti della linea Ortona - Crocetta si rimanda alla lettura dell'appendice B.2.2.

Fascia	# Segmenti	%
High	0	0
Moderate	54	43,20
Low	71	56,80

Tabella 16: Analisi statistica della classificazione in fasce di rischio dei segmenti della linea Ortona - Crocetta

4.2.3 Linea Marina di San Vito - Castel di Sangro

Si riportano in Tabella 17 i primi dieci (di 125) segmenti di tale tratta ferroviaria ordinati in maniera decrescente rispetto il valore di *exposure*.

n.	# Segmento	Valore di Exp. del Segmento
1	64	20,43
2	78	17,92
3	79	17,10
4	73	17,04
5	68	15,80
6	76	15,55
7	56	15,54
8	77	14,32
9	70	14,28
10	75	13,91

Tabella 17: Classifica dei primi dieci (di 105) segmenti appartenenti alla linea Marina di San Vito - Castel di Sangro

La Figura 40 mostra invece una visione d'insieme della classificazione in fasce di rischio dei segmenti che compongono la linea.

Analizzando i risultati esposti in Tabella 18 si evince che in questa linea solo il 0,95% dei segmenti si trova in fascia *High* e quindi deve essere sottoposto a controlli più frequenti vista la sua esposizione al rischio idrogeologico. Per una visione completa della classificazione dei segmenti della linea Marina di San Vito - Castel di Sangro si rimanda alla lettura dell'appendice B.2.3.



Figura 40: Visione d'insieme della classificazione in fasce di rischio dei segmenti della linea Marina di San Vito - Castel di Sangro

Fascia	# Segmenti	%
High	1	0,95
Moderate	87	82,86
Low	17	16,19

Tabella 18: Analisi statistica della classificazione in fasce di rischio dei segmenti della linea Marina di San Vito - Castel di Sangro

4.2.4 Linea Pescara - Roma

Si riportano in Tabella 19 i primi dieci (di 170) segmenti di tale tratta ferroviaria ordinati in maniera decrescente rispetto il valore di *exposure*.

n.	# Segmento	Valore di Exp. del Segmento
1	47	76,54
2	46	74,56
3	158	62,02
4	78	59,00
5	50	46,78
6	88	45,87
7	157	45,87
8	48	45,59
9	98	38,57
10	89	36,85

Tabella 19: Classifica dei primi dieci (di 170) segmenti appartenenti alla linea Pescara - Roma

La Figura 41 mostra invece una visione d'insieme della classificazione in fasce di rischio dei segmenti che compongono la linea.

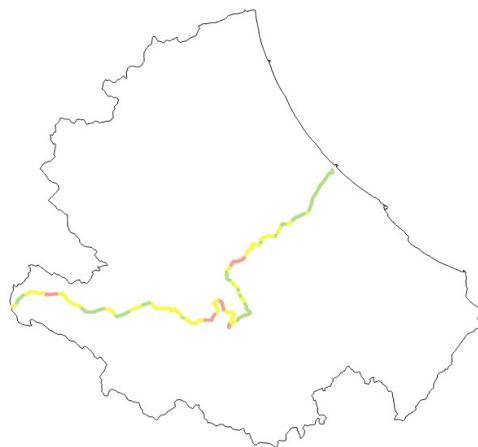


Figura 41: Visione d'insieme della classificazione in fasce di rischio dei segmenti della linea Pescara - Roma

Analizzando i risultati esposti in Tabella 20 si evince che in questa linea solo il 12,94% dei segmenti si trova in fascia *High* e quindi deve essere sottoposto a controlli più frequenti vista la sua esposizione al rischio idrogeologico. Per una visione completa della classificazione dei segmenti della linea Pescara - Roma si rimanda alla lettura dell'appendice B.2.4.

Fascia	# Segmenti	%
High	22	12,94
Moderate	87	51,18
Low	61	35,88

Tabella 20: Analisi statistica della classificazione in fasce di rischio dei segmenti della linea Pescara - Roma

4.2.5 Linea Roccasecca - Avezzano

Si riportano in Tabella 21 i primi dieci (di 46) segmenti di tale tratta ferroviaria ordinati in maniera decrescente rispetto il valore di *exposure*.

n.	# Segmento	Valore di Exp. del Segmento
1	35	50,76
2	42	36,80
3	36	34,80
4	37	34,77
5	32	27,77
6	24	25,48
7	33	24,48
8	30	20,27
9	43	19,46
10	34	19,14

Tabella 21: Classifica dei primi dieci (di 46) segmenti appartenenti alla linea Roccasecca - Avezzano

La Figura 42 mostra invece una visione d'insieme della classificazione in fasce di rischio dei segmenti che compongono la linea.

Analizzando i risultati esposti in Tabella 22 si evince che in questa linea solo il 17,39% dei segmenti si trova in fascia *High* e quindi deve essere sottoposto a controlli più frequenti vista la sua esposizione al rischio idrogeologico. Per una visione completa della classificazione dei segmenti della linea Roccasecca - Avezzano si rimanda alla lettura dell'appendice B.2.5.



Figura 42: Visione d'insieme della classificazione in fasce di rischio dei segmenti della linea Roccasecca - Avezzano

Fascia	# Segmenti	%
High	8	17,39
Moderate	35	76,09
Low	3	6,52

Tabella 22: Analisi statistica della classificazione in fasce di rischio dei segmenti della linea Roccasecca - Avezzano

4.2.6 Linea Archi Stazione - Atessa

Si riportano in Tabella 23 i primi dieci (di 15) segmenti di tale tratta ferroviaria ordinati in maniera decrescente rispetto il valore di *exposure*.

La Figura 43 mostra invece una visione d'insieme della classificazione in fasce di rischio dei segmenti che compongono la linea.

Analizzando i risultati esposti in Tabella 24 si evince che in questa linea non ci sono segmenti che destano preoccupazione circa il rischio di smottamenti. Per una visione completa della classificazione dei segmenti della linea Archi Stazione - Atessa si rimanda alla lettura dell'appendice B.2.6.

n.	# Segmento	Valore di Exp. del Segmento
1	14	4,63
2	15	4,20
3	2	3,75
4	13	3,39
5	3	2,47
6	12	2,39
7	1	2,04
8	11	1,85
9	10	1,70
10	7	1,53

Tabella 23: Classifica dei primi dieci (di 15) segmenti appartenenti alla linea Archi Stazione - Atessa



Figura 43: Visione d'insieme della classificazione in fasce di rischio dei segmenti della linea Archi Stazione - Atessa

Fascia	# Segmenti	%
High	0	0
Moderate	15	100
Low	0	0

Tabella 24: Analisi statistica della classificazione in fasce di rischio dei segmenti della linea Archi Stazione - Atessa

4.2.7 Linea Sulmona - Carpinone

Si riportano in Tabella 25 i primi dieci (di 86) segmenti di tale tratta ferroviaria ordinati in maniera decrescente rispetto il valore di *exposure*.

n.	# Segmento	Valore di Exp. del Segmento
1	57	82,19
2	10	64,49
3	58	61,57
4	56	55,16
5	62	48,71
6	36	48,04
7	42	47,93
8	18	42,72
9	12	41,53
10	19	39,95

Tabella 25: Classifica dei primi dieci (di 86) segmenti appartenenti alla linea Sulmona - Carpinone

La Figura 44 mostra invece una visione d'insieme della classificazione in fasce di rischio dei segmenti che compongono la linea.

Analizzando i risultati esposti in Tabella 26 si evince che in questa linea solo il 32,56% dei segmenti si trova in fascia *High* e quindi deve essere sottoposto a controlli più frequenti vista la sua esposizione al rischio idrogeologico. Per una visione completa della classificazione dei segmenti della linea Sulmona - Carpinone si rimanda alla lettura dell'appendice B.2.7.

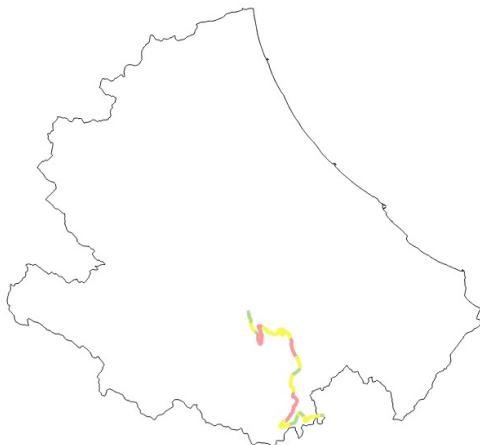


Figura 44: Visione d'insieme della classificazione in fasce di rischio dei segmenti della linea Sulmona - Carpinone

Fascia	# Segmenti	%
High	28	32,56
Moderate	44	51,16
Low	14	16,28

Tabella 26: Analisi statistica della classificazione in fasce di rischio dei segmenti della linea Sulmona - Carpinone

4.2.8 Linea Sulmona - L'Aquila - Rieti

Si riportano in Tabella 27 i primi dieci (di 81) segmenti di tale tratta ferroviaria ordinati in maniera decrescente rispetto il valore di *exposure*.

n.	# Segmento	Valore di Exp. del Segmento
1	56	130,14
2	55	107,79
3	63	103,77
4	62	93,95
5	53	71,89
6	42	53,91
7	71	49,65
8	40	44,97
9	70	40,37
10	32	39,55

Tabella 27: Classifica dei primi dieci (di 81) segmenti appartenenti alla linea Sulmona - L'Aquila - Rieti

La Figura 45 mostra invece una visione d'insieme della classificazione in fasce di rischio dei segmenti che compongono la linea. Questa è la tratta che tra tutte presenta i valori di *exposure* dei segmenti più alti.

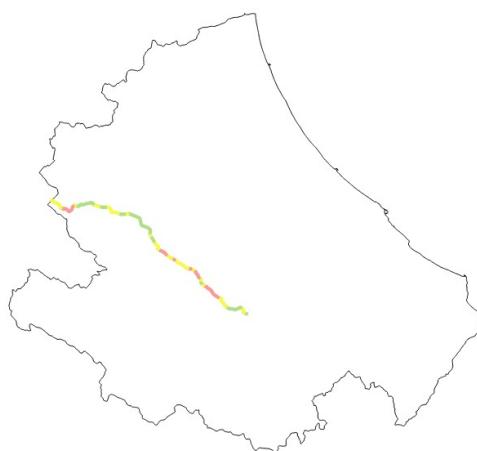


Figura 45: Visione d'insieme della classificazione in fasce di rischio dei segmenti della linea Sulmona - L'Aquila - Rieti

Analizzando i risultati esposti in Tabella 28 si evince che in questa linea solo il 23,46% dei segmenti si trova in fascia *High* e quindi deve essere sottoposto a controlli più frequenti vista la sua esposizione al rischio idrogeologico. Per una visione completa

della classificazione dei segmenti della linea Sulmona - L'Aquila - Rieti si rimanda alla lettura dell'appendice [B.2.9](#).

Fascia	# Segmenti	%
High	19	23,46
Moderate	33	40,74
Low	29	35,80

Tabella 28: Analisi statistica della classificazione in fasce di rischio dei segmenti della linea Sulmona - L'Aquila - Rieti

4.2.9 Linea Giulianova - Teramo

Si riportano in Tabella 29 i primi dieci (di 26) segmenti di tale tratta ferroviaria ordinati in maniera decrescente rispetto il valore di *exposure*.

n.	# Segmento	Valore di Exp. del Segmento
1	17	2,99
2	14	2,19
3	23	2,15
4	13	1,96
5	16	1,72
6	15	1,40
7	22	1,24
8	24	1,12
9	25	0,80
10	12	0,72

Tabella 29: Classifica dei primi dieci (di 26) segmenti appartenenti alla linea Giulianova - Teramo

La Figura 46 mostra invece una visione d'insieme della classificazione in fasce di rischio dei segmenti che compongono la linea.

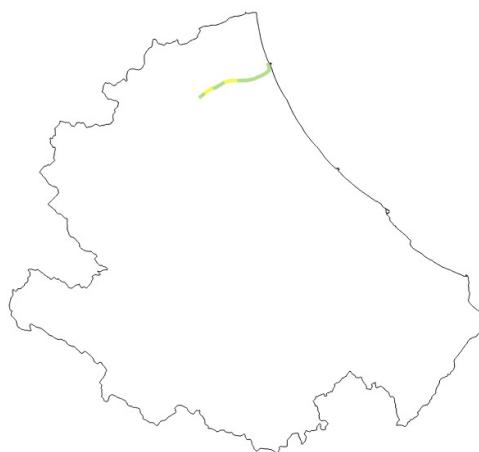


Figura 46: Visione d'insieme della classificazione in fasce di rischio dei segmenti della linea Giulianova - Teramo

Analizzando i risultati esposti in Tabella 30 si evince che in questa linea non ci sono segmenti che destano preoccupazione circa il rischio di smottamenti. Per una visione completa della classificazione dei segmenti della linea Giulianova - Teramo si rimanda alla lettura dell'appendice B.2.8.

Fascia	# Segmenti	%
High	0	0
Moderate	8	30,77
Low	18	69,23

Tabella 30: Analisi statistica della classificazione in fasce di rischio dei segmenti della linea Giulianova - Teramo

4.2.10 *Discussione dei risultati ottenuti*

Per la validazione dei risultati ottenuti dall'applicazione del metodo NMCE, discusso nella Sezione 2.3.1, sono stati utilizzati il software QGIS, presentato in appendice A.2, e GoogleMaps, con lo scopo di verificare, almeno dal punto di vista visivo, la classificazione in fasce di rischio ottenuta. Un ulteriore controllo è stato realizzato confrontando la fascia di rischio associata al segmento della linea ferroviaria e la fascia di rischio associata alla stazione che insiste su di esso: In alcuni casi la fascia del segmento è in contrasto con la fascia assegnata alla stazione ferroviaria dal NMC. Questi casi corrispondono ai casi critici di cui abbiamo ampiamente parlato nella sezione 4.1.2. Ad esempio la stazione di Aversa sappiamo essere stata classificata erroneamente nella fascia *Moderate* dal NMC. Invece il NMCE classifica i segmenti in prossimità della stazione nella fascia *High* come si evince dalla Fig.47

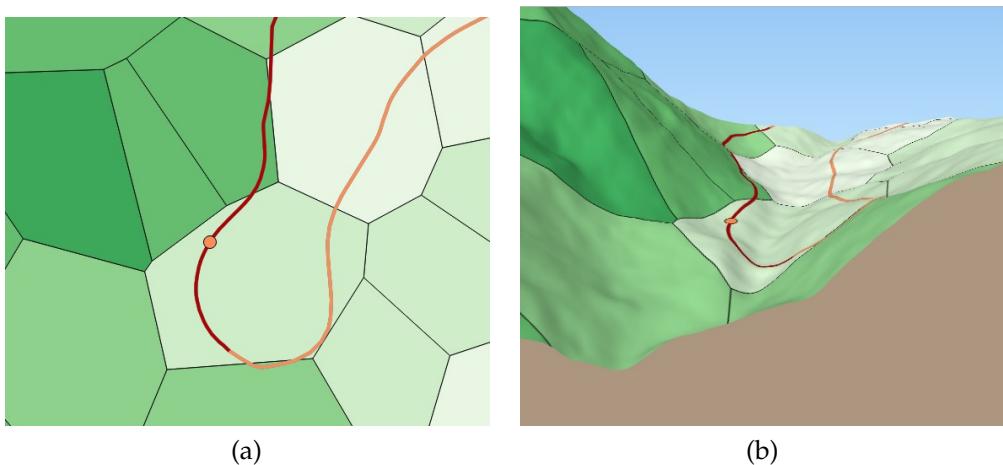


Figura 47: Vista 2D (a) e 3D (b) della stazione di Aversa e della linea ferroviaria che l'attraversa

Il metodo NMCE presenta inoltre una criticità che determina in alcuni casi la presenza di *falsi positivi*: L'assenza di un dataset che informi circa la presenza di gallerie ferroviarie: Il metodo, non avendo conoscenza della loro presenza o meno, non riesce ad individuare le situazioni in cui i binari della linea ferroviaria proseguono all'interno di una galleria e crede che la tratta prosegua arrampicandosi sul pendio, generando così valori di *exposure* errati. In Figura 48 è possibile osservare una di queste situazioni

critiche, corrispondenti alla tratta ferroviaria Pescara - Roma, dal km 154 al km 157 ovvero la galleria *Colli di Monte Bove* nei pressi dell'omonima stazione ferroviaria, mentre in Figura 49 è possibile osservare la stessa tratta fotografata tramite GoogleMaps.

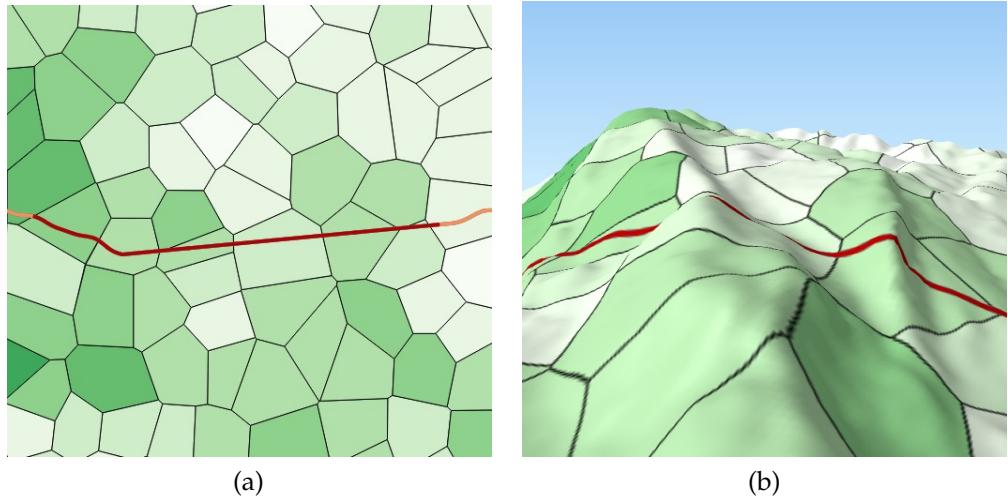


Figura 48: Vista 2D (a) e 3D (b) della tratta Pescara - Roma dal km 154 al 157



Figura 49: Instantanea di GoogleMaps della galleria *Colli di Monte Bove*

CONSIDERAZIONI FINALI E PROPOSTE PER LAVORI FUTURI

In questo capitolo andremo a riportare alcune considerazioni circa i metodi discussi per il ranking delle tratte ferroviarie e delle stazioni che su di esse insistono al fine di proporre spunti per i lavori futuri. Inoltre verranno evidenziate alcune inesattezze riscontrate nei dataset utilizzati.

In conclusione è presente un riepilogo del lavoro svolto.

5.1 PROPOSTE PER LAVORI FUTURI SUL METODO DI RANKING DELLE STAZIONI FERROVIARIE

5.2 PROPOSTE PER LAVORI FUTURI SUL METODO DI RANKING DELLE TRATTE FERROVIARIE

5.3 CONSIDERAZIONI SUI DATASET

Nella valutazione delle performance dei metodi proposti non è possibile prescindere dalla consapevolezza che i dataset utilizzati potrebbero essere incompleti o potrebbero presentare imprecisioni. Uno spunto per dei lavori futuri potrebbe essere quello di utilizzare un dataset della rete ferroviaria e delle stazioni che insistono su di essa aggiornato, tale da limitare le inesattezze dei risultati. Infatti, in seguito ad accurate ricerche effettuate in rete, risulta evidente che alcune tratte, nonchè alcune stazioni situate lungo di esse, sono inattive o in fase di smantellamento. I portali che abbiamo utilizzato maggiormente sono quello proposto da Trenitalia (<http://www.rfi.it/rfi/LINEE-STAZIONI-TERRITORIO/Nelle-regioni/Abruzzo/La-rete-oggi-in:-Abruzzo>) e <http://www.ferrovieabbandonate.it/>.

Qui di seguito alcune foto di stazioni ferroviarie ormai dismesse o in disuso: Archi (Fig. 50) , Perano (Fig. 51), Pettorano sul Gizio (Fig. 52) e Civitaluparella (Fig. 53).



Figura 50: Stazione di Archi



Figura 51: Stazione di Perano



Figura 52: Stazione di Pettorano



Figura 53: Stazione di Civitaluparella

E' necessaria inoltre un'altra riflessione circa i dataset utilizzati, quella circa il dataset delle zone in cui è stato deframmentato il territorio abruzzese. Il dataset in nostro possesso è l'output di un metodo che sfrutta un algoritmo che si basa sul *Diagramma di Voronoi* per decomporre il territorio, per cui le zone ottenute potrebbero non rispecchiare in maniera coerente le caratteristiche morfologiche del terreno. Bisogna inoltre considerare che il valore numerico che quantifica la pericolosità di frana di ogni zona è strettamente legato alle curve di livello che sono presenti in tale zona e non tiene conto di quelle che sono le caratteristiche litologiche del terreno. Uno spunto per i lavori futuri potrebbe essere quello di utilizzare un dataset delle zone che presenta maggiore affidabilità circa la morfologia e le caratteristiche del territorio.

5.4 CONSIDERAZIONI FINALI

L'obiettivo di questo studio è stato fornire una classifica circa le stazioni e le linee ferroviarie in base al loro livello di esposizione al rischio frane. Sono stati proposti rispettivamente due metodi al fine di raggiungere tale scopo. Successivamente è stata progettata una base di dati spaziale atta a contenere i dataset presi in input. Nel nostro caso di studio questi ultimi hanno interessato la regione Abruzzo. La base di dati spaziale è stata implementata tramite il DBMS PostgreSQL servendoci dell'estensione spaziale PostGIS. Sono state implementate delle UDF, e definite apposite query, tramite le quali poter raggiungere l'obiettivo preposto. Dopo un'analisi dei risultati ottenuti, si è proceduto alla discussione circa le criticità dei metodi e dei dataset in input. E' opportuno sottolineare come i metodi proposti non abbiano la pretesa di restituire valori assoluti circa l'esposizione al pericolo di frane, bensì quello di fornire una metodologia valida applicabile e ovviamente migliorabile per determinare

delle classifiche grazie alle quali è possibile individuare le stazioni e le tratte ferrovia-
rie che sono maggiormente esposte al pericolo di smottamenti. I risultati finali sono
migliorabili sulla base degli spunti offerti durante la trattazione.

Parte I
APPENDICI

APPENDICE A

A.1 DATASET DELLE ZONE SVILUPPATO DAL GRUPPO

Prima di approdare al dataset delle zone di cui abbiamo discusso nella Sez. 3.3, abbiamo ipotizzato e implementato un metodo in grado di restituirci un dataset delle zone in cui suddividere l’Abruzzo prendendo come punto di partenza il dataset delle zone prodotto dall’esecuzione della seguente *query* su PostGreSQL:

```
CREATE TABLE GeoArea_split8 (
    id serial PRIMARY KEY,
    geom geometry (Polygon, 3004))
```

```
INSERT INTO GeoArea_split8 (geom)
SELECT ST_Subdivide(geom,8)
FROM Geo_Area
```

L’invocazione della funzione **ST_Subdivide()** sulla geometria che descrive *GeoArea* genera 24142 zone in cui è suddiviso il territorio abruzzese, di cui è possibile avere una visione d’insieme in Fig. 54. L’esame della Fig. 55 fornisce una visione più nitida

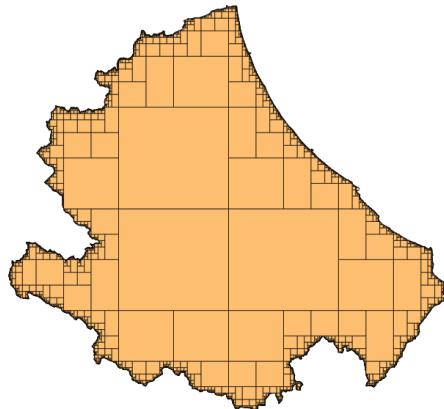


Figura 54: Visione d’insieme delle 24142 zone generate

del tipo di suddivisione effettuata da **ST_Subdivide()**. Esso in prossimità del *boundary* della *GeoArea* costruisce poligoni di forma irregolare aventi al massimo otto vertici, mentre tutte le altre aree sono dei quadrati di area crescente via via che ci si sposta verso il centro della *GeoArea*. Al fine di realizzare un’analisi quantitativa dell’esito del partizionamento della *GeoArea* si riportano in tabella 32 e in tabella 31 rispettivamente le dieci aree con area maggiore e le dieci con aree minore.

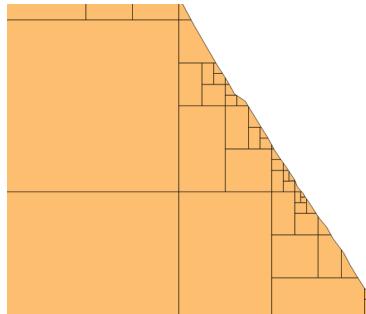


Figura 55: Particolare delle zone in prossimità del *boundary*

Area (kmq)	Perimetro (km)
$3,46 \times 10^{-11}$	$3,55 \times 10^{-5}$
$1,14 \times 10^{-10}$	$5,74 \times 10^{-5}$
$3,11 \times 10^{-10}$	$8,64 \times 10^{-5}$
$3,21 \times 10^{-10}$	$8,84 \times 10^{-5}$
$6,29 \times 10^{-10}$	$19,22 \times 10^{-5}$
$6,46 \times 10^{-10}$	$14,01 \times 10^{-5}$
$6,86 \times 10^{-10}$	$14,38 \times 10^{-5}$
$8,90 \times 10^{-10}$	$16,65 \times 10^{-5}$
$11,81 \times 10^{-10}$	$19,91 \times 10^{-5}$
$13,05 \times 10^{-10}$	$17,51 \times 10^{-5}$

Tabella 31: Top 10 aree più piccole

Come si può vedere chiaramente i valori del perimetro e dell'area delle varie zone presentano davvero molti ordini di grandezza di differenza, ragion per cui è necessario un duplice approccio di elaborazione allo scopo di ottenere un dataset delle zone tale per cui le varie zone abbiano un'estensione quantomeno paragonabile tra loro:

- Frazionamenti successivi dei poligoni interni "enormi" ;
- Accorpamenti successivi dei poligoni lungo il bordo.

A.1.1 *Processamento dei poligoni interni*

Al fine di realizzare un algoritmo in grado di realizzare dei frazionamenti successivi dei poligoni interni "enormi" presenti nel dataset delle zone che abbiamo preso come punto di partenza abbiamo realizzato uno studio della funzione **ST_Subdivide()**, con lo scopo di comprendere quali sono i suoi limiti applicativi. Da questo studio è emerso che tale funzione risulta essere inapplicabile per 1257 delle 24142 zone in cui abbiamo diviso il territorio. Questi 1257 elementi presentano un'area maggiore di 0,1 kmq e hanno numero di vertici inferiore ad 8, motivo per cui non sono processabili

Area (kmq)	Perimetro (km)
1231,2	140,5
1231,2	140,5
1231,2	140,5
307,8	70,2
307,8	70,2
307,8	70,2
307,8	70,2
307,8	70,2
307,8	70,2
307,8	70,2

Tabella 32: Top 10 aree più grandi

ulteriormente tramite la funzione `ST_Subdivide()`.

Abbiamo ricercato dunque un'altra funzione che potesse venirci in aiuto per raggiungere il nostro obiettivo e consultando il manuale PostGIS 2.3.2dev abbiamo ritenuto opportuno utilizzare la funzione `ST_Segmentize` per processare queste 1257 zone "problematiche". Applicando tale funzione abbiamo ottenuto 37875 zone geometricamente omogenee la cui area massima è pari a 0,5 kmq. E' stato necessario, di conseguenza, implementare un meccanismo di aggregazione di tali zone con lo scopo di ottenere zone geometricamente eterogenee. Il metodo di aggregazione che abbiamo proposto e implementato sfrutta approccio *random*:

Per 1/3 delle zone, scelte in maniera random tra tutte le zone "problematiche", abbiamo utilizzato la funzione `ST_Buffer()`, con centro situato nel centroide della zona e raggio pari a 400 m, per individuare le zone confinanti con essa, candidate ad essere aggregate; Tra tutte le zone candidate ad essere aggregate, individuo in maniera random una di queste e tramite la funzione `ST_Relate` determino se esse sono effettivamente confinanti. Nel caso in cui la loro intersezione è una linea/geometria di dimensione 1, avviene effettivamente l'aggregazione. Questa operazione di aggregazione avviene un massimo di tre volte e porta da una situazione come quella rappresentata in Fig 56a a una situazione come quella visibile in Fig 56b.

A.1.2 Processamento dei poligoni lungo il bordo

Come abbiamo già discusso nella Sez. A.1 le zone presenti sul bordo hanno un'area distante molti ordini di grandezza rispetto l'obiettivo che ci siamo prefissati. Di conseguenza è stato necessario implementare un meccanismo di aggregazione di tali zone con lo scopo di ottenere delle aree con estensione più accettabile e vicina al valore medio che ci siamo posti come obiettivo:

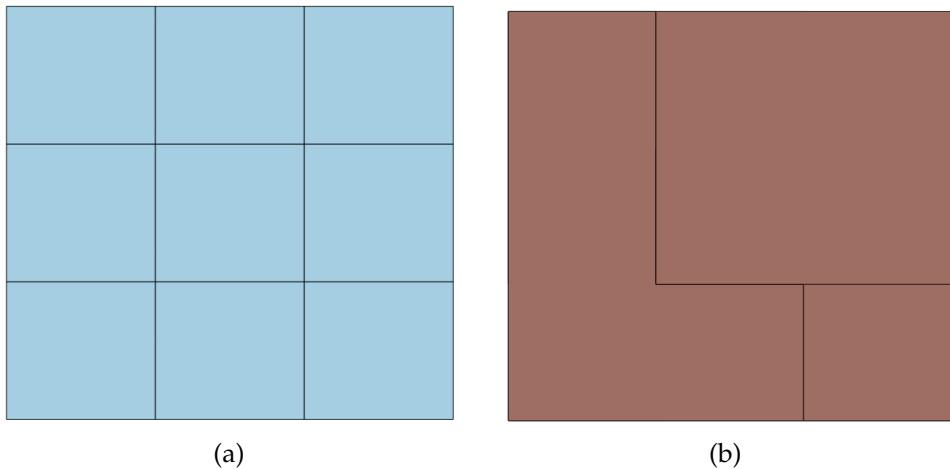


Figura 56: Zone prima dell'aggregazione (a) e zone dopo l'aggregazione (b)

- Per ogni zona problematica del bordo, che andremo a chiamare z_{bordo} , andiamo a realizzare un buffer con un certo raggio (che a livello implementativo abbiamo fissato a 150m) con lo scopo di costruire un cerchio.
- Tale cerchio verrà intersecato, tramite la funzione **ST_Intersect()**, con tutte le zone problematiche del bordo al fine di individuare quali sono le zone che intersecano la sua geometria, che andremo a chiamare z_{cerchio} .
- Per ogni z_{cerchio} , sfruttando la funzione **ST_Relate** e la definizione di *DE-9IM matrix pattern*, riusciamo a stabilire se risulta essere adiacente con la zona z_{bordo} (la funzione **ST_Relate** restituirà *true* o *false* se l'intersezione tra le due zone, rappresentate come poligoni, è una linea/geometria di dimensione 1).
- Una volta individuate tutte le zone adiacenti alla zona presa in esame, che indicheremo con $z_{\text{adiacente}}$, realizzo un'operazione di aggregazione ovvero unisco la z_{bordo} con la $z_{\text{adiacente}}$ più piccola. Se dall'unione risulta che l'area è ancora troppo piccola ripeto l'operazione di aggregazione.

Dopo il processamento i poligoni lungo il bordo vengono quindi aggregati passando da una situazione come quella mostrata in Fig 57a a una situazione come quella visibile in Fig 57b.

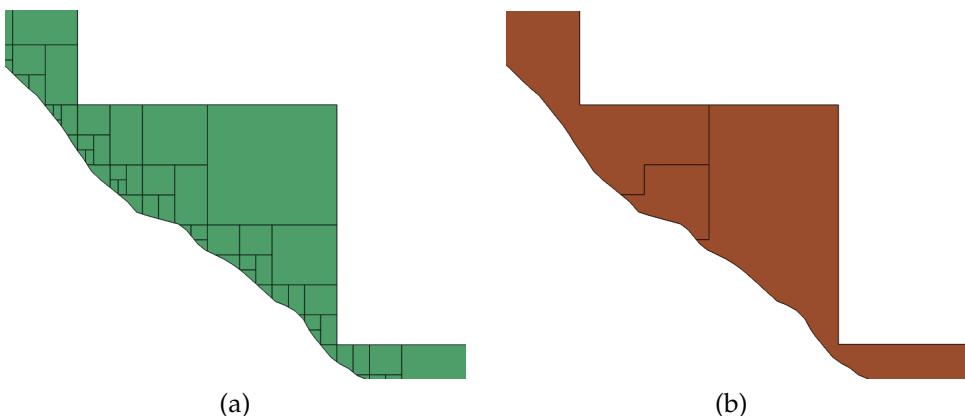


Figura 57: Zone prima dell'aggregazione (a) e zone dopo l'aggregazione (b)

A.1.3 Conclusioni

Le operazioni realizzate per realizzare il dataset così come le abbiamo pensate, risultano essere molto pesanti e i tempi di esecuzione sono molto lunghi, ragion per cui non sono state mai eseguite per intero ma per alcune aree.

A.2 QUANTUMGIS

Uno strumento software che è risultato indispensabile nello studio metodologico-sperimentale descritto in questo documento è il software QuantumGIS (QGIS) nella versione 2.18.2. Trattasi di un software open source, scritto in C+ e distribuito sotto licenza GNU General Public License, orientato ad agevolare coloro che hanno esigenza di operare su dati geografici/territoriali. La versione che abbiamo utilizzato supporta moltissimi formati raster e vettoriali, supporto che si amplia con l'uso dei plugins esterni, come ad esempio Qgis2threejs e QuickWKT.

Il plugin **Qgis2threejs** è un visualizzatore 3D basato sulla tecnologia WebGL e sulla libreria Javascript three.js che ci è risultato molto utile nello studio della morfologia del terreno e di conseguenza nella validazione dei risultati ottenuti dai metodi descritti in questo documento. Un altro plugin che ci è risultato utile nello studio è **QuickWKT**, ovvero un plugin in grado di mostrare facilmente i dati WKT / WKB in QGIS.

L'utilità di tale GeoViewer è evidente in quanto i dati vettoriali e raster e le tabelle spaziali utilizzati dalle *query* implementate su PostGreSQL sono difficilmente interpretabili e risulta quindi evidente la necessità di avere un tool in grado di visualizzarli. QGIS utilizza la libreria OGT per leggere e creare vettori (comunemente chiamati *layers*) a partire da ESRI shapefiles o da dati spaziali PostGis. E' possibile creare un layer QGIS a partire da dati spaziali memorizzati in un database PostGreSQL/PostGIS e i vari layer aperti in un certo istante della sessione di lavoro sono elencati nella finestra di QGIS denominata *legenda* e vengono contestualmente sovrapposti nella finestra grande sulla destra nello stesso ordine con il quale sono elencati nella *legenda*.

Uno strumento di QGIS che ci è risultato molto utile è "*Informazioni Elementi*", che ci consente di interagire con gli elementi presenti nel layer rappresentato nella finestra grande per conoscerne le proprietà geometriche e descrittive.

RISULTATI DI LABORATORIO

B.1 CLASSIFICAZIONE COMPLETA DELLE STAZIONI TRAMITE IL NMC

Tabella 33: Risultati classificazione NMC parte 1

ID	Nome	Exposure	Fascia
114	Acciano	49,853849	3
54	Fontecchio	42,676392	3
44	Sant'Ilario	31,731346	3
38	Pettorano Sul Gizio	31,141373	3
94	Isca d'Archi	22,708767	3
66	Prezza	20,544748	3
33	Bussi	19,983268	2
81	Capistrello	18,569988	2
91	Civitaluparella	16,317314	2
63	Sella di Corno	14,241012	2
65	Aversa	13,354986	2
43	Roccaraso	13,342687	2
62	Vigliano d'Abruzzo	13,147649	2
52	Beffi	11,80442	2
84	Civita D'Antino - Morino	10,787065	2
76	Cappelle Magliano	10,690382	2

Tabella 34: Risultati classificazione NMC parte 2

ID	Nome	Exposure	Fascia
74	Paterno - San Pelino	8,838893	2
53	Tione degli Abruzzi	8,719812	2
90	Quadri	8,613139	2
60	L'Aquila	8,530002	2
41	Palena	8,469246	2
83	Civitella Roveto	7,722004	2
92	Villa Santa Maria	7,227218	2
89	Gamberale	7,184135	2
111	Guardigrele	6,229817	2
110	Filetto	5,920856	2
51	Molina Aterno	5,630178	2
6	Silvi	5,591903	2
102	Castel Frentano	5,586048	2
45	Alfredena - Scontrone	5,390527	2
67	Goriano Sicoli	5,358967	2
18	Vasto San Salvo	5,314123	2
32	Tocco - Castiglione	5,255392	2
55	Fagnano - Campana	4,917836	2
78	Tagliacozzo	4,456948	2
13	San Vito - Lanciano	4,444776	2
82	Canistro	4,388832	2
97	Atessa	4,239604	2
9	Francavilla al mare	4,215249	2

Tabella 35: Risultati classificazione NMC parte 3

ID	Nome	Exposure	Fascia
31	Torre de passeri	4,184996	2
85	Morrea	3,939813	2
15	Torino di Sangro - Paglieta	3,498431	2
109	Orsogna	3,480277	2
47	Castel Di Sangro	3,362495	2
68	Carrito - Ortona	3,267118	2
72	Aielli	3,248848	2
112	San Vincenzo	3,152791	2
96	Perano	3,069793	2
95	Archi	3,059541	2
80	Carsoli	3,056247	2
39	Cansano	3,006398	2
40	Campo di Giove	2,914821	2
104	Treglio	2,86405	2
69	Pescina	2,773715	2
101	Crocetta	2,772219	2
105	San Vito Chietino	2,319073	2
79	Lanciano	2,047489	2
12	Ortona - Sangritana	1,977955	2
34	Popoli - Vittorito	1,968973	2
14	Fossacesia	1,960503	2
88	Ateleta	1,402293	2
48	Castel Di Sangro	1,399907	2

Tabella 36: Risultati classificazione NMC parte 4

ID	Nome	Exposure	Fascia
73	Celano - Ovindoli	1,347801	2
36	Sulmona	1,32685	2
99	Casoli	1,305557	2
100	Sant'Eusanio del Sangro	1,25947	2
22	Bellante - Ripattoni	1,196643	2
27	Manoppello	1,149626	2
35	Pratola Peligna	0,877992	1
26	Brecciarola	0,851859	1
20	Teramo	0,827735	1
71	Cerchio	0,824056	1
113	Vasto	0,757341	1
16	Casalbordino - Pollutri	0,718942	1
29	Alanno	0,610406	1
23	Notaresco	0,595669	1
93	Bomba	0,586459	1
57	San Demetrio Vestini	0,543126	1
64	Bugnara	0,514268	1
30	Scafa - San Valentino - Caramanico Term	0,513323	1
37	Sulmona - Introdacqua	0,429095	1
25	Chieti	0,423106	1
11	Ortona	0,403358	1
42	Rivisondoli - Pescocostanzo	0,36118	1
86	Balsorano	0,350808	1
5	Pineto - Atri	0,350332	1

Tabella 37: Risultati classificazione NMC parte 5

ID	Nome	Exposure	Fascia
10	Foro	0,326424	1
98	Altino	0,308774	1
107	Selceroli	0,3056	1
70	Collarmele	0,228687	1
50	Raiano	0,201354	1
28	Rosciano	0,172428	1
4	Roseto Degli Abruzzi	0,166173	1
3	Giulianova	0,165897	1
56	Villa Sant'Angelo	0,081536	1
108	Arielli	0,025756	1
21	San Nicolò a Tordino	0,003282	1
24	Mosciano Sant'Angelo	0,002453	1
49	Pratola Peligna Superiore	0,001562	1
58	Fossa	0,000758	1
103	Lanciano	0,000711	1
2	Tortoreto	0,000539	1
106	Villa Caldari	0,000233	1
8	Pescara Centrale	0,000213	1

Tabella 38: Risultati classificazione NMC parte 6

ID	Nome	Exposure	Fascia
46	Montenero - Valcocchiara	0,000137	1
1	Alba Adriatica - Nereto - Controguerra	0	1
7	Montesilvano	0	1
17	Porto di Vasto	0	1
19	San Salvo	0	1
59	Paganica	0	1
61	Sassa - Torninparte	0	1
75	Avezzano	0	1
77	Scurcola Marsicana	0	1
87	San Pietro Avellana	0	1

B.2 CLASSIFICAZIONE COMPLETA DELLE LINEE FERROVIARIE TRAMITE IL NMCE

Di seguito si mostrano i risultati ottenuti tramite il NMCE per le 9 linee ferroviarie forniti nei dati di input. Per ognuna di esse verranno mostrate due tabelle che riassumono i valori di *exposure* dei segmenti che le compongono.

La prima tabella mostra il numero di segmenti che compongono la specifica linea ferroviaria e la distribuzione, in percentuale, dei segmenti all'interno delle fasce già viste nella Sez.?? Tab. 11.

La seconda tabella è una vista sui risultati puntuali dei vari segmenti, nella colonna *Numbers* sono stati inseriti i numeri relativi dei segmenti, utilizzando il carattere - per indicare intervalli di number consecutivi.

B.2.1 Bologna-Bari

Numero di segmenti	% in Low	% in Moderate	% in High
125	56,8%	43,2%	0%

Tabella 39: Riepilogo risultati segmenti per la linea Bologna-bari

	Numero di segmenti	Numbers	max exposure	avg exposure
LOW	71	2-23 25-32 35-38 44-49 54-57 60-63 65-68 72 93-94 101-105 107-113 121-125	0,9443758754	0,3237692482
MODERATE	54	1 24 33-34 39-43 50-53 58-59 64 69 70-71 73-92 95-100 102 106 114-120	8,58597068	2,808810913
HIGH	0			

B.2.2 Ortona-Crocetta

Numero di segmenti	% in Low	% in Moderate	% in High
36	36%	64%	0%

Tabella 40: Sintesi dei valori di *Riepilogo risultati segmenti per la linea Ortona-Crocetta*

	Numero di segmenti	Numbers	max exposure	avg exposure
LOW	13	3-15	0,7777983649	0,245232976
Moderate	23	1-2 16-36	10,21509432	3,759937833
HIGH	0			

B.2.3 Marina di San Vito - Castel di Sangro

Numero di segmenti	% in Low	% in Moderate	% in High
105	16%	83%	1%

Tabella 41: Riepilogo risultati segmenti per la linea Marina di San Vito - Castel di Sangro

	Numero di segmenti	Numbers	max exposure	avg exposure
LOW	17	14-21 35 39 43 93-97 103	0,7777983649	0,4015290067
Moderate	87	1-13 22-34 36-38 40-42 44-63 65-92 98-102 104-105	10,21509432	6,124790459
HIGH	1	64	20,4253592	20,4253592

B.2.4 Roma-Pescara

Numero di segmenti	% in Low	% in Moderate	% in High
170	35%	52%	13%

Tabella 42: Riepilogo risultati segmenti per la linea Roma-Pescara

	Numero di segmenti	Numbers	max exposure	avg exposure
LOW	61	1-16 18-22 25-26 30 36 53-56 60-73 108 122-124 129-133 138 139 140-145 166-169	0,9748385253	0,2258576073
MODERATE	87	17 23 24 27-29 31-35 37-44 51-52 57-59 61 63 64 74-76 79-85 91-95 99 101-107 109-121 125-128 134-137 146-153 159-165 170	18,27710597	5,991583624
HIGH	22	45-50 77-78 86-90 96-98 100 154-158	76,54443068	37,56807977

B.2.5 Avezzano-Roccasecca

Numero di segmenti	% in Low	% in Moderate	% in High
46	6%	76%	18%

Tabella 43: Riepilogo risultati segmenti per la linea Avezzano-Roccasecca

	Numero di segmenti	Numbers	max exposure	avg exposure
LOW	3	1 2 46	0,01349595782	0,004498652606
MODERATE	35	3-23 25-29 31 34 38-41 43-45	19,46157978	9,640153421
HIGH	8	24 30 32 33 35-37 42	50,76284705	31,89190056

B.2.6 Archi

stazione-Atessa

Numero di segmenti	% in Low	% in Moderate	% in High
16	0%	100%	0%

Tabella 44: Riepilogo risultati segmenti per la linea Archi stazione-Atessa

	Numero di segmenti	Numbers	max exposure	avg exposure
LOW	0		0	0
MODERATE	16	1-15	4,62602218	2,296455041
HIGH	0		0	0

B.2.7 Sulmona-Carpinone

Numero di segmenti	% in Low	% in Moderate	% in High
86	16%	52%	32%

Tabella 45: Riepilogo risultati segmenti per la linea Sulmona-Carpinone

	Numero di segmenti	Numbers	max exposure	avg exposure
LOW	14	1-4 8 46-48 72-75 77 86	0,623711422	0,2976164776
Moderate	44	5-7 20-35 42-45 49-54 66-71 76 78-85	19,71590823	7,368289452
HIGH	28	9-19 36-41 55-65	82,18520303	38,98580056

B.2.8 Teramo-Giulianova

Numero di segmenti	% in Low	% in Moderate	% in High
26	70%	30%	0%

Tabella 46: Riepilogo risultati segmenti per la linea Teramo-Giulianova

	Numero di segmenti	Numbers	max exposure	avg exposure
LOW	18	1-12 18-21 25-26	0,8041957699	0,186238294
Moderate	8	13-17 22-24	2,986196534	1,845508738
HIGH	0			

Numero di segmenti	% in Low	% in Moderate	% in High
81	36%	41%	23%

Tabella 47: Riepilogo risultati segmenti per la linea Rieti-L'aquila-Sulmona

	Numero di segmenti	Numbers	max exposure	avg exposure
LOW	29	44-48 68 59 61 64 65 72 78-81 1-3 5 6 12 13 16-18 22-24 20	0,9736608166	0,3750273597
MODERATE	33	43 41 49-52 67 69 21 27-31 34 38 54 57 58 60 4 7-11 14 15 26 74-77	18,5260294680	7,7033145327
HIGH	19	53 55 56 62 63 66 70 71 19 25 33 39 40 42 32 35-37 73	130,1376566585	52,2874599615

B.3 CLASSIFICAZIONE UFFICIALE DELLE STAZIONI UTILIZZATA PER LA VALIDAZIONE

Tabella 48: Classificazione ufficiale parte 1

ID	Nome	Fascia
44	Sant'Ilario	3
65	Aversa	3
54	Fontecchio	3
38	Pettorano	3
	Sul Gizio	
114	Acciano	3
62	Vigliano d'Abruzzo	3
4	Roseto	2
	Degli Abruzzi	
50	Raiano	2
98	Altino	2
10	Foro	2
11	Ortona	2
93	Bomba	2
27	Manoppello	2
100	Sant'Eusanio del Sangro	2
99	Casoli	2
73	Celano - Ovindoli	2
88	Ateleta	2
14	Fossacesia	2
12	Ortona - Sangritana	2
79	Lanciano	2
105	San Vito Chietino	2
101	Crocetta	2

Tabella 49: Classificazione ufficiale parte 2

ID	Nome	Fascia
69	Pescina	2
104	Treglio	2
40	Campo di Giove	2
39	Cansano	2
80	Carsoli	2
112	San Vincenzo	2
72	Aielli	2
68	Carrito - Ortona	2
109	Orsogna	2
15	Torino di Sangro - Paglieta	2
85	Morrea	2
31	Torre de passeri	2
9	Francavilla al mare	2
97	Atessa	2
82	Canistro	2
13	San Vito - Lanciano	2
78	Tagliacozzo	2
55	Fagnano - Campana	2
32	Tocco - Castiglione	2
18	Vasto San Salvo	2
67	Goriano Sicoli	2
45	Alfredena - Scontrone	2
102	Castel Frentano	2
6	Silvi	2

Tabella 50: Classificazione ufficiale parte 3

ID	Nome	Fascia
51	Molina Aterno	2
110	Filetto	2
111	Guardigrele	2
89	Gamberale	2
92	Villa Santa Maria	2
83	Civitella Roveto	2
41	Palena	2
60	L'Aquila	2
90	Quadri	2
53	Tione degli Abruzzi	2
74	Paterno - San Pelino	2
76	Cappelle Magliano	2
84	Civita D'Antino - Morino	2
52	Beffi	2
43	Roccaraso	2
63	Sella di Corno	2
91	Civilaluparella	2
81	Capistrello	2
33	Bussi	2
66	Prezza	2
94	Isca d'Archi	2
1	Alba Adriatica - Nereto - Controguerra	1
7	Montesilvano	1
17	Porto di Vasto	1
19	San Salvo	1
59	Paganica	1
61	Sassa - Torninparte	1
75	Avezzano	1
77	Scurcola Marsicana	1
87	San Pietro Avellana	1

Tabella 51: Classificazione ufficiale parte 4

ID	Nome	Fascia
46	Montenero - Valcocchiara	1
8	Pescara Centrale	1
106	Villa Caldari	1
2	Tortoreto	1
103	Lanciano	1
58	Fossa	1
49	Pratola Peligna Superiore	1
24	Mosciano Sant'Angelo	1
21	San Nicolò a Tordino	1
108	Arielli	1
56	Villa Sant'Angelo	1
3	Giulianova	1
28	Rosciano	1
70	Collarmele	1
107	Selceroli	1
5	Pineto - Atri	1
86	Balsorano	1
42	Rivisondoli - Pescocostanzo	1
25	Chieti	1
37	Sulmona - Introdacqua	1
30	Scafa - San Valentino - Caramanico Term.	1
64	Bugnara	1
57	San Demetrio Vestini	1
23	Notaresco	1
29	Alanno	1
16	Casalbordino - Pollutri	1

Tabella 52: Classificazione ufficiale parte 5

ID	Nome	Fascia
113	Vasto	1
71	Cerchio	1
20	Teramo	1
26	Brecciarola	1
35	Pratola Peligna	1
22	Bellante - Ripattoni	1
36	Sulmona	1
48	Castel Di Sangro	1
34	Popoli - Vittorito	1
95	Archi	1
96	Perano	1
47	Castel Di Sangro	1

UDF

C.1 UDF DEL NMC

In questa sezione verranno mostrate tutte le User Defined Function (da qui in avanti indicate con l'acronimo UDF) che implementano NMC e che quindi calcolano del valore di exposure delle stazione ferroviarie.

Il linguaggio utilizzato nelle UDF è PL/pgSQL offerto da PostgreSQL, in particolare sono state fruttatele una serie di funzioni fornite dall'estensione spaziale PostGIS.

La struttura utilizzata per l'esecuzione delle UDF consiste in una funzione principale denominata *esposizione* che richiama le altre UDF. Queste vengono eseguite consecutivamente in quanto l'elaborazione dell'una prevede l'utilizzo dei risultati dell'elaborazione della precedente.

All'interno dell'UDF *esposizione* per ogni stazione analizzata vengono richiamate nell'ordine le UDF *quotaMediaZoneVicine*, *stimaQuotaStazione* e *dislivelloPiuVicino*. Le 4 UDF citate vanno a implementare i 4 algoritmi descritti nella sezione 2.2.3. In Particolare *quotaMediaZoneVicine* implementa *averageElevationNearZones*, *stimaQuotaStazione* implementa *elevationBuild*, *dislivelloPiuVicino* implementa *slopeFactor* ed infine *esposizione* implementa *computeExposure*.

Poiché lo scopo di questa sezione è mostrare come sono state implementate le cose e non quello di spiegarne la logica di funzionamento, all'interno delle UDF per semplicità realizzativa è stato utilizzato un naming diverso rispetto a quello utilizzato precedentemente. Per questo motivo prima di ogni UDF verrà mostrata una tabella riassuntiva che mostra come questi nomi sono stati mappati.

L'invocazione alla main function *esposizione* eseguita durante la sperimentazione è la seguente:

1 `SELECT esposizione('stazioni', FALSE, 5, 500);`

Dai parametri della funzione si osserva che il fattore *alpha_slope* è settato a 5, mentre il raggio del buffer è posto pari a 500 m.

C.1.1 *esposizione*

Rappresenta la main function che richiama le altre UDF. **Input:**

- *tabInput text* corrisponde al nome della tabella contenente i punti sul quale si vuole calcolare l'exposure.
- *stessaTab BOOL* corrisponde a un booleano che se posto a *TRUE* permette di salvare i risultati nella stessa tabella di input altrimenti li conserva nella tabella

di default *tabella_risultati*. I parametri dell'UDF *tabInput* e *stessaTab* sono stati inseriti per dare un maggior grado di libertà al codice in modo da poter essere riutilizzato come verrà fatto in seguito dal NMCE.

- *raggioBuffer FLOAT* rappresenta il raggio del buffer circolare espresso in *m*.
- *alphaSlope FLOAT* rappresenta il parametro che da un peso al fattore pendenza.
- *raggioBuffer FLOAT* rappresenta il raggio del buffer circolare centrato sulla stazione espresso in *m*.

Output: VOID

Tabella 53: Tabella di corrispondenza dei nomi per l'UDF esposizione

Schema logico	UDF
elevation	quota
slope	pendenza
experiment_station	tabella_risultati

```

1 CREATE OR REPLACE FUNCTION esposizione(tabInput text, stessaTab BOOLEAN, alphaSlope FLOAT, raggioBuffer FLOAT)
2   RETURNS VOID
3   LANGUAGE plpgsql
4   AS $$
5   DECLARE
6     -- variabili di ausilio per il calcolo -
7     stazione RECORD;
8     raggio FLOAT;
9     buffer geometry;
10    idLineaPiuVicina INT;
11    fattorePendenzaPiuVicina FLOAT;
12    areaVicina RECORD;
13    percentualeCopertura FLOAT;
14    exp FLOAT;
15    pesoPendenza FLOAT;
16    quotaArray INT [2];
17    quotaStimataStazione INT;
18
19    BEGIN
20      pesoPendenza:=alphaSlope; -- pesopendenza rappresenta alphaSlope
21      raggio:=raggioBuffer; -- parametro che indica il raggio del buffer circolare centrato sulla stazione -
22      -- pulizia della tabella per evitare risultati doppi -
23      DELETE FROM tabella_risultati;
24      -- creazione delle tabelle di appoggio tempo, tempol e zoneVicine -
25      CREATE TABLE zoneVicine( id SERIAL PRIMARY KEY, geom geometry(MULTIPOLYGON), quotamedia FLOAT, szk FLOAT);
26      CREATE TABLE tempo(id SERIAL PRIMARY KEY, geom geometry(MULTILINESTRING), elevation INT);
27      CREATE TABLE tempol(id INT PRIMARY KEY, geom geometry(MULTILINESTRING), elevation INT, distanza FLOAT);
28
29      -- ciclo su tutte le stazioni -
30      FOR stazione IN EXECUTE 'SELECT * FROM '|| tabInput ||' WHERE gid=3' LOOP
31        exp:=0;
32        buffer:=st_buffer(stazione.geom, raggio);
33
34        -- chiamate alle udf che vanno a calcolare i coefficienti necessari per il calcolo dell'exposure -
35        PERFORM quotaMediaZoneVicine(buffer);
36        quotaArray:=(SELECT stimaQuotaStazione(stazione.gid, stazione.geom));
37        idLineaPiuVicina:=quotaArray[1];
38        quotaStimataStazione:=quotaArray[0];

```

```

37     fattorePendenzaPiuVicina:=(SELECT dislivelloPiuVicino(buffer, raggio, stazione.geom, quotaStimataStazione
            , idLineaPiuVicina));
38
39 -- ciclo sulle zone che che intersecano il buffer -
40 FOR areaVicina IN (SELECT * FROM zoneVicine) LOOP
41     percentualeCopertura:=0;
42     percentualeCopertura:=(SELECT st_area(st_intersection(areaVicina.geom,buffer))/st_area(areaVicina.geom)
            );
43     IF st_intersects(stazione.geom,areaVicina.geom) OR st_distance(stazione.geom,areaVicina.geom)<1
        THEN
            exp:=exp+percentualeCopertura*areaVicina.szk*((@quotaStimataStazione-areaVicina.quotamedia));
        ELSE
            exp=exp+(percentualeCopertura*areaVicina.szk*((@quotaStimataStazione-areaVicina.quotamedia))/(
                st_distance(stazione.geom,areaVicina.geom))^1);
        END IF;
    END LOOP;
45     exp:=exp+(fattorePendenzaPiuVicina*pesoPendenza);
46     INSERT INTO tabella_risultati(id, quota, pendenza, exposure) VALUES (stazione.gid, quotaStimataStazione,
            (fattorePendenzaPiuVicina*pesoPendenza), exp);
47     IF stessaTab THEN
        EXECUTE ( 'UPDATE'|| tabInput ||' SET exposure='|| exp ||' WHERE'|| tabInput ||'.id='|| stazione.gid
            ||');
    END IF;
49     END LOOP;
50
51 -- cancellazione tabelle di appoggio -
52 DROP TABLE zoneVicine;
53 DROP TABLE tempo;
54 DROP TABLE tempol;
55 END;
56 $$;

```

C.1.2 *quotaMediaZoneVicine*

La funzione calcola la quota delle zone che intersecano il buffer attraverso la media delle *elevation* delle linee di livello che intersecano ognuna di queste zone. Inoltre la funzione conserva nella tabella *zoneVicine* tutte le zone che intersecano il buffer con la quota media calcolata in modo da consentire alle successive UDF un immediato utilizzo di questi risultati.

Input:

- *buffer GEOMETRY* rappresenta la geometria del buffer circolare centrato sulla stazione in esame.

Output: VOID

Tabella 54: Tabella di corrispondenza dei nomi per l'UDF *quotaMediaZoneVicine*

Schema logico	UDF
Zones	zone
Isoipse	lineelvl

```

1 CREATE or REPLACE FUNCTION quotaMediaZoneVicine(buffer GEOMETRY) RETURNS VOID
2 LANGUAGE plpgsql

```

```

3  as $$ 
4  DECLARE
5    -- variabili di ausilio per il calcolo -
6    zona RECORD;
7    linea RECORD;
8    quotamedia FLOAT;
9    contatore INT;
10   BEGIN
11     -- inizializzazione delle variabili per il calcolo della quota media delle zone -
12     quotamedia:=0;
13     contatore:=0;
14     -- pulizia della tabella di appoggio -
15     DELETE FROM zoneVicine;
16     -- ciclo su tutte le partizioni della geoarea che intersecano il buffer -
17     FOR zona IN (SELECT * FROM zone) LOOP
18       IF(st_intersects(zona.geom, buffer)) THEN
19         -- ciclo su tutte le linee di livello che intersecano la zona in esame -
20         FOR linea IN (SELECT * FROM lineelvl) LOOP
21           IF st_intersects(zona.geom,linea.geom) THEN
22             -- somma della quota delle linee di livello -
23             quotamedia:=quotamedia+linea.elevation;
24             contatore:=contatore+1;
25           END IF;
26         END LOOP;
27         IF contatore>0 THEN
28           quotamedia=quotamedia/contatore;
29         END IF;
30         INSERT INTO zoneVicine(geom, quotamedia, szk)VALUES (zona.geom, quotamedia, zona.szk);
31         quotamedia:=0;
32         contatore:=0;
33       END IF;
34     END LOOP;
35   END;
36 $$;

```

C.1.3 *stimaQuotaStazione*

La funzione calcola una stima della quota della stazione attraverso una media ponderata tra l'*elevation* delle linee di livello più vicine ad essa. Inoltre la funzione conserva nella tabella *tempo* tutte le intersezione tra le linee di livello e le zone che intersecano il buffer centrato sulla stazione, in modo da consentire alle successive UDF un immediato utilizzo di questi risultati.

Input:

- *gidStazione* INTEGER rappresenta l'identificativo della stazione in esame.
- *geomStazione* GEOMETRY rappresenta la geometria della stazione in esame.

Output: *INT[]* la funzione ritorna un array di interi di due elementi dove il primo elemento corrisponde alla quota stimata della stazione, mentre il secondo contiene l'identificativo della linea più vicina all'interno della tabella *tempo*.

```

1 CREATE or REPLACE FUNCTION stimaQuotaStazione(gidStazione INTEGER, geomStazione GEOMETRY) RETURNS INT[]
2 LANGUAGE plpgsql
3 as $$ 
4 DECLARE
5   -- variabili di ausilio per il calcolo -
6   areaVicina RECORD;

```

Tabella 55: Tabella di corrispondenza dei nomi per l'UDF stimaQuotaStazione

Schema logico	UDF
Isoipse	lineelvl

```

7    linea RECORD;
8    lineaInZonaVicina RECORD;
9    dmin FLOAT;
10   dquasimin FLOAT;
11   idmin INT;
12   idquasimin INT;
13   dist FLOAT;
14   altezzaStazione INT;
15   risultatiFun INT [2];
16
16 BEGIN
17   -- pulizia della tabella -
18   DELETE FROM tempo;
19   -- ciclo sulle zone che intersecano il buffer centrato sulla stazione -
20   FOR areaVicina IN (SELECT * FROM zoneVicine) LOOP
21     -- ciclo sulle linee di livello che intersecano la zona in esame -
22     FOR linea IN (SELECT * FROM lineelvl) LOOP
23       IF st_intersects(areaVicina.geom,linea.geom) THEN
24         -- inserimento nella tabella tempo l'intersezione tra la zona e la linea, con la quota associata alla
25         -- linea -
26         INSERT INTO tempo( geom, elevation) VALUES((SELECT st_multi(st_intersection(areaVicina.geom,linea.
27           geom))), linea.elevation);
28       END IF;
29     END LOOP;
30   END LOOP;
31   -- dmin conterrà il valore della distanza tra la linea di livello più vicina e la stazione -
32   dmin:=10000;
33   -- dquasimin conterrà il valore della distanza tra la seconda linea di livello più vicina e la stazione -
34   dquasimin:=10000;
35   -- idmin conterrà l'identificativo della linea a distanza minore dalla stazione -
36   idmin:=0;
37   -- idquasimin conterrà l'identificativo della seconda linea a distanza minore dalla stazione -
38   idquasimin:=0;
39   -- ciclo sui segmenti di linea che intersecano le zone che intersecano con il buffer centrato sulla
40   -- stazione -
41   FOR lineaInZonaVicina IN (SELECT * FROM tempo) LOOP
42     dist:=st_distance(geomStazione, lineaInZonaVicina.geom);
43     IF dist<=dmin THEN
44       dquasimin:=dmin;
45       idquasimin:=idmin;
46       dmin:=dist;
47       idmin:=lineaInZonaVicina.id;
48     END IF;
49     IF dist>dmin AND dist<=dquasimin THEN
50       dquasimin:=dist;
51       idquasimin:=lineaInZonaVicina.id;
52     END IF;
53   END LOOP;
54   IF(st_intersects(st_makeline(st_closestpoint((SELECT tempo.geom FROM tempo WHERE id=idquasimin),((
55     geomStazione)),geomStazione),(SELECT tempo.geom FROM tempo WHERE tempo.id=idmin))) THEN
56     dquasimin:=dmin;
57     idquasimin:=idmin;
58   END IF;
59   -- calcolo della quota della stazione come media ponderata delle quote delle linee di livello più vicina ad
60   -- essa -
61   altezzaStazione:=((SELECT tempo.elevation FROM tempo WHERE id=idmin)*dquasimin+(SELECT tempo.elevation FROM
62     tempo WHERE id=idquasimin)*dmin)/(dmin+dquasimin);

```

```

58     UPDATE stazioni SET quota=altezzaStazione WHERE stazioni.gid=gidStazione;
59     risultatiFun[0]=altezzaStazione;
60     risultatiFun[1]=idmin;
61     RETURN risultatiFun;
62 END;
63 $$;
```

C.1.4 *dislivelloPiuVicino*

La funzione calcola la somma delle pendenze tra le linee di livello lungo il vettore direzionale.

Input:

- *buffer GEOMETRY* rappresenta la geometria del buffer circolare centrato sulla stazione in esame.
- *raggioBuffer FLOAT* rappresenta il raggio del buffer circolare espresso in *m*.
- *geomStazione GEOMETRY* rappresenta la geometria della stazione in esame.
- *quotaStazione INT* rappresenta la quota stimata della stazione in esame.
- *idLineaPiuVicina INT* rappresenta l'identificativo della linea di livello più vicina all'interno della tabella *tempo*.

Output: *FLOAT* la funzione restituisce la somma delle pendenze calcolata.

Tabella 56: Tabella di corrispondenza dei nomi per l'UDF *dislivelloPiuVicino*

Schema logico	UDF
Isoipse	lineelvl

```

1 CREATE OR REPLACE FUNCTION dislivelloPiuVicino(buffer GEOMETRY, raggioBuffer FLOAT, geomStazione GEOMETRY,
2                                                 quotaStazione INT, idLineaPiuVicina INT ) RETURNS FLOAT
3 LANGUAGE plpgsql
4 as $$
5 DECLARE
6     -- variabili di ausilio per il calcolo -
7     linea RECORD;
8     puntoVicino GEOMETRY;
9     azimuth FLOAT;
10    lunghezzaLinea FLOAT;
11    vettoreDirezionale GEOMETRY;
12    lineaInBuffer RECORD;
13    primo BOOL;
14    monotonia BOOL;
15    salita BOOL;
16    stazioneStessaQuotaLinea BOOL;
17    sommaPendenza FLOAT;
18    geomUltimaLinea GEOMETRY;
19    elevationUltimaLinea INT;
20    BEGIN
21        -- pulizia della tabella di appoggio -
22        DELETE FROM temp01;
```

```

22 -- ciclo sulle linee di livello che intersano il buffer centrato sulla stazione -
23 FOR linea IN (SELECT * FROM lineelvl) LOOP
24   IF st_intersects(buffer,linea.geom) THEN
25     -- inserimento in una tabella dell'intersezione tra la linea e il buffer, con la quota associata e l'id -
26     INSERT INTO tempol(id, geom, elevation) SELECT linea.gid, (SELECT st_multi(st_intersection(buffer,linea.
27       geom))), linea.elevation WHERE NOT exists(SELECT tempol.id FROM tempol WHERE tempol.id=linea.gid);
28   END IF;
29 END LOOP;
30   -- costruzione del vettore direzionale -
31 puntoVicino:=st_closestpoint((SELECT tempo.geom FROM tempo WHERE id=idLineaPiuVicina),(geomStazione));
32 azimuth:=st_azimuth(geomStazione,puntoVicino);
33 lunghezzaLinea:=raggioBuffer+1;
34 vettoreDirezionale:=st_makeline(geomStazione, st_translate(geomStazione, sin(azimuth)*lunghezzaLinea, cos(
35   azimuth)*lunghezzaLinea));
36   -- ciclo sui segmenti di linee di livello che intersecano il buffer -
37 FOR lineaInBuffer IN(SELECT * FROM tempol)LOOP
38   IF st_intersects(lineaInBuffer.geom, vettoreDirezionale) THEN
39     -- aggiornamento delle ennumple in tempol con la distanza con la stazione calcolata lungo il vettore
40     -- direzionale -
41     UPDATE tempol SET distanza=(SELECT st_distance(geomStazione, st_intersection(lineaInBuffer.geom,
42       vettoreDirezionale))) WHERE tempol.id=lineaInBuffer.id;
43   ELSE
44     DELETE FROM tempol WHERE tempol.id=lineaInBuffer.id;
45   END IF;
46 END LOOP;
47   -- inizializzazione di variabili booleane utilizzate per interrompere l'algoritmo -
48 primo:=TRUE;
49 monotonia:=TRUE;
50 salita:=FALSE;
51 stazioneStessaQuotaLinea:=FALSE;
52 sommaPendenza:=0;
53 FOR lineaInBuffer IN(SELECT * FROM tempol ORDER BY distanza)LOOP
54   IF (monotonia=TRUE) THEN
55     -- calcola della pendenza tra la stazione e la prima linea di livello incontrata lungo il vettore
56     -- direzionale -
57     IF (primo=TRUE) THEN
58       sommaPendenza:=sommaPendenza+((@(lineaInBuffer.elevation-quotaStazione))/(st_distance(geomStazione,
59         lineaInBuffer.geom)));
60       IF lineaInBuffer.elevation>quotaStazione THEN
61         salita:=TRUE;
62       END IF;
63       IF lineaInBuffer.elevation=quotaStazione THEN
64         stazioneStessaQuotaLinea:=TRUE;
65       END IF;
66       primo:=FALSE ;
67
68     -- calcola della pendenza tra una linea di livello e la successiva -
69   ELSE
70     IF (stazioneStessaQuotaLinea=FALSE) THEN
71       IF ((salita=TRUE) AND lineaInBuffer.elevation>elevationUltimaLinea) OR ((salita=FALSE) AND
72         lineaInBuffer.elevation<elevationUltimaLinea) THEN
73         sommaPendenza:=sommaPendenza+(@(lineaInBuffer.elevation-elevationUltimaLinea))/st_distance(
74           st_intersection(geomUltimaLinea, vettoreDirezionale), st_intersection(lineaInBuffer.geom,
75             vettoreDirezionale));
76       ELSE
77         monotonia:=FALSE ;
78       END IF;
79     ELSE
80       IF lineaInBuffer.elevation>elevationUltimaLinea THEN
81         salita:=TRUE ;
82       END IF;
83       sommaPendenza:=sommaPendenza+(@(lineaInBuffer.elevation-elevationUltimaLinea))/st_distance(
84         st_intersection(geomUltimaLinea, vettoreDirezionale), st_intersection(lineaInBuffer.geom,

```

```

    vettoreDirezionale));
77     stazioneStessaQuotaLinea:=FALSE;
78   END IF;
79 END IF;
80
81 geomUltimaLinea:=lineaInBuffer.geom;
82 elevationUltimaLinea:=lineaInBuffer.elevation;
83 END IF;
84 END LOOP;
85 RETURN sommaPendenza;
86 END;
87 $$;

```

C.2 UDF DEL NMCE

In questa sezione verranno mostrate tutte le UDF che implementano NMCE e che contribuiscono al calcolo valore di exposure delle linee ferroviarie.

Così come per le stazioni ferroviarie la struttura utilizzata per l'esecuzione delle UDF relative alle linee ferroviarie consiste in una funzione principale denominata *esposizione_linee* che richiama le altre UDF. Queste vengono eseguite consecutivamente in quanto l'elaborazione dell'una prevede l'utilizzo dei risultati dell'elaborazione della precedente.

All'interno dell'UDF *esposizione_linee* vengono richiamate nell'ordine le UDF *semplificaLinee*, *campiona*, *dividiInSegmenti*, *esposizione* e *expSegmenti*. Poiché lo scopo di questa sezione è mostrare come sono state implementate le cose e non quello di spiegarne la logica di funzionamento, all'interno delle UDF per semplicità realizzativa è stato utilizzato un naming diverso rispetto a quello utilizzato precedentemente. Per questo motivo prima di ogni UDF verrà mostrata una tabella riassuntiva che mostra come questi nomi sono stati mappati. L'invocazione alla main function *esposizione_linee* eseguita durante la sperimentazione è la seguente:

```
1 SELECT esposizione_linee(300, 1000);
```

C.2.1 *esposizione_linee*

Rappresenta la main function che richiama le altre UDF.

Input:

- *passoCampionamento* INTEGER definisce il passo di campionamento espresso in *m* delle linee ferroviarie.
- *lunghezzaSegmento* INTEGER definisce la lunghezza che si vuole dare ai segmenti che formano le linee ferroviarie.

Output: VOID

```
1 CREATE or REPLACE FUNCTION esposizione_linee(passoCampionamento INT, lunghezzaSegmento INT) RETURNS VOID
2 LANGUAGE plpgsql
3 as $$
4 DECLARE
```

```

5 BEGIN
6   -- creazione tabella di appoggio -
7   CREATE TABLE only_line(id SERIAL PRIMARY KEY, old_id INT, geom geometry(LINESTRING), p INTEGER[]);
8   PERFORM semplificaLinee();
9   PERFORM campiona(passoCampionamento);
10  PERFORM dividiInSegmenti(lunghezzaSegmento);
11  -- richiamo della main function del NMC
12  PERFORM esposizione('punti_linea', TRUE, 5, 500);
13  PERFORM expSegmenti();
14  -- cancellazione tabella di appoggio -
15  DROP TABLE only_line;
16 END;
17 $$;

```

C.2.2 *semplificaLinee*

La funzione a partire dalle *MultiLineString* che rappresentano le linee ferroviarie ricava delle geometrie più semplici ossia *Linestring* e le salva nella tabella *only_line*.

Tabella 57: Tabella di corrispondenza dei nomi per l'UDF *semplificaLinee*

Schema logico	UDF
Raylway	linee

```

1 CREATE OR REPLACE FUNCTION semplificaLinee() RETURNS VOID
2 LANGUAGE plpgsql
3 AS $$
4 DECLARE
5   -- variabili di ausilio -
6   linea RECORD;
7   lineaSemplice GEOMETRY;
8 BEGIN
9   ALTER SEQUENCE only_line_id_seq RESTART WITH 1;
10  -- ciclo sulle linee ferroviarie -
11  FOR linea IN (SELECT * FROM linee) LOOP
12    lineaSemplice:=(SELECT st_linemerge(linea.geom));
13    -- si tenta di riunire multilinestring in linestring, se non è possibile si spezza la multilinestring in
14    -- più linestring -
15    IF st_geometertype(lineaSemplice)='ST_MultiLineString' THEN
16      INSERT INTO only_line(geom, old_id, p) VALUES ((st_dump(st_linemerge(linea.geom))).geom, linea.gid, (
17        st_dump(st_linemerge(linea.geom))).path) ;
18    ELSE
19      INSERT INTO only_line(geom, old_id) VALUES (lineaSemplice, linea.gid);
20    END IF;
21  END LOOP;
22 END;
23 $$;

```

C.2.3 *campiona*

Funzione che campiona le le linee semplici presenti nella tabella *only_line* contenente le *Linestring* ricavate dalle linee ferroviarie.

Input:

- *passo* INTEGER definisce il passo di campionamento.

Output: VOID

Tabella 58: Tabella di corrispondenza dei nomi per l'UDF campiona

Schema logico	UDF
Sampling_point	punti_linea

```

1 CREATE or REPLACE FUNCTION campiona(passo INT) RETURNS VOID
2 LANGUAGE plpgsql
3 as $$
4 DECLARE
5     -- variabili di ausilio -
6     linea RECORD;
7     percentuale FLOAT;
8     lunghezzalinea FLOAT;
9     lineaPrecedente INT;
10    i INT;
11 BEGIN
12     ALTER SEQUENCE punti_linea_gid_seq RESTART WITH 1;
13     lineaPrecedente:=0;
14     -- ciclo sulle linee semplici ricavate dalle linee ferroviarie -
15     FOR linea in (SELECT * FROM only_line ORDER BY id) LOOP
16         lunghezzaLinea:=st_length(linea.geom);
17         percentuale:=(passo*1)/lunghezzaLinea;
18         i:=1;
19         -- ciclo fino a che con il campionamento non si supera la lunghezza della linestring -
20         WHILE (passo*i)<lunghezzaLinea LOOP
21             INSERT INTO punti_linea(old_id, geom) VALUES (linea.old_id, (SELECT st_lineinterpolatepoint(linea.geom,
22                                         percentuale*i)));
23             i:=i+1;
24         END LOOP;
25     END LOOP;
26 END;
27 $$;
```

C.2.4 dividInSegmenti

Funzione che divide le linee semplici, presenti nella tabella *only_line* contenente le *Linestring* ricavate dalle linee ferroviarie, in segmenti di lunghezza fissa.

Input:

- *passo* INTEGER definisce il la lunghezza dei segmenti che si vuole ottenere.

Output: VOID

Tabella 59: Tabella di corrispondenza dei nomi per l'UDF dividInSegmenti

Schema logico	UDF
Experiment_railway	segmenti

```

1 CREATE or REPLACE FUNCTION dividiInSegmenti(passo INT) RETURNS VOID
2 LANGUAGE plpgsql
3 as $$ 
4 DECLARE
5     -- variabili di ausilio -
6     linea RECORD;
7     lineaPrecedente INT;
8     lunghezzaLinea FLOAT;
9     percentuale FLOAT;
10    i INT;
11    startFraction FLOAT;
12    endFraction FLOAT;
13 BEGIN
14     passo:=1000;
15     lineaPrecedente:=0;
16     ALTER SEQUENCE segmenti_id_seq RESTART WITH 1;
17     -- ciclo sulle linee semplici ricate dalle linee ferroviarie -
18     FOR linea IN (SELECT * FROM only_line ORDER BY old_id, p DESC ) LOOP
19         -- numero viene riazzeroato quando si comincia la divisione di una nuova linea -
20         IF lineaPrecedente!=linea.old_id THEN
21             ALTER SEQUENCE segmenti_numero_seq RESTART WITH 1;
22         END IF;
23         lineaPrecedente:=linea.old_id;
24         lunghezzaLinea:=st_length(linea.geom);
25         percentuale:=(passo*1)/lunghezzaLinea;
26         i:=1;
27         -- ciclo fino al superamento della lunghezza della linea -
28         WHILE (passo*i)<lunghezzaLinea LOOP
29             startFraction:=percentuale*(i-1);
30             endFraction:=(percentuale*(i-1))+percentuale;
31             INSERT INTO segmenti(old_id, geom) VALUES (linea.old_id, ST_LineSubstring(linea.geom, startFraction,
32                                         endFraction));
32             i:=i+1;
33         END LOOP;
34         -- controlli per l'ultimo segmento della linea -
35         IF (passo*i)>=lunghezzaLinea AND passo*(i-1)!=lunghezzaLinea AND percentuale<=1 THEN
36             INSERT INTO segmenti(old_id, geom) VALUES (linea.old_id, ST_LineSubstring(linea.geom, endFraction, 1));
37         END IF;
38         IF percentuale>1 THEN
39             INSERT INTO segmenti(old_id, geom) VALUES (linea.old_id, ST_LineSubstring(linea.geom, 0, 1));
40         END IF;
41     END LOOP;
42 END;
43 $$;

```

C.2.5 expSegmenti

Funzione che per ogni segmento delle linee ferroviarie calcola un valore di *exposure* pari alla media del valore di *exposure* dei punti campionati che interseca.

Tabella 6o: Tabella di corrispondenza dei nomi per l'UDF dividiInSegmenti

Schema logico	UDF
Experiment_railway	segmenti
Sampling_point	punti_linea

```

1 CREATE or REPLACE FUNCTION expSegmenti() RETURNS VOID
2 LANGUAGE plpgsql
3 as $$ 
4 DECLARE
5   -- variabili di ausilio -
6   segmento RECORD;
7   punto RECORD;
8   exp FLOAT;
9   contatore INT;
10 BEGIN
11   -- ciclo su tutti i segmenti -
12   FOR segmento IN (SELECT * FROM segmenti) LOOP
13     contatore:=0;
14     exp:=0;
15     -- ciclo su tutti i punti della stessa linea del segmento -
16     FOR punto in (SELECT * FROM punti_linea WHERE punti_linea.old_id=segmento.old_id) LOOP
17       IF (st_distance(segmento.geom, punto.geom)<1) THEN
18         exp:=exp+punto.exposure;
19         contatore:=contatore+1;
20       END IF;
21     END LOOP;
22     IF contatore!=0 THEN
23       exp:=exp/contatore;
24     END IF;
25     UPDATE segmenti SET exp_segmento=exp WHERE segmenti.id=segmento.id;
26   END LOOP;
27 END;
28 $$;

```