

UNIVERSITÀ DEGLI STUDI GENOVA



# Riconoscimento automatico di difetti per la diagnostica predittiva su sistemi di isolamento

Facoltà di ingegneria  
Dipartimento di ingegneria navale, elettrica, elettronica e tecnologie  
dell'informazione

Christian Gianoglio

18 aprile 2019

## *Abstract*

Una macchina elettrica, durante il ciclo di vita, presenta fenomeni di invecchiamento nel sistema di isolamento. Questi si caratterizzano come impulsi di corrente, conosciuti come "scariche parziali", e sono sia il sintomo sia la causa del deterioramento dell'isolamento stesso. Per questo motivo vengono programmate azioni di manutenzione periodiche per limitare i danni che possono provocare. Risulta però utile monitorare continuamente lo stato di una macchina, sia per non fermarla per una manutenzione non necessaria, sia per valutare online la condizione dell'isolamento, in modo tale da intervenire immediatamente nel caso in cui un difetto grave si manifesti improvvisamente. Le scariche parziali possono essere più o meno pericolose in funzione di alcuni fattori quali l'intensità, la frequenza con cui si manifestano e la posizione all'interno di un motore elettrico. Risulta dunque necessario distinguere la/e sorgente/i che generano tali fenomeni.

Perciò, in questa tesi, vengono presentati diversi approcci e tecniche per il riconoscimento automatico di difetti, sia con algoritmi di apprendimento supervisionato che non. Nel primo caso si identificano soluzioni di apprendimento rapido che possono essere realizzate su dispositivi hardware, con un ottimo compromesso tra capacità di generalizzazione e occupazione di area. Nel secondo, vengono confrontati diversi algoritmi presenti in letteratura e proposta una scelta alternativa dei parametri in ingresso ad essi, che porta a risultati soddisfacenti.

# Indice

<b>Abstract</b>	<b>i</b>
<b>Elenco delle Figure</b>	<b>v</b>
<b>Elenco delle Tabelle</b>	<b>ix</b>
<b>1 Introduzione</b>	<b>1</b>
1.1 Contributo . . . . .	3
<b>2 Il fenomeno delle scariche parziali</b>	<b>5</b>
2.1 Cause dell'attività di scarica . . . . .	5
2.2 Sistemi di misura delle scariche parziali . . . . .	6
2.3 Tipologia di scariche parziali . . . . .	7
2.3.1 Scariche interne . . . . .	8
2.3.2 Scariche esterne . . . . .	11
2.3.3 Impulso di corrente associato alla scarica . . . . .	12
2.4 Misura delle scariche parziali . . . . .	13
2.4.1 Misure ON-LINE . . . . .	14
2.4.2 Misure OFF-LINE . . . . .	14
2.5 Cenni alla normativa IEC 60270 . . . . .	15
2.5.1 Grandezze principali definite dalla norma . . . . .	15
2.5.2 Circuiti normati per la misura delle scariche parziali: metodo diretto	17
2.5.3 PD Pattern . . . . .	19
2.5.4 Limiti nelle misure condotte secondo normativa . . . . .	20
<b>3 Stato dell'arte sulla estrazione delle features e sulla classificazione delle scariche parziali</b>	<b>21</b>
3.1 Estrazione delle features . . . . .	21
3.1.1 Parametri statistici . . . . .	21
3.1.2 Trasformata di Fourier . . . . .	23
3.1.3 Principal Component Analysis . . . . .	24
3.1.4 t-SNE . . . . .	25
3.1.5 Sommario sulle tecniche di estrazione delle features . . . . .	26
3.2 Algoritmi per la classificazione non supervisionata . . . . .	26
3.2.1 K-means . . . . .	26
3.2.2 DBSCAN . . . . .	27
3.2.3 OPTICS . . . . .	28
3.3 Algoritmi per la classificazione supervisionata . . . . .	30

3.3.1	Support Vector Machine (SVM) . . . . .	30
3.3.2	Reti neurali . . . . .	32
3.4	Sommario delle tecniche di classificazione . . . . .	34
<b>4</b>	<b>Misura delle scariche e separazione dal rumore</b>	<b>35</b>
4.1	Sistemi per la misura condotta delle scariche . . . . .	35
4.2	Sistemi per la misura irradiata delle scariche . . . . .	37
4.2.1	Antenna SolBox <sup>®</sup> . . . . .	38
4.2.2	Provini e sistema di alimentazione . . . . .	39
4.2.3	Sistema di misura scariche condotte . . . . .	40
4.2.4	Sistema di misura per le scariche irradiate . . . . .	41
4.2.5	Misure di scariche parziali . . . . .	41
4.3	Analisi sulla quantizzazione delle scariche e sul rumore di fondo . . . . .	46
4.3.1	Mappa Equivalent Time–Equivalent Bandwidth . . . . .	47
4.3.2	Sistema di acquisizione . . . . .	47
4.3.3	Test sperimentali . . . . .	48
4.4	Comparazione tra sistemi di misura che impiegano un diverso numero di bit per la quantizzazione dei segnali misurati . . . . .	52
4.4.1	Modello matematico . . . . .	53
4.4.2	Sistema di acquisizione . . . . .	53
4.4.3	Test sperimentali . . . . .	54
<b>5</b>	<b>Approcci di apprendimento veloce</b>	<b>58</b>
5.1	Reti neurali feedforward basate sulla randomicità . . . . .	59
5.2	Cenni sulla teoria dell'apprendimento con le funzioni di similarità . . . . .	62
5.3	Convergenza tra l'apprendimento con le funzioni di similarità e l'Extreme Learning Machine . . . . .	64
5.3.1	Parametro di forma della funzione di attivazione . . . . .	65
5.3.2	Campionamento dei landmark . . . . .	67
5.3.3	Riepilogo . . . . .	67
5.4	Analisi Geometrica . . . . .	68
5.4.1	Funzioni di attivazione generiche . . . . .	69
5.4.2	Funzione soglia e prodotto scalare . . . . .	72
5.5	Algoritmi per un mapping efficiente . . . . .	74
5.5.1	Algoritmo per il campionamento del fattore di forma . . . . .	75
5.5.1.1	Risultati sperimentali . . . . .	76
5.5.1.2	Considerazioni . . . . .	86
5.5.2	Algoritmo per il parametro della soglia . . . . .	86
5.5.2.1	Risultati sperimentali . . . . .	87
5.5.3	Analisi Comparativa . . . . .	91
<b>6</b>	<b>Implementazione hardware</b>	<b>93</b>
6.1	Realizzazione digitale . . . . .	95
6.1.1	Implementazione seriale del neurone . . . . .	97
6.1.2	Realizzazione parallela del neurone . . . . .	99
6.1.3	Confronto tra i due design: utilizzo dell'area vs latenza . . . . .	101
6.2	Architettura basata su generatore di numeri pseudo casuali . . . . .	102
6.3	Analisi sperimentale . . . . .	103

---

6.3.1	Implementazione dell'architettura seriale . . . . .	104
6.3.2	Confronto tra l'implementazione seriale e parallela . . . . .	105
6.3.3	Confronto tra le implementazioni con ROM e PRNG . . . . .	109
6.4	Considerazioni teoriche finali . . . . .	111
6.5	Esempio applicativo sulla classificazione delle scariche parziali . . . . .	112
6.5.1	Test Sperimentali . . . . .	113
6.5.2	Test delle performance hardware . . . . .	115
6.5.3	Considerazioni finali . . . . .	115
<b>7</b>	<b>Classificazione non supervisionata</b>	<b>117</b>
7.1	Acquisizione sperimentale dei dati . . . . .	117
7.2	Estrazione delle features . . . . .	119
7.2.1	Mapping dei dati con t-SNE . . . . .	120
7.2.2	Mapping dei dati con PCA . . . . .	122
7.2.3	Mapping dei dati con TW . . . . .	123
7.3	Classificazione non supervisionata . . . . .	123
7.3.1	K-means . . . . .	124
7.3.2	DBSCAN . . . . .	128
7.3.3	OPTICS . . . . .	130
7.4	Osservazioni . . . . .	132
<b>8</b>	<b>Impianto pilota per misure di scariche parziali su turbine eoliche</b>	<b>135</b>
8.1	Struttura della turbina eolica Vestas . . . . .	135
8.2	Misure per fini diagnostici . . . . .	137
8.3	Sistema remoto per monitoraggio a lungo termine . . . . .	137
8.3.1	Monitoraggio Temporale . . . . .	140
<b>9</b>	<b>Conclusioni e sviluppi futuri</b>	<b>142</b>
9.1	Sviluppi futuri . . . . .	143
	<b>Bibliografia</b>	<b>144</b>

# Elenco delle figure

2.1	Scarica con emissione di luce. . . . .	6
2.2	Circuito fisico (immagine a sinistra) ed elettrico (figura a destra) equivalenti di un difetto all'interno dell'isolamento. . . . .	8
2.3	Sequenza di scarica in un difetto all'interno di un dielettrico solido. . . . .	10
2.4	Treeing elettrico. . . . .	11
2.5	Schematizzazione scarica corona. . . . .	11
2.6	Esempi scariche superficiali. . . . .	12
2.7	Simulazione impulso di scarica. . . . .	13
2.8	Esempio di impulso di scarica reale. . . . .	13
2.9	Circuito con impedenza di misura in serie al condensatore di accoppiamento. . . . .	18
2.10	Circuito con impedenza di misura in serie all'oggetto in prova. . . . .	18
2.11	Esempi di due PD pattern. . . . .	19
3.1	Esempio di classificatore creato con SVM in uno spazio bidimensionale. . . . .	30
3.2	Esempio di rete neurale a singolo strato nascosto. . . . .	33
4.1	Sensore induttivo di tipo <i>HFCT</i> . . . . .	36
4.2	Circuito di misura condotto. . . . .	37
4.3	Antenna SolBox <sup>®</sup> . . . . .	38
4.4	Provino twisted pair. . . . .	39
4.5	Tensione di alimentazione in uscita dall'AWADIT. . . . .	40
4.6	Sistema per la misura delle scariche condotte. . . . .	41
4.7	Confronto di scariche parziali misurate irradiate e condotte su provino twisted pair. . . . .	42
4.8	Confronto scarica parziale misurata con sistema condotto (a sinistra) e con antenna (a destra). . . . .	43
4.9	Spettro delle scariche captate dalle due antenne. . . . .	44
4.10	Acquisizione dettagliata di una scarica irradiata. . . . .	45
4.11	Acquisizione dettagliata del rumore prodotto con l'inverter. . . . .	45
4.12	Spettro dei segnali di scarica e rumore misurati in laboratorio. . . . .	46
4.13	Sistema di acquisizione delle scariche. . . . .	47
4.14	Esempio di scarica acquisita. . . . .	48
4.15	Mappa TW normalizzata con segnali ben quantizzati. . . . .	49
4.16	Clustering su grafico TW variando il fattore di scala G. . . . .	50
4.17	Clustering con fattore di ampiezza G=0.15: errore nel riconoscimento dei segnali. . . . .	51
4.18	Distanza tra i centroidi dei cluster delle scariche e del rumore in funzione di G. . . . .	51

4.19	Scarica quantizzata per diversi valori di $G$ . . . . .	52
4.20	Sistema di acquisizione. . . . .	54
4.21	Acquisizione in un periodo della tensione di alimentazione a $50Hz$ . . . . .	54
4.22	Esempio di segnali aventi un'ampiezza piccola. . . . .	55
4.23	Grafico TW a 8 e 12 bit sui dati NACF proiettati. . . . .	56
4.24	L'algoritmo k-means trova i centroidi nel nuovo spazio e crea i cluster. I dati classificati erroneamente vengono rappresentati con dei cerchi verdi. . . . .	56
4.25	Classificazione dei segnali a 8 bit utilizzando l'indice di similarità $\rho$ . I cerchi verdi sono i segnali di scarica che l'algoritmo attribuisce erroneamente al cluster del rumore. . . . .	57
5.1	Esempio di dati rimappati con la TBF: (a) spazio originale di input; (b) dati rimappati con un valore appropriato di $r$ ; (c) dati rimappati con un valore non appropriato di $r$ . . . . .	71
5.2	Un esempio di dati nello spazio di input 2D: (a) separatore lineare con errore $\epsilon = 0$ ; (b) stesso separatore lineare ma diverso bias $r$ ; (c) attivazione ottenuta utilizzando il separatore lineare (a) e il segno come attivazione (linea arancione) o la sigmoide riscalata (linea blu) (d) lo stesso formato della figura (c) ma il separatore lineare della figura (b) . . . . .	74
5.3	Risultati degli esperimenti eseguiti sul dataset Ionosphere: a) sigmoide; b) RBF; c) multiquadrica . . . . .	79
5.4	Risultati degli esperimenti sul dataset Glass: a) sigmoide; b) RBF; c) multiquadrica . . . . .	81
5.5	Risultati degli esperimenti utilizzando il dataset Landsat: a) sigmoide; b) RBF; c) multiquadrica . . . . .	82
5.6	Risultati degli esperimenti utilizzando il dataset Covertype: a) sigmoide; b) RBF; c) multiquadratica . . . . .	84
5.7	Risultati degli esperimenti utilizzando il dataset CodRNA: a) sigmoide; b) RBF; c) multiquadratica . . . . .	85
6.1	Architettura dell'implementazione digitale del predittore: approccio sequenziale. . . . .	97
6.2	Esempio di flusso delle operazioni per l'architettura sequenziale. . . . .	98
6.3	Architettura per l'implementazione digitale del predittore: approccio parallelo. . . . .	99
6.4	Esempio di flusso del processo dell'architettura parallela. . . . .	100
6.5	Configurazioni che corrisponderebbero all'occupazione totale di area quando si realizza un predittore in base all'implementazione seriale del neurone. . . . .	104
6.6	Test su CPLD: configurazioni che corrisponderebbero all'occupazione totale di area durante la realizzazione hardware del predittore: (a) approccio sequenziale, (b) approccio in pipeline. . . . .	106
6.7	Test su FPGA: configurazioni che corrisponderebbero all'occupazione totale di area durante la realizzazione hardware del predittore: (a) approccio sequenziale, (b) approccio parallelo. . . . .	108
6.8	Test su CPLD: configurazioni che corrisponderebbero all'occupazione totale di area quando si realizza su hardware il predittore con memorie rom standard (diamanti come marcatori) o PRNG (marcatori circolari): (a) approccio sequenziale; (b) approccio parallelo. . . . .	110

6.9	Test su CPLD: configurazioni che corrisponderebbero all'occupazione totale di area quando si realizza su hardware il predittore con memorie rom standard (diamanti come marcatori) o PRNG (marcatori circolari): (a) approccio sequenziale, (b) approccio parallelo. . . . .	111
6.10	Sistema di acquisizione dei segnali di scarica. . . . .	113
7.1	Sistema di acquisizione. . . . .	118
7.2	Bobina con isolamento in nastro micato. . . . .	118
7.3	Segnali provenienti dalle sorgenti di scarica delle due bobine. I segnali riportati sono stati campionati in un periodo della sinusoide a $50Hz$ . . . . .	119
7.4	PD Pattern relativi alle sorgenti di scarica delle due bobine. . . . .	120
7.5	t-SNE applicato ai dati originali. . . . .	121
7.6	t-SNE applicato alle funzioni di autocorrelazione. . . . .	121
7.7	PCA applicata ai dati originali. . . . .	122
7.8	PCA applicata alle funzioni di autocorrelazione. . . . .	123
7.9	TW applicato ai dati originali. . . . .	124
7.10	TW applicato alle funzioni di autocorrelazione. . . . .	124
7.11	Distanze dei dati dai rispettivi centroidi al variare del numero di cluster nello spazio TW dei dati originali. . . . .	125
7.12	Distanze dei dati dai rispettivi centroidi al variare del numero di cluster nello spazio TW delle funzioni di autocorrelazione. . . . .	125
7.13	Distanze dei dati dai rispettivi centroidi al variare del numero di cluster nello spazio t-SNE delle funzioni di autocorrelazione. . . . .	126
7.14	Risultato del clustering sui dati originali proiettati nello spazio TW. . . . .	126
7.15	Risultato del clustering sulle funzioni di autocorrelazione proiettate nello spazio TW. . . . .	127
7.16	Risultato del clustering sulle funzioni di autocorrelazione proiettate nello spazio t-SNE con la distanza di Chebychev. . . . .	127
7.17	Risultato del clustering con DBSCAN sui dati originali proiettati nello spazio TW. . . . .	128
7.18	Risultato del clustering con DBSCAN sulle funzioni di autocorrelazione proiettate nello spazio TW. . . . .	129
7.19	Risultato del clustering con DBSCAN sulle funzioni di autocorrelazione proiettate nello spazio t-SNE con la distanza di Chebychev. . . . .	129
7.20	Reachability distances ordinate dei dati originali proiettati nello spazio TW. . . . .	130
7.21	Reachability distances ordinate delle funzioni di attivazione proiettate nello spazio TW. . . . .	131
7.22	Reachability distances ordinate delle funzioni di autocorrelazione proiettate nello spazio t-SNE con la distanza di Chebychev. . . . .	131
7.23	Risultato del clustering con OPITCS sui dati originali proiettati nello spazio TW, con un valore di soglia pari a 0.075. . . . .	132
7.24	Risultato del clustering con OPITCS sulle funzioni di autocorrelazione proiettate nello spazio TW, con un valore di soglia pari a: a) 0.062 e b) 0.085. . . . .	133
7.25	Risultato del clustering con OPITCS sulle funzioni di autocorrelazione proiettate nello spazio TW, con un valore di soglia pari a 16. . . . .	134
8.1	Schema della navicella della turbina eolica. . . . .	136
8.2	Schema a blocchi del sistema di monitoraggio. . . . .	138



---

8.3	Posizionamento delle antenne all'interno della navicella: a) testata 2 del generatore, b) parte superiore del trasformatore. . . . .	139
8.4	Posizionamento delle due ferriti attorno ai cavi in uscita dallo statore. . .	139
8.5	Acquisizione effettuata dopo 2 mesi. . . . .	140
8.6	Acquisizione effettuata dopo 4 mesi. . . . .	140
8.7	Acquisizione effettuata dopo 6 mesi. . . . .	141

# Elenco delle tabelle

3.1	Funzioni kernel . . . . .	32
5.1	Funzioni di attivazione classiche . . . . .	69
5.2	Confronto tra la ELM e SVM per il dataset Ionosphere. . . . .	80
5.3	Confronto tra ELM e SVM per il dataset Glass . . . . .	80
5.4	Confronto tra ELM e SVM per il dataset Landasat. . . . .	83
5.5	Confronto tra ELM e SVM per il dataset Covertype. . . . .	83
5.6	Confronto tra ELM e SVM per il dataset Cod-RNA. . . . .	86
5.7	Performance di generalizzazione con la $l_2$ come termine di regolarizzazione. . . . .	89
5.8	Performance di generalizzazione e sparsità. . . . .	90
6.1	Occupazione di area - CPLD 5M1270Z . . . . .	105
6.2	Implementazione su CPLD: $D_{25}/D_N$ in funzione di $N$ . . . . .	107
6.3	Realizzazione su FPGA: $D_{25}/D_N$ in funzione di $N$ . . . . .	109
6.4	Features per ciascuna distribuzione. . . . .	113
6.5	Errori percentuali di generalizzazione . . . . .	114
6.6	Realizzazione su FPGA. . . . .	115
6.7	Realizzazione su CPLD. . . . .	115

# Capitolo 1

## Introduzione

I sistemi di isolamento delle macchine elettriche sono soggetti a continue sollecitazioni di tipo termico, elettrico, meccanico e ambientale (TEMA) che causano l'invecchiamento degli stessi. Perciò, durante la normale attività della macchina, è possibile che si osservi la formazione di scariche parziali nell'isolamento, che si presentano sotto forma di impulsi di corrente. Questi segnali sono il sintomo e la causa del degrado del sistema di isolamento e possono portare al fenomeno della scarica totale, ovvero un cortocircuito tra due conduttori normalmente isolati. Il danno economico che ne deriva è molto ingente considerando il fermo impianto e la riparazione, quando possibile, o la sostituzione del macchinario nel peggiore delle ipotesi.

Per prevenire la scarica totale vengono effettuati controlli programmati che causano però il fermo dell'impianto portando a perdite economiche e disagi. Inoltre risulta impossibile prevedere un guasto all'interno di una apparecchiatura poiché, se si considera ad esempio il caso di due motori apparentemente identici, essi possono presentare comportamenti totalmente differenti se sottoposti a condizioni ambientali diverse.

Per questi motivi si rende necessario disporre di un sistema di diagnostica predittiva in grado di monitorare e di identificare il tipo o i tipi di difetti presenti nel sistema di isolamento. In questo modo si è in grado di prevenire i danni e non fermare l'impianto per una manutenzione non necessaria. Il sistema di diagnostica potrà fornire agli operatori informazioni sullo stato dell'apparecchiatura monitorata, e quindi decidere se intervenire immediatamente sulla macchina prossima al guasto.

Per il riconoscimento automatico dei difetti è stato necessario adottare tecniche di apprendimento che rientrano nel campo del *Machine Learning* (ML). Lo scopo di questi algoritmi è di inferire proprietà su dati mai visti, dopo una fase di apprendimento chiamata *training*, durante la quale viene imparata la relazione  $f$  tra i dati e le proprietà

da inferire. In generale,  $f$  è una funzione parametrica: in base al tipo di algoritmo scelto esistono diversi parametri che devono essere imparati. Alcuni tra gli algoritmi più conosciuti sono le reti neurali, la Support Vector Machine (SVM), le random forest, il K-means. Durante la fase di training, basata su un insieme di dati, i parametri della funzione  $f$  vengono scelti in modo tale da produrre la migliore predizione su dati mai visti dal modello. Quindi l'algoritmo produce un classificatore (una regola  $f$ ) che fornisca predizioni su nuovi dati mai utilizzati durante la fase di training.

La qualità del processo di inferenza è strettamente relazionata alla rappresentazione dei dati, ovvero l'insieme delle *features* estratte dal dominio di input. Questo pre-step, chiamato "estrazione delle features", è l'aspetto cruciale per la fase di apprendimento perché le informazioni derivate dai dati di input devono essere rappresentative dei punti nello spazio originale.

Gli algoritmi di apprendimento automatico possono essere divisi in due grandi famiglie: supervisionati e non. I primi vengono utilizzati quando si conosce l'etichetta dei dati di training, ovvero si hanno a disposizione tutti i risultati della  $f$  che il modello dovrebbe fornire. Una volta che i parametri dell'algoritmo sono stati imparati, viene fornito un nuovo dato in ingresso al sistema di classificazione al fine di ottenere la classe di appartenenza. Esempi di questi metodi sono le reti neurali, la SVM e le random forest. I secondi si impiegano quando non si conosce l'etichetta dei dati di training, per cui non si hanno informazioni sul numero di classi in cui possono essere divisi i dati. Gli algoritmi più conosciuti per affrontare problemi di questo tipo sono quelli di *clustering*, quali K-means.

La tesi è organizzata come segue: il Capitolo 2 introduce il concetto di scarica parziale, mostrando le cause per cui si genera tale fenomeno e la normativa di riferimento in cui si descrivono alcuni sistemi di misura; il Capitolo 3 descrive lo stato dell'arte relativo alle tecniche di estrazione delle features e agli algoritmi di classificazione supervisionata e non delle scariche; nel Capitolo 4 vengono confrontati i sistemi di misura condotta e irradiata, con un'analisi sulla quantizzazione dei segnali; nel Capitolo 5 si discutono gli approcci di apprendimento veloci basati su reti neurali a singolo strato; il Capitolo 6 si riferisce all'implementazione hardware del predittore proposto nel Capitolo 5, mostrando un'applicazione di classificazione supervisionata al problema delle scariche; nel Capitolo 7 si confrontano varie tecniche di estrazione delle features e alcuni algoritmi di classificazione non supervisionata per il problema delle scariche. Infine, nel Capitolo 8 viene proposto un prototipo di sistema di acquisizione per la diagnostica su turbine eoliche.

## 1.1 Contributo

Il contributo di questa tesi può essere riassunto nei punti seguenti:

1. tecniche di apprendimento supervisionato basate su reti neurali che sfruttano la randomicità nella scelta parametri del modello (Capitolo 5 e Capitolo 6);
2. confronto di algoritmi per l'estrazione delle features e per la classificazione non supervisionata (Capitolo 7);
3. progettazione di un sistema di acquisizione remota per la diagnostica in ambienti ostili (Capitolo 8).

Nella prima parte è stata svolta un'analisi riguardante il ruolo della randomicità nel processo di training di un modello di apprendimento, e sull'affinità tra due schemi basati sulla Extreme Learning Machine (ELM) e sulle funzioni di similarità. Questi paradigmi condividono un approccio comune all'apprendimento induttivo che combina un mapping dei dati esplicito con un separatore lineare, anche se sembrerebbero adottare strategie differenti nel design dello strato di mapping. La ricerca mostra che la teoria dell'apprendimento con le funzioni di similarità può portare ad una nuova interpretazione del paradigma della ELM, che sfocia in un quadro teorico comune. Sono stati ottenuti nuovi approfondimenti sul modello ELM, soprattutto per quanto riguarda la strategia per l'impostazione dei parametri dei neuroni. I risultati sperimentali confermano che i nuovi metodi migliorano gli approcci convenzionali, in particolare nel trade-off tra l'accuratezza della classificazione e complessità della macchina, ovvero la dimensionalità dello spazio rimappato.

Inoltre, ci si è posti come obiettivo l'implementazione di reti neurali a singolo strato su dispositivi digitali a basso costo e con risorse limitate. Per questo scopo la hard-limiter è stata scelta come funzione di attivazione dei neuroni poiché, dal punto di vista hardware, è interpretabile come un multiplexer. Al fine di avere un buon bilanciamento tra numero di neuroni e accuratezza della classificazione è stato necessario adottare delle strategie ad-hoc per la scelta dei parametri nella fase di training. I risultati confermano che il design di queste reti neurali porta ad un'implementazione digitale efficiente del predittore su dispositivi con risorse limitate. Infine una delle architetture proposte è stata impiegata per la classificazione supervisionata di alcune tipologie di difetti.

Nella seconda parte sono state confrontate diverse tecniche di mapping di segnali temporali in spazi a bassa dimensionalità per permettere una facile visualizzazione. I dati proiettati sono stati raggruppati con diversi algoritmi di clustering, cercando quello che individui una separazione migliore rispetto agli altri. L'algoritmo del K-means, si è

dimostrato utile per calcolare i parametri di ingresso di altri due modelli di apprendimento non supervisionato. Dal punto di vista della separazione questo algoritmo non ha portato a risultati soddisfacenti poiché la forma dei cluster negli spazi rimappati non è omogenea. Metodi basati sulla densità spaziale dei punti comportano buoni risultati nella separazione di diverse fonti di scariche.

Nell'ultima parte viene presentato un prototipo di un sistema di acquisizione remota di segnali su turbine eoliche. Questo sistema è costituito da quattro sensori collegati ad un oscilloscopio pilotato da un computer industriale, dotato di connessione internet per la trasmissione dati. Per monitorarlo regolarmente è stato sviluppato un software ad-hoc che permetta sia la riconfigurazione manuale dei parametri relativi all'oscilloscopio, sia il campionamento regolare dei segnali captati dai sensori e la loro trasmissione ad un server.

## Capitolo 2

# Il fenomeno delle scariche parziali

Una scarica parziale viene definita dalla normativa CEI EN 60270 [1] come una:

“Scarica elettrica localizzata che attraversa solo parzialmente l’isolamento tra conduttori e che può essere o no adiacente a un conduttore.”

È necessario precisare che le scariche parziali sono non solo il sintomo, ma anche la causa del parziale cedimento dell’isolamento interposto tra due conduttori a potenziale differente. Con il passare del tempo, se non viene eseguita un’azione manutentiva, l’isolamento tenderà a cedere completamente, di fatto cortocircuitando i due conduttori, portando così al fenomeno della scarica totale che provoca il fermo macchina. E’ dunque necessario valutare il grado di pericolosità di tale attività in modo tale da prevenire la rottura completa dell’isolamento.

### 2.1 Cause dell’attività di scarica

Le scariche derivano da concentrazioni locali di campo elettrico e compaiono come impulsi di corrente con durata che va da qualche nanosecondo ( $ns$ ) a qualche microsecondo ( $\mu s$ ), e sono accompagnate dall’emissione di suono, luce e calore. Il campo elettrico, durante il fenomeno delle scariche parziali, è tale da superare la rigidità dielettrica del materiale isolante, ma il potenziale richiesto per innescare una scarica parziale risulta inferiore al valore per cortocircuitare completamente i due conduttori.

Siccome le scariche avvengono a causa di difetti nel sistema di isolamento, sono utilizzabili come indicatore della presenza dei difetti stessi. Ogni volta che si innesca una scarica parziale si crea un surriscaldamento locale dovuto al passaggio di corrente che porta a deformazioni chimiche e fisiche dell’isolamento nella zona interessata. Queste

deformazioni possono essere temporanee o permanenti: è possibile che si possano depositare sulla superficie residui carboniosi riducendo la resistività dell'isolante, e generare sostanze come ozono ( $O_3$ ) e ossidi di azoto ( $NO_2$ ), chimicamente molto attive. Questi fattori accelerano il deterioramento del materiale isolante, con un incremento dell'attività di scarica.

Esistono materiali auto-ripristinanti che riportano il loro stato a quello precedente alla scarica, risultando così molto più resistenti ai fenomeni di invecchiamento dell'isolamento elettrico. Alcuni esempi di questi materiali sono i gas e gli oli. Resine e polimeri, al contrario, non sono auto-ripristinanti, in quanto il passaggio di corrente al loro interno porta ad un degradamento irreversibile. In un ambito industriale sono adoperate entrambe le tecnologie per la costruzione degli isolamenti delle macchine.

## 2.2 Sistemi di misura delle scariche parziali

Le scariche parziali possono essere misurate in diversi modi, a seconda del sistema su cui si verificano. Dato che ad ogni scarica è associata un'emissione di luce, è possibile osservare i fenomeni esterni al buio: se l'attività di scarica è sufficientemente intensa si nota ad occhio nudo un alone di luce azzurra attorno al punto in cui sta avvenendo (Fig. 2.1), spesso accompagnato da un tipico fruscio.



FIGURA 2.1: Scarica con emissione di luce.

Ovviamente si tratta di misure qualitative, non utilizzabili in un sistema reale (considerando anche il fatto che molte tipologie di scariche avvengono all'interno del macchinario). Esistono però sistemi di misura che sfruttano l'emissione di suono per la localizzazione delle scariche, sfruttando microfoni molto direzionali e sensori di tipo piezoelettrico [2].



I segnali acquisiti vengono amplificati e traslati in frequenze udibili da un operatore. Questo sistema è poco sensibile a scariche di bassa intensità e notevolmente soggetto al rumore ambientale.

Esistono metodi ottici che, sfruttando fotomoltiplicatori e apparati fotografici, possono localizzare scariche parziali con sensibilità inferiori a  $0.1 \text{ pC}$  soltanto in condizioni particolari [3].

Per gli isolamenti in olio è possibile effettuare analisi chimiche ad intervalli periodici, in modo tale da rilevare sostanze disciolte nel liquido [4].

Possono essere impiegati metodi di tipo elettromagnetico che si basano sull'utilizzo di opportune antenne: esse captano il segnale elettromagnetico generato da un breve impulso di corrente a cui è associato l'evento di scarica. Con questo tipo di misura si ottengono soltanto informazioni qualitative sulla scarica avvenuta, poiché il segnale misurato ha subito effetti di attenuazioni, distorsioni e riflessione multipla che modificano l'impulso originale.

Per ottenere informazioni quantitative sulle scariche è necessario misurare la quantità di carica associata al singolo impulso. Facendo passare il conduttore all'interno di trasformatori amperometrici (TA), realizzati con ferriti o con avvolgimenti di tipo Rogovsky, è possibile trasdurre l'impulso di corrente [5]. Con questo metodo, tuttavia, non si può rilevare la quantità di carica associata alla scarica. Per ottenere informazioni quantitative e sul trasferimento della carica è necessario collegare il sistema di misura direttamente all'alimentazione della macchina. Gli impulsi vengono captati mediante un partitore capacitivo e un'impedenza RLC. Il sistema viene calibrato ad ogni nuova misura: utilizzando un impulso con carica nota il valore di tensione misurato viene associato al valore di carica che lo ha generato.

### 2.3 Tipologia di scariche parziali

I difetti del sistema di isolamento possono derivare sia da problemi di costruzione sia dalle sollecitazioni termiche, elettriche, meccaniche e ambientali (TEMA) che causano l'invecchiamento dei materiali. Ad esempio, considerando macchine con isolamento in resina epossidica, si possono creare, in fase di produzione, dei piccoli vuoti nella resina che successivamente causeranno scariche parziali durante il normale funzionamento [6]. La contaminazione delle superfici da parte di polveri conduttive può portare a scariche parziali sulla superficie stessa [7]. Inoltre i macchinari durante l'installazione o il trasporto possono subire stress meccanici che creano piccole crepe nell'isolamento o tra gli

elementi che dovrebbero rimanere allo stesso potenziale. Infine può accadere che un conduttore non sia correttamente arrotondato o una terminazione abbia degli spigoli vivi, in questi casi la distribuzione del campo potrebbe risultare non sufficientemente uniforme, dando origine a scariche di tipo corona [8]. Come appena osservato i fenomeni che scatenano l'attività delle scariche parziali possono essere molti e di natura differente, ma sono suddivisibili in due grandi sottofamiglie: scariche interne e scariche esterne al sistema di isolamento.

### 2.3.1 Scariche interne

Durante la fabbricazione di un dielettrico solido può succedere che impurità (es. bolle di gas o particelle estranee) vengano inglobate all'interno dell'isolamento. Queste, caratterizzate con molta probabilità da una costante dielettrica inferiore a quella dell'isolamento, portano ad avere concentrazioni di campo, superando quindi la rigidità del dielettrico localmente al difetto. Si generano così delle scariche parziali limitate inizialmente al luogo dell'impurità. In Fig. 2.2 viene riportato il modello del circuito equivalente dei difetti inglobati in un dielettrico solido.

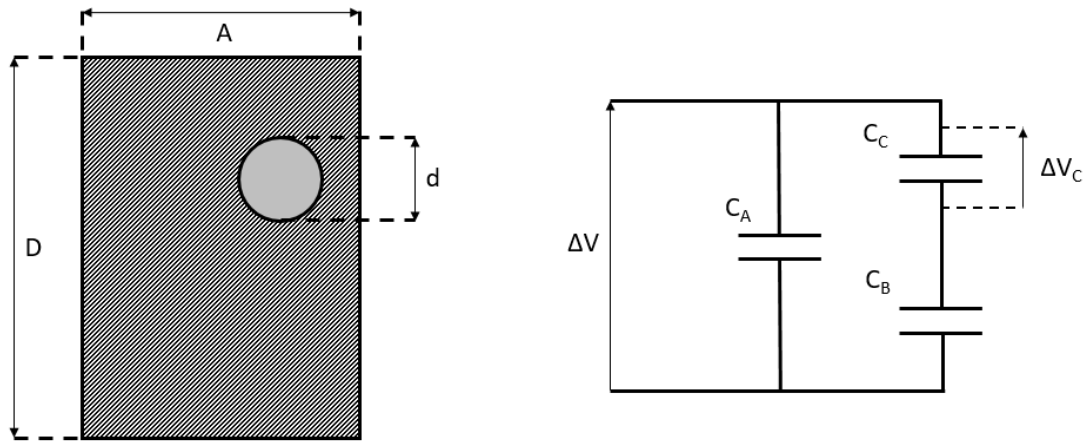


FIGURA 2.2: Circuito fisico (immagine a sinistra) ed elettrico (figura a destra) equivalenti di un difetto all'interno dell'isolamento.

Essendo il difetto completamente inglobato in un materiale isolante e supponendo che l'altezza del difetto  $d$  sia molto minore rispetto all'altezza dell'isolamento  $D$  ( $d \ll D$ ), il campo all'interno del difetto  $E_C$  risulterebbe proporzionale al campo in cui è immerso l'isolamento stesso ed è esprimibile come:

$$E_C = \epsilon_r E_A \quad (2.1)$$

dove  $E_A$  è il campo elettrico nel dielettrico circostante ed  $\epsilon_r$  la costante dielettrica del materiale isolante, quindi risulta che  $E_C > E_A$ , essendo  $\epsilon_r > 1$ .

Nel circuito elettrico equivalente,  $C_A$ ,  $C_C$  e  $C_B$  rappresentano, rispettivamente, la capacità dell'intero isolamento, quella del difetto e quella restante in prossimità del difetto. Si possono dunque calcolare le capacità  $C_B$  e  $C_C$  come:

$$C_B = \frac{\epsilon_0 \epsilon_r A}{D - d} \quad (2.2)$$

$$C_C = \frac{\epsilon_0 A}{d} \quad (2.3)$$

considerando la sezione  $A$  uguale per entrambi i casi. In questo modo si valuta la tensione ai capi del difetto ( $V_C$ ) come:

$$V_C = \frac{C_B}{C_B + C_C} V_A \quad (2.4)$$

Sostituendo la (2.2) e la (2.3) dentro la (2.4), si ottiene:

$$V_C = \frac{V_A}{1 + \frac{1}{\epsilon_r} \left( \frac{D}{d} - 1 \right)} \quad (2.5)$$

da cui si può ricavare la tensione ai capi del materiale isolante (tensione critica  $V_{A_S}$ ) tale per cui una scarica si innesca all'interno della cavità:

$$V_{A_S} = \epsilon_C \cdot d \left[ 1 + \frac{1}{\epsilon_r} \left( \frac{D}{d} - 1 \right) \right] \quad (2.6)$$

dove  $\epsilon_C$  è la rigidità dielettrica del difetto.

Applicando una tensione sinusoidale  $V_A$  ai capi del dielettrico la cavità sarà soggetta alla tensione  $V_C$  proporzionale a  $V_A$  secondo la (2.5). Se  $V_C$  supera la cosiddetta tensione di innesco  $V_{INN}$  (ricavabile dalla conoscenza delle condizioni ambientali iniziali e dall'utilizzo delle curve di Paschen), un elettrone si rende libero dando inizio ad una "valanga elettronica", causando quindi una scarica da un lato all'altro del difetto. La tensione successivamente si riduce fino a quando raggiunge quella di estinzione  $V_{EST}$ . A questo punto il processo può ripetersi o arrestarsi in base al cambiamento che ha subito il sito di scarica: se vengono rilasciati residui carboniosi conduttivi all'interno del difetto si forma una superficie equipotenziale tale da estinguere il processo di scarica, se la cavità rimane libera o lo strato carbonioso conduttivo non è tale da coprire interamente la superficie allora il processo continua. Questo viene rappresentato in Fig. 2.3. Nell'immagine si osserva che ogniqualvolta la tensione  $V_C$  supera quella di innesco  $V_{INN}$ , sia positiva che negativa, essa crolla alla tensione di estinzione  $V_{EST}$ , dando luogo ad un impulso di corrente. L'impulso è molto breve e solitamente ha una durata inferiore al  $\mu s$  (nella

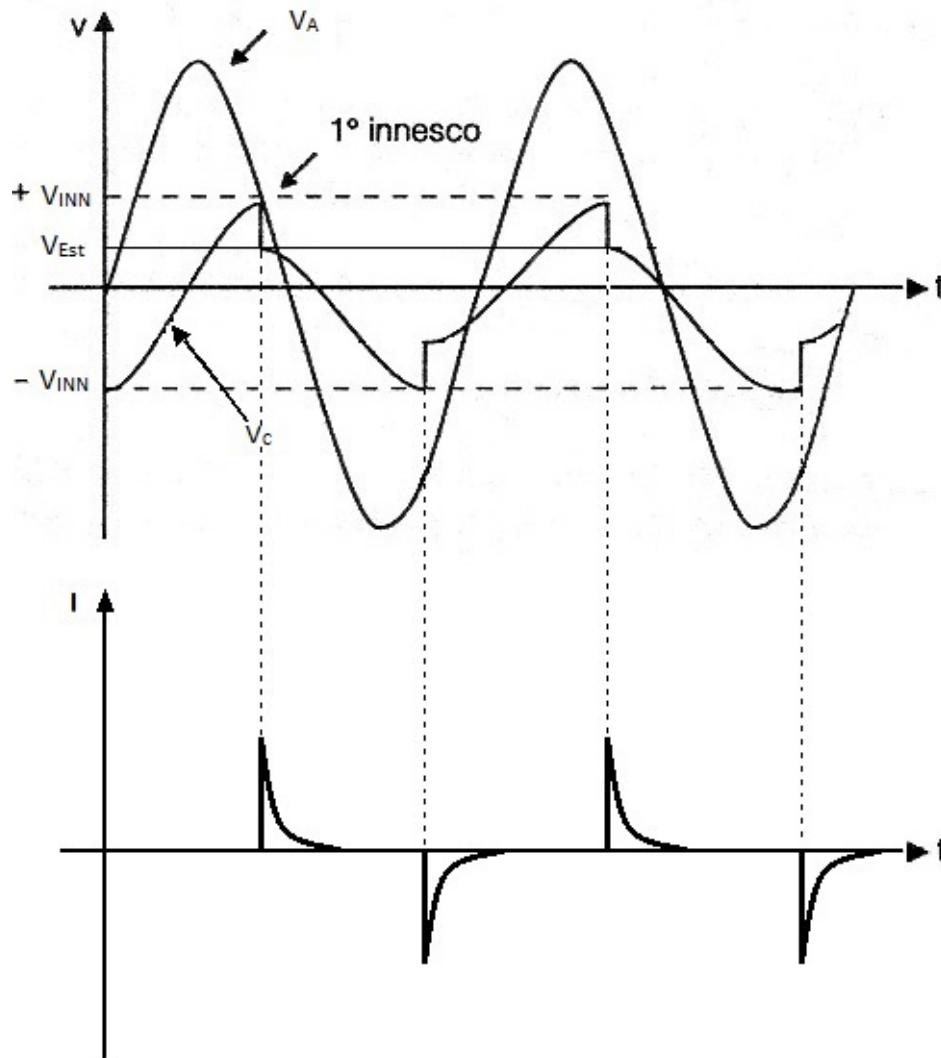


FIGURA 2.3: Sequenza di scarica in un difetto all'interno di un dielettrico solido.

Fig. 2.3 il fenomeno è stato volutamente allargato per renderlo visibile sul periodo di alimentazione da  $50\text{Hz}$ ).

Il passaggio di elettroni durante la scarica porta ad un innalzamento della temperatura della cavità e al rilascio di energia, provocando cambiamenti ambientali in grado di rompere i legami chimici delle pareti del vacuolo. L'erosione che ne deriva porta ad un allargamento del difetto, intensificando dunque l'attività di scarica e dando luogo a ramificazioni (*treeing elettrico*, Fig. 2.4). D'altro canto è possibile che il cambiamento chimico della superficie interna della cavità porti ad una carbonizzazione della stessa, che si rende equipotenziale con l'isolamento che lo circonda, estinguendo dunque il fenomeno.

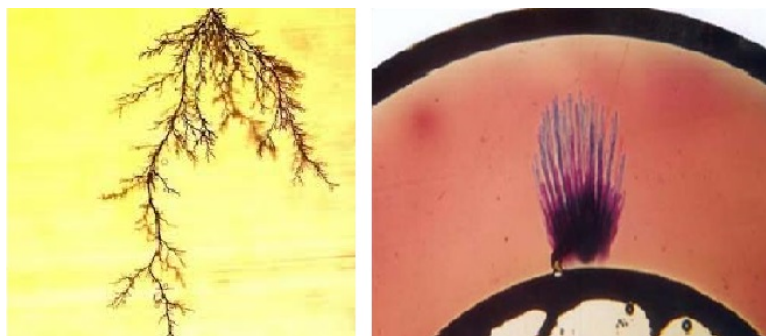


FIGURA 2.4: Treeing elettrico.

### 2.3.2 Scariche esterne

Le scariche esterne avvengono all'interfaccia tra dielettrici differenti. Esistono due tipi di fenomeni relativi a questa categoria: le scariche per effetto corona e quelle superficiali. Le prime si verificano quando la rigidità del gas isolante viene superata localmente. Infatti, quando il gradiente di potenziale supera un determinato valore, il fluido isolante si ionizza e diventa conduttore. Solitamente sono causate da elettrodi appuntiti, quali superfici che presentano spigoli, permettendo quindi una grande concentrazione di carica locale. Quando le tensioni sono elevate questo fenomeno può portare alla scarica totale creando un percorso induttivo nel gas. Se la ionizzazione continua quello che può accadere è l'innesco di un arco elettrico. Le scariche di tipo corona sono accompagnate dalla produzione di ozono e generano rumore udibile. L'ozono, combinandosi chimicamente, può corrodere le superfici metalliche esterne delle apparecchiature formando depositi che creano scariche elettriche disruptive. In Fig. 2.5 è riportata la schematizzazione delle scariche generate per effetto corona e in Fig. 2.1 un esempio reale.

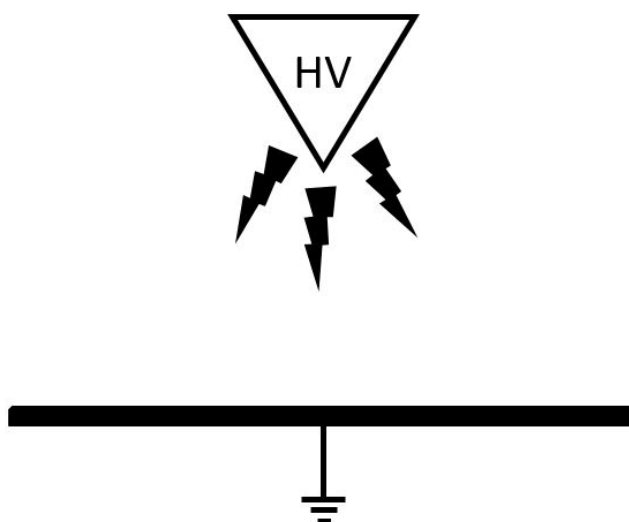


FIGURA 2.5: Schematizzazione scarica corona.

Le scariche superficiali si manifestano all'interfaccia di due dielettrici differenti dove è presente una elevata concentrazione di campo parallela alla superficie (Fig. 2.6a). Sono causate da contaminazioni e umidità che diminuiscono la rigidità dell'isolamento. La scarica si propaga lungo la superficie e, con il passare del tempo, può tracciare dei percorsi, riducendone così la resistività (Fig. 2.6b) fino a quando non avviene la scarica totale.

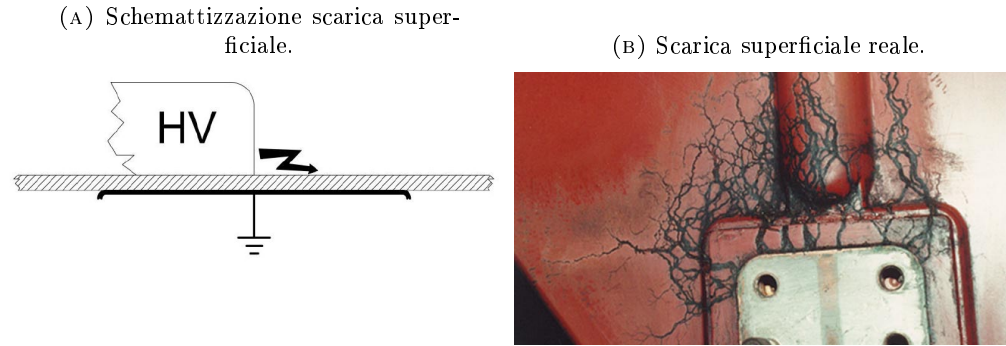


FIGURA 2.6: Esempi scariche superficiali.

### 2.3.3 Impulso di corrente associato alla scarica

L'impulso di corrente associato alla scarica è un fenomeno dell'ordine di qualche  $\mu s$ , con tempi di salita inferiori di circa dieci volte rispetto al tempo di discesa. L'impulso ideale potrebbe essere formulato nel seguente modo:

$$I(t) = I_0 \left( e^{-at} - e^{-bt} \right) \quad (2.7)$$

Dove  $a$  è la costante associata al tempo di salita e  $b$  quella relativa al tempo di discesa. Ipotizzando un tempo di salita di  $100 \text{ ns}$ , un tempo di discesa di  $1 \mu s$  e normalizzando l'ampiezza ad 1, si può rappresentare l'impulso come in Fig. 2.7.

In un caso reale l'impulso di scarica è normalmente soggetto a rumore e distorsioni dovute sia alla propagazione in un mezzo dispersivo sia al sistema di misura dell'impulso. In Fig. 2.8 viene riportato un esempio di scarica misurata su una bobina.

Confrontando quest'ultimo caso con quello ideale, si nota come i tempi di salita siano paragonabili. Dopo che la scarica ha raggiunto il suo picco si osservano due comportamenti differenti: nel caso ideale l'impulso si esaurisce con legge esponenziale, mentre in quello reale il segnale presenta oscillazioni prima di estinguersi. Queste sono dovute principalmente al sistema di misura, trattandosi di un circuito  $RLC$ , e al mezzo di propagazione.

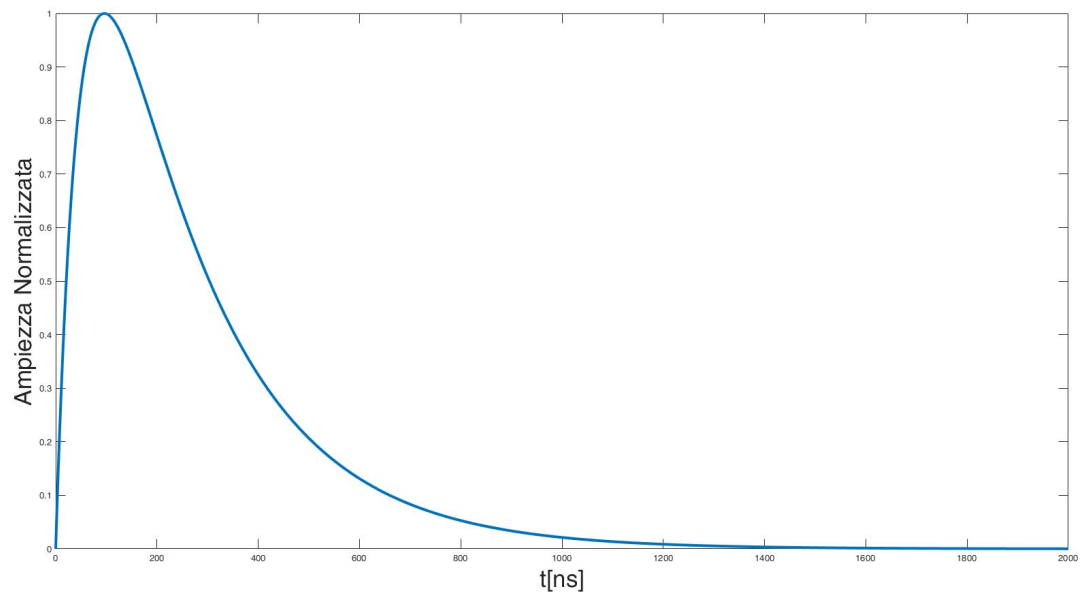


FIGURA 2.7: Simulazione impulso di scarica.

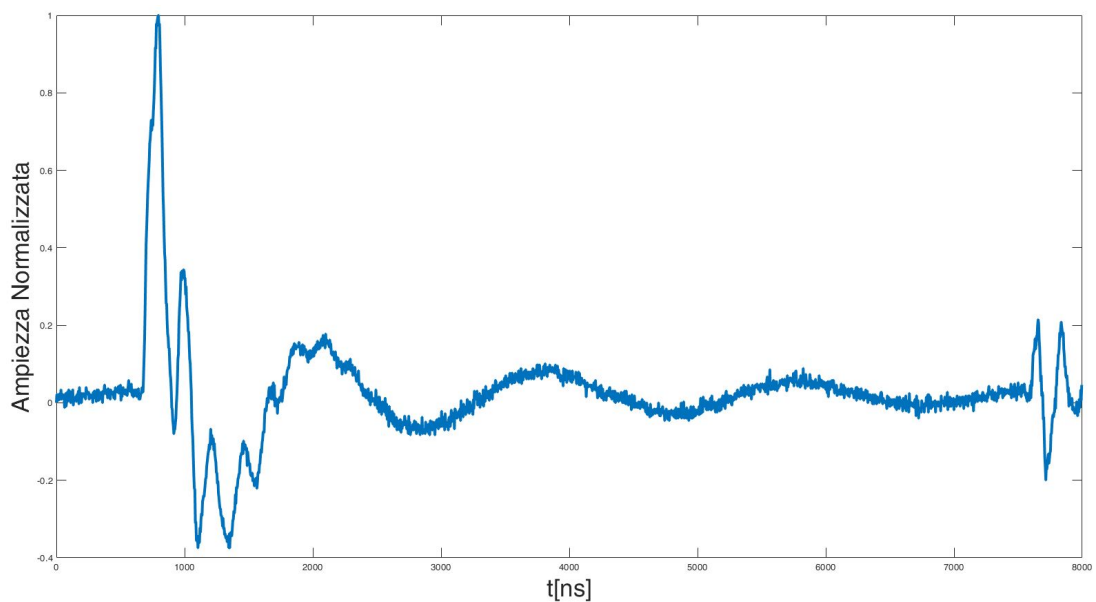


FIGURA 2.8: Esempio di impulso di scarica reale.

## 2.4 Misura delle scariche parziali

Come accennato nella Sezione 2.2, ad ogni fenomeno di scarica è associato un impulso di corrente. Per rilevare scariche parziali è quindi necessario riuscire a distinguere questi impulsi dal rumore ambientale e, in alcune applicazioni, anche misurarne l'ampiezza. Esistono due metodi per effettuare una diagnostica predittiva su un componente: si possono eseguire misure mentre il dispositivo è nelle normali condizioni di funzionamento (ON-LINE) o fornendo opportuni stimoli al sistema di isolamento (OFF-LINE). Nessuno

dei due è migliore dell'altro, entrambi hanno dei pregi da tenere in considerazione, a scapito di alcuni difetti.

#### **2.4.1 Misure ON-LINE**

Le misure ON-LINE vengono effettuate durante l'effettivo funzionamento della macchina. Il vantaggio di questa tipologia di misure è che il macchinario sarà sottoposto ai reali stress di tipo meccanico ed elettrico oltre alle effettive condizioni di temperatura e umidità. La principale difficoltà è dovuta al fatto che i sensori devono essere installati sul macchinario senza alterare il suo normale funzionamento. Inoltre le misure saranno facilmente soggette a rumore ambientale elevato oltre ad essere influenzate dagli altri macchinari (scariche provenienti da una macchina potrebbero essere visibili anche su un'altra).

Si potrebbero preferire le scariche ON-LINE a quelle OFF-LINE poiché non è necessario fermare l'impianto, quindi senza perdite di denaro. Il fermo avverrà solamente se è necessario operare una manutenzione al macchinario.

#### **2.4.2 Misure OFF-LINE**

Le misure OFF-LINE vengono eseguite con il sistema fermo, ovvero non alimentato nelle condizioni nominali di funzionamento. Nei sistemi trifase sarà dunque possibile analizzare una fase alla volta, eliminando di fatto i fenomeni di crosstalk. Il rumore ambientale risulterà inferiore, oltre a garantire il fatto che i fenomeni di scarica provenienti da altre macchine non vengono misurati. Questo permette di caratterizzare dettagliatamente la scarica consentendo di calibrare il sistema di misura.

Per avere un'evoluzione accurata dell'attività di scarica sarebbe opportuno misurare il sistema di isolamento fin dalla sua messa in servizio, poiché potrebbero esistere difetti provenienti dal processo di costruzione. Successivamente si continuerà a monitorare la macchina, controllandone così l'evoluzione.

Risulta facilmente intuibile che con le misure OFF-LINE alcuni fenomeni di scarica potrebbero non essere visibili, dato che le condizioni dell'isolamento (temperatura, umidità, stress elettro-meccanico) non sono quelle del normale funzionamento del macchinario.



## 2.5 Cenni alla normativa IEC 60270

La normativa IEC 60270 pone delle regole per la misura delle scariche parziali in ambito industriale. Nella norma sono descritti i criteri e circuiti da realizzare per la misurazione condotta delle scariche. Le prove possono essere effettuate sia in corrente continua sia in alternata. Il limite massimo della frequenza di alimentazione per cui si possano effettuare misurazioni nel rispetto della normativa è di  $400\text{Hz}$ . La norma si pone come obiettivo la riproducibilità della misura, ovvero il fatto che questa sia indipendente dal sistema di misura, ma senza considerare i fini diagnostici.

### 2.5.1 Grandezze principali definite dalla norma

Esistono alcune grandezze di riferimento per la misura delle scariche parziali.

- ***Scarica Parziale (SP)***

Scarica elettrica localizzata che attraversa solo parzialmente l'isolamento tra conduttori e che può essere o non adiacente a un conduttore. Come accennato nelle sezioni precedenti la scarica elettrica è localizzata in un punto del sistema isolante dove la concentrazione del gradiente del potenziale elettrico risulta superiore alla rigidità dielettrica dell'isolamento.

- ***Impulso elettrico relativo alla scarica***

Impulso di corrente che si verifica come conseguenza di una SP all'interno dell'oggetto in prova. Gli impulsi possono essere rilevati con opportuni circuiti di misura, sia in corrente che in tensione, con il collegamento di una opportuna impedenza.

- ***Carica apparente  $q$***

Carica misurabile associata all'evento di scarica. Quando avviene la scarica si avrà un breve impulso ai terminali dell'oggetto in prova, misurabile con il circuito apposito. In funzione dello strumento utilizzato si avrà più o meno sensibilità nella misura. La carica relativa all'impulso non sarà dunque quella reale messa in gioco dalla scarica stessa. E' possibile però calibrare il sistema di misura iniettando un impulso di corrente noto e misurando la risposta del sistema. In questo modo è sempre possibile ottenere misure ripetibili. L'unità di misura della carica apparente è il picoCoulomb ( $pC$ ).

- ***Corrente media di scarica  $I$***

Grandezza derivata e somma dei valori assoluti delle ampiezze delle singole cariche apparenti  $q_i$ , durante un certo intervallo di tempo di riferimento  $T_{ref}$ , diviso per

l'intervallo di tempo stesso  $T_{ref}$ :

$$I = \frac{1}{T_{ref}} \sum_{i=1}^n |q_i| \quad (2.8)$$

Essa si esprime generalmente in coulomb al secondo ( $C/s$ ) oppure in ampere ( $A$ ).

- **Angolo di fase  $\Phi_i$  e istante  $t_i$  in cui si verifica un impulso di SP**

La relazione tra l'istante di misura di un impulso rispetto all'angolo di fase (espresso in gradi sessagesimali) della tensione di alimentazione è:

$$\Phi_i = 360 * \left( \frac{t_i}{T} \right) \quad (2.9)$$

dove  $t_i$  è il tempo misurato tra l'istante del passaggio per lo zero del fronte montante della tensione di prova, precedente alla scarica, e l'impulso di scarica parziale.  $T$  è il periodo della tensione di prova. La definizione della (2.9) pone le basi per la costruzione dei diagrammi di distribuzione degli impulsi di scarica, chiamati *PD Pattern*.

- **Rumore di fondo**

Tutti i segnali rilevati dalla strumentazione non derivanti dall'oggetto in prova generano il cosiddetto "rumore di fondo". Può essere costituito da rumore di tipo bianco all'interno del sistema di misura, da trasmissioni radio, da altri segnali continui od impulsivi anche derivati dalla rete di alimentazione.

- **Tensione di innesco delle scariche parziali ( $U_i$ )**

Tensione applicata alla quale si osservano per la prima volta nell'oggetto in prova scariche parziali ripetitive, quando la tensione applicata all'oggetto è aumentata gradualmente da un valore inferiore al quale non si osservano scariche parziali. In pratica, la tensione di innesco  $U_i$  è la minima tensione applicata alla quale l'ampiezza di una grandezza dell'impulso di scarica parziale diviene uguale o maggiore di uno specificato basso valore.

- **Tensione di estinzione delle scariche parziali ( $U_e$ )**

Tensione applicata alla quale scariche parziali ripetitive cessano di verificarsi nell'oggetto in prova, quando la tensione applicata all'oggetto è ridotta gradualmente da un valore superiore al quale si osservano scariche parziali. In pratica, la tensione di estinzione  $U_e$  è la minima tensione applicata alla quale l'ampiezza di una grandezza scelta dell'impulso di scarica parziale diviene uguale o minore di uno specificato basso valore.

### 2.5.2 Circuiti normati per la misura delle scariche parziali: metodo diretto

Tutti i metodi di rilevazione delle scariche parziali sono basati sulle grandezze che accompagnano la scarica parziale durante la sua evoluzione, dall'emissione di luce o di onde sonore alla degradazione chimica dei materiali. In realtà il fenomeno che viene sfruttato è quello legato alla nascita di transistori nel circuito di misura provocato a sua volta dalle scariche parziali.

La misura di quest'ultime poggia le sue basi sull'acquisizione dell'istante iniziale in cui ha inizio il fenomeno e sulla registrazione della diminuzione di tensione ai capi dell'oggetto in prova così da poter risalire alla carica.

Poiché l'ampiezza della carica che si intende rilevare è nell'ordine dei picocoulomb ( $pC$ ) e poiché, generalmente, si trattano oggetti con capacità dell'ordine dei nanofarad ( $nF$ ), le variazioni di tensione da misurare possono non superare le decine di microvolt ( $\mu V$ ). Durante una prova di scariche parziali oltre ai suddetti segnali, si possono manifestare dei disturbi o di tipo elettromagnetico o causati dalla presenza di armoniche nella tensione di alimentazione. Si evince di conseguenza che il primo obiettivo di un sistema di rilevamento di scariche parziali consiste nel sopprimere tali interferenze e raggiungere un'adeguata sensibilità.

La norma 60270 indica due possibili circuiti riportati in Fig. 2.9 e in Fig. 2.10, in base alla posizione circuitale dell'impedenza di misura. E' necessario definire alcune grandezze per comprendere al meglio i due circuiti:

- $U$ : alimentazione ad alta tensione
- $Z_{mi}$ : impedenza di ingresso del sistema di misura
- $CC$ : cavo di collegamento
- $C_a$ : oggetto in prova
- $C_k$ : condensatore di accoppiamento
- $CD$ : dispositivo di accoppiamento
- $MI$ : strumento di misura
- $Z$ : impedenza del filtro

Per entrambi  $Z_{mi}$  è l'impedenza di misura ai capi della quale viene valutato il segnale opportunamente amplificato e filtrato della componente a frequenza industriale e quindi inviato al blocco di misura  $MI$ .

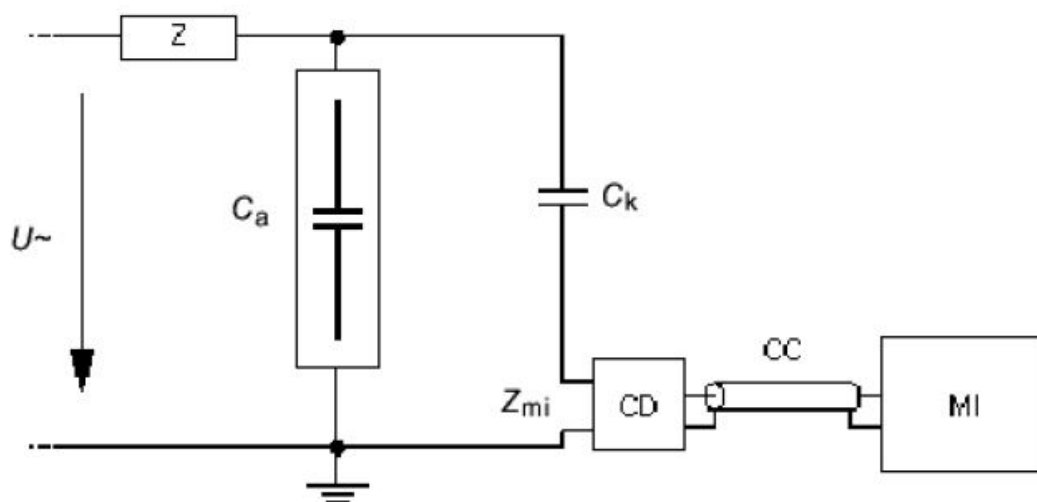


FIGURA 2.9: Circuito con impedenza di misura in serie al condensatore di accoppiamento.

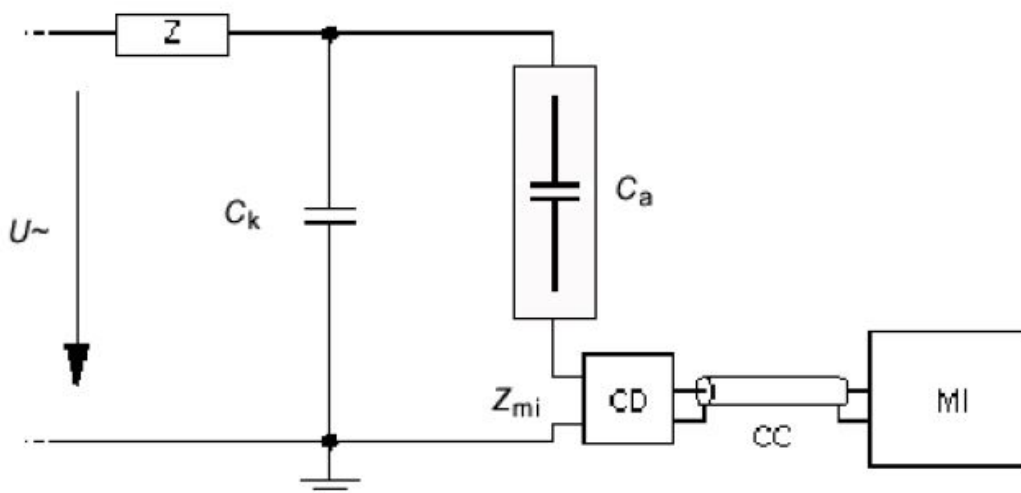


FIGURA 2.10: Circuito con impedenza di misura in serie all'oggetto in prova.

$Z$  è l'impedenza del generatore che costituisce un filtro passa basso, presenta quindi bassa impedenza a  $50\text{Hz}$  ed un'impedenza elevata alle frequenze tipiche delle scariche parziali. In pratica disaccoppia il circuito dal generatore quando si innesca la scarica nell'oggetto di prova e consente di filtrare il rumore proveniente dall'alimentazione.

Si nota la presenza di una capacità  $C_k$ , ovvero la cosiddetta capacità di accoppiamento (condensatore di accoppiamento) interessata dalle componenti a frequenza elevata che non coinvolgono il generatore. Logicamente essa deve essere esente da scariche parziali fino al valore massimo della tensione di prova.

$Z_{mi}$  può essere messa in serie a  $C_k$  o in serie al provino  $C_a$ ; la prima soluzione è adatta nel caso in cui sia prescritto che l'oggetto in prova debba funzionare con un capo a potenziale

di terra, oppure se risulta molto probabile una scarica totale dell'oggetto in prova che potrebbe in tal caso danneggiare l'impedenza di misura e gli strumenti di rilevazione posti a valle, provocando danni all'operatore.

Nella prima soluzione proposta, con  $Z_{mi}$  in serie a  $C_k$ , la corrente di scarica che attraversa il provino non interessa direttamente l'impedenza di misura  $Z_{mi}$  quindi la sensibilità di misura risulta inferiore rispetto alla precedente. Nonostante questo obiettivo punto a favore della seconda soluzione spesso è preferita la prima perché permette di agire con un maggior grado di sicurezza: si evita la possibilità che una scarica provochi una sovratensione ai capi del sistema di misura a valle, provocando danni a persone e cose.

In generale l'impedenza di misura  $Z_{mi}$  è costituita da un resistore ed un condensatore in parallelo, o da un circuito oscillante  $RLC$  parallelo.

### 2.5.3 PD Pattern

Le scariche parziali possono essere rappresentate su un grafico 3D (Fig. 2.11): l'asse x rappresenta la fase dell'alimentazione per cui avviene una scarica, l'asse y rappresenta l'ampiezza massima (anche negativa) della scarica e la terza dimensione indica la numerosità di fenomeni che è avvenuta ad una determinata coppia fase-ampiezza, quantificabile tramite una scala cromatica. Dal grafico è possibile dedurre la tipologia di difetto presente nel sistema di isolamento [9].

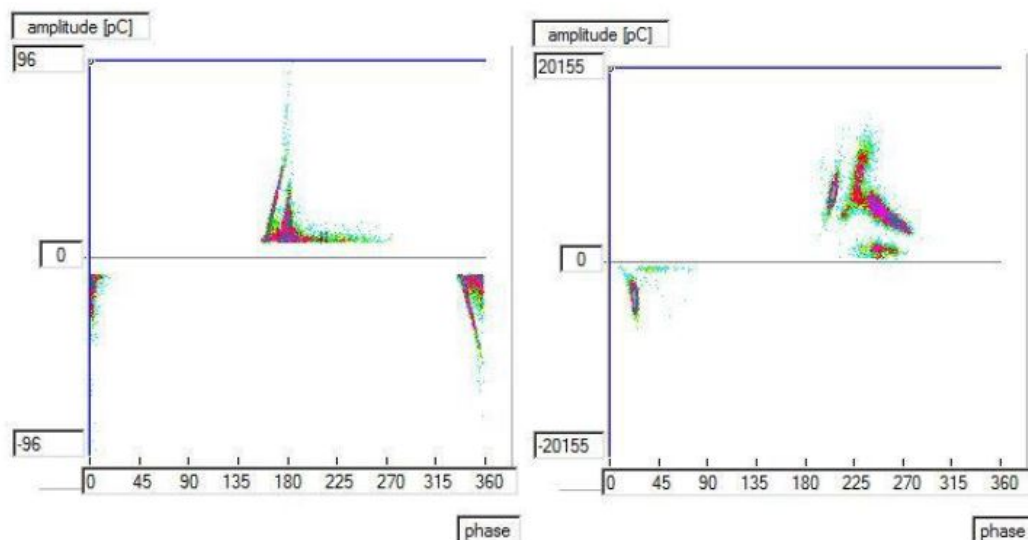


FIGURA 2.11: Esempi di due PD pattern.

Il problema fondamentale di questa analisi è che potrebbero presentarsi due difetti contemporanei sullo stesso sistema di isolamento non distinguibili direttamente sul PD pattern, poiché uno dei due prevale sull'altro presentando attività più intense, di fatto nascondendo nel grafico il secondo. Un altro problema in cui si potrebbe incorrere è la presenza di un elevato rumore ambientale che verrebbe riportato sul PD pattern, nascondendo nuovamente la presenza di difetti. Per ovviare a questi problemi sono state sviluppate nel corso degli anni e durante questa tesi di dottorato tecniche per la separazioni di difetti dalle altre sorgenti di scarica e/o dal rumore. Queste tecniche verranno descritte e approfondite nei capitoli successivi.

#### **2.5.4 Limiti nelle misure condotte secondo normativa**

La norma IEC 60270 si pone come obiettivo la misurazione della massima ampiezza della scarica parziale che si ripete, ossia l'intensità della carica apparente della scarica che si verifica periodicamente con intensità maggiore rispetto alle altre, all'interno di un periodo della tensione di alimentazione. Il limite di questa metodologia è che non esiste nessuna correlazione con la numerosità dei fenomeni che avvengono all'interno dell'isolamento, quindi la maggior parte delle cause di degradamento all'interno dei materiali isolanti non viene rilevata.

Si riporta un esempio numerico per comprendere meglio il limite della normativa: si supponga di avere due oggetti che presentano difetti simili. Si decide di fissare la soglia minima di misura della carica apparente a  $10 \text{ pC}$ . Nel primo oggetto, in un determinato intervallo di tempo, vengono misurate 100 scariche aventi intensità  $8 \text{ pC}$ . Nel secondo oggetto, nello stesso arco di tempo, vengono captate 5 scariche di intensità  $12 \text{ pC}$ . Nonostante la corrente media nel primo oggetto sia molto più elevata rispetto al secondo, per la norma il primo oggetto risulta esente da scariche parziale quando, probabilmente, l'isolamento di questo cederà prima del secondo dato che viene stressato maggiormente.

Per concludere si può affermare che la normativa non si pone come obiettivo primario il fine diagnostico: durante il dottorato di ricerca si sono presi in considerazione i limiti di questa norma, presentando tecniche di misura e analisi alternative.

## Capitolo 3

# Stato dell'arte sulla estrazione delle features e sulla classificazione delle scariche parziali

In questo capitolo verranno presentati alcuni degli algoritmi allo stato dell'arte per l'estrazione delle features dai segnali di scarica, la classificazione supervisionata e non supervisionata. Gli algoritmi presentati sono stato oggetto di studio nel percorso del dottorato di ricerca, proponendo anche nuove alternative che saranno presentate nei capitoli successivi.

### 3.1 Estrazione delle features

In questa sezione verranno descritte le tecniche di estrazione delle features utilizzate in questa tesi per ridurre la dimensionalità dei segnali di scarica e per estrarre informazioni utili a distinguere diverse tipologie di difetto e/o segnali di scarica dal rumore di fondo.

#### 3.1.1 Parametri statistici

Data una quantità di impulsi sufficiente, il PD Pattern (sottosezione 2.5.3) è una buona rappresentazione per distinguere le varie tipologie di difetto. Sul PD Pattern possono essere estratte informazioni statistiche (i momenti) in base alla forma stessa del grafico nei due cicli della tensione di alimentazione [10–12].

Si assuma, per esempio, che il PD Pattern venga diviso in  $N$  finestre sulle ascisse (la fase) e che  $x_i$  sia una feature specifica (ad esempio l'ampiezza e la carica). Allora la

media di  $x_i$  su le  $N$  finestre di fase sarà:

$$\mu = \sum_{i=1}^N \frac{x_i}{N}$$

Il momento di ordine due risulterà:

$$\sigma = \frac{\sum_{i=1}^N (x_i - \mu)^2}{N}$$

La Skewness  $\gamma$  si può calcolare come:

$$\gamma = \frac{\sum_{i=1}^N (x_i - \mu)^3}{\sigma^3 \cdot N}$$

Infine la Kurtosi  $\kappa$  si esprime come:

$$\kappa = \frac{\sum_{i=1}^N (x_i - \mu)^4}{\sigma^4 \cdot N} - 3$$

Nelle definizioni precedenti la Skewness e la Kurtosi sono calcolate rispetto alla distribuzione normale  $N(\mu; \sigma^2)$ . La Skewness è la misura di asimmetria dei dati rispetto alla distribuzione normale che ha  $\gamma = 0$ . Valori negativi per la Skewness indicano che i dati sono distorti verso sinistra, ovvero che la coda di sinistra è più lunga rispetto a quella di destra; valori positivi indicano che i dati sono distorti verso destra. La Kurtosi è un indicatore della forma della distribuzione. Se una distribuzione ha la stessa forma di una distribuzione normale, la Kurtosi è zero. Valori negativi indicano una distribuzione più piatta rispetto alla distribuzione normale, mentre valori positivi indicano una distribuzione più appuntita.

Le scariche solitamente avvengono sia nel semi-ciclo positivo sia in quello negativo della tensione di alimentazione sinusoidale: per questo motivo è solito calcolare le quantità sopracitate in entrambi i semi-cicli.

Altre quantità quali moda e mediana possono essere utilizzate alcune volte come features: la moda è l'elemento che si ripete con maggiore frequenza, mentre la mediana corrisponde all'elemento che divide a metà la distribuzione, ovvero il campione alla cui sinistra e alla cui destra si trovano un numero uguale di elementi.

Infine un altro parametro statistico molto utilizzato come feature è il parametro  $\beta$  proveniente dalla distribuzione di Weibull. In lavori precedenti [13–16] è stato mostrato che l'ampiezza di una scarica si adatta bene alla distribuzione di Weibull a due parametri.



In generale l'espressione della distribuzione cumulativa è espressa come:

$$F(x, \alpha, \beta) = 1 - \exp \left[ - \left( \frac{x}{\alpha} \right)^\beta \right] \quad (3.1)$$

Mentre la funzione densità di probabilità come:

$$f(x, \alpha, \beta) = \frac{\beta}{\alpha} \left( \frac{x}{\alpha} \right)^{\beta-1} \exp \left[ - \left( \frac{x}{\alpha} \right)^\beta \right] \quad (3.2)$$

dove  $\alpha$  e  $\beta$  sono rispettivamente i parametri di scala e di forma, mentre  $x$  è la variabile aleatoria in considerazione.

I parametri  $\alpha$  e  $\beta$  possono essere calcolati tramite una regressione lineare con i minimi quadrati, oppure con il criterio di massima verosimiglianza. Entrambi vengono applicati alla (3.1) espressa in termini logaritmici:

$$\log(-\ln(1 - F(x))) = -\beta \log \alpha + \beta \log x \quad (3.3)$$

I parametri statistici sono stati impiegati in letteratura per la trasformazione dei segnali in vettori numericamente più contenuti per permettere all'algoritmo utilizzato per la classificazione di essere più veloce. Alcuni esempi si possono trovare in [17–24].

### 3.1.2 Trasformata di Fourier

Le scariche parziali sono di natura fenomeni trattabili come serie temporali. Inoltre il contenuto in frequenza potrebbe essere usato come fattore discriminante sia per creare dei filtri opportuni per l'eliminazione del rumore sia per distinguere diverse tipologie di scariche a parità di sistema e condizioni di misura.

La trasformata di Fourier converte un segnale dal suo dominio originale (il tempo in questo caso) nella rappresentazione nel dominio delle frequenze. Nella pratica viene impiegato l'algoritmo della *Fast Fourier Transform* (FFT) che permette il calcolo delle componenti in frequenza di un segnale in un modo molto più rapido rispetto alla semplice *Discrete Fourier Transform* (DFT).

La DFT si può calcolare come:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N} \quad \text{con} \quad k = 0, \dots, N-1 \quad (3.4)$$

dove  $X_0, \dots, X_{N-1}$  sono numeri complessi. Il costo computazionale è pari a  $O(n^2)$ : ci sono  $N$  output  $X_k$  ed ogni output richiede una somma di  $N$  termini. La FFT permette di ottenere gli stessi risultati in  $O(n \log n)$  operazioni.

In [25] la FFT è stata utilizzata per trasformare i segnali temporali in un grafico a due dimensioni per permettere l'utilizzo di un algoritmo di clustering su uno spazio delle features ridotto. Nel Capitolo 4 verrà utilizzato questo mapping per la distinzione delle scariche dal rumore.

In [26] è stata calcolata la FFT sui segnali campionati nel tempo. Successivamente è stata effettuata una riduzione dello spazio delle features utilizzando la *Principal Component Analysis* (PCA), che verrà analizzata nel corso di questo capitolo. Infine è stato impiegato un algoritmo di clustering gerarchico per la classificazione finale delle scariche.

### 3.1.3 Principal Component Analysis

La PCA è una tecnica statistica per l'estrazione delle features e per la riduzione della dimensionalità [27]. Questa tecnica trasforma lo spazio dei dati in modo tale che la componente principale sia quella che abbia la varianza più grande possibile nel nuovo spazio. La componente successiva sarà quella che avrà la seconda varianza più grande con il vincolo che sia ortogonale a quella precedente e così via.

Si rappresentino i dati con  $m$  osservazioni e  $n$  features in forma matriciale  $\mathbf{X} \in \mathbb{R}^{m \times n}$ . Si assuma che  $\mathbf{X}$  sia normalizzata (ciascuna colonna con media 0), risulta dunque che  $\mathbf{C} = (1/n)\mathbf{X}^T\mathbf{X}$  sia la covarianza della matrice  $\mathbf{X}$ : quindi  $\mathbf{C} \in \mathbb{R}^{n \times n}$ . Successivamente si calcolino gli autovalori e gli autovettori per la matrice  $\mathbf{C}$  in base alla decomposizione degli autovalori. Per esempio,  $\lambda$  è l'autovalore e  $\boldsymbol{\nu} \in \mathbb{R}^{n \times 1}$  è l'autovettore corrispondente se  $\lambda\boldsymbol{\nu} = \mathbf{C}\boldsymbol{\nu}$ . Si selezionino i primi  $p$  autovalori (i più grandi) e i rispettivi autovettori: si chiami  $\mathbf{V} \in \mathbb{R}^{n \times p}$  la matrice costituita dai  $p$  autovettori ( $p \leq n$ ). Infine, i dati dallo spazio  $n$ -dimensionale  $\mathbf{X}$  sono mappati in uno spazio  $p$ -dimensionale  $\mathbf{X}_{pca}$  mediante la relazione  $\mathbf{X}_{pca} = \mathbf{X}\mathbf{V}$ .

La PCA è stata applicata principalmente a due tipologie di rappresentazione delle scariche:

#### 1) PCA applicata a segnali nel tempo

Nel primo esempio [28] ciascun impulso di scarica è rappresentato come un vettore a 2000 dimensioni. La PCA viene applicata per ridurre le features da 2000 a 2 per permettere una visualizzazione semplice e per classificare le scariche. In [26] ciascun segnale è rappresentato come un vettore a 5000 dimensioni. Successivamente il

vettore viene ridotto da 5000 a 2500 dimensioni trasformando i segnali dal dominio del tempo al dominio delle frequenze con la FFT (la riduzione è dovuta al criterio di Nyquist). La PCA viene infine utilizzata per ridurre ulteriormente le features da 2500 a 6 per permettere un *clustering* computazionalmente meno oneroso. In [29] la decomposizione mediante *wavelet* genera 6 coefficienti legati all'energia di ciascun livello di decomposizione. La PCA viene utilizzata per ridurre ulteriormente la dimensionalità da 6 a 3 per la fase di clustering e per la visualizzazione.

## 2) PCA applicata al PD Pattern

In [18] il vettore delle features del PD pattern ha 600 dimensioni (200 finestre di fase con 3 features in ogni finestra). La PCA riduce la dimensionalità da 600 a 7 per la fase di classificazione. In [17, 30] ciascun PD pattern ha un vettore delle features con 1440 dimensioni (360 finestre di fase con 4 features ciascuna). La PCA è nuovamente utilizzata per ridurre le dimensioni al fine di classificare le sorgenti di scarica.

### 3.1.4 t-SNE

L'algoritmo del t-SNE (*t-distributed Stochastic Neighbor Embedding*) è un'altra tecnica di riduzione delle dimensionalità sviluppata per la visualizzazione di dati ad alta dimensionalità [31]. Lo scopo del t-SNE è quello di proiettare i dati in uno spazio a bassa dimensionalità cercando di conservare la distanza reciproca dei punti dello spazio originale. Per questo scopo il t-SNE costruisce la distribuzione di probabilità  $\mathcal{P}$  sopra una coppia di oggetti ad alta dimensionalità in modo tale che due oggetti simili hanno un valore di probabilità alto, mentre punti dissimili hanno un valore di probabilità basso. Sia  $p_{ij}$  la probabilità per la coppia  $(x_i, x_j)$  nello spazio originale. Allo stesso modo il t-SNE definisce una distribuzione di probabilità  $\mathcal{Q}$  sui punti appartenenti allo spazio a bassa dimensionalità, e sia  $q_{ij}$  la probabilità per la coppia  $(y_i, y_j)$  corrispondente nello spazio a bassa dimensionalità. Supponendo che le due distribuzioni di probabilità  $\mathcal{P}$  e  $\mathcal{Q}$  siano simili, due punti  $x_i$  e  $x_j$  vicini nello spazio ad alta dimensionalità saranno similmente vicini nello spazio a bassa dimensionalità.

Quindi il t-SNE usa la divergenza di Kullback-Leibler (KL) in (3.5) per misurare la distanza tra le due distribuzioni  $\mathcal{P}$  e  $\mathcal{Q}$ . Minimizzando la divergenza KL tra  $\mathcal{P}$  e  $\mathcal{Q}$ , il t-SNE riduce le dimensionalità del dato originale mantenendo la distanza reciproca tra i punti.

$$KL(\mathcal{P}||\mathcal{Q}) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}} \quad (3.5)$$

In [32, 33] e in [34] viene impiegata la decomposizione con le funzioni wavelet applicata direttamente alle scariche nel tempo. Per ciascun livello di filtraggio viene calcolato il

coefficiente di energia. Infine viene utilizzato il t-SNE per ridurre ulteriormente lo spazio delle features a 3 dimensioni per la visualizzazione e il clustering dei difetti. In [18] il vettore delle features per ciascun PD pattern ha 600 dimensioni, e il t-SNE riduce le dimensionalità da 600 a 7 per la classificazione delle scariche.

### 3.1.5 Sommario sulle tecniche di estrazione delle features

Le features statistiche conservano un significato fisico quando vengono estratte, infatti sono legate alla forma dell'impulso o alla forma del PD pattern e alle distribuzioni delle quantità rappresentate. Anche la trasformata di Fourier ha un significato fisico in quanto mostra il contenuto in frequenza delle scariche campionate.

Invece la PCA e il t-SNE sono algoritmi di riduzione della numerosità delle features che combinano le features dello spazio originario rendendo così un'interpretazione fisica difficile.

Ciascuna di queste tecniche presenta pregi e difetti anche in funzione dell'algoritmo di classificazione utilizzato a posteriori per la distinzione delle diverse tipologie di scariche.

## 3.2 Algoritmi per la classificazione non supervisionata

In questa sezione verranno presentate alcune tecniche di classificazione non supervisionata per la separazione delle diverse tipologie di difetto. L'ipotesi principale è che impulsi provenienti dalla stessa sorgente abbiano caratteristiche simili.

Gli algoritmi di clustering, utili a questo scopo, dividono gli oggetti in gruppi tramite una metrica che misura la similarità o la dissimilarità tra gli oggetti stessi. La metrica è solitamente concepita in termini di distanza in uno spazio multidimensionale. Di conseguenza l'appartenenza o meno di un elemento ad un certo insieme dipende da quanto l'elemento considerato è distante dall'insieme stesso. Questi algoritmi sono ampiamente utilizzati in letteratura per separare sorgenti multiple di difetti.

### 3.2.1 K-means

L'algoritmo del K-means è basato sulla definizione dei centroidi e sulla distanza reciproca tra i punti e i centroidi stessi. È una tecnica iterativa basata su diversi step:

- **Primo step:**  $K$  centroidi iniziali sono scelti random nello spazio delle features.

- **Secondo step:** vengono formati  $K$  cluster, assegnando ciascun oggetto al centroide più vicino.
- **Terzo step:** viene ricalcolato il centroide di ciascun cluster.

L'algoritmo procede iterativamente eseguendo il secondo e terzo step finché converge. Il processo è un algoritmo euristico che ottimizza la funzione:

$$\arg \min_{\mathbf{C}} \sum_{i=1}^K \sum_{\mathbf{x}_j \in \mathbf{C}_i} \|\mathbf{x}_j - \mathbf{o}_i\|^2 \quad (3.6)$$

dove  $\mathbf{x}_j$  è l'elemento preso in considerazione,  $\mathbf{C} = \{\mathbf{C}_1, \dots, \mathbf{C}_k\}$  è l'insieme dei  $K$  cluster e  $\mathbf{o}_i$  è il centroide del cluster  $\mathbf{C}_i$ .

In [21, 35–38] il K-means viene utilizzato come algoritmo di classificazione su spazi delle features differenti. Ad esempio, in [21], i segnali delle scariche vengono rappresentati da un vettore delle features contenente i momenti statistici relativi al PD pattern. In [37] gli autori utilizzano la funzione di autocorrelazione normalizzata per rappresentare le scariche (la funzione di autocorrelazione sintetizza features sia nel tempo che nel dominio delle frequenze). Inoltre la similarità tra due oggetti è misurata con il coefficiente di correlazione di Pearson calcolata sulle funzioni di autocorrelazione normalizzate. In [36] gli autori estraggono le features in base alla forma dell'impulso (tempo di salita e di discesa, massima ampiezza, etc.) e poi utilizzano il K-means per generare i cluster. Infine raggruppano i cluster in base alla distribuzione delle frequenze dei vari impulsi per ottenere il risultato finale. In [35] vengono estratte 9 features dai segnali catturati con un sensore UHF in base all'ampiezza e alla distanza temporale tra due impulsi consecutivi. In [38] le scariche vengono rappresentate mediante l'ampiezza e l'angolo di fase, e vengono applicate regole per scegliere la tipologia di difetto di ciascun cluster in base alla forma dell'impulso.

Il K-means è un algoritmo semplice e molto utilizzato ma mostra alcune limiti. Innanzitutto la soluzione trovata è un minimo locale, data la natura randomica della scelta iniziale dei centroidi. Inoltre è opportuno specificare il numero dei cluster  $K$ . Nel Capitolo 7 verranno mostrati alcuni accorgimenti per ovviare a questi problemi.

### 3.2.2 DBSCAN

Il DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) è un algoritmo di clustering basato sulla densità spaziale dei punti [39]. L'algoritmo ha due parametri in ingresso: il raggio  $\epsilon$  e il numero minimo di punti  $N_{min}$ . Dato un punto  $P$  e una metrica per la distanza  $d$  allora  $N(P) = \{Q | d(P, Q) \leq \epsilon\}$  è l'insieme dei punti vicini a

$P$ , e si affermi che  $P$  e tutti i punti contenuti in  $N(P)$  siano connessi. Se due punti differenti  $X$  e  $Y$  sono connessi ad un terzo punto  $Z$ , allora anche  $X$  e  $Y$  sono connessi. Se  $size(N(P)) \geq N_{min}$  allora DBSCAN genera un cluster da  $P$  includendo tutti i punti che sono connessi a  $P$ . Una volta che un cluster è stato identificato, il DBSCAN seleziona un nuovo punto  $P'$  che non appartiene a nessun cluster e ripete la procedura precedente.

L'algoritmo del DBSCAN è stato impiegato per separare diverse sorgenti di scariche [29, 32, 34, 40–42]. In [29] i segnali nel tempo sono stati filtrati con le funzioni wavelet (5 livelli di filtraggio quindi 6 coefficienti calcolati come l'energia dei segnali filtrati). Successivamente è stata utilizzata una PCA per ridurre la dimensionalità da 6 a 3. Infine i segnali sono stati raggruppati con il DBSCAN e visualizzati in uno spazio 3D. In [32], similmente a prima, i segnali sono stati filtrati con le funzioni wavelet ma in 9 livelli. Successivamente gli autori hanno utilizzato il t-SNE per ridurre le dimensioni da 10 a 3 e il DBSCAN per classificare le diverse tipologie di difetto. In [40] la funzione di autocorrelazione normalizzata è stata calcolata per ogni segnale. Successivamente la riduzione di spazio viene effettuata prendendo i 3 picchi principali della varianza calcolata su tutte le funzioni di autocorrelazione. Infine il DBSCAN viene utilizzato per separare le diverse sorgenti di scariche. In [34] viene applicato il t-SNE ai coefficienti di energia estratti dal filtraggio effettuato con le funzioni wavelet. Nello spazio 3-D risultante viene applicato il DBSCAN per classificare le scariche e i risultati vengono comparati con altri algoritmi di clustering. In [41, 42] le features vengono estratte dalla funzione cumulativa di energia calcolata sia nel dominio del tempo che delle frequenze. Nel nuovo spazio delle features viene utilizzato l'algoritmo del DBSCAN per distinguere le diverse fonti di rumore.

Rispetto al K-means, il DBSCAN ha due vantaggi: innanzitutto non è necessario specificare il numero di cluster, in secondo luogo può creare cluster di diversa forma (non separabili linearmente) che non possono essere trovati con il K-means.

### 3.2.3 OPTICS

L'algoritmo di clustering OPTICS (*Ordering Points To Identify the Clustering Structure*) forma cluster basandosi sulla densità spaziale dei punti [43]. L'idea di base è molto simile al DBSCAN, ma cerca di risolvere uno dei problemi più grandi legati a questo: creare cluster nei dati con densità che varia. Per fare ciò i dati vengono ordinati in modo tale che i punti più vicini nello spazio siano vicini anche nel vettore ordinato.

Come il DBSCAN anche OPTICS riceve in ingresso due parametri: il raggio  $\epsilon$  e il numero minimo di punti  $MinPts$ . Si affermi che un punto  $p$  sia un *corepoint* se almeno  $MinPts$  punti giacciono nel suo intorno  $\epsilon N_\epsilon(p)$ . Contrariamente al DBSCAN, OPTICS

considera punti che fanno parte di un cluster occupato più densamente, così a ciascun punto è assegnata una *core distance* che descrive la distanza al *MinPts*-esimo punto più vicino. Si definisca la *core distance* come:

$$core-dist_{\epsilon, MinPts} = \begin{cases} \text{NON DEFINITA} & \text{se } |N_{\epsilon}(p)| < MinPts \\ MinPts - \text{esima distanza più piccola a } N_{\epsilon}(p) & \text{altrimenti} \end{cases} \quad (3.7)$$

La *reachability distance* di un altro punto  $o$  da un punto  $p$  è la quantità maggiore tra la distanza tra  $o$  e  $p$  o la *core distance* di  $p$ :

$$reach-dist_{\epsilon, MinPts}(o, p) = \begin{cases} \text{NON DEFINITA} & \text{se } |N_{\epsilon}(p)| < MinPts \\ \max(core-dist_{\epsilon, MinPts}(p), \text{dist}(p, o)) & \text{altrimenti} \end{cases} \quad (3.8)$$

Sia la *core distance* che la *reachability distance* non sono definite se i punti non sono sufficientemente densi nello spazio. Se  $\epsilon$  è sufficientemente largo, questo non succede mai, però tutti i punti dello spazio giaceranno nello stesso cluster e la richiesta del calcolo di  $N_{\epsilon}(p)$  restituirà sempre tutto il dataset, avendo dunque una complessità computazionale di  $O(n^2)$ . Quindi il parametro  $\epsilon$  è necessario per eliminare i punti che non sono densamente concentrati e per aumentare le prestazioni computazionali dell'algorithm.

Mostrando su un grafico bidimensionale chiamato *reachability plot* le *reachability distances* in funzione dei punti ordinati è possibile individuare delle valli che corrispondono ai cluster formati. Più profonde sono queste valli più è denso il cluster a cui appartengono i punti. I cluster possono essere estratti mettendo una soglia sull'asse  $y$  del grafico, oppure utilizzando algoritmi per la ricerca di punti di ginocchio o calcolando il gradiente di salita e di discesa. In questo modo, in realtà, OPTICS non necessita del parametro  $\epsilon$  in ingresso poiché crea l'insieme delle distanze ordinate rispetto ai punti vicini utilizzando soltanto il parametro *MinPts*. Quindi, in un secondo momento, si può definire una soglia per la creazione dei cluster, avendo già a disposizione il grafico delle *reachability distances* ordinate.

L'algorithm OPTICS è stato utilizzato solamente negli ultimi anni per distinguere le diverse sorgenti di scarica: due esempi sono riportati in [34, 44]. In [34], come detto precedentemente, i dati sono stati proiettati in uno spazio 3D con l'algorithm del t-SNE, per poi essere raggruppati con il DBSCAN e con OPTICS (il secondo mostra risultati migliori). In [44] gli autori utilizzano diverse strategie per l'estrazione delle features e impiegano OPTICS per distinguere le diverse sorgenti di scarica valutando quale sia lo spazio delle features migliore per classificare correttamente le diverse tipologie di scariche.

### 3.3 Algoritmi per la classificazione supervisionata

In questa sezione verranno presentate alcune tecniche di classificazione supervisionata per la separazione delle diverse tipologie di difetto.

#### 3.3.1 Support Vector Machine (SVM)

La SVM è un algoritmo allo stato dell'arte per la classificazione supervisionata [45]. In Fig. 3.1,  $y = \mathbf{w} \cdot \mathbf{x} + b$  è un iperpiano che può essere usato per la classificazione, mentre  $2/\|\mathbf{w}\|$  è il margine per questo iperpiano.

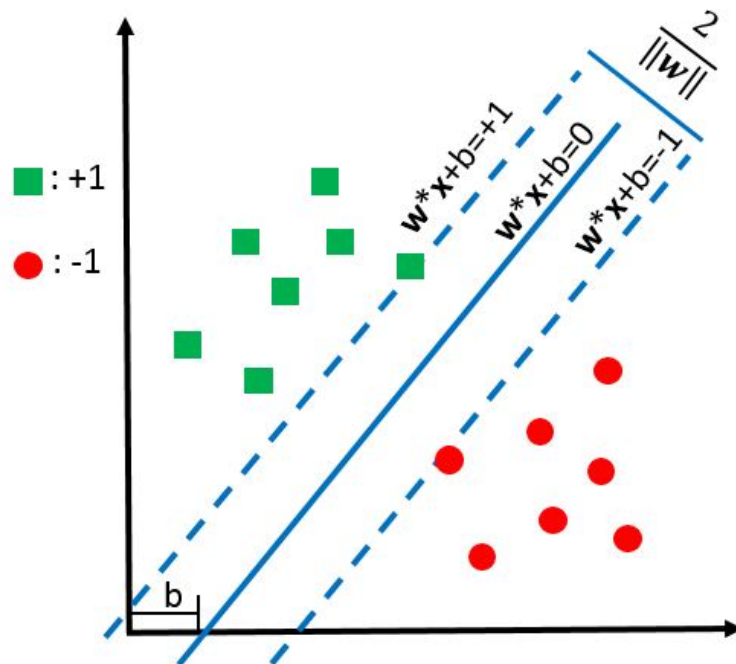


FIGURA 3.1: Esempio di classificatore creato con SVM in uno spazio bidimensionale.

Il principio base della SVM è quello di trovare l'iperpiano a massimo margine che separi i dati, risolvendo il problema di ottimizzazione convesso, composto da due termini, espresso nella (3.9).

$$\mathbf{w}^* = \arg \min_{\mathbf{w} \in \mathbb{R}^n, b} \left[ \overbrace{\lambda \|\mathbf{w}\|^2}^{1^\circ \text{ termine}} + \overbrace{\frac{1}{n} \sum_{i=1}^m \max\{0, 1 - y_i \cdot (\mathbf{w} \cdot \mathbf{x}_i - b)\}}^{2^\circ \text{ termine}} \right] \quad (3.9)$$

Il primo termine  $\|\mathbf{w}\|^2$  indica l'ampiezza del margine, mentre il secondo è un termine di penalità che tiene conto dei dati non classificati correttamente ( $y_i$  è la vera etichetta dell'istanza  $i$ -esima e  $\mathbf{w} \cdot \mathbf{x}_i - b$  è l'etichetta della predizione). Il parametro  $\lambda$  è il termine



di regolarizzazione che tiene conto del trade-off tra l'ampiezza del margine e che i dati giacciono dal lato corretto del margine. Il problema di ottimizzazione può essere risolto usando la programmazione quadratica.

La Fig. 3.1 mostra un caso lineare per la SVM. Per trovare iperpiani non lineari, possono essere applicate diverse funzioni di kernel per mappare lo spazio originario degli input in uno spazio a maggiore dimensionalità dove possono essere utilizzati iperpiani per la classificazione [45]. Affinché si possano utilizzare le funzioni kernel è necessario riformulare il problema (3.9) nella sua formulazione duale, passando per quella primale.

Per ogni  $i = 1, \dots, n$  si introduca una variabile  $\zeta_i = \max(0, 1 - y_i \cdot (\mathbf{w} \cdot \mathbf{x}_i - b))$ . Si osservi che  $\zeta_i$  è il più piccolo numero non negativo che soddisfa  $y_i \cdot (\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1 - \zeta_i$ .

Allora si può riscrivere il problema (3.9) nella sua formulazione primale:

$$\begin{aligned} \min_{\mathbf{w}, b} \quad & \lambda \|\mathbf{w}\|^2 + \frac{1}{n} \sum_{i=1}^n \zeta_i \\ \text{soggetto a} \quad & y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1 - \zeta_i \text{ e } \zeta_i \geq 0, \forall i \end{aligned} \quad (3.10)$$

Includendo i vincoli nella formulazione (3.10), impiegando i moltiplicatori di Lagrange e utilizzando le condizioni di Karush-Kuhn-Tucker è possibile riformulare il problema primale nel suo duale:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i \alpha_i (\mathbf{x}_i \cdot \mathbf{x}_j) y_j \alpha_j \\ \text{soggetto a} \quad & \sum_{i=1}^n \alpha_i y_i = 0 \text{ e } 0 \leq \alpha_i \leq \frac{1}{2n\lambda}, \forall i \end{aligned} \quad (3.11)$$

Gli  $\alpha_i \neq 0$  sono chiamati vettori di supporto.

A questo punto è possibile definire la funzione kernel come:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) \quad (3.12)$$

che può essere facilmente sostituita nella (3.13) ottenendo il problema di ottimizzazione:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i \alpha_i \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) y_j \alpha_j \\ & = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i \alpha_i k(\mathbf{x}_i, \mathbf{x}_j) y_j \alpha_j \\ \text{soggetto a} \quad & \sum_{i=1}^n \alpha_i y_i = 0 \text{ e } 0 \leq \alpha_i \leq \frac{1}{2n\lambda}, \forall i \end{aligned} \quad (3.13)$$

Le principali funzioni kernel sono sintetizzate in tabella 3.1.

In [23], SVM è stata utilizzata per identificare il luogo dei difetti del sistema di isolamento basandosi su parametri provenienti dalle scariche parziali. Il problema è stato

TABELLA 3.1: Funzioni kernel

Funzione kernel	$k(\mathbf{x}_i, \mathbf{x}_j)$
<i>Polynomial</i>	$= (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$
<i>Gaussian Radial Basis Function (RBF)</i>	$= \exp(-\gamma \ \mathbf{x}_i - \mathbf{x}_j\ ^2)$
<i>Sigmoid</i>	$= \tanh(\alpha \mathbf{x}_i^T \mathbf{x}_j + c)$

semplificato indicando se il difetto è vicino ad un elettrodo ad alta tensione o ad uno a bassa, per cui risulta un problema di classificazione binaria. In [46], particelle contaminanti nei trasformatori in olio sono separate in 4 classi in base alla loro lunghezza, raggio e tipo di metallo. Le features sono state collezionate dai segnali di scarica generata dalla collisione delle particelle con l'elettrodo. La SVM è stata utilizzata per classificare le particelle.

In [17, 19, 20, 47] SVM è stata utilizzata per distinguere diverse fonti di scarica. Per esempio in [17] sono state estratti due insiemi di features dal PD pattern dei segnali. Su ciascun insieme SVM ha ottenuto un'accuratezza migliore nel classificare le scariche rispetto alle reti neurali e alle *self-organizing map*. In [47], la SVM ha ricevuto in ingresso features provenienti dalla densità di potenza spettrale delle scariche e del rumore. L'algoritmo è stato testato con diverse funzioni kernel per il mapping dei dati.

Alcune varianti della SVM, ad esempio *least square SVM* [48] e la *fuzzy-SVM* [18] sono usate per la classificazione delle scariche. Per la LS-SVM,  $(y_i - \mathbf{w} \cdot \mathbf{x}_i - b)^2$  è stato utilizzato come termine di penalità nella (3.9). Per la fuzzy-SVM, ciascun dato presenta un termine fuzzy in modo tale che ciascun punto di input possa dare un contributo diverso nel calcolo dell'iperpiano di separazione.

### 3.3.2 Reti neurali

Le reti neurali sono modelli di apprendimento ispirati al sistema nervoso centrale degli animali. Solitamente una rete neurale consiste in uno strato di input, uno strato di output e uno o più strati nascosti. La Fig. 3.2 mostra un esempio di rete neurale con un solo strato nascosto. I nodi, chiamati neuroni, connettono tutti quelli dello strato precedente a quelli dello strato successivo con pesi differenti. Inoltre, ciascun neurone nello strato di input è una feature di input mentre ogni neurone dello strato di output è una classe predetta. Un neurone calcola la somma pesata dei segnali in ingresso e possiede una funzione di attivazione non lineare (simile alle funzioni kernel) per determinare se è attivo oppure no.

Durante la fase di training, generalmente si impara un mapping non lineare tra l'input e l'output, che è determinato dai pesi dei collegamenti e i neuroni della rete. In particolare

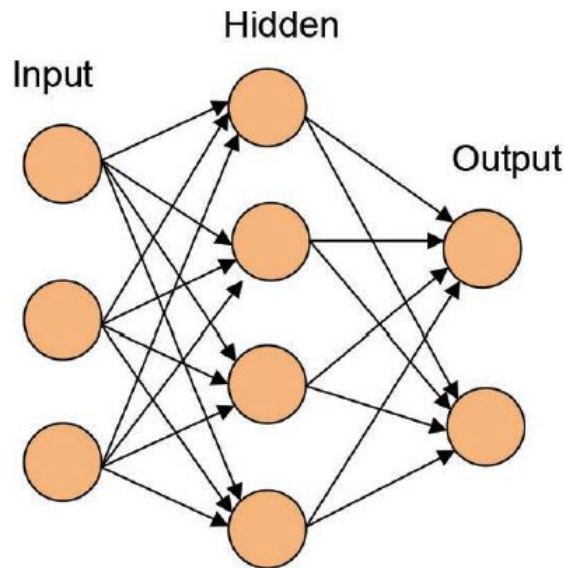


FIGURA 3.2: Esempio di rete neurale a singolo strato nascosto.

la *back-propagation* è un algoritmo molto comune sviluppato per imparare questi pesi. Quelli imparati durante la fase di training verranno utilizzati per la classificazione di nuovi dati.

In [49] gli autori hanno effettuato un'analisi delle tecniche basate su reti neurali per la classificazione delle scariche prima del 2013. Per esempio in [11, 12] utilizzano le reti neurali con le features statistiche estratte dai segnali di scarica. Successivamente diverse tecniche basate sulle reti neurali sono state utilizzate per l'analisi dei dati provenienti dalle scariche parziali [17, 18, 50–55].

In [52], i dati di ingresso della rete neurale sono le percentuali di gas generate dalla DGA (*Dissolved and free Gas Analysis*), mostrando risultati migliori rispetto al metodo utilizzato nella norma di riferimento. In [54] gli autori hanno collezionato 8 tipi di scariche parziali che si verificano nei sistemi di isolamento in carta-olio alterati da processi d'invecchiamento. Successivamente hanno estratto features dal dominio del tempo e delle frequenze dei segnali di scarica. Una rete neurale è stata utilizzata per classificare diverse tipologie di difetto. In [50], le funzioni wavelet e i criteri statistici sono stati utilizzati per generare le features. Una rete neurale è stata utilizzata per separare le scariche corona da altri tipi di scariche. In [51] sono stati acquisiti segnali provenienti da 4 tipologie di difetti. Ogni segnale è stato filtrato con le funzioni wavelet. Ad ogni livello di filtraggio sono stati estratti i parametri statistici fino al quarto ordine. Infine è stata utilizzata una rete neurale probabilistica per classificare le varie tipologie di difetto. In [55] sono estratte le features statistiche dai PD pattern. Successivamente diverse tipologie di difetto sono classificate mediante un *ensemble* di reti neurali, utilizzando solamente una parte dei dati di training in ciascuna rete.

In [56] è stata realizzata un'analisi dei progressi effettuati fino al 2016 con le ANN (*Artificial Neural Network*) per il riconoscimento delle scariche parziali.

### **3.4 Sommario delle tecniche di classificazione**

Nelle Sezioni 3.2 e 3.3 sono state presentate alcune tecniche di apprendimento automatico sia supervisionate che non. Quelle mostrate in questo capitolo sono le più conosciute in letteratura. Ovviamente diversi ricercatori hanno effettuato modifiche più o meno sostanziose agli algoritmi di base per raggiungere performance migliori sia dal punto di vista computazionale sia dal punto di vista dell'accuratezza dei risultati.

Nel Capitolo 5 verrà presentata la teoria di una rete neurale a singolo strato che sfrutta la randomicità per il mapping dei dati. Inoltre nel Capitolo 6 l'algoritmo verrà adattato per essere realizzato in un dispositivo hardware, mostrando un'applicazione nel riconoscimento di diverse tipologie di scariche parziali. Nel Capitolo 7 verranno utilizzate e confrontate diverse tecniche allo stato dell'arte per la separazione non supervisionata dei difetti.

## Capitolo 4

# Misura delle scariche e separazione dal rumore

In questo capitolo verranno descritti i sistemi di misura per le scariche di tipo condotto, quali impedenze RLC e sensori induttivi, e sistemi di misura per le scariche di tipo irradiato, mediante l'impiego di antenne.

Inoltre verranno fatte considerazioni sulla quantizzazione delle scariche in fase di misura, mostrando le differenze in fase di separazione tra il rumore di fondo e il segnali di scarica tra un sistema a 8 bit e uno a 12 bit. La separazione è stata effettuata utilizzando un semplice algoritmo di clustering.

### 4.1 Sistemi per la misura condotta delle scariche

I sistemi per la misura di scariche condotte possono essere divisi principalmente in due categorie:

#### 1) Impedenze RLC

Si consideri il circuito di Fig. 2.9 nel Capitolo 2. La  $Z_{mi}$ , ovvero l'impedenza di misura, potrebbe essere costituita da una pura resistenza  $R$ , da un circuito  $RC$  o da un circuito  $RLC$ . La scelta è influenzata dal tipo di misura che si sta effettuando: se si scegliesse una pura resistenza sarebbe possibile realizzare misurazioni in alta frequenza avendo così un dettaglio maggiore sul fronte di salita dell'impulso. La controindicazione è che si avrebbe un'elevata corrente di ricircolo nella maglia  $C_a - C_k - Z_{mi}$  del circuito di misura, perciò l'oscillazione dovuta ad un impulso potrebbe durare centinaia di  $ns$  rischiando di sovrapporsi ad una scarica successiva.

Nel caso in cui si scegliesse un'impedenza di tipo  $RC$  l'impulso verrebbe integrato: il fronte di salita avrebbe una durata superiore rispetto a quello captato con un semplice resistenza (nel dominio delle frequenze l'impulso presenterebbe componenti in più bassa frequenza rispetto a segnali misurati con la resistenza). In funzione della costante di tempo del circuito  $RC$  si potrebbe limitare il fenomeno di oscillazione dovuto alla corrente di ricircolo, ma si potrebbe abbassare troppo l'ampiezza del segnale misurato. Inoltre il circuito  $RC$  ha un effetto passa-basso sui segnali misurati, per cui la tensione di alimentazione deve essere opportunamente filtrata poiché fonte di errore nella misura.

Con un'impedenza  $RLC$  si hanno fenomeni di oscillazione nel circuito di misura, ma si introduce un effetto passa-alto sulla tensione di alimentazione eliminando i disturbi in bassa frequenza.

In ogni caso capacità e induttanze parassite introducono fenomeni di ricircolo di corrente e dunque di oscillazioni e smorzamenti nei segnali captati. Solitamente si cerca di attenuare questi fenomeni con dei filtraggi opportuni, studiando la risposta del sistema quando sottoposto ad impulsi noti (fase di calibrazione). Inoltre il filtraggio è necessario per eliminare eventuali fonti di rumore esterne.

## 2) Sensori induttivi

I principali sensori induttivi utilizzati per la misura delle scariche parziali sono gli *High Frequency Current Transformers* (HFCT), chiamati anche "ferriti". Un esempio viene riportato in Fig. 4.1.



FIGURA 4.1: Sensore induttivo di tipo  $HFCT$ .

La banda passante di questi sensori, che può arrivare fino a qualche centinaio di  $MHz$ , è in funzione della ferrite utilizzata per la misura e dal numero di spire. La risposta di un HFCT è di tipo oscillatorio e nel caso in cui si impiegasse per la misurazione di scariche parziali essa può esaurirsi anche nell'ordine dei  $\mu s$ . Per questo motivo si rende necessaria una fase di filtraggio per limitare la risposta

oscillatoria del sensore, poiché una durata eccessiva di un singolo segnale potrebbe non permettere la captazione di altre scariche.

La scelta per l'utilizzo tra un'impedenza di misura e una ferrite dipende principalmente dalla disponibilità di disalimentare il macchinario che si vuole diagnosticare per poter applicare un'impedenza di misura opportuna per poi alimentare il macchinario separatamente. Se non è possibile disalimentare la macchina dalla rete allora si adoperano le ferriti di più facile installazione, in quanto possono essere posizionate intorno ad un cavo di terra senza interferire con il normale funzionamento del macchinario, quindi senza necessitare il fermo macchina.

Il sistema di misura condotto in generale può essere schematizzato come in Fig. 4.2:

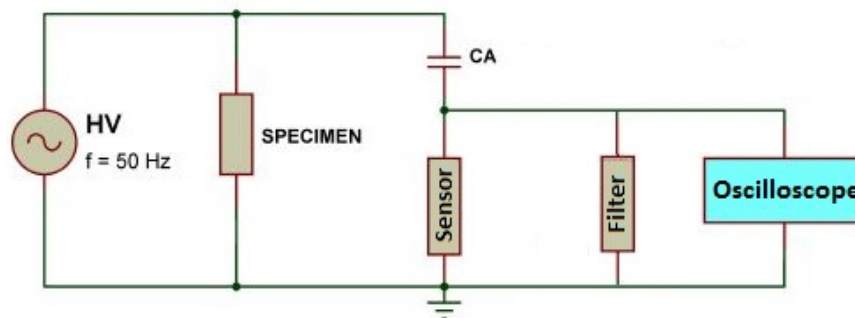


FIGURA 4.2: Circuito di misura condotto.

dove  $HV$  è l'alimentazione sinusoidale, ad alta tensione, tipicamente a  $50Hz$ ,  $SPECIMEN$  è il sistema isolante che si sta controllando,  $CA$  è il condensatore di accoppiamento,  $Sensor$  è il sensore che si sta utilizzando per la misura (impedenza o HFCT),  $Filter$  è il filtro per rimuovere disturbi e per smorzare le oscillazioni, infine  $Oscilloscope$  è l'oscilloscopio che permette di campionare e di salvare i segnali misurati per una successiva fase di analisi.

## 4.2 Sistemi per la misura irradiata delle scariche

Un sistema per la misura delle scariche mediante antenne nasce dall'esigenza di effettuare una diagnostica a basso costo e soprattutto in modo non invasivo. Inoltre il sistema di misura risulterebbe molto compatto, quindi adatto per essere posizionato in ambienti ostili quali turbine eoliche.

La misura tramite antenna è sostanzialmente analoga a quella di tipo condotto: infatti quando si verifica una scarica parziale, oltre all'impulso di corrente che scorre all'interno dei conduttori si ha anche una parte di segnale irradiata nell'aria. La limitazione

principale dell'utilizzo di questi sensori è la perdita dell'informazione sull'ampiezza della scarica, in quanto l'ampiezza del segnale irradiato dipende dal punto in cui la scarica avviene, dal materiale interposto e dai fenomeni di riflessione.

L'utilizzo di un'antenna esterna è utile quando si vuole verificare la presenza o meno delle scariche parziali e osservare l'evoluzione del fenomeno nel corso del tempo. Per un'analisi qualitativa è opportuno eseguire misurazioni condotte sul macchinario.

Durante questa tesi di dottorato è stata impiegata l'antenna SolBox<sup>®</sup>, progettata dall'università degli studi di Genova e ampiamente descritta in [57], per la realizzazione di un prototipo di un sistema di acquisizione remota su turbine eoliche. Questo sistema verrà descritto più approfonditamente nel Capitolo 8.

#### 4.2.1 Antenna SolBox<sup>®</sup>

L'antenna SolBox<sup>®</sup> riportata in Fig. 4.3 è un'antenna ad ampia banda di tipo planare che può essere realizzata mediante un foglio di metallo o, come nel caso studiato, mediante un comune circuito stampato. Queste caratteristiche la rendono non solo molto compatta, ma permettono anche la sua realizzazione direttamente sullo stesso circuito stampato insieme agli stadi di filtraggio e amplificazione.



FIGURA 4.3: Antenna SolBox<sup>®</sup>.

L'antenna ha una banda passante tra  $600 \div 1200 MHz$  ed, essendo molto direttiva, capta i segnali nella direzione perpendicolare ad essa. Inoltre è pensata per essere montata con



uno schermo sul lato posteriore, e deve essere incastolata in un contenitore metallico, in modo tale da ridurre il più possibile il rumore ambientale. Per la captazione dei segnali di interesse il primo aspetto da tenere in considerazione è quello di isolare i segnali desiderati dal rumore, per poterli caratterizzare e successivamente studiare.

Sono stati condotti degli esperimenti in laboratorio per verificare il funzionamento dell'antenna, confrontando i segnali captati con il sistema di misura condotto.

#### 4.2.2 Provini e sistema di alimentazione

Le scariche parziali sono state generate utilizzando dei provini *twisted pair* (Fig. 4.4), realizzati secondo la norma CEI EN 60851-5 [58] e un'alimentazione *Pulse Width Modulation* (PWM).

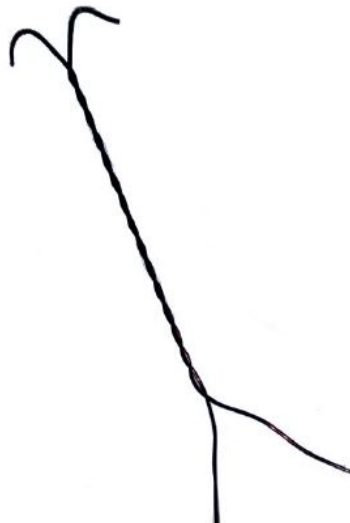


FIGURA 4.4: Provino twisted pair.

La tensione di alimentazione utilizzata per stressare i provini emula la tensione di un inverter bipolare a due livelli; inoltre sono presenti gli overshoot dovuti alle connessioni presenti tra inverter e utilizzatore. La forma d'onda è stata generata con il generatore di funzioni Tektronix AWG2005, il cui segnale viene amplificato mediante AWADIT. Quest'ultimo è un amplificatore a valvole con guadagno 1000 realizzato da R.V.R Elettronica appositamente per l'Università degli studi di Genova. La tensione applicata al provino è riportata in Fig. 4.5: è possibile vedere come gli overshoot siano presenti su ogni transizione. Si tratta di un'alimentazione PWM con una frequenza della fondamentale di  $3kHz$  e una frequenza di commutazione di  $24kHz$ . I fronti di salita e discesa hanno un  $dv/dt$  pari a  $1.5kV/\mu s$ , mentre il valore massimo raggiunto è pari a  $6kV_{pp}$ .



FIGURA 4.5: Tensione di alimentazione in uscita dall'AWADIT.

Questo tipo di tensione emula lo scenario peggiore a cui può essere sottoposto un sistema di isolamento in una situazione reale. I picchi di tensione rappresentano una riflessione totale della forma d'onda, e sono esattamente il doppio dei 3kV della fondamentale.

### 4.2.3 Sistema di misura scariche condotte

Per verificare che le misure effettuate mediante le antenne siano consistenti, viene utilizzato un sistema tradizionale di tipo condotto: i segnali di scarica sono acquisiti con l'oscilloscopio LeCroy WaveJet 300A, tramite un isolatore capacitivo e una resistenza da  $50\Omega$ . Con questo strumento è possibile scegliere l'accoppiamento resistivo per ogni canale: per le antenne viene impostato a  $50\Omega$ , mentre per la tensione di alimentazione a  $1M\Omega$ . Per eliminare il residuo della tensione di alimentazione e misurare solamente i segnali relativi alle scariche parziali, il segnale viene inviato all'oscilloscopio tramite un filtro di tipo passa-alto, con una frequenza di taglio pari a  $25MHz$ . Per avere un riferimento con la tensione di alimentazione una sonda HV è collegata all'uscita dell'amplificatore AWADIT. In Fig. 4.6 è riportato l'intero sistema utilizzato per compiere la misura.

Il provino è stato inserito all'interno di un forno alla temperatura di  $125^{\circ}C$ , in questo modo è stato possibile emulare le condizioni di temperatura che si hanno durante il normale funzionamento di un motore.

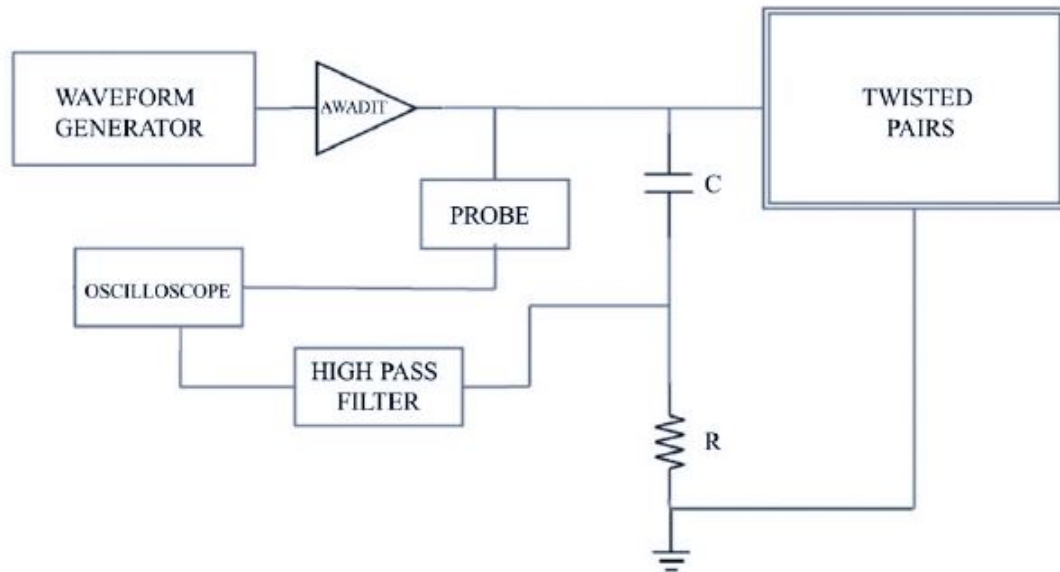


FIGURA 4.6: Sistema per la misura delle scariche condotte.

#### 4.2.4 Sistema di misura per le scariche irradiate

Per la misura delle scariche irradiate sono state installate due antenne in prossimità del forno utilizzato per contenere i provini. La struttura del forno, una doppia parete metallica con interposto un materiale isolante, emula bene la situazione di un motore reale. L'antenna SolBox<sup>®</sup> è stata appoggiata alla parete metallica, mentre un'antenna Log-Periodica, con banda passante tra  $400 \div 1300 MHz$ , è stata posizionata in prossimità. Siccome la copertura metallica del forno e, più in generale, dei motori elettrici non è ideale, risulta possibile captare i segnali di scarica. L'impiego di una seconda antenna è utile per una verifica comparativa dei segnali misurati.

#### 4.2.5 Misure di scariche parziali

La prima prova effettuata ha avuto come scopo quello di testare il corretto funzionamento dell'antenna, in modo da avere conferma che i segnali di scarica siano presenti e misurabili nella posizione in cui è montata. Sono state messe a confronto le misure realizzate mediante il sistema tradizionale descritto nella Sezione 4.2.3 e l'antenna Solbox<sup>®</sup>, per verificare che i segnali acquisiti siano quelli relativi all'attività di scarica [59].

L'oscilloscopio impiegato per comparare i due sistemi non possiede una larghezza di banda sufficiente per un confronto qualitativo delle misure. Per ovviare parzialmente a questo problema l'oscilloscopio è stato impostato in modalità *peak detect*: in questo modo è possibile avere una misura quantitativa dei segnali captati.

In Fig. 4.7 vengono riportati i risultati del test comparativo:

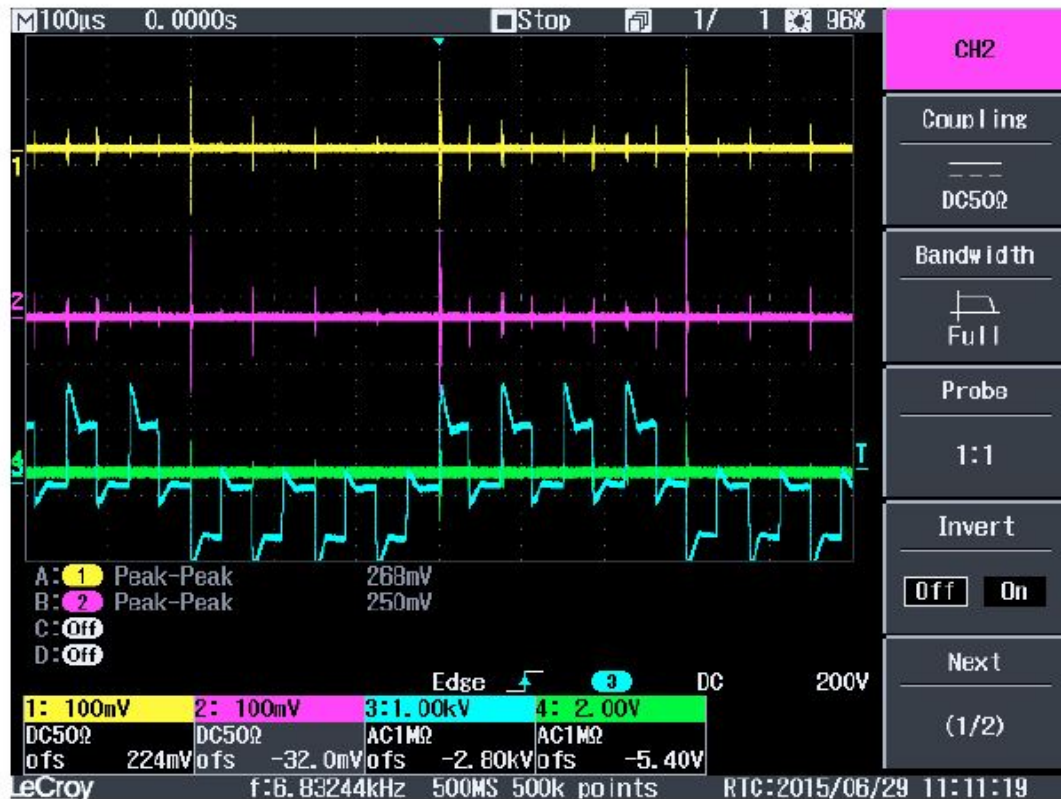


FIGURA 4.7: Confronto di scariche parziali misurate irradiate e condotte su provino twisted pair.

La traccia azzurra rappresenta la tensione di alimentazione, mentre quella verde riporta la misura effettuata mediante il sistema tradizionale. Le tracce gialla e viola sono relative rispettivamente all'antenna SolBox<sup>®</sup> e a quella Log-Periodica.

Si nota immediatamente una corrispondenza temporale perfetta tra i tre segnali dei sensori: tutte le scariche rilevate con il sistema condotto vengono captati anche con le due antenne. È importante osservare gli istanti  $200\mu s$  e  $800\mu s$ : il segnale misurato dal sistema condotto nel primo istante presenta un'ampiezza che risulta essere circa la metà rispetto al secondo. Questa osservazione non è più valida se si osservano, negli stessi punti, i segnali captati con le due antenne poiché presentano ampiezze comparabili. Per cui risulta che soluzioni ottenute con i due sistemi non sono legate in modo diretto.

Per una misura dei segnali di scarica più accurata è stato impiegato l'oscilloscopio Tektronix TDS 7254B, che permette solamente accoppiamenti resistivi a  $50\Omega$ . In Fig. 4.8 viene riportata una scarica misurata con entrambi i sistemi di acquisizione.

È possibile notare l'ampia differenza di bande tra i due segnali, osservando che la scarica misurata con l'antenna presenta oscillazioni in altissima frequenza. Inoltre l'ampiezza del

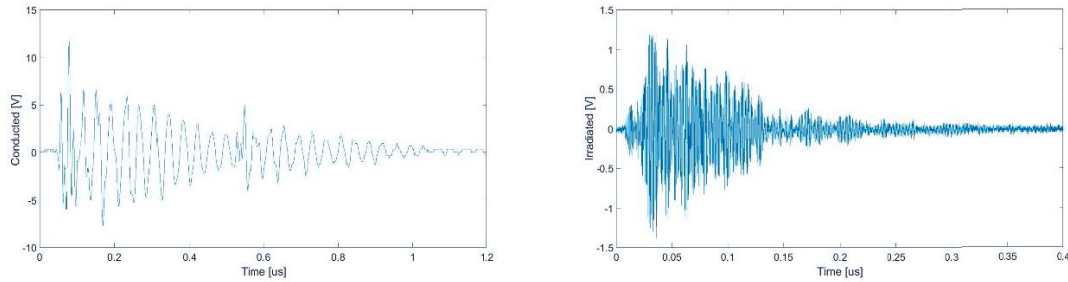


FIGURA 4.8: Confronto scarica parziale misurata con sistema condotto (a sinistra) e con antenna (a destra).

segnale acquisito con l'antenna è una decina di volte inferiore rispetto a quello acquisito con il sistema condotto.

Dopo aver verificato la corrispondenza tra segnali condotti e irradiati, sono state effettuate misure in frequenza delle due antenne mediante l'analizzatore di spettro Agilent N1996A. La modalità utilizzata è la *Max-Hold* sia per quanto riguarda il rumore di fondo sia per le misure di scariche parziali, con una durata di 30 secondi. Questo è necessario in quanto il fenomeno di scarica è casuale e molto veloce. Affinché un segnale sia osservabile è necessario che il filtro dell'analizzatore sia sintonizzato sulle frequenze corrette. Per rilevare le scariche parziali si deve dunque utilizzare una velocità di "spazzolata" sufficientemente elevata che comporta ad avere una bassa risoluzione (in questo caso  $50\text{MHz}$ ), ma più che sufficiente per definire le zone di interesse.

In Fig. 4.9a e in Fig. 4.9b vengono riportate le analisi in frequenza delle scariche captate rispettivamente dall'antenna SolBox<sup>®</sup> e dalla Log-Periodica.

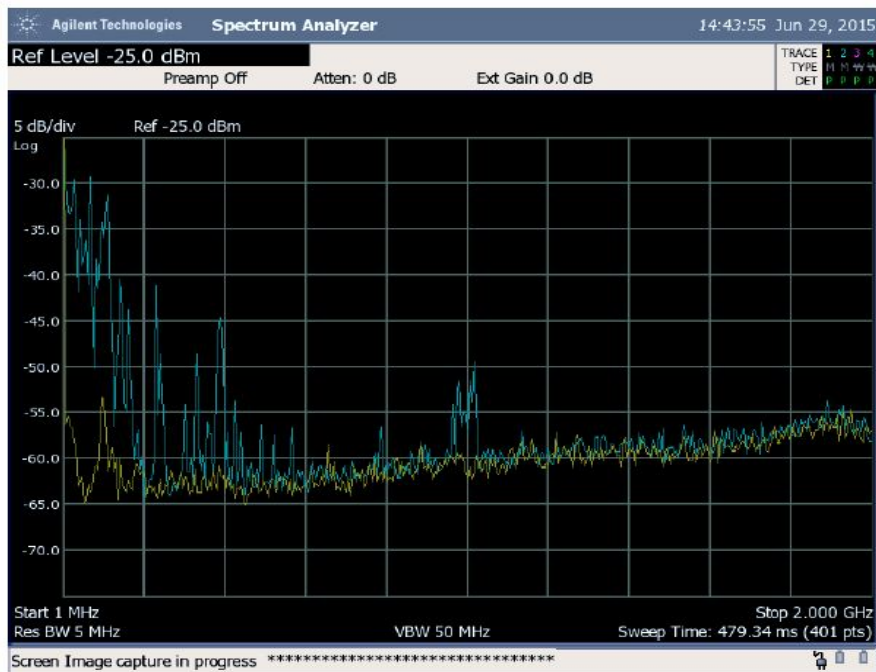
Entrambe le acquisizioni sono state effettuate confrontando il rumore di fondo (traccia gialla) con la misura realizzata in presenza del fenomeno di scariche parziali (traccia blu). Si può osservare come, per entrambe le antenne, si abbia uno spettro distribuito specialmente alle più basse frequenze: per l'antenna SolBox<sup>®</sup> fino a  $600\text{MHz}$  e fino a  $200\text{MHz}$  per la Log-Periodica. Si nota inoltre un residuo del segnale localizzato in prossimità di  $1\text{GHz}$ , più marcato utilizzando l'antenna SolBox<sup>®</sup>.

Procedendo con l'analisi del fenomeno sono state effettuate delle misure impiegando l'oscilloscopio MSO-X 6004A che, con una banda passante da  $6\text{GHz}$ , ha consentito la misura corretta della singola scarica, riportata in Fig. 4.10.

È stato inoltre possibile valutare il disturbo introdotto dall'inverter (Fig. 4.11).

Rielaborando le misure ottenute e facendone la trasformata FFT mediante Matlab, è stato possibile mettere a confronto i due segnali in un unico grafico, riportato in Fig. 4.12.

(A) Spettro scariche antenna SolBox®.



(B) Spettro scariche antenna Log-Periodica.

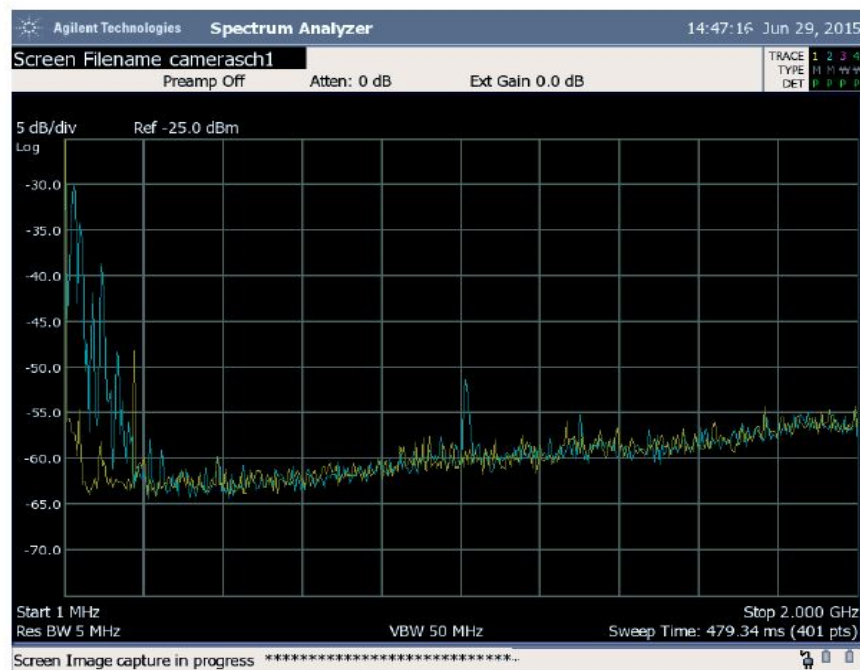


FIGURA 4.9: Spettro delle scariche captate dalle due antenne.

Dalla Fig. 4.12 si osserva come lo spettro delle scariche parziali (traccia blu), misurato con l'antenna SolBox®, sia concentrato leggermente sopra alla frequenza di  $1\text{GHz}$ , mentre il rumore dovuto all'inverter rimane al di sotto di  $500\text{MHz}$  (traccia verde), non disturbando la misura, a patto di effettuare un corretto filtraggio. Dalla comparazione delle due immagini (Fig. 4.10 e Fig. 4.11) inoltre si nota come il disturbo prodotto dall'inverter

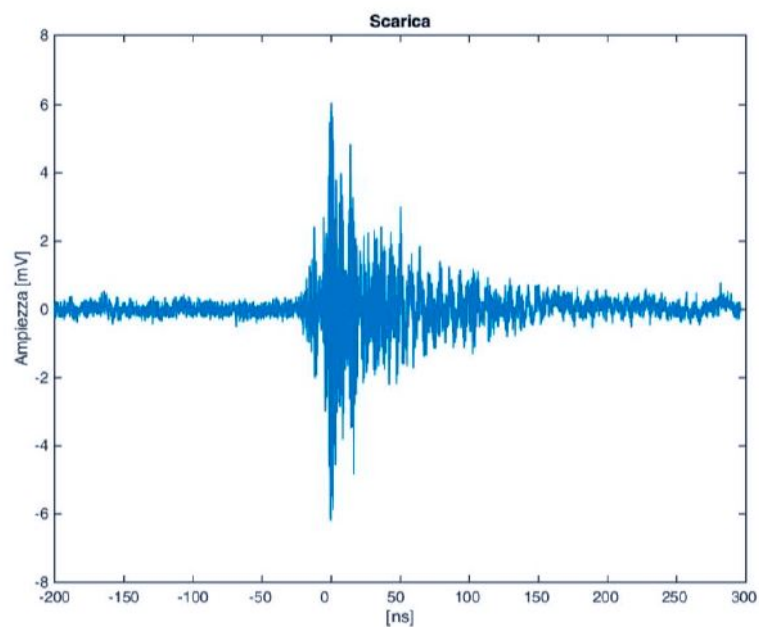


FIGURA 4.10: Acquisizione dettagliata di una scarica irradiata.

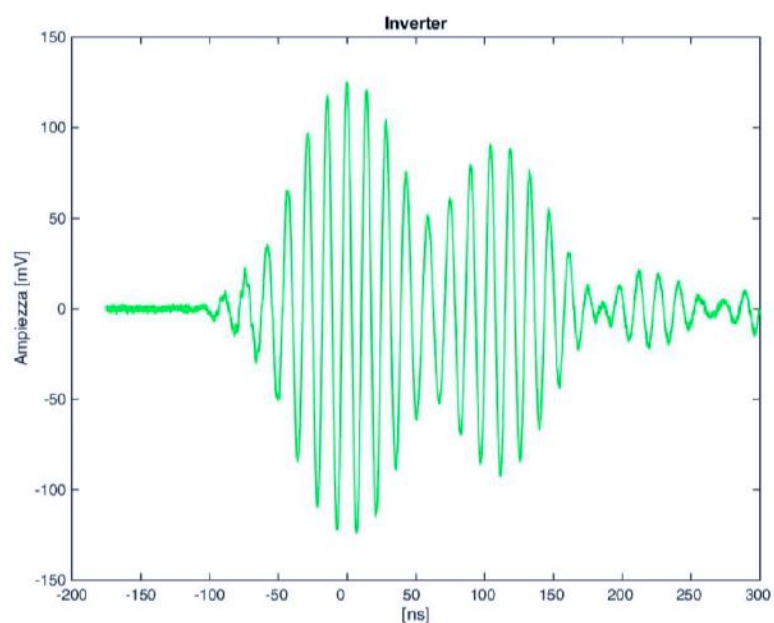


FIGURA 4.11: Acquisizione dettagliata del rumore prodotto con l'inverter.

abbia ampiezza nettamente maggiore rispetto al segnale di scarica, questo perché le correnti in gioco durante una commutazione sono nettamente maggiori rispetto a quelle derivanti dall'attività di scarica parziale.

Si rimanda alla tesi di dottorato [57] per una spiegazione approfondita sul sistema di filtraggio progettato per il sistema SolBox®.

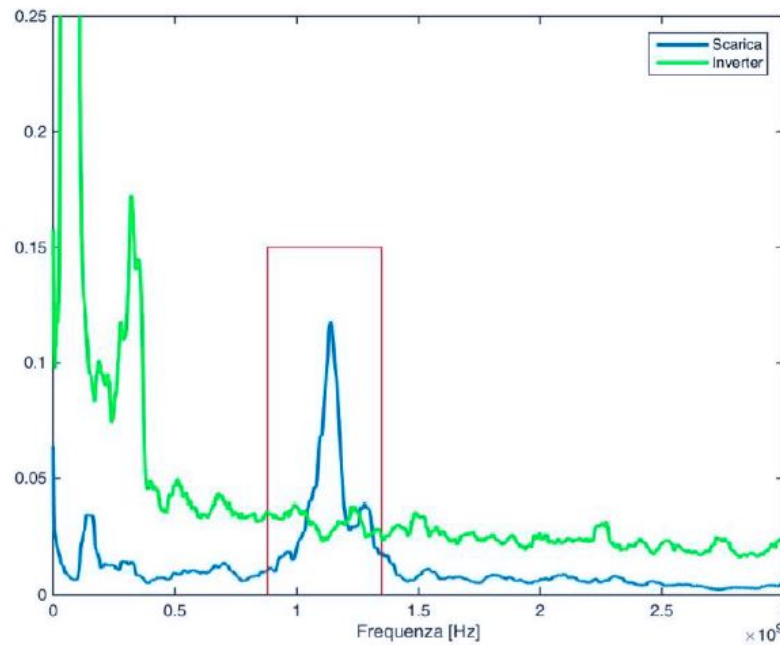


FIGURA 4.12: Spettro dei segnali di scarica e rumore misurati in laboratorio.

### 4.3 Analisi sulla quantizzazione delle scariche e sul rumore di fondo

In [60] è stato condotto un lavoro preliminare sull'influenza della quantizzazione durante l'attività di scarica in presenza di rumore ambientale.

Mentre si verificano le scariche parziali vengono misurati segnali di ampiezze differenti per l'analisi PRPDA (*Phase Resolved Partial Discharges Acquisition*). Solitamente si hanno molti segnali di ampiezza piccola e pochi con ampiezza elevata. Siccome il guadagno dell'amplificatore (se presente) e i livelli di quantizzazione del convertitore AC devono rimanere costanti per tutto il periodo di acquisizione, è necessario impostare il convertitore analogico-digitale (ADC) sull'ampiezza massima dei segnali che si possono acquisire, in modo da non far saturare il convertitore stesso.

Solitamente questi, utilizzati per l'acquisizione delle scariche, hanno una risoluzione di 8 bit: è facile ipotizzare che questa quantità non sia sufficiente per la distinzione delle scariche di ampiezza più piccola dal rumore di fondo. Lo scopo principale di questo studio preliminare è stato trovare la scarica di ampiezza più piccola per cui è ancora possibile distinguere le scariche dal rumore ambientale utilizzando un semplice algoritmo di clustering sulla mappa TW [25].



### 4.3.1 Mappa Equivalent Time–Equivalent Bandwidth

Per utilizzare un semplice algoritmo di clustering è necessario innanzitutto proiettare i dati in un nuovo spazio. A questo scopo è stato scelto di impiegare la mappa *Equivalent Time–Equivalent Bandwidth* anche conosciuta come mappa TW, che permette una proiezione di segnali nel tempo in un grafico 2D. Per ogni segnale è necessario calcolare le due coordinate, una nel dominio del tempo (T) e una in quello delle frequenze (W), in modo tale da rappresentare una scarica in un punto del grafico.

Si supponga che ciascun segnale abbia  $N$  campioni, il baricentro temporale  $t_0$  può essere ottenuto come:

$$t_0 = \frac{\sum_{i=0}^N t_i s_i(t_i)^2}{\sum_{i=0}^N s_i(t_i)^2} \quad (4.1)$$

dove  $s_i(t_i)$  è il campione del segnale al tempo  $t_i$ .

Grazie alla (4.1) è possibile definire una nuova quantità che rappresenta l'*equivalent time-length*:

$$T^2 = \frac{\sum_{i=0}^N (t_i - t_0)^2 s_i(t_i)^2}{\sum_{i=0}^N s_i(t_i)^2} \quad (4.2)$$

Utilizzando la trasformata di Fourier è possibile definire l'*equivalent bandwidth*:

$$W^2 = \frac{\sum_{i=0}^N f_i^2 |X_i(f_i)|^2}{\sum_{i=0}^N |X_i(f_i)|^2} \quad (4.3)$$

dove  $X_i(f_i)$  è la componente  $i$ -th delle frequenze del segnale.

### 4.3.2 Sistema di acquisizione

Il sistema di acquisizione dei segnali impiegati in questo studio è riportato in Fig. 4.13.

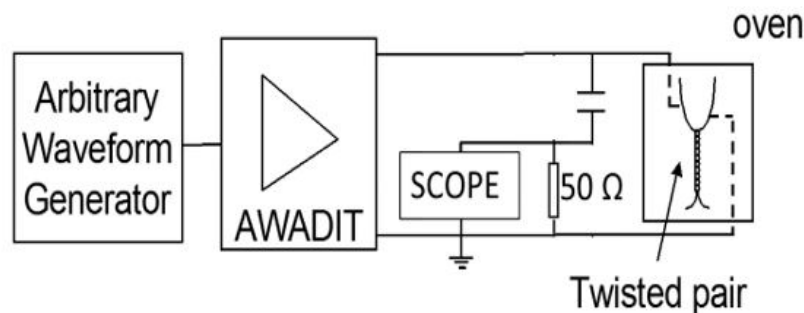


FIGURA 4.13: Sistema di acquisizione delle scariche.

Le scariche sono state generate applicando una tensione sinusoidale amplificata dall'AWADIT ad un provino *twisted pair* al fine di ottenere delle scariche di tipo superficiale.

Il provino è stato posto all'interno di un forno alla temperatura di  $120^{\circ}C$ . Le misure sono state condotte su una resistenza pura a  $50\Omega$ , in modo tale da limitare distorsioni e attenuazioni che potrebbero essere causate da un'impedenza RLC. Per mantenere un adattamento di impedenza perfetto con l'oscilloscopio, avente ingresso a  $50\Omega$ , è stato impiegato un cavo RG58. Ogni segnale è stato campionato con una frequenza di  $2GS/s$  per  $50\mu s$ , utilizzando un convertitore ADC a 8 bit.

La Fig. 4.14 riporta un esempio di segnale catturato.

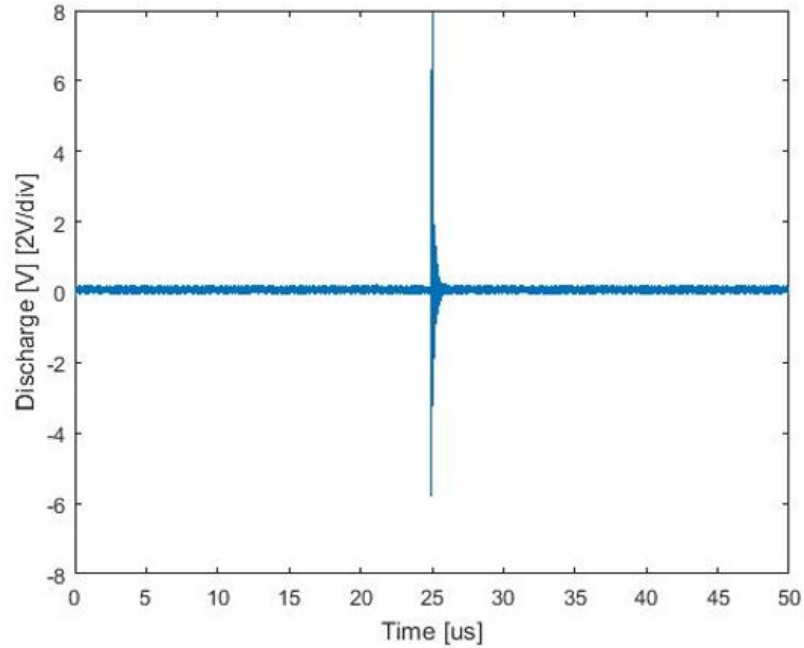


FIGURA 4.14: Esempio di scarica acquisita.

### 4.3.3 Test sperimentali

Per effettuare l'analisi TW sono stati acquisiti 30 segnali di scarica e 30 segnali di rumore ambientale. La scarica più ampia, riportata in Fig. 4.14, ha un'ampiezza di  $7930mV$ , per cui si è deciso di impostare la risoluzione dell'oscilloscopio a  $2V/div$  per evitare la saturazione del convertitore. Il primo grafico TW (Fig. 4.15) è stato costruito impiegando segnali di scarica ben quantizzati.

Il K-means è stato utilizzato per la separazione delle due categorie nel nuovo spazio TW. Le "X" nere sul grafico rappresentano i centroidi dei due cluster. In questo è possibile distinguere facilmente le due classi (punti rossi per le scariche, punti blu per il rumore).

Per analizzare l'effetto della quantizzazione si è deciso di moltiplicare iterativamente i segnali di scarica per un coefficiente  $G < 1$  via via sempre più piccolo, senza cambiare

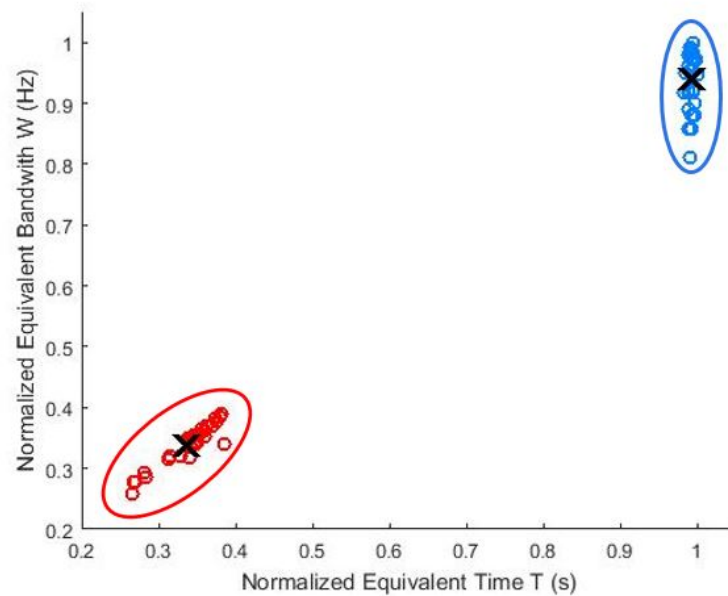


FIGURA 4.15: Mappa TW normalizzata con segnali ben quantizzati.

l'ampiezza del rumore ambientale e senza alterare la risoluzione dell'ADC. In questo modo è possibile simulare l'acquisizione di scariche piccole in presenza di quelle più grandi, con un fondo scala per non permettere la saturazione dell'ADC. Mantenendo lo stesso fondo scala dei segnali originali, le scariche moltiplicate per il fattore  $G$  devono essere ri-quantizzate.

In Fig. 4.16a e in Fig. 4.16b si può notare come, variando il coefficiente  $G$ , il cluster relativo ai segnali delle scariche si avvicini al cluster relativo al rumore.

Il limite di questo metodo per la separazione delle scariche dal rumore risulta essere la sovrapposizione dei segnali delle due classi nella mappa TW: per un valore sufficientemente piccolo di  $G$  il rumore diventa predominante nello spazio TW. Un'eccessiva riduzione del fattore  $G$  rende impossibile il riconoscimento delle scariche dal rumore impiegando l'algoritmo del K-means (Fig. 4.17).

La Fig. 4.18 mostra la distanza euclidea tra i centroidi delle scariche e del rumore in funzione del parametro  $G$ . È possibile definire una soglia sotto la quale i livelli di quantizzazione delle scariche non sono più sufficienti per garantire un buon risultato di separazione con questo metodo.

Infine in Fig. 4.19 è stata confrontata la stessa scarica moltiplicata per 4 valori di  $G$ , di cui 3 sono sopra la soglia (Fig. 4.19a, Fig. 4.19b, Fig. 4.19c) e una sotto (Fig. 4.19d).

L'ampiezza delle scariche è riportata in livelli di quantizzazione piuttosto che nella scala dei volt. L'asse dei tempi è stato zoomato intorno alla scarica per sottolineare le differenze nella ri-quantizzazione dei segnali. Ciò che si osserva chiaramente è che il segnale in

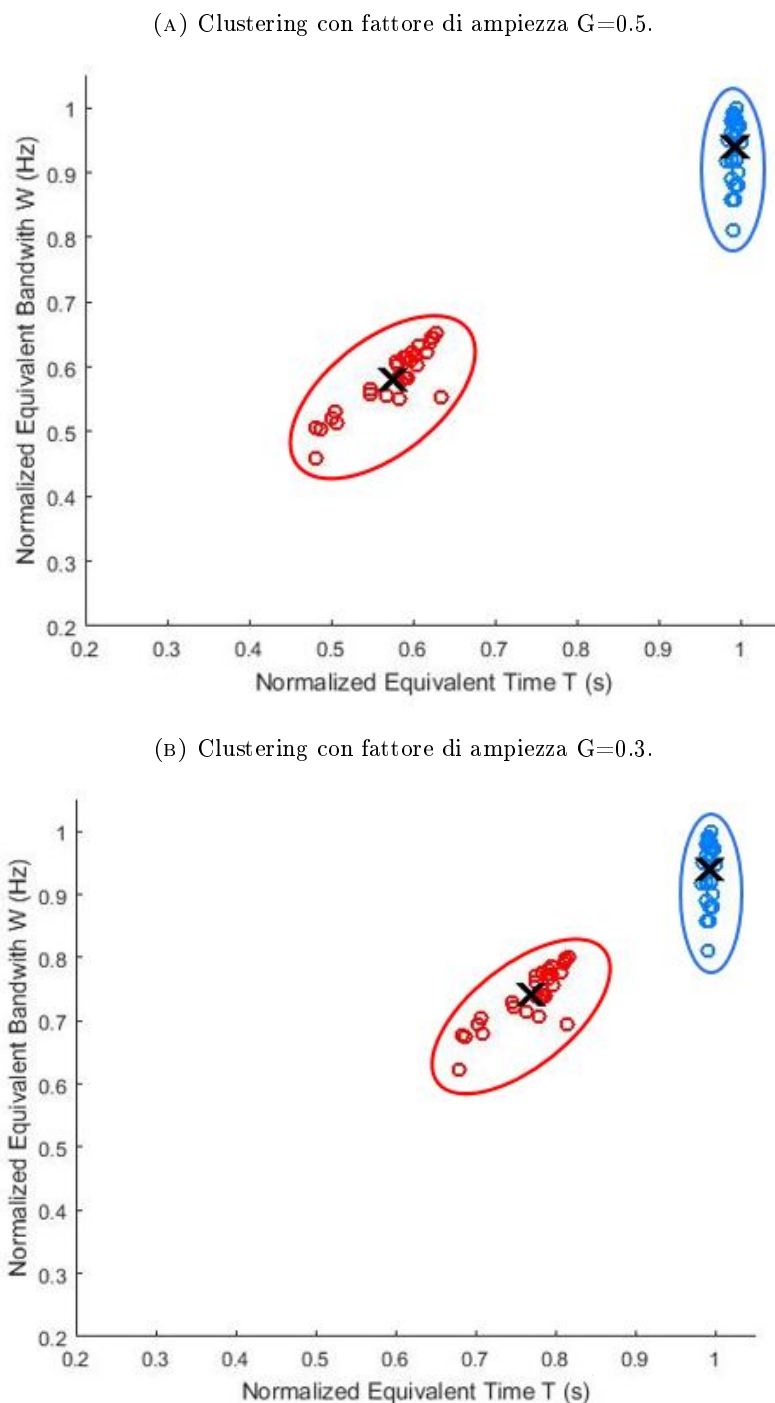
FIGURA 4.16: Clustering su grafico TW variando il fattore di scala  $G$ .

Fig. 4.19d, riconosciuto come rumore dall'algoritmo, in realtà è chiaramente classificabile come scarica.

È stato condotto un ulteriore studio al fine di migliorare il metodo di separazione tra scariche di piccola ampiezza e rumore di fondo, impiegando sempre il K-means come algoritmo di clustering.

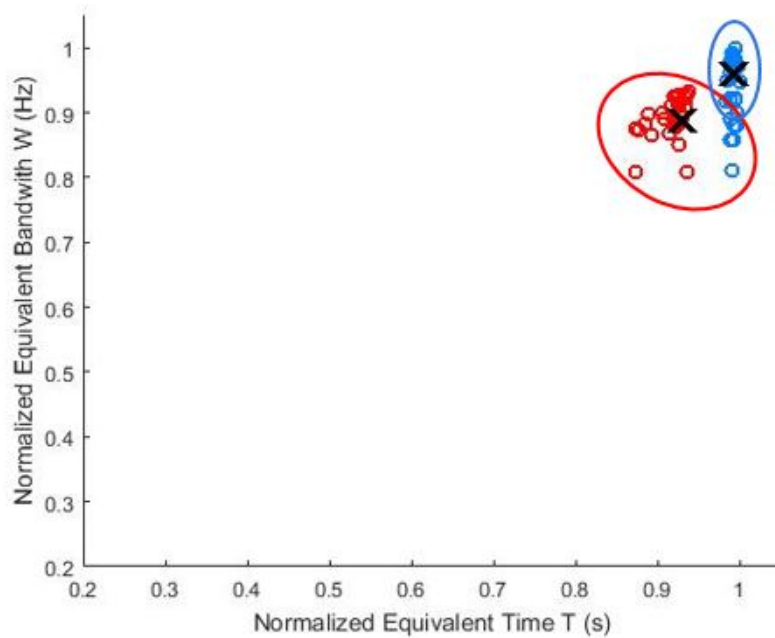


FIGURA 4.17: Clustering con fattore di ampiezza  $G=0.15$ : errore nel riconoscimento dei segnali.

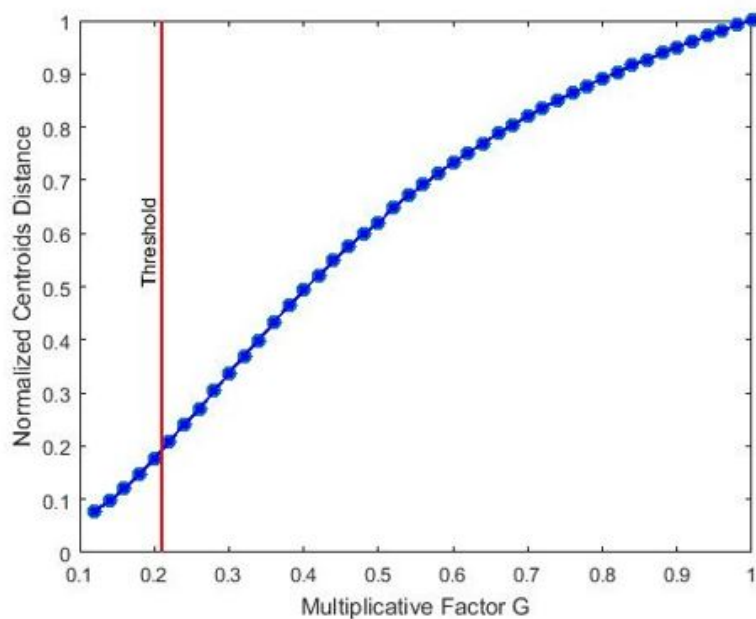
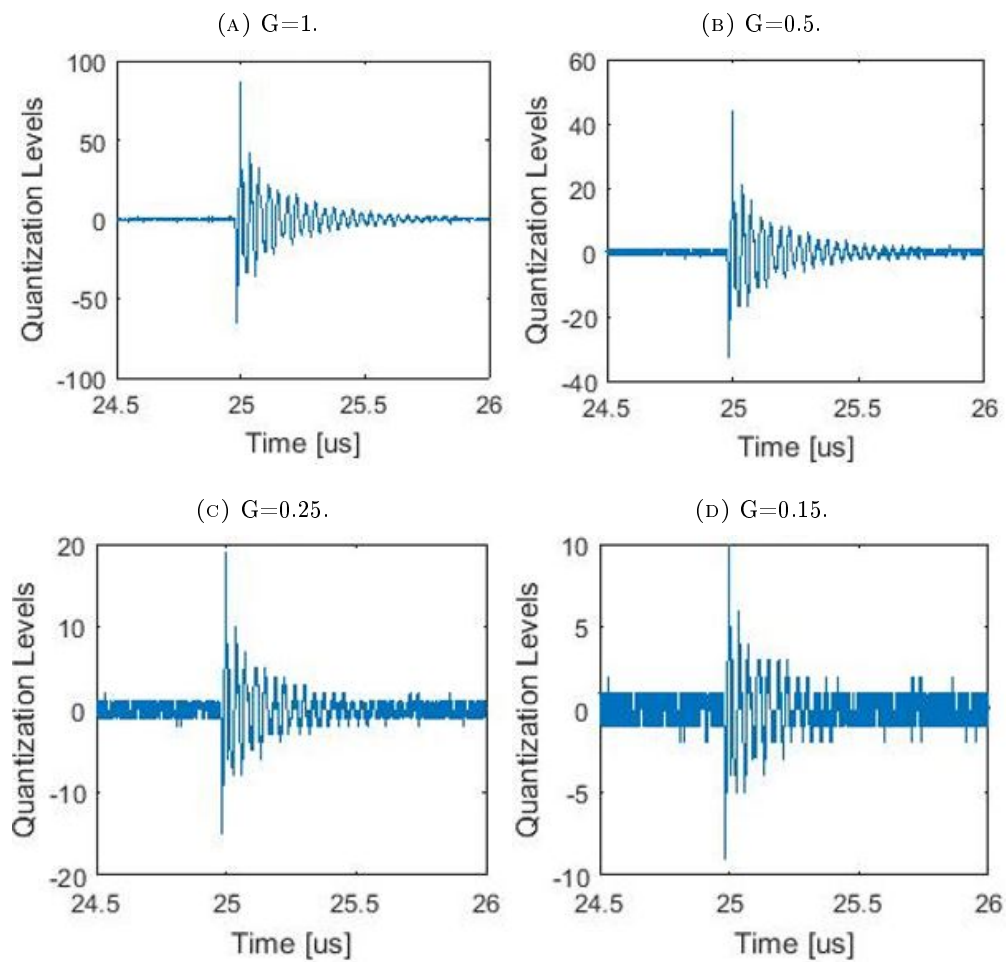


FIGURA 4.18: Distanza tra i centroidi dei cluster delle scariche e del rumore in funzione di  $G$ .

FIGURA 4.19: Scarica quantizzata per diversi valori di  $G$ .

#### 4.4 Comparazione tra sistemi di misura che impiegano un diverso numero di bit per la quantizzazione dei segnali misurati

Nella sezione precedente è stato mostrato che utilizzando un semplice algoritmo di clustering come il K-means in uno spazio 2D chiamato TW è possibile separare i segnali di scarica dal rumore se questi possiedono un'ampiezza sufficientemente grande da garantire una "buona" quantizzazione. Questa osservazione ha portato ad eseguire nuovi test sperimentali sfruttando un numero maggiore di bit nella conversione AD, avendo così, a parità di fondo scala, una risoluzione più elevata. Lo scopo di questo secondo studio [61] è stato trovare le differenze tra due sistemi che adottano un diverso numero di bit (8 e 12 rispettivamente) nel riconoscimento delle scariche dal rumore.

#### 4.4.1 Modello matematico

Su tutti i segnali è stata calcolata la funzione di autocorrelazione normalizzata (NACF). Essa è in grado di sintetizzare le informazioni temporali e in frequenza dei segnali campionati. Dato un segnale la NACF fornisce la stessa informazione in termini di contenuto armonico del segnale stesso (la fondamentale della NACF è uguale alla fondamentale del segnale). Inoltre la funzione di correlazione di una sinusoide o una sinusoide smorzata è sempre una sinusoide o una sinusoide smorzata. Il segnale derivato dal calcolo della NACF risulta meno affetto dal rumore in alta frequenza.

Si assuma che ogni segnale  $S$  abbia  $N$  campioni ( $S = \{s_1, s_2, s_3, \dots, s_N\}$ ), allora la NACF del segnale può essere calcolato come:

$$R[k] = \frac{\sum_{i=1}^N s(i)s(i-k)}{N} \frac{1}{R[0]} \quad (4.4)$$

dove  $R[0]$  è l'energia totale del segnale  $S$ .

Utilizzando la (4.1), la (4.2) e la (4.3) è possibile mappare i nuovi dati nello spazio TW. Una volta proiettati nel nuovo spazio, viene utilizzato il K-means per trovare gli archetipi delle due classi (segnali relativi alle scariche o rumore), ovvero i due punti più rappresentativi dei due cluster.

L'ultimo step consiste nel calcolo del coefficiente di correlazione  $\rho$  tra tutti i segnali NACF e i due archetipi. In questo modo è possibile separare i dati: ciascun dato viene classificato come l'archetipo con cui ha la maggiore correlazione  $\rho$ . Il coefficiente di correlazione è molto usato come indice di similarità e può essere definita come:

$$\rho = \frac{\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^N (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^N (y_i - \bar{y})^2}} \quad (4.5)$$

#### 4.4.2 Sistema di acquisizione

In Fig. 4.20 viene mostrato il sistema di acquisizione. Per ottenere le scariche superficiali viene applicata una tensione sinusoidale con ampiezza elevata ad un elettrodo punta-piano. L'oscilloscopio impiegato per l'acquisizione dei segnali ha una banda passante nel range  $0 \div 200MHz$ , una frequenza di campionamento massima pari a  $2GS/s$  e la possibilità di impostare i livelli di quantizzazione a 8 o 12 bit.

L'impedenza di misura è una pura resistenza a  $50\Omega$ . Il residuo della tensione di alimentazione a  $50Hz$  viene filtrato con un filtro passa-alto del primo ordine. I segnali acquisiti sono stati triggerati con la sinusoide a  $50Hz$  e sono stati campionati con una frequenza

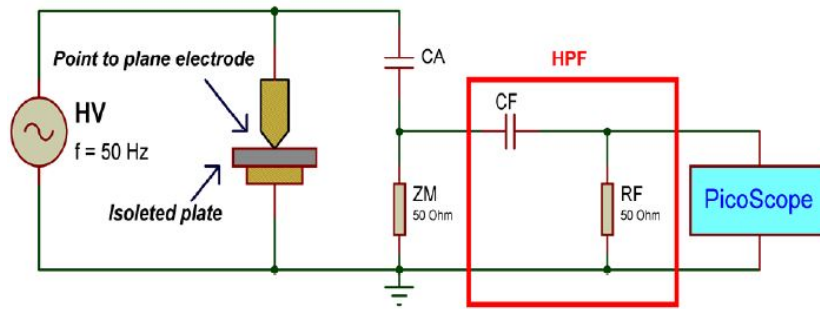
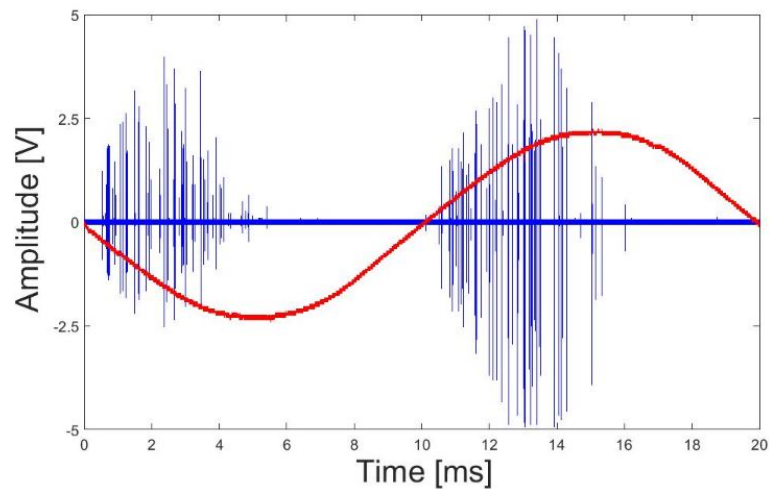


FIGURA 4.20: Sistema di acquisizione.

pari a  $500MS/s$  usando 8 bit come risoluzione, a  $250MS/s$  impiegando 12 bit per la quantizzazione.

In Fig. 4.21 viene riportata l'acquisizione dei segnali in un periodo della tensione di alimentazione a  $50Hz$ . Il fondo scala del convertitore ADC nei due sistemi è stato impostato a  $\pm 5V$  in modo tale da catturare anche le scariche più grandi senza far saturare il convertitore.

FIGURA 4.21: Acquisizione in un periodo della tensione di alimentazione a  $50Hz$ .

#### 4.4.3 Test sperimentali

Una volta che i due dataset (8 bit e 12 bit) sono stati acquisiti, è necessario creare le due classi (segnali di scarica e rumore) per la verifica del metodo descritto nella Sezione 4.4.1. Le scariche sono state estratte dai periodi della sinusoide a  $50Hz$ . La durata di ciascun segnale è di  $2\mu s$ . Allo stesso modo sono stati estratti, in numero uguale, i segnali relativi al rumore.



La Fig. 4.22 riporta alcuni esempi di scariche e rumore per sottolineare visivamente le differenze della quantizzazione sui segnali.

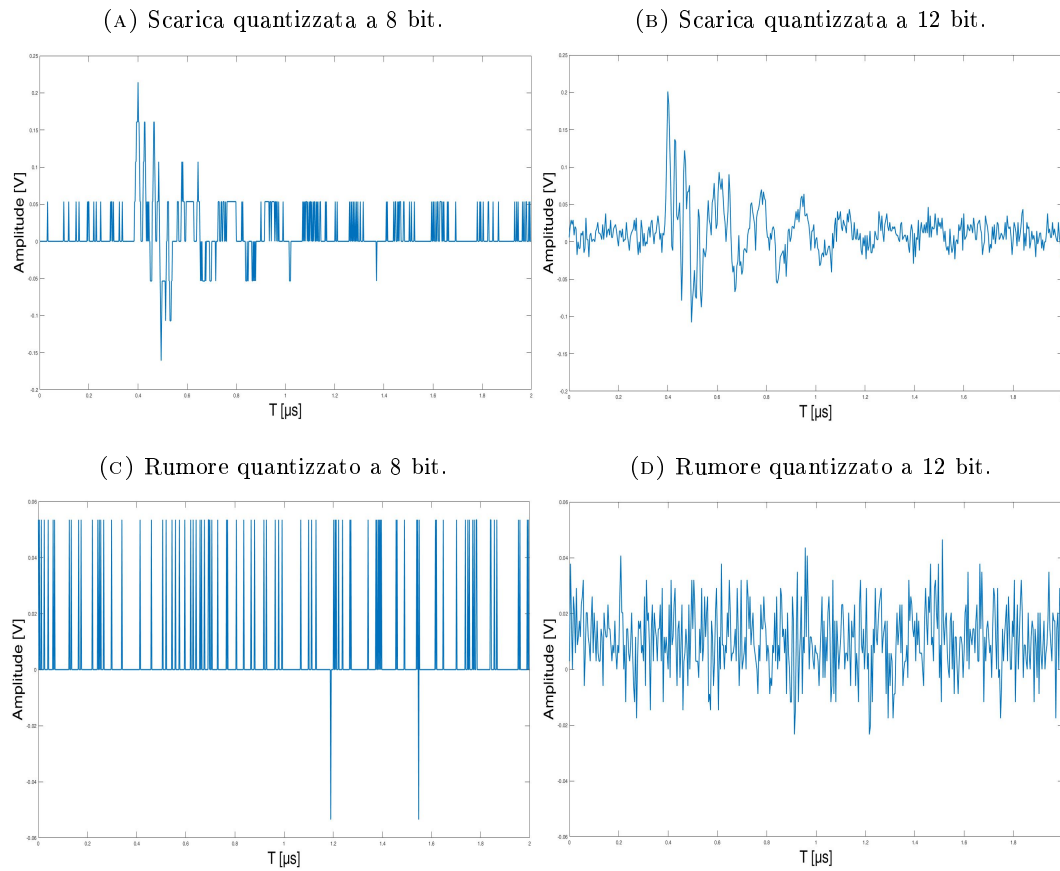


FIGURA 4.22: Esempio di segnali aventi un'ampiezza piccola.

Comparando quelli di Fig. 4.22c e Fig. 4.22d si può notare che la quantizzazione a 8 bit non è sufficiente a caratterizzare la vera forma del segnale poiché l'ampiezza è rappresentabile soltanto con un numero limitato di livelli di quantizzazione; mentre con il sistema a 12 bit, avendo a disposizione molti più livelli nello stesso range di ampiezze, è possibile delineare il profilo del rumore più dettagliatamente. Per quanto riguarda i casi in Fig. 4.22a e Fig. 4.22b le principali differenze si notano nella quantizzazione delle code dei segnali stessi.

Quello che ci si aspetta è che il metodo di separazione (Sezione 4.4.1) possa commettere errori nella classificazione di scariche di piccola ampiezza, categorizzandole come rumore quando si impiega il sistema ad 8 bit per l'acquisizione dei segnali. Per dimostrare questa ipotesi per ogni segnale è stata calcolata la NACF in accordo con la (4.1). I nuovi dati sono stati proiettati sul grafico TW per essere facilmente visualizzabili: in Fig. 4.23a viene mostrata la mappa TW delle NACF calcolate sui segnali campionati a 8 bit, mentre in Fig. 4.23b viene mostrata la mappa TW delle NACF calcolate su quelli campionati a 12

bit. I cerchi blu in entrambi i grafici rappresentano i segnali NACF relativi alle scariche, mentre i cerchi rossi sono relativi al rumore.

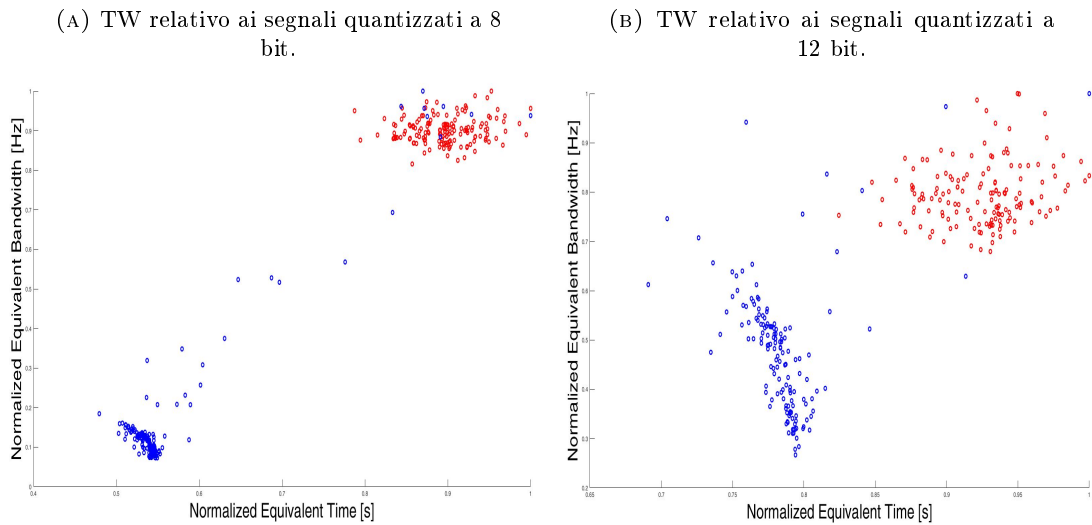


FIGURA 4.23: Grafico TW a 8 e 12 bit sui dati NACF proiettati.

Si può osservare che alcuni segnali relativi alle scariche sono inglobati nel cluster relativo al rumore. Questi segnali verranno sicuramente classificati erroneamente dal K-means.

Il K-means è stato utilizzato per calcolare i centroidi dei due cluster e vengono rappresentati con delle "X" nere sul grafico TW (Fig. 4.24a e Fig. 4.24b). I cerchi verdi rappresentano i segnali classificati erroneamente dall'algoritmo di clustering. È importante sottolineare che utilizzando solo il K-means per la separazione dei dati si ottengono errori sia nel sistema a 8 bit che in quello a 12 bit.

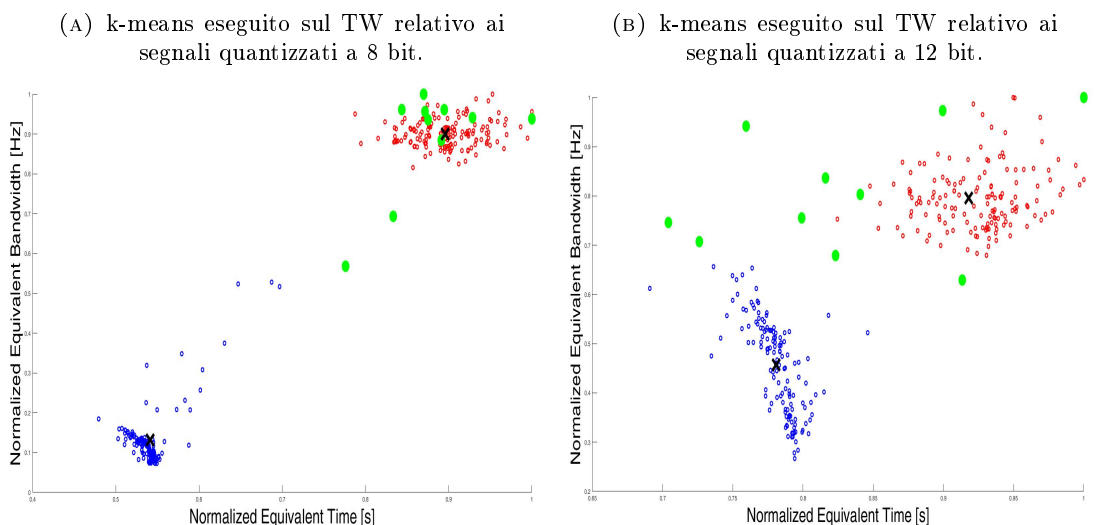


FIGURA 4.24: L'algoritmo k-means trova i centroidi nel nuovo spazio e crea i cluster. I dati classificati erroneamente vengono rappresentati con dei cerchi verdi.

Per superare il problema della classificazione errata viene calcolato il coefficiente di correlazione tra tutti i segnali NACF e i due NACF dei centroidi utilizzando la (4.5) in entrambi i sistemi. Il coefficiente di correlazione viene utilizzato come indice di similarità: la classe di appartenenza di ciascun segnale viene attribuita a quella che presenta un  $\rho$  maggiore tra i due calcolati rispetto ai centroidi.

Usando il coefficiente di correlazione come indice di similarità tutti i segnali quantizzati a 12 vengono separati correttamente. Alcuni campionati a 8 bit continuano ad essere attribuiti alla classe sbagliata anche se in numero minore rispetto a prima (Fig. 4.25). L'errore di classificazione è dovuto al numero di livelli non sufficiente per rappresentare segnali di piccola ampiezza. Le scariche classificate erroneamente sono quelle che in Fig. 4.24a sono pressoché incluse nel cluster del rumore.

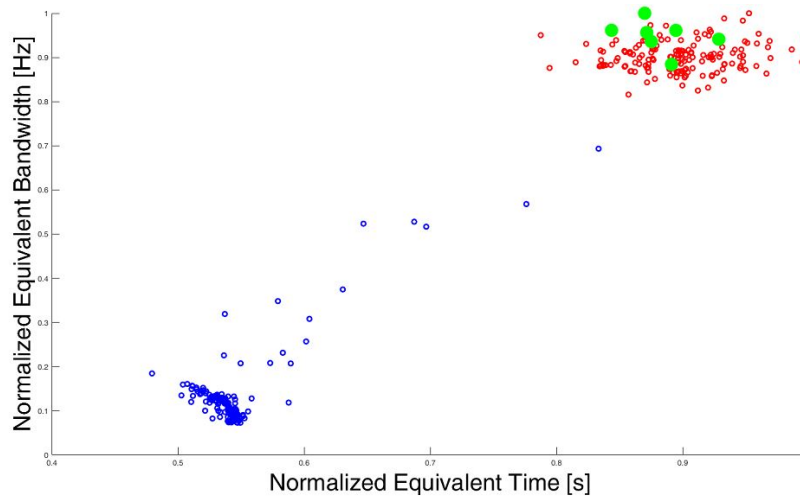


FIGURA 4.25: Classificazione dei segnali a 8 bit utilizzando l'indice di similarità  $\rho$ . I cerchi verdi sono i segnali di scarica che l'algoritmo attribuisce erroneamente al cluster del rumore.

Questo metodo è semplice e veloce da utilizzare e risulta efficace nella separazione di scariche dal rumore nel caso in cui si adoperi un sistema con un numero sufficiente di bit allo scopo di rappresentare segnali di piccola ampiezza. Nel caso in cui il rumore fosse talmente elevato da risultare confrontabile con le scariche di maggiore ampiezza è necessario adoperare un filtraggio adeguato dei segnali o tecniche software come la *Wavelet Decomposition*, molto utilizzata in letteratura, che richiedono uno sforzo computazionale molto più elevato.

## Capitolo 5

# Approcci di apprendimento veloce

La grande disponibilità di dati in ambito industriale crea nuove opportunità nell'area dell'apprendimento statistico [62–64]. Le architetture basate sul deep learning [64] hanno dimostrato di raggiungere ottimi risultati in termini di generalizzazione. Nel campo del riconoscimento delle scariche parziali questi algoritmi hanno iniziato a diffondersi solamente negli ultimi anni [65–67]. D'altro canto si riscontrano alcuni problemi: 1) l'implementazione della fase di addestramento e di inferenza richiedono un'elevata potenza computazionale [68]; 2) la fase di training è estremamente sensibile alla parametrizzazione; infine, 3) per addestrare una rete deep è necessario disporre di un dataset con un numero di campioni elevatissimo.

Gli aspetti computazionali influenzano, tuttavia, anche gli approcci standard di apprendimento automatico, che in generale possono riguardare training set di medie dimensioni (migliaia di campioni). Questo problema diventa rilevante quando la procedura di addestramento deve essere eseguita su dispositivi con risorse limitate, ad esempio sistemi elettronici embedded. In questo scenario, è necessario fare affidamento su un paradigma in grado di tenere in considerazione il compromesso tra complessità del predittore e capacità di generalizzazione. A questo scopo, i modelli che sfruttano le *Single Hidden Layer Feedforward Networks* (SLFN) e le *Random Basis Functions* (RBF) hanno ottenuto risultati significativi, perché 1) richiedono risorse computazionali modeste [69], 2) la natura convessa del problema di ottimizzazione semplifica la fase di implementazione [63] e 3) la quantità limitata di parametri liberi coinvolti consente di addestrare questi modelli anche quando i dataset disponibili sono piccoli.

Le *Random Radial Basis Functions* [70], la *Random Vector Functional-Link* (RVFL) [71], le *Extreme Learning Machines* (ELMs) [72, 73], e le *Weighted Sum of Random Kitchen Sinks* [74] rappresentano esempi molto interessanti di tale approccio. Questi modelli condividono l'idea comune di inizializzare lo strato *hidden* usando la randomicità per

supportare una trasformazione fissa dei dati di input. Di conseguenza, il processo di apprendimento dovrebbe solo regolarizzare un separatore lineare nello strato superiore (vale a dire, nello spazio rimappato). Questa semplificazione non influenza le capacità di approssimazione universale delle reti risultanti [63]. Tra questi paradigmi, ELM è quello che presenta il più solido background teorico e la maggior parte della letteratura si riferisce alla *random based neural network* come ELM. Per questi motivi, nel seguito, si farà riferimento ai modelli basati sulla randomicità come ELM o viceversa.

Questo capitolo mira ad aprire nuove prospettive sugli approcci di apprendimento elencati precedentemente, dimostrando che esiste un parallelismo tra le *Random Basis Functions* e le funzioni di similarità [75]. A questo proposito, il primo contributo fornito da questa tesi è una nuova analisi dello spazio dell'ipotesi della ELM che sfrutta la convergenza tra ELM e la teoria dell'apprendimento con funzioni di similarità (SIM). Di conseguenza, questa ricerca mostra che nel modello ELM standard ogni neurone dello strato hidden rimappa un dato di input indipendentemente dagli altri; inoltre, la funzione di attivazione (*basis function*) agisce come una funzione di similarità che rimappa i dati di input secondo una nozione di similarità casuale rispetto a un punto di riferimento casuale (*landmark*). Come conseguenza principale, mentre lo strato che rimappa i dati non richiede alcun adattamento dei parametri, molti neuroni potrebbero coprire porzioni dello spazio dei dati irrilevanti o essere parametrizzati male, compromettendo così l'efficienza nel comprendere la struttura del problema in questione. Ciò è ancora più importante se si considera che le complessità computazionali delle operazioni di addestramento e di runtime dipendono dalla dimensione dello strato di *mapping*. Pertanto, questa ricerca dimostra che in linea di principio il framework ELM fornisce uno spazio di ipotesi più ampio rispetto alla teoria standard delle funzioni di similarità. Tuttavia, lo svantaggio risiede in un piccolo deterioramento dei limiti di apprendimento.

Il secondo contributo fornito dal presente capitolo è dato da una serie di nuove strategie di training per il modello ELM. L'obiettivo finale è superare le limitazioni imposte dal mapping standard. Le strategie di training proposte si basano ancora su landmark casuali; viceversa, il parametro di configurazione della funzione di attivazione/similarità viene scelto in base a procedure di selezione che mitigano la presenza di configurazioni patologiche. Tale risultato si ottiene evitando strategie standard che sono impegnative dal punto di vista computazionale, come la *backpropagation* o la *model selection*.

## 5.1 Reti neurali feedforward basate sulla randomicità

Le reti neurali *feedforward* [76] rappresentano uno dei paradigmi più utilizzati grazie al loro trade-off tra il carico computazionale e la capacità di generalizzazione del predittore

finale. Considerando un dato di input  $\mathbf{x} \in \mathbb{R}^D$ , la struttura di una SLFN è:

$$f(\mathbf{x}) = \sum_{j=1}^N \beta_j h_j(\mathbf{x}, \boldsymbol{\omega}_j, \sigma_j) \quad (5.1)$$

dove  $N$  è il numero di neuroni nello strato hidden,  $h_j$  è una funzione di attivazione non lineare,  $\boldsymbol{\omega}_j$  sono i parametri dello strato hidden,  $\sigma_j$  è il parametro di forma del  $j$ -esimo neurone e  $\boldsymbol{\beta}$  è il vettore dei pesi di output.

Dato un insieme di dati etichettati  $\mathcal{T} = \{(\mathbf{x}_i, y_i), i = 1, \dots, Z, \mathbf{x}_i \in \mathbb{R}^D, y_i \in [-1; 1]\}$ , il training di questi modelli consiste nell'ottimizzare i parametri della rete per minimizzare la funzione costo  $L$ :

$$\min_{\boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\Omega}} L(\mathcal{T}, f(\boldsymbol{\beta}, \boldsymbol{\Omega}, \boldsymbol{\sigma})) \quad (5.2)$$

In pratica, l'addestramento consiste nella soluzione di un problema di ottimizzazione che, in base alla funzione costo specifica  $L$  e al predittore  $f$ , può presentare diversi livelli di complessità.

Anche se è stato proposto nel 1986, l'algoritmo di *backpropagation* (BP) [77] è alla base delle procedure più efficaci e utilizzate per l'addestramento delle reti neurali. Questa tecnica di ottimizzazione si basa su un aggiornamento iterativo dei parametri dei modelli  $\{\boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\Omega}\}$  nella direzione opposta rispetto al gradiente della funzione costo. Questa tecnica garantisce una soluzione ottima a livello globale solo quando sia  $L$  che  $f$  sono convessi rispetto ai parametri. Nonostante la sua efficacia, l'algoritmo della BP nella fase di training di una SLFN converge lentamente e sono presenti problemi di minimi locali poiché  $f$  non è convessa per costruzione. La letteratura fornisce numerosi approcci per affrontare questo problema [78, 79], ma in generale vengono introdotti nuovi iper-parametri che rendono difficile il processo di ottimizzazione. Inoltre, i requisiti imposti dal calcolo del gradiente limitano le funzioni di attivazione  $h$  consentite a quelle differenziabili.

Una strategia diffusa, per affrontare i problemi sopracitati, si basa sui cosiddetti modelli casuali [70–72, 74]. L'idea comune alla base di tutti questi paradigmi consiste nell'eseguire una trasformazione non lineare e casuale dei dati di input. Ciò si ottiene impostando casualmente i parametri  $\boldsymbol{\Omega}$  e  $\boldsymbol{\sigma}$ . Questa scelta progettuale influisce notevolmente sul carico computazionale della fase di addestramento perché  $f$  diventa convessa rispetto ai parametri di training. Di conseguenza, il problema di ottimizzazione si trasforma nel trovare un separatore lineare  $\boldsymbol{\beta}$  nel nuovo spazio. Dato che  $f$  è convessa, se la funzione costo è anch'essa convessa, il problema di ottimizzazione ammette un'unica soluzione ottima. Tra le funzioni costo convesse più utilizzate in letteratura troviamo il *Means Square Error* (MSE):

$$\min_{\boldsymbol{\beta}} \{\|\mathbf{y} - \mathbf{H}\boldsymbol{\beta}\|^2\} \quad (5.3)$$

dove  $\mathbf{H}$  indica una matrice  $Z \times N$ , con  $h_{ij} = h_j(\mathbf{x}_i, \boldsymbol{\omega}_j, \sigma_j)$ .

Il successo del MSE è strettamente legato al fatto che il problema di ottimizzazione risultante, non solo diventa convesso, ma ammette anche una soluzione in forma chiusa:

$$\boldsymbol{\beta} = \mathbf{H}^{-1} \mathbf{y} \quad (5.4)$$

È interessante notare che la capacità di approssimazione universale [80] viene mantenuta e l'insieme delle funzioni di attivazione  $h$  ammissibili viene esteso a un set più ampio rispetto a quelli consentiti dalla BP [63]. Il problema della stabilità numerica e dell'*overfitting* influenzano la soluzione del problema di apprendimento. La teoria spiegata in [81] dimostra che le tecniche di regolarizzazione possono migliorare ulteriormente le prestazioni di generalizzazione dell'approccio, allo stesso tempo il problema della stabilità numerica può essere limitato rendendo la soluzione "smooth". Di conseguenza, si sostituisce la funzione di costo (5.3) con:

$$\min_{\boldsymbol{\beta}} \{ \|\mathbf{y} - \mathbf{H}\boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|^2 \} \quad (5.5)$$

dove  $\lambda$  è il parametro di regolarizzazione che controlla il rapporto tra il *fitting* dei dati e la "smoothness" della soluzione. Questa versione regolarizzata (RMSE) della funzione costo ammette ancora una soluzione analitica in forma chiusa. Quando  $Z \leq N$ , si ha:

$$\boldsymbol{\beta} = \mathbf{H}^T (\lambda \mathbf{I} + \mathbf{H}\mathbf{H}^T)^{-1} \mathbf{y} \quad (5.6)$$

mentre, quando  $Z > N$  si ha:

$$\boldsymbol{\beta} = (\lambda \mathbf{I} + \mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{y} \quad (5.7)$$

È importante sottolineare il fatto che, nonostante la soluzione in forma chiusa mostrata in (5.4) e (5.5), il calcolo della soluzione non coinvolge l'inversione esplicita della matrice, ma si basa su algoritmi per la soluzione di sistemi di equazioni lineari che sono computazionalmente più efficienti [82]. È interessante notare che l'errore quadratico medio regolarizzato non è l'unica funzione costo convessa disponibile per regolarizzare il separatore lineare; in [83] gli autori hanno confrontato diverse funzioni costo in termini di capacità di generalizzazione del predittore finale e i risultati hanno dimostrato che RMSE poteva non essere la soluzione ottimale in alcuni casi. Nonostante quest'ultima osservazione, RMSE vanta una letteratura solida e ben strutturata con algoritmi ottimizzati che consentono un'implementazione efficiente in uno scenario con risorse limitate. Per questo motivo, nel seguito, verrà considerata la funzione costo RMSE.

La procedura di training completa della ELM viene presentata nell'Algoritmo 1.

---

**Algoritmo 1** Schema di apprendimento basato sulla randomicità nello strato hidden.

---

**Input**

- Un insieme di dati di training etichettato  $\mathcal{T} = \{(\mathbf{x}, y)_i; i = 1, \dots, Z\}$ .
- Un insieme di funzioni di attivazione  $\mathbf{h}$ .
- Un numero di neuroni  $N$ .

**0. Inizializzazione**

Estrazione randomica di un insieme  $\{\boldsymbol{\omega}_i, \sigma_i\}, i = 1, \dots, N$ .

**1. Mapping**

Mapping dei dati  $\mathbf{x} \in \mathcal{T}$  in un nuovo spazio usando la seguente funzione:

$$\phi(\mathbf{x}) = \{h_1(\mathbf{x}, \boldsymbol{\omega}_1, \sigma_1), \dots, h_N(\mathbf{x}, \boldsymbol{\omega}_N, \sigma_N)\}$$

**2. Apprendimento**

Training di un predittore lineare  $\phi : \mathcal{X} \rightarrow \mathbb{R}^N$ .

---

## 5.2 Cenni sulla teoria dell'apprendimento con le funzioni di similarità

La teoria dell'apprendimento con le funzioni di similarità presentata in [75] consiste in un quadro teorico che estende il paradigma matematico dell'apprendimento con le funzioni *kernel* [84]. Questa teoria generale stabilisce le condizioni sufficienti per una funzione di similarità per consentire di "apprendere bene", senza richiedere che la funzione sia semi-definita positiva o addirittura simmetrica.

La seguente coppia di definizioni riassume gli elementi cruciali del quadro teorico; il primo affronta la nozione di funzione di similarità "a coppie",  $K$ , mentre il secondo introduce la nozione più semplice, ma intuitiva di una buona funzione di similarità.

**Definizione 1.** [75] Una funzione di similarità su  $\mathcal{X}$  è una qualsiasi funzione a coppie  $K : \mathcal{X} \times \mathcal{X} \rightarrow [-1, 1]$ .  $K$  è definita come una funzione di similarità simmetrica se  $\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}, K(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}', \mathbf{x})$ .

**Definizione 2.** [75]  $K$  è una buona funzione di similarità  $(\epsilon, \gamma)$  forte per un problema di apprendimento  $\mathbf{P}$  se con una probabilità di almeno  $(1 - \epsilon)$  una grande quantità di campioni  $\mathbf{x}$  soddisfa:

$$E_{\mathbf{x}' \sim \mathbf{P}}[K(\mathbf{x}, \mathbf{x}') | y(\mathbf{x}) = y(\mathbf{x}')] \geq E_{\mathbf{x}' \sim \mathbf{P}}[K(\mathbf{x}, \mathbf{x}') | y(\mathbf{x}) \neq y(\mathbf{x}')] + \gamma \quad (5.8)$$



La definizione di buona funzione di similarità  $(\epsilon, \gamma)$  impone una serie di assunzioni rigorose che non sono applicabili allo scenario reale se  $K$  è fissato a priori.

Una buona funzione di similarità  $(\epsilon, \gamma)$  soddisfa vincoli meno stringenti (ereditati di nuovo da [75]):

**Definizione 3.** [75] *Una funzione di similarità,  $K$ , è una buona funzione di similarità  $(\epsilon, \gamma)$  per il problema di apprendimento  $\mathbf{P}$  se esiste una funzione di pesi  $\omega$  su  $\mathcal{X}$  ( $\omega(\mathbf{x}') \in [0, 1]$ ) per tutti  $\mathbf{x}' \in \mathcal{X}$  tale che con una probabilità di almeno  $(1 - \epsilon)$  una grande quantità di campioni  $\mathbf{x}$  soddisfi:*

$$E_{\mathbf{x}' \sim \mathbf{P}}[\omega(\mathbf{x}')K(\mathbf{x}, \mathbf{x}')|y(\mathbf{x}) = y(\mathbf{x}')] \geq E_{\mathbf{x}' \sim \mathbf{P}}[\omega(\mathbf{x}')K(\mathbf{x}, \mathbf{x}')|y(\mathbf{x}) \neq y(\mathbf{x}')] + \gamma \quad (5.9)$$

La definizione sopra riportata utilizza una funzione di pesi,  $\omega$ , per bilanciare il significato relativo di ciascun campione,  $\mathbf{x}'$ . Quando si considera l'eventuale algoritmo di apprendimento, la definizione richiede l'esistenza di una funzione di pesi,  $\omega$ , sebbene ciò non implichi che tale funzione sia nota a priori. In altri termini, qualsiasi espressione di similarità è, in linea di principio, una buona funzione di similarità  $(\epsilon, \gamma)$  e c'è bisogno di un criterio adatto per trovare la funzione di pesi  $\omega$  che riduca al minimo  $\epsilon$  e massimizzi  $\gamma$ .

In pratica, si dovrebbe sostituire l'*expectation* nella (5.9) con una media su un insieme di landmark, ad esempio, i campioni  $\mathbf{x}'$ . Di conseguenza, la funzione dei pesi  $\omega$  soddisferà la Definizione 3 in corrispondenza dei landmark. Il risultato di questa osservazione è che la nozione di una buona funzione di similarità consente di impostare uno schema di apprendimento basato su uno spazio di ipotesi. A questo scopo, si ha bisogno di:

- $L$  landmark, cioè un sottoinsieme del set di dati originale che viene estratto in modo casuale dalla distribuzione del dominio  $p(\mathcal{X})$  che caratterizza  $\mathbf{P}$ . Entrambi i modelli etichettati e senza etichetta forniscono una fonte ammissibile di landmark.
- Una funzione di similarità  $K$ .

Per costruire lo spazio di ipotesi, lo spazio del dominio  $\mathcal{X}$  viene prima proiettato in un nuovo spazio  $\mathbb{R}^L$ . Di conseguenza, per ogni punto  $\mathbf{x}$ , si calcolano le similarità  $K$  tra  $\mathbf{x}$  e ogni landmark. Nella seconda fase viene addestrato un predittore lineare nel nuovo spazio  $\mathbb{R}^L$ . Lo spazio di ipotesi risultante può essere formalizzato come:

$$f(\mathbf{x}) = \sum_{j=1}^L \omega_j K(\mathbf{x}, \mathbf{l}_j) \quad (5.10)$$

dove i pesi  $\omega_j$  sono calcolati regolarizzando un predittore lineare. L'Algoritmo 2 delinea la procedura di apprendimento associata.

---

**Algoritmo 2** Schema di apprendimento che impiega la teoria delle buone funzioni di similarità  $(\epsilon, \gamma)$ .

---

**Input**

- Un insieme di dati di training etichettato  $\mathcal{T} = \{(\mathbf{x}, y)_i; i = 1, \dots, Z\}$ .
- Una funzione di similarità  $K$ .
- Numero di landmark  $L$ .

**0. Inizializzazione**

Si estraggano  $L$  campioni random  $\mathcal{L} = \{\mathbf{l}_n; n = 1, \dots, L\}$  da  $\mathcal{T}$ .

**1. Mapping**

Si rimappino tutti i campioni  $\mathbf{x} \in \mathcal{T}$  usando la seguente funzione di mapping:

$$\phi(\mathbf{x}) = \left\{ \frac{1}{\sqrt{L}}K(\mathbf{x}, \mathbf{l}_1), \dots, \frac{1}{\sqrt{L}}K(\mathbf{x}, \mathbf{l}_L) \right\}$$

**2. Apprendimento**

si addestrano un predittore lineare  $\phi : \mathcal{X} \rightarrow \mathbb{R}^L$

---

L'Algoritmo 2 si basa sulla funzione di similarità  $K$  per rimappare lo spazio originale in un nuovo spazio in cui i dati sono separati da un margine (possibilmente grande) con errore  $\epsilon$ . Quindi, il compito di aggiustare la funzione dei pesi  $\omega$  è assegnato al predittore lineare. Le capacità di apprendimento di questa procedura sono state formalmente analizzate in [75]: se si imposta  $L = 16 \cdot \ln(4/\epsilon^*)/\gamma^2$ , con probabilità almeno di  $(1 - \epsilon^*/2)$  esiste un separatore nello spazio delle funzioni con un errore piccolo ( $\leq \epsilon + \epsilon^*$ ) e con un margine grande ( $\geq \gamma/2$ ).

### 5.3 Convergenza tra l'apprendimento con le funzioni di similarità e l'Extreme Learning Machine

Esistono diversi punti in comune tra i predittori che derivano dai paradigmi introdotti nella Sezione 5.1 e nella Sezione 5.2 e le loro procedure di apprendimento [69, 85, 86]. In effetti, entrambi gli approcci presentati negli Algoritmi 1 e 2 si basano su una procedura divisa in due fasi: in primo luogo, i dati vengono proiettati in uno spazio con dimensione

esplicita,  $L$  per SIM e  $N$  per ELM. Il secondo passo della procedura consiste nella messa a punto di un separatore lineare nel nuovo spazio.

In linea di principio, tutte le differenze consistono nelle diverse strategie di proiezione utilizzate, tuttavia un'attenta analisi rivela che esistono molti punti in comune. Comprendere queste similitudini può stimolare nuove idee e sviluppi nel campo, e per queste ragioni, nel seguito, lo spazio dell'ipotesi di ELM sarà reinterpretato in vista dell'apprendimento con le funzioni di similarità. È importante notare che l'interpretazione proposta non è l'unica ammissibile, ma offre un diverso punto di vista sulla capacità di apprendimento del modello rispetto ai lavori precedenti [81, 87, 88].

In generale, tali convergenze diventano evidenti quando il modello ELM utilizza funzioni di attivazione che possono essere reinterpretate come funzioni di similarità [85]. Quindi,  $\phi(\mathbf{x}, \mathbf{r}_j, \chi_j)$  è funzione di attivazione/similarità parametrica che rimappa  $\mathbf{x}$  in  $\mathbb{R}$  usando un punto  $\mathbf{r}_j \in \mathbb{R}^D$  come landmark;  $\chi_j$  è il parametro di configurazione per  $\phi$ . Di conseguenza, lo spazio di ipotesi (5.1) e lo spazio di ipotesi (5.10) possono rappresentare due istanze dello spazio di ipotesi seguente:

$$f(\mathbf{x}) = \sum_{j=1}^N \beta_j \phi(\mathbf{x}, \mathbf{r}_j, \chi_j) \quad (5.11)$$

Dal punto di vista dell'apprendimento con le funzioni di similarità (5.10), i vettori  $\mathbf{r}_j$  in (5.10) incorporano i dati appartenenti alla distribuzione (sconosciuta)  $\mathcal{P}$ . Inoltre,  $\chi_1 = \chi_2 = \dots = \chi_N$ , come spazio di ipotesi (5.10) non ammette configurazioni multiple per la funzione di similarità. Nel caso dello spazio di ipotesi ELM (5.1), al contrario, i landmark  $\mathbf{r}_j$  e i parametri  $\chi_j$  derivano da una selezione casuale. Pertanto, i vettori  $\mathbf{r}_j$  possono incorporare qualsiasi dato appartenente a  $\mathbb{R}^D$  e la funzione di similarità adotta una parametrizzazione diversa per ogni landmark.

Lo scopo è di mostrare, formalmente, che la connessione tra i due paradigmi è stretta. Di seguito, le Sezioni 5.3.1 e 5.3.2, rispettivamente, mostreranno che in principio il modello ELM può ereditare i termini di apprendimento dello spazio di ipotesi (5.1) anche se 1)  $\chi_1 \neq \chi_2 \neq \dots \neq \chi_N$ , e 2)  $\mathbf{r}_j \in \mathbb{R}^D$ .

### 5.3.1 Parametro di forma della funzione di attivazione

Lo spazio di ipotesi stabilito dalla teoria dell'apprendimento con le funzioni di similarità in linea di principio non ammette un parametro di configurazione per la funzione  $K$ . Quindi, nel caso di una funzione di similarità appartenente a una famiglia parametrica, è necessario impostare implicitamente  $\chi_1 = \chi_2 = \dots = \chi_N = \chi$ , dove  $\chi$  è un valore predeterminato. D'altra parte, il modello ELM imposta in modo casuale ogni  $\chi_j$ . Il

teorema 5.3.1 dimostra in effetti che lo strato di mapping della ELM può avere un separatore lineare con un errore  $\epsilon$  basso e un margine  $\gamma$  ampio:

**Teorema 5.3.1.** *Si consideri un insieme composto da  $N$  landmarks  $\{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N\}$  scelti randomicamente da  $\mathcal{T}$  e lo spazio rimappato  $\rho(\mathbf{x}) = \{\phi(\mathbf{x}, \mathbf{r}_1, \chi_1), \phi(\mathbf{x}, \mathbf{r}_2, \chi_2), \dots, \phi(\mathbf{x}, \mathbf{r}_N, \chi_N)\}$ . Sia  $\tilde{\phi}$ :*

$$\tilde{\phi}(\mathbf{x}, \mathbf{r}) = \frac{1}{L}(\delta(\mathbf{r}, \mathbf{r}_1)\phi(\mathbf{x}, \mathbf{r}_1, \chi_1) + \delta(\mathbf{r}, \mathbf{r}_2)\phi(\mathbf{x}, \mathbf{r}_2, \chi_2) + \dots + \delta(\mathbf{r}, \mathbf{r}_N)\phi(\mathbf{x}, \mathbf{r}_N, \chi_N)) \quad (5.12)$$

dove  $\delta(\mathbf{r}, \mathbf{r}_j)$  è la delta di Kronecker. Se  $\tilde{\phi}$  è una buona funzione di similarità  $(\epsilon, \gamma)$  per un generico campione di taglia  $N$  del problema di apprendimento  $\mathbf{P}$ , allora con probabilità  $(1 - \epsilon)$  il mapping  $\rho: \mathcal{X} \rightarrow \mathbb{R}^N$  con  $N = (8 \cdot \log(1/\delta))/\gamma^2$  ha la proprietà che la distribuzione indotta  $\rho(\mathbf{P})$  in  $\mathbb{R}^N$  ha un separatore avente un errore massimo  $(\epsilon + \delta)$  con margine  $\gamma/2$ .

*Dimostrazione.* Seguendo la dimostrazione del Teorema 3 in [75],  $\tilde{\psi}: \mathcal{X} \rightarrow \mathbb{R}^N$  definita come  $\tilde{\psi}(\mathbf{x}) = \frac{\tilde{\rho}(\mathbf{x})}{\sqrt{N}}$ ,  $\tilde{\rho}(\mathbf{x}) = \{\tilde{\phi}(\mathbf{x}, \mathbf{r}_1, \chi_1), \tilde{\phi}(\mathbf{x}, \mathbf{r}_2, \chi_2), \dots, \tilde{\phi}(\mathbf{x}, \mathbf{r}_N, \chi_N)\}$ . Allora con probabilità  $(1 - \delta)$ , la distribuzione indotta  $\tilde{\psi}(\mathbf{P}) \in \mathbb{R}^N$  avrà un separatore con errore massimo  $(\epsilon + \delta)$  e un margine almeno  $\gamma/2$ . Sia  $\tilde{\boldsymbol{\beta}}$  il vettore corrispondente a tale separatore, e si traduca questo in  $\hat{\boldsymbol{\beta}} \in \mathbb{R}^{N \times N}$  sostituendo ciascuna coordinata  $\tilde{\beta}_i$  con gli  $N$  valori  $(\frac{1}{N}\tilde{\beta}_i, \frac{1}{N}\tilde{\beta}_i, \dots, \frac{1}{N}\tilde{\beta}_i)$ . Dato che  $\|\hat{\boldsymbol{\beta}}\| = \|\tilde{\boldsymbol{\beta}}\|$  allora il margine nello spazio  $\mathbb{R}^{N \times N}$  è  $\gamma/2$ . Per costruzione  $\rho(\mathbf{x}) = \tilde{\rho}(\mathbf{x})$ , allora lo stesso separatore lineare è valido per il problema di apprendimento  $\mathbf{P}$ .  $\square$

Il Teorema 5.3.1 afferma, senza alcuna perdita di generalità, che  $\mathbf{r} \in \mathcal{T}$ . La Sezione 5.3.2 mostrerà che quando si campiona  $\mathbf{r}$  in  $\mathbb{R}^D$  è necessario aumentare adeguatamente il numero di landmark  $N$ . È interessante notare che questo teorema mostra come il modello ELM possa anche sfruttare la nozione di "buona" funzione di similarità. Nello spazio di ipotesi (5.10) l'assunto è che tale funzione di similarità sia "buona" su tutto il dominio di input. D'altra parte, il modello ELM sfrutta una funzione di similarità che diventa "buona" adattando la sua configurazione al landmark. Infine, vale la pena sottolineare che il modello ELM può anche adottare livelli di mapping in cui  $\Phi_1 \neq \Phi_2 \neq \dots \neq \Phi_N$ , ovvero dove ogni neurone sfrutta una specifica funzione di attivazione/similarità. Il Teorema 5.3.1 può essere facilmente esteso a questa configurazione.

### 5.3.2 Campionamento dei landmark

La teoria dell'apprendimento con le funzioni di similarità presuppone che i landmark  $\mathbf{r}_j$  debbano trovarsi all'interno del dominio di input  $\mathcal{X} \subseteq \mathbb{R}^D$ . Di conseguenza, occorrono  $N = (8 \cdot \log(1/\delta))/\gamma^2$  landmark per ottenere un separatore nello spazio rimappato con un errore inferiore a  $\epsilon + \epsilon^*$  e con un margine maggiore di  $\gamma/2$ . La strategia ELM, al contrario, estende il dominio ammissibile per i punti di riferimento a tutto  $\mathbb{R}^D$ . Formalmente, la probabilità di ottenere un punto di riferimento appartenente a  $\mathcal{X}$  campionando randomicamente  $\mathbb{R}^D$  può essere modellata come una distribuzione di Bernoulli:

$$P_{\mathbf{r} \in \mathcal{X}} = P(\mathbf{r} \in \mathcal{X} | \mathbf{r} \in \mathbb{R}^D) \leq 1 \quad (5.13)$$

Questo a sua volta significa che dati  $\hat{N}$  pattern random, la probabilità di avere almeno  $N$  landmark ammissibili, nel senso della teoria delle funzioni di similarità, può essere espressa come la cumulativa di una distribuzione binomiale:

$$F(N, \hat{N}, P_{\mathbf{r} \in \mathcal{X}}) = \sum_{i=0}^N \binom{N}{i} (P_{\mathbf{r} \in \mathcal{X}})^i (1 - P_{\mathbf{r} \in \mathcal{X}})^{\hat{N}-i} \quad (5.14)$$

allora con probabilità  $\zeta = 1 - F(N, \hat{N}, \alpha)$  almeno  $N$  campioni appartengono a  $\mathcal{X}$ .

In pratica, ciò significa che applicando la strategia ELM sono necessari almeno  $\hat{N} > N$  landmark per ottenere un separatore nello spazio rimappato con un errore  $\leq \epsilon + \epsilon^*$  e un margine ampio  $\geq \gamma/2$ . Ovviamente, il valore esatto di  $\hat{N}$  è sconosciuto poiché, in generale,  $P_{\mathbf{r} \in \mathcal{X}}$  è sconosciuto. È importante notare che questa è un'analisi del caso peggiore, perché si sta assumendo che i pattern che non appartengono al dominio di input  $\mathcal{X}$  non siano affatto coinvolti nella fase di apprendimento.

### 5.3.3 Riepilogo

Le Sezioni 5.3.1 e 5.3.2 affermano che i due paradigmi di apprendimento non solo condividono alcune somiglianze nella forma dei predittori finali, ma soprattutto possono essere reinterpretati in un unico quadro teorico con alcune importanti differenze nella logica alla base dei modelli.

Il modello ELM estende il dominio di campionamento dei punti di riferimento accettabili da  $\mathcal{T}$  a  $\mathcal{X}$  per ignorare qualsiasi vincolo sulla dimensione  $N$  dello strato di mapping. Questa scelta può rivelarsi particolarmente utile in presenza di set di dati limitati, vale a dire quando  $\mathcal{T}$  potrebbe non coprire adeguatamente la distribuzione (sconosciuta)  $\mathbf{P}$  che caratterizza il problema dell'apprendimento, senza incorrere nell'overfitting. Viceversa,

la disponibilità di serie di dati di grandi dimensioni può compromettere i vantaggi di tale strategia.

La scelta di consentire più configurazioni della nozione di similarità può limitare, in linea di principio, la complessità computazionale del processo di training. Quando si utilizza una funzione di similarità parametrizzata, la *model selection* è l'unico metodo efficace per trovare efficacemente l'iper-parametro tra una varietà di candidati. Ciò comporterebbe chiaramente un sovraccarico computazionale. Con la ELM si aggira questo problema utilizzando la randomizzazione nella configurazione degli iper-parametri.

In breve, l'approccio speculativo adottato dal modello ELM nella scelta di  $\{\mathbf{r}_j, \chi_j\}$  mira a bilanciare lo spazio del campione e l'efficienza computazionale. D'altra parte, la Sezione 5.4 mostra che le scelte casuali dei parametri possono portare a configurazioni inadatte. Si corre il rischio di avere un gran numero di unità di mapping inefficaci, cioè coppie  $\{\mathbf{r}_j, \chi_j\}$  che non supportano una corretta proiezione dei dati secondo la nozione di similarità incorporata.

## 5.4 Analisi Geometrica

La complessità computazionale di una SLFN è direttamente correlata al numero di neuroni  $N$ . In effetti,  $N$  influenza sia il tempo di addestramento che il costo computazionale del predittore. Le procedure che sfruttano la randomicità privilegiano chiaramente una fase di training veloce a scapito di una minore precisione del predittore finale, quando  $N$  è fisso. Le strategie di *pruning* o di selezione [89–92], computazionalmente onerose, per il setup dello strato hidden sono l'unica opzione disponibile quando l'obiettivo è il miglior compromesso tra il numero di neuroni e la capacità di generalizzazione dei predittori che derivano da modelli basati sulla randomizzazione. D'altro canto, un'attenta gestione della strategia di campionamento può in gran parte migliorare le prestazioni del predittore con un numero modesto di neuroni, poiché può aiutare ad evitare configurazioni patologiche. Ad esempio, i neuroni che forniscono sempre lo stesso output indipendentemente dall'input sono inutili.

Nelle due sezioni seguenti, viene presentata una serie di considerazioni sul ruolo svolto dai parametri dello strato hidden per diverse funzioni di attivazione. Nella Sezione 5.4.1 viene discusso il ruolo svolto da diversi parametri nelle funzioni di attivazione classiche, nella Sezione 5.4.2 vengono presentate le peculiarità della funzione soglia.

### 5.4.1 Funzioni di attivazione generiche

Sia  $g$  una funzione di trasferimento monotona decrescente, allora si ha:

$$\phi(\mathbf{x}, \mathbf{r}, \chi) = g(I(M(\mathbf{x}, \mathbf{r}), \chi)) \quad (5.15)$$

La (5.15) mostra che la funzione di attivazione/similarità applica la funzione di trasferimento  $g$  su un input  $I(M(\mathbf{x}, \mathbf{r}), \chi)$ . In questa notazione  $M$  indica la metrica che processa il dato  $\mathbf{x}$  e il landmark  $\mathbf{r}$ , mentre l'input  $I$  deriva dall'azione del parametro di forma  $\chi$  sulla metrica. La Tabella 5.1 fornisce alcuni esempi di funzioni di attivazione che soddisfano i teoremi sulla capacità di approssimazione universale di ELM [63]. Tutte le funzioni presentate nella tabella Tabella 5.1 possono essere convertite in una funzione monotona decrescente con un'inversione di segno dove necessario.

TABELLA 5.1: Funzioni di attivazione classiche

Funzione di attivazione	$\phi(I)$
<i>Triangular Basis Function</i> (TBF)	$\begin{cases} 1 & I = 0 \\ -I & 0 < I < 1 \\ 0 & I \geq 1 \end{cases}$
<i>Radial Basis Function</i> (RBF)	$= e^{-I^2}$
<i>MultiQuadric</i> (MQ)	$= \sqrt{I^2}$
<i>Inverse MultiQuadric</i> (IMQ)	$= \frac{1}{\sqrt{1+I^2}}$
<i>Sigmoid</i> (SGM)	$= \frac{1}{1+e^{-I}}$
<i>Threshold</i> (THR)	$\begin{cases} 1 & I > 0 \\ -1 & I \leq 0 \end{cases}$

La seguente proprietà indica un attributo di base della classe delle funzioni di similarità monotone (5.15):

**Proprietà 1.** *Il posizionamento relativo nel dominio di similarità di due campioni  $\mathbf{x}_1, \mathbf{x}_2$  rispetto a  $\mathbf{r}$  è indipendente da  $\chi$  e  $g$ .*

Ciò significa che, dato un landmark  $\mathbf{r}$  e una coppia di campioni  $\{\mathbf{x}_1, \mathbf{x}_2\}$  tali che  $M(\mathbf{x}_1, \mathbf{r}) < M(\mathbf{x}_2, \mathbf{r})$ , la seguente proprietà vale per qualsiasi valore positivo di  $\chi$ :

$$g(I(M(\mathbf{x}_1, \mathbf{r}), \chi)) \geq g(I(M(\mathbf{x}_2, \mathbf{r}), \chi)) \quad (5.16)$$

Per capire le conseguenze della Proprietà 1, sia  $\nu_C^-$  la distanza  $M$  tra  $\mathbf{r}$  e il campione più vicino della classe '-1', e  $\nu_O^-$  può essere la distanza  $M$  tra  $\mathbf{r}$  e il campione più lontano della classe '-1'. Allo stesso modo,  $\nu_C^+$  e  $\nu_O^+$  siano le quantità corrispondenti per la

classe '+1'. Quindi, un prerequisito per evitare la sovrapposizione di classi nel dominio di similarità è:

$$\nu_C^- < \nu_O^- < \nu_C^+ < \nu_O^+ \quad (5.17)$$

o, in modo equivalente:

$$\nu_C^+ < \nu_O^+ < \nu_C^- < \nu_O^- \quad (5.18)$$

In pratica, il prerequisito per ottenere la configurazione di classi completamente separate nel dominio di similarità è utilizzare un landmark  $\mathbf{r}$  che soddisfi le condizioni di cui sopra. Infatti,  $g$  e  $\chi$  non possono influenzare l'ordine relativo dei campioni nel dominio di similarità.

Un secondo attributo della classe delle funzioni di similarità (5.15) è formalizzato dalla seguente proprietà:

**Proprietà 2.** *La similarità tra qualsiasi campione  $\mathbf{x}$  e  $\mathbf{r}$  è una funzione sia di  $f$  che di  $\chi$ .*

Si scelga TBF come funzione di attivazione/similarità con  $I(M(\mathbf{x}, \mathbf{r}), \chi) = \chi \|\mathbf{x} - \mathbf{r}\|^2$ . La distanza euclidea valuta la dissimilarità tra un campione di input,  $\mathbf{x}$  e il campione di riferimento (cioè il landmark)  $\mathbf{r}$ . La forma specifica della funzione di trasferimento TBF, converte le dissimilarità in similarità. Pertanto, quando  $\mathbf{x} = \mathbf{r}$ , si ha la massima similarità. D'altro canto, se  $\mathbf{x} \neq \mathbf{r}$ , il grado di similarità non può diminuire linearmente all'aumentare della distanza euclidea.

La Fig. 5.1 illustra l'azione di proiezione dei dati eseguita da questa funzione su un caso di studio semplice.

La Fig. 5.1a mostra i campioni che si trovano su uno spazio bidimensionale, in cui le classi sono rappresentate da motivi triangolari ('-1') e quadrati ('+1'). Il simbolo del diamante segna la posizione del landmark  $\mathbf{r}$ . La Fig. 5.1b presenta l'azione di mapping eseguita dalla TBF, che attribuisce somiglianza nulla a qualsiasi campione che si trova a una distanza superiore a  $1/\chi$  dal landmark. La figura mostra anche le quattro quantità di base definite sopra, cioè  $\nu_C^+$ ,  $\nu_O^+$ ,  $\nu_C^-$  e  $\nu_O^-$  e presenta la configurazione migliore: il landmark è stato selezionato in modo che un margine apprezzabile separi i  $Z^+$  campioni della classe '+1' dagli  $Z^-$  campioni della classe '-1'. Inoltre, la funzione di trasferimento  $g$  conserva (o addirittura enfatizza) tale margine. Questo significa che tale configurazione porterebbe ad una buona funzione di similarità  $(0, \gamma)$  forte.

La Fig. 5.1c illustra un controesempio. Se  $\chi$  assume un valore tale che  $1/\chi < \nu_O^+$ , una parte dei campioni della classe '+1' diventa indistinguibile, nel dominio di similarità, dai campioni della classe '-1'. Questa situazione può verificarsi anche in presenza di un



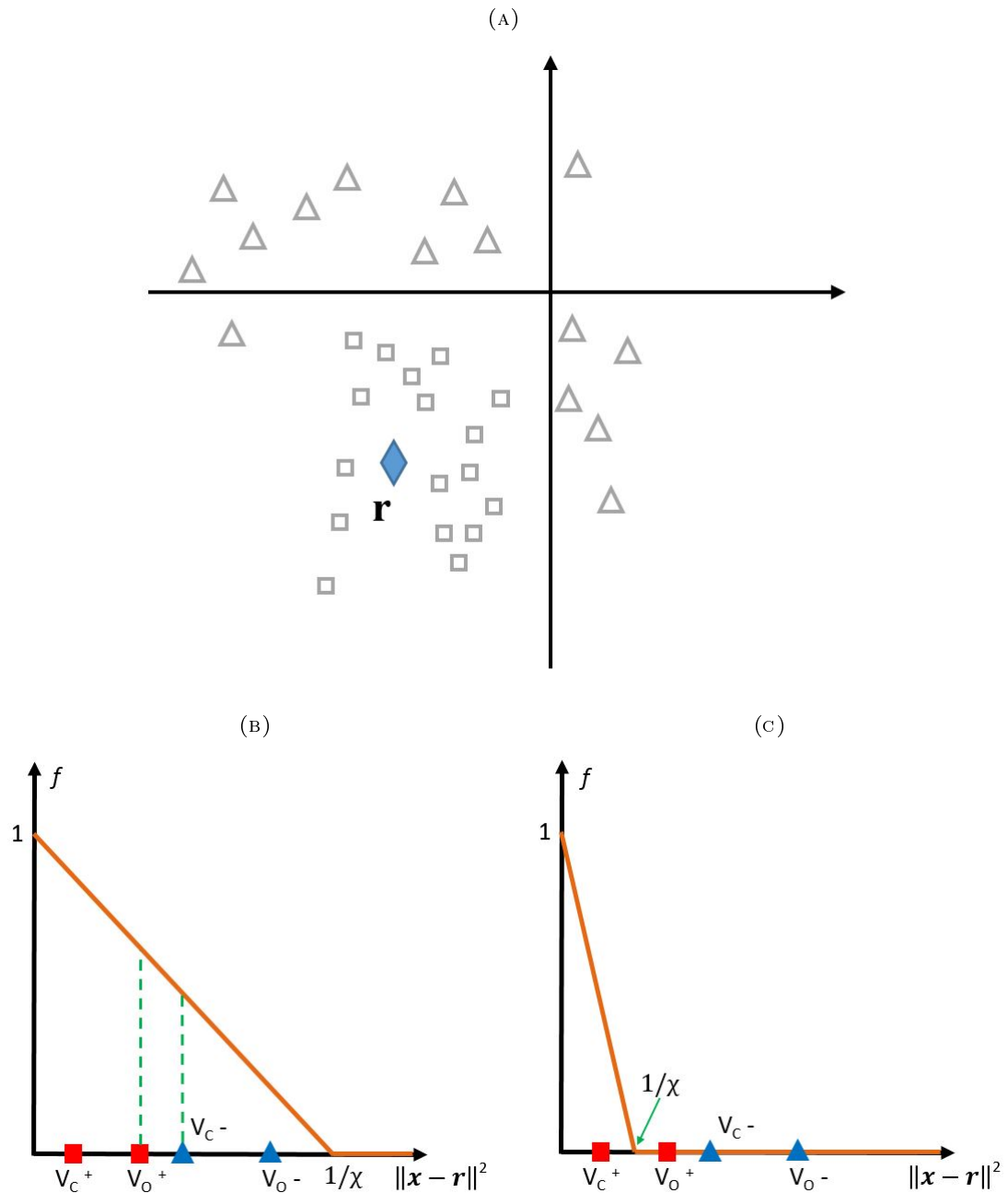


FIGURA 5.1: Esempio di dati rimappati con la TBF: (a) spazio originale di input; (b) dati rimappati con un valore appropriato di  $r$ ; (c) dati rimappati con un valore non appropriato di  $r$

ampio margine tra le due classi rispetto a  $\mathbf{r}$ . Il caso peggiore si verifica quando si imposta  $1/\chi < \nu_C^+$ : tutti i campioni diventano identici dopo il remapping, poiché la similarità tra  $\mathbf{r}$  e qualsiasi campione diventa sempre nulla.

La discussione di cui sopra si applica in generale a qualsiasi funzione di attivazione descritta dall'equazione (5.15). La coppia  $\{g, \chi\}$  sceglie una similarità specifica basata sul mapping  $M(\mathbf{x}, \mathbf{r})$ , il cui comportamento e le cui proprietà derivano dai contributi combinati di due fattori principali:

- innanzitutto, grazie alla natura monotona di  $g$  (come per la (5.15)) e alla mappatura radiale, l'ordinamento non è alterato dalla nozione di similarità. In termini di  $(\epsilon, \gamma)$ , il punto di riferimento  $\mathbf{r}$  stabilisce il livello di sovrapposizione tra le due classi e quindi imposta il valore minimo raggiungibile da  $\epsilon$ .
- In secondo luogo, la metrica di similarità definita da  $\{g, \chi\}$  determina il divario tra un campione e i suoi punti vicini.

### 5.4.2 Funzione soglia e prodotto scalare

Tra le funzioni di attivazione esistenti, la soglia è probabilmente la più conveniente se l'obiettivo è quello di un'implementazione digitale. D'altra parte, questa funzione non può essere scelta quando il processo di training prevede l'uso dell'algoritmo BP perché non è differenziabile. Nella letteratura sono stati proposti molti approcci approssimati [93, 94] inducendo un degradamento della performance. È interessante notare che Huang et al. [95] hanno dimostrato che la teoria ELM vale anche per le funzioni di attivazione *hard-limit* che estendono di conseguenza la capacità di approssimazione universale alle reti direttamente addestrate con la funzione soglia.

In generale, la funzione di soglia può essere considerata come un caso degenero della sigmoide in cui la pendenza della parte che non satura è infinita. In questa sezione vengono analizzati i dettagli sulla distribuzione geometrica del prodotto scalare e sulla sua interazione con la funzione sigmoide. Come risultato principale, è dimostrato che le reti basate sulla funzione sigmoide contengono una percentuale considerevole di neuroni che agisce come unità di soglia.

La funzione di attivazione del prodotto scalare può essere riscritta come:

$$I(M(\mathbf{x}, \mathbf{r}), \chi) = \mathbf{x}^t \mathbf{r} + \chi = \mathbf{x}^t \mathbf{r} + \mathbf{l}^t \mathbf{r} = \|\mathbf{r}\|(\mathbf{x}^t \hat{\mathbf{r}} + \mathbf{l}^t \hat{\mathbf{r}}) = \|\mathbf{r}\|(\mathbf{x}^t \hat{\mathbf{r}} - r) \quad (5.19)$$

dove  $\hat{\mathbf{r}}$  è un vettore unitario e  $r \in \mathbb{R}$  è la proiezione di un punto generico  $\mathbf{l}$  su  $\mathbf{r}$ . Di conseguenza,  $I(M(\mathbf{x}, \mathbf{r}), \chi)$  può essere reinterpretato come  $I(\mathbf{x}, \mathbf{r}, r)$ . Se si considera  $\mathbf{x} \in [0, 1]^D$  senza alcuna perdita di generalità, la proiezione scalare di  $\mathbf{x}$  su  $\hat{\mathbf{r}}$  si trova in un intervallo che dipende dalla dimensionalità  $D$  dello spazio di input; ad esempio,  $\mathbf{x}^t \hat{\mathbf{r}} \in [-\sqrt{D}, \sqrt{D}]$ .

Secondo l'equazione (5.19), in una funzione sigmoide,  $SGM(I) \in [0, 1]$ , la porzione di elevata pendenza della curva giace approssimativamente nel range:

$$\left[ r - \frac{5}{\|\mathbf{r}\|}, r + \frac{5}{\|\mathbf{r}\|} \right] \quad (5.20)$$

Dal momento che non sarebbe conveniente centrare la sigmoide fuori da questo intervallo, si può scegliere,  $r \in [-\sqrt{D}, \sqrt{D}]$ .

La (5.20) dimostra che  $\|\mathbf{r}\|$ , ha il ruolo di variare la forma di SGM(I). Per valutare l'effetto di  $\|\mathbf{r}\|$ , si consideri  $\epsilon$  come il rapporto tra la parte non saturante della funzione sigmoide (5.19) e l'intervallo  $(2\sqrt{D})$  in cui si trovano i pattern proiettati.

$$\epsilon \leq \frac{10}{2\sqrt{D}\|\mathbf{r}\|} = \frac{5}{\sqrt{D}\|\mathbf{r}\|} \quad (5.21)$$

Il limite superiore  $\epsilon$ , espresso nella (5.21), vale quando la quantità (5.20) può trovarsi parzialmente al di fuori dell'intervallo  $[-\sqrt{D}, +\sqrt{D}]$  (ad esempio, quando  $r = \pm\sqrt{D}$ ). Ovviamente  $\epsilon = 0$  significa che la funzione sigmoide degenera in una *hard-limiter* (cioè,  $\|\mathbf{r}\| \rightarrow \infty$ ).

Nel modello ELM,  $\mathbf{r}$  è ricavato solitamente da una distribuzione uniforme (ad esempio,  $P(r_d) = U(-1, 1)$ ). Estendendo opportunamente il Teorema del Limite Centrale [96], si può approssimare la distribuzione di  $\|\mathbf{r}\|$  come una distribuzione normale nell'intervallo  $[0, \sqrt{D}]$  il cui valore atteso è  $\mu_{\|\mathbf{r}\|} = \sqrt{\text{Var}(r_d)}\sqrt{D} = 0.57\sqrt{D}$ . L'espressione (5.21) mostra che, durante l'addestramento della ELM standard, 50% delle funzioni di attivazione nello strato hidden hanno  $\epsilon \leq 0.3$  fino a  $D > 30$ . Quando  $D = 100$  si ha  $\epsilon < 0.09$ . Di conseguenza, una ELM che impiega la funzione sigmoide presenta sempre un sottoinsieme di neuroni che in realtà coinvolgono meccanismi di soglia, anche se  $\|\mathbf{r}\| \in [0, \sqrt{D}]$ . L'incidenza di questi ultimi aumenta quando cresce la dimensionalità  $D$  dello spazio di input. Quindi si potrebbe creare una ELM che includa solo funzioni hard-limiter, con lo scopo di ottenere un predittore esplicitamente progettato per implementazioni digitali a basso costo. Tale strategia potrebbe portare a una ridotta capacità di generalizzazione rispetto a una configurazione che si basa su funzioni sigmoide. La discussione di cui sopra suggerisce che questo divario potrebbe non essere sostanziale.

Infine, è importante notare il diverso ruolo giocato dal parametro  $r$  nella soglia e nella sigmoide. Si consideri l'esempio mostrato nella Fig. 5.2a dove viene mostrata l'azione eseguita da una singola unità di remapping avente la forma  $g(\hat{\mathbf{r}} \cdot \mathbf{x} + r)$ . Nell'esempio, i campioni si trovano in uno spazio bidimensionale, in cui le classi sono rappresentate da rombi ('-1') e croci ('+1'). Lo spazio è diviso in due parti da un iperpiano identificato dalla coppia  $\{\hat{\mathbf{r}}, r\}$  che separa perfettamente le due classi. La Fig. 5.2b rappresenta lo stesso esempio ma con un valore diverso per il bias  $r$  dell'iperpiano; in questo caso tutti i campioni giacerebbero sullo stesso lato del separatore lineare. La Fig. 5.2c e la Fig. 5.2d si riferiscono, rispettivamente, all'azione di mapping eseguita dall'iperpiano presentato nella Fig. 5.2a e nella Fig. 5.2b. In entrambe le figure, sull'asse  $y$ , è mostrato il mapping ottenuto utilizzando la funzione sigmoide (linea blu) e la funzione segno (linea

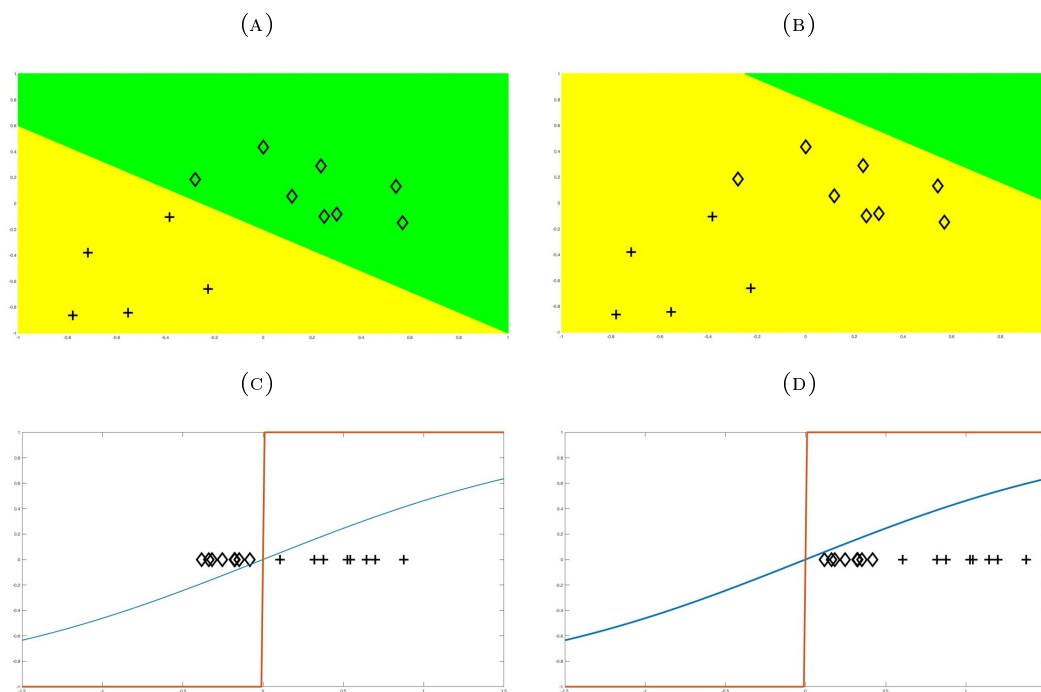


FIGURA 5.2: Un esempio di dati nello spazio di input 2D: (a) separatore lineare con errore  $\epsilon = 0$ ; (b) stesso separatore lineare ma diverso bias  $r$ ; (c) attivazione ottenuta utilizzando il separatore lineare (a) e il segno come attivazione (linea arancione) o la sigmoide riscalata (linea blu) (d) lo stesso formato della figura (c) ma il separatore lineare della figura (b)

arancione). I dati rimappati sono linearmente separabili con errore nullo per entrambe le non linearità nella Fig. 5.2c. Invece, nella Fig. 5.2d, il nuovo spazio ottenuto usando la funzione SGM mantiene la proprietà di separabilità lineare con errore nullo (sempre considerando l'asse  $y$ ), mentre la proiezione ottenuta usando la funzione di soglia non è separabile a causa della configurazione del bias  $r$  nell'iperpiano della Fig. 5.2b.

Questo semplice esempio ha chiarito le diverse implicazioni del parametro  $r$  con diverse non linearità e sottolinea il fatto che una selezione casuale di questo parametro può facilmente portare a un'unità di mapping inutile (i neuroni) anche quando il parametro  $\hat{r}$  è ottimale.

## 5.5 Algoritmi per un mapping efficiente

Questo capitolo fornisce una serie di algoritmi per una configurazione efficiente dei parametri dello strato hidden. Tutte le soluzioni proposte condividono l'obiettivo comune di bilanciare in modo efficiente il trade-off tra il costo computazionale dell'eventuale predittore e la capacità di generalizzazione, senza un significativo incremento del costo della fase di training. Per raggiungere questo obiettivo, le strategie proposte si focalizzano

sul parametro  $\chi$ , in questo modo si evita qualsiasi ricalcolo delle operazioni vettoriali relativo a  $M(\mathbf{r}, \mathbf{x})$ .

### 5.5.1 Algoritmo per il campionamento del fattore di forma

La Sezione 5.4 illustra chiaramente il diverso ruolo giocato da  $\{\mathbf{r}, \chi\}$  se combinato con differenti tipi di non linearità. In particolare, la sottosezione 5.4.1 intensifica il concetto che la selezione di un parametro non coerente  $\chi$  potrebbe portare a situazioni patologiche. Ad esempio, un'unità di mapping che comprime tutti i pattern  $\mathbf{x} \in \mathcal{X}$  in un singolo punto  $p \in \mathbb{R}$  non è utile in termini di apprendimento. L'obiettivo finale del livello di mappatura è di proiettare i campioni di input in un nuovo spazio in cui i modelli positivi e negativi sono ben separabili. In un semplice spazio unidimensionale, se  $\mu_+$  indica il baricentro dei modelli positivi in  $\mathbb{R}$  e  $\mu_-$  il corrispondente baricentro dei modelli negativi, dovrebbe valere la seguente condizione:

$$\mu_+^{(j)} \neq \mu_-^{(j)}, \quad j = 1, \dots, N \quad (5.22)$$

dove  $N$  è il numero di unità di mapping, cioè il numero di landmark nello strato di mapping. Idealmente, la migliore unità di proiezione garantisce chiaramente un ampio margine tra  $\mu^+$  e  $\mu^-$  con una piccola varianza intra-classe.

In generale, selezionando casualmente i valori da assegnare a  $\chi_j$  si potrebbe finire in configurazioni imperfette, indipendentemente dalla scelta specifica della funzione di similarità/attivazione. Nel modello ELM, questo significa che un sottoinsieme di neuroni può essere inutile in termini di apprendimento. Pertanto, potrebbe essere difficile garantire un buon trade-off tra le prestazioni di generalizzazione e la complessità computazionale del predittore finale, che diventa critico quando si prende in considerazione l'implementazione del sistema di classificazione sui dispositivi elettronici.

È necessario confrontare la strategia convenzionale applicata da ELM per l'impostazione di  $\chi$  con una strategia che garantisca la conformità con il vincolo (5.22). L'obiettivo di questa nuova strategia è controllare la configurazione dei parametri,  $\chi$ , che è assegnata a un'unità di mapping  $j$ . Una configurazione è considerata "imperfetta" quando:

$$\left| \mu_+^{(j)} - \mu_-^{(j)} \right| < \tau \quad (5.23)$$

dove  $\tau$  è un valore di soglia che imposta la tolleranza ammessa quando si viola il vincolo. L'algoritmo convalida solo il  $\chi_j$ , poiché la funzione di attivazione/similarità  $\phi$  e il landmark  $\mathbf{r}_j$  assegnati all'unità  $j$ -esima non variano.

L'Algoritmo 3 delinea la procedura associata, che in realtà riscrive il passaggio 0 (*Initialize*) dell'Algoritmo 1. Il passo *Initialize* riceve un set di training etichettato  $\mathcal{T} = \{(\mathbf{x}, y)_i; i = 1, \dots, Z\}$ , una funzione di attivazione/similarità  $\phi$ , un intervallo di valori ammissibile per  $\chi$  e un valore  $N$ , per definire la dimensionalità del livello di mappatura.

Per ogni unità di mapping, l'algoritmo genera il landmark corrispondente utilizzando la strategia ELM convenzionale. Quindi, la routine per l'impostazione del  $\chi_j$  procede in due passaggi: 1) viene scelto un valore casuale all'interno dell'intervallo di input e 2) il valore viene convalidato. La routine si interrompe quando il passaggio 2) viene completato correttamente. Alla fine, l'algoritmo restituisce i valori del parametro libero  $\chi_j$ , per ogni unità di mappatura  $j$ -esima (neurone  $j$ -esimo). Possono essere applicati criteri diversi per valutare l'efficacia di un'unità di mappatura nella separazione di campioni positivi da quelli negativi. Si evita il "punto morto" semplicemente dividendo per due il valore della soglia  $\tau$  ogni volta che una configurazione viene rifiutata.

Nell'Algoritmo 3, i due insiemi  $\mathcal{X}_+$  e  $\mathcal{X}_-$  sono caratterizzati dai valori medi, ma possono essere utilizzate altre grandezze come mediana o percentili di ordine  $p$ . Inoltre, l'algoritmo non considera l'interazione tra le diverse unità di mapping e non considera la varianza inter e intra classe. La logica dietro queste scelte è di introdurre un sovraccarico trascurabile nel costo computazionale della fase di addestramento. In termini di complessità computazionale l'algoritmo sembra ancora meno efficiente della procedura ELM originale, dal momento che il numero di tentativi richiesti per trovare un setup valido di  $\chi_j$  per ogni unità di mappatura non è prevedibile. D'altra parte, la strategia dell'Algoritmo 3 può offrire vantaggi significativi per il compromesso tra la capacità di generalizzazione e la dimensionalità dello strato hidden.

### 5.5.1.1 Risultati sperimentali

La sezione sperimentale mira a valutare la capacità dell'Algoritmo 3 di migliorare le prestazioni del modello ELM in termini di accuratezza durante la classificazione. L'obiettivo è verificare che, applicando l'Algoritmo 3, si possa raggiungere un livello di mappatura che supporti meglio la ricerca di una buona funzione di similarità  $(\epsilon, \gamma)$  coerente. Allo scopo di valutare in modo rigoroso tale aspetto, sono stati coinvolti cinque benchmark diversi nella fase sperimentale [97]: *Ionosphere*, *Glass Identification*, *Statlog Landsat Satellite*, *Coverttype* e *Cod-RNA*. Ogni sessione è stata pensata per fornire un confronto equo tra le prestazioni di generalizzazione dei due modelli ELM: quello che applica la strategia convenzionale nella configurazione dei parametri liberi e quella che sfrutta l'Algoritmo 3. Pertanto, in ogni esperimento, invece di utilizzare le tecniche di model selection standard

---

**Algoritmo 3** Strategia di training migliorata usando la ricerca random.
 

---

**Input**

- Un dataset etichettato  $\mathcal{T} = \{(\mathbf{x}, y)_i; i = 1, \dots, Z\}$ .
- Un numero di neuroni  $N$ .
- Un range di valore ammissibili per  $\chi : [\chi_{inf}, \chi_{sup}]$ .
- Un valore di soglia  $\tau$ .

**0. Inizializzazione**

- **a.** Si generi un insieme di landmark random  $\mathbf{r}_j, j = 1, \dots, N$ :
 

```

for  $j = 1$  to  $N$  do
  for  $d = 1$  to  $D$  do
     $\mathbf{r}_{j,d} = rand(-1;1)$ 
  end for
end for

```
- **b.** Per ogni neurone, si imposti  $\chi_j$  utilizzando i seguenti passi:
 

```

 $\mathcal{X}_+ = \{\}, \mathcal{X}_- = \{\}$ 
 $ok = 0$ 
while  $ok == 0$  do
   $\chi_j = rand(\chi_{inf}, \chi_{sup})$ 
  for  $i = 1$  to  $Z$  do
     $a_i = \phi(\mathbf{x}_i, \mathbf{r}_j, \chi_j)$ 
    if  $y_i == 1$  then
       $a_i \rightarrow \mathcal{X}_+$ 
    else
       $a_i \rightarrow \mathcal{X}_-$ 
    end if
  end for
   $\mu_+ = mean(\mathcal{X}_+)$ 
   $\mu_- = mean(\mathcal{X}_-)$ 
  if  $|\mu_+ - \mu_-| \geq \tau$  then
     $ok = 1$ 
  else
     $\tau = \tau/2$ 
  end if
end while

```

**1. Mapping**

Si rimappino tutti i campioni  $\mathbf{x} \in \mathcal{T}$  usando la seguente funzione di mapping:

$$\boldsymbol{\psi}(\mathbf{x}) = \{\phi(\mathbf{x}, \mathbf{r}_1, \chi_1), \dots, \phi(\mathbf{x}, \mathbf{r}_N, \chi_N)\}$$

**2. Apprendimento**

Si addestri un separatore lineare nello spazio  $\boldsymbol{\psi} : \mathcal{X} \rightarrow \mathbb{R}^N$

---

[62], si definisce una configurazione comune per entrambe le configurazioni che stabiliscano gli intervalli ammissibili di  $\lambda$  (cioè il parametro di regolarizzazione), e della dimensionalità  $N$  dello spazio di proiezione dei dati (cioè il numero di landmark/neuroni). Le configurazioni sono:  $\lambda = \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10^1, 10^2, 10^3, 10^4, 10^5, 10^6\}$ ,  $N = \{50, 100, 200, 500, 1000\}$ .

**Dataset Ionosphere** Il set di dati Ionosphere include un totale di 351 pattern che si trovano in uno spazio 34-dimensionale; il dataset originale è abbastanza sbilanciato, in quanto una delle due classi fornisce solo 126 pattern su 351. Nella progettazione sperimentale presente, sia il set di training che il set di test includono 50 pattern per classe; tutte le 34 *features* sono normalizzate nell'intervallo  $[-1, 1]$ . Nella sessione sono state impiegate tre funzioni di attivazione/similarità diverse: sigmoide, RBF e la multiquadrica. Pertanto, i parametri liberi sono il bias per la sigmoide e il fattore di *spread* per le altre due. L'implementazione migliorata della ELM ha sfruttato l'Algoritmo 3 con soglia  $\tau = 0.5$ . Quindi, un'unità di mappatura è considerata efficace solo quando il divario tra il baricentro  $\mu_+$  dei campioni rimappati positivi e il baricentro  $\mu_-$  di quelli negativi è maggiore di 0.5; si noti che i pattern rimappati si trovano in uno spazio con range  $[-1, 1]$ .

La Fig. 5.3 fornisce i risultati dei tre esperimenti. La Fig. 5.3a si riferisce a quello in cui le unità di mapping implementano la funzione sigmoide; l'asse  $x$  indica il numero di neuroni  $N$ , mentre l'asse  $y$  fornisce l'errore di classificazione (espresso in percentuale rispetto alla dimensione del set di test). Il grafico confronta le prestazioni della ELM standard (asterischi come marker) con le prestazioni della ELM migliorata (cerchi come marker). Per ogni  $N$  viene valutata la performance di un predittore in base alla configurazione (cioè  $\lambda$ ) che porta al miglior errore di classificazione medio sul set di test; il valore medio è calcolato su 50 serie, cioè 50 diverse randomizzazioni del livello di mappatura. Il grafico fornisce anche l'intervallo di confidenza  $\pm\sigma$ . È stato applicato lo stesso formato alla Fig. 5.3b, che si riferisce all'esperimento in cui le unità di mapping implementano la RBF, e alla Fig. 5.3c, che si riferisce a quello in cui i neuroni realizzano la funzione multiquadrica.

Complessivamente, i grafici mostrano che la ELM migliorata può, nella maggior parte dei casi, presentare risultati più performanti rispetto alla ELM standard in termini di classificazione. Infatti, il miglioramento varia in base alla funzione di attivazione/similarità e del numero di neuroni. A questo proposito, la Tabella 5.2 riporta, per ciascuna funzione di attivazione/similarità, il miglior predittore, cioè quello che ha ottenuto l'errore di classificazione più basso. Ogni riga indica l'impostazione del predittore (standard/migliorata), l'errore di classificazione e il numero di neuroni. Inoltre, l'ultima riga della tabella riporta la performance ottenuta da una *Support Vector Machine* (SVM) [62] sullo



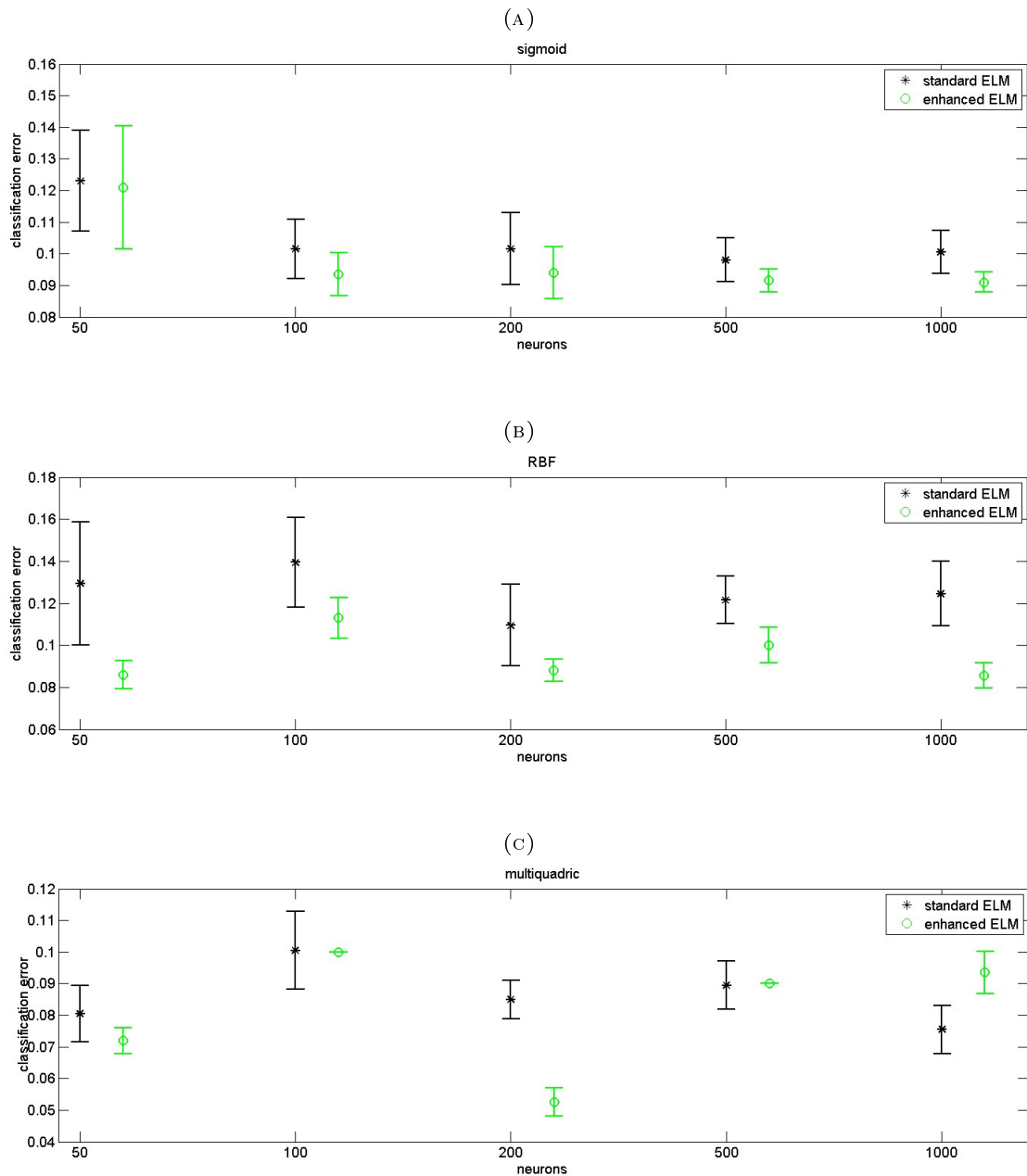


FIGURA 5.3: Risultati degli esperimenti eseguiti sul dataset Ionosphere: a) sigmoide; b) RBF; c) multiquadrica

stesso problema. I risultati si riferiscono a un'implementazione basata sul kernel RBF: la tabella fornisce l'errore di classificazione ottenuto dopo la model selection [98, 99] insieme alla configurazione corrispondente  $(C, \sigma)$ .

Per la funzione sigmoide, nonostante l'accuratezza migliore si abbia con  $N = 1000$ , nella Fig. 5.3c si osserva che già con  $N = 100$  si ottengono ottimi risultati per la ELM migliorata.

TABELLA 5.2: Confronto tra la ELM e SVM per il dataset Ionosphere.

Funzione	Predittore Migliore	Errore di Classificazione	Configurazione
Sigmoide	ELM migliorata	9.1	$N = 1000$
RBF	ELM migliorata	8.6	$N = 50$
Multiquadratica	ELM migliorata	5.2	$N = 200$
SVM		7.0	$(C, \sigma) = (10^{-1}, 1)$

**Dataset Glass** Il dataset *Glass Identification* include 214 campioni che si trovano in uno spazio a 9 dimensioni. Il benchmark coinvolge un problema multi-classe, in quanto nel set di dati sono rappresentate sei serie differenti; gli esperimenti presentati qui, però, riguardano solo un problema di classificazione binaria. Nel progetto sperimentale proposto, sia il set di training che il set di test includono 30 pattern per classe estratti a caso dal dataset originale. Tutte le 9 features sono state normalizzate nell'intervallo  $[-1, 1]$ . Nella sessione, come in quella precedente, sono state coinvolte tre diverse funzioni di attivazione/similarità: funzione sigmoide, RBF e multiquadratica. L'implementazione migliorata della ELM ha impiegato l'Algoritmo 3 con soglia  $\tau = 0.5$ .

La Fig. 5.4 mostra i risultati dei tre esperimenti. Il formato adottato per i grafici replica quello utilizzato nella Fig. 5.3. Complessivamente le figure mostrano che la ELM migliorata fornisce risultati migliori rispetto alla ELM standard solo in alcuni casi. Tuttavia è interessante notare che, quando si utilizza la funzione sigmoide, la ELM standard ha ottenuto la sua migliore performance con 1000 neuroni (errore di classificazione del 16.7%). Il predittore corrispondente, basato sull'Algoritmo 3, è stato in grado di ottenere lo stesso errore di classificazione utilizzando 100 neuroni.

La Tabella 5.3 riporta il confronto tra i predittori basati sulla ELM e il predittore basato su SVM. La ELM migliorata basata sulla funzione sigmoide ha ottenuto le stesse prestazioni di SVM (errore di classificazione del 16.7%); la ELM migliorata con la funzione multiquadratica ha ottenuto un errore di classificazione abbastanza simile.

TABELLA 5.3: Confronto tra ELM e SVM per il dataset Glass

Funzione	Predittore Migliore	Errore di Classificazione	Configurazione
Sigmoide	ELM migliorata	16.7	$N = 100$
RBF	ELM standard	19.1	$N = 50$
Multiquadratica	ELM migliorata	17.9	$N = 50$
SVM		16.7	$(C, \sigma) = (10^{-1}, 1)$

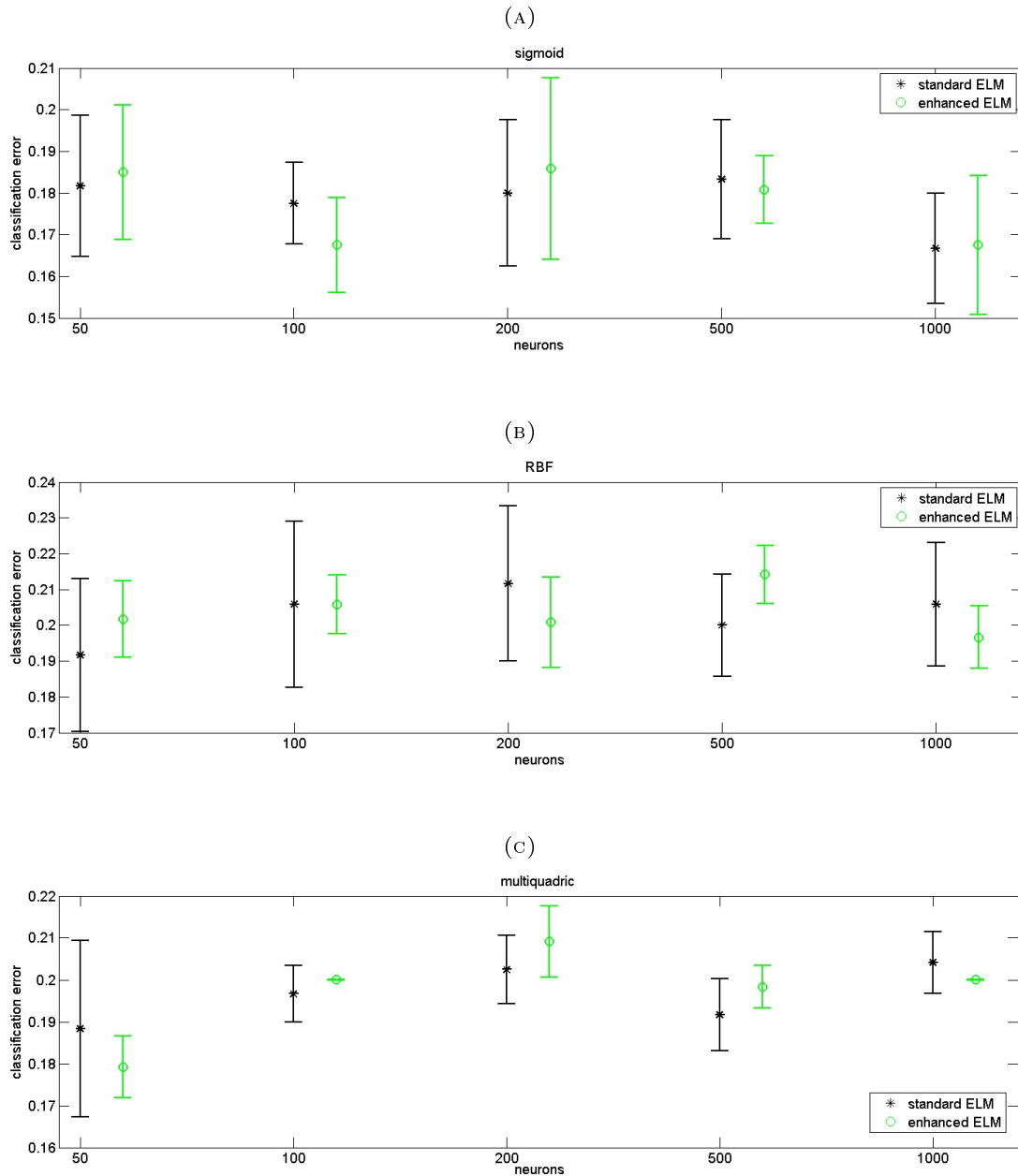


FIGURA 5.4: Risultati degli esperimenti sul dataset Glass: a) sigmoide; b) RBF; c) multiquadrica

**Dataset Landsat** Il dataset Landsat fornisce un set di training costituito da 4435 campioni e uno di test con 2000; i dati sono tratti da uno spazio 36-dimensionale. Il benchmark originale coinvolge un problema multi-classe, ma gli esperimenti presenti ne riguardano solo uno di classificazione binaria: classe 4 contro classe 7. Nel progetto sperimentale proposto, il set di addestramento comprende 300 campioni per classe estratti a caso dal database di training originale; il set di test include 150 campioni per classe estratti casualmente dal database di test originale. Tutte le 36 features sono state normalizzate nell'intervallo  $[-1, 1]$ .

Come visto precedentemente nella sessione sono state coinvolte tre diverse funzioni di attivazione/similarità: funzione sigmoide, RBF e multiquadrica. L'implementazione migliorata della ELM ha impiegato l'Algoritmo 3 con soglia  $\tau = 0.5$ . La Fig. 5.5 fornisce i risultati dei tre esperimenti. Il formato adottato per i grafici replica quello utilizzato nella Fig. 5.3. Le figure evidenziano che nella maggior parte dei casi la ELM migliorata mostra risultati più soddisfacenti rispetto alla ELM standard. In effetti, il divario tra ELM standard e ELM migliorata è significativo, in particolare impiegando la funzione RBF e la multiquadrica.

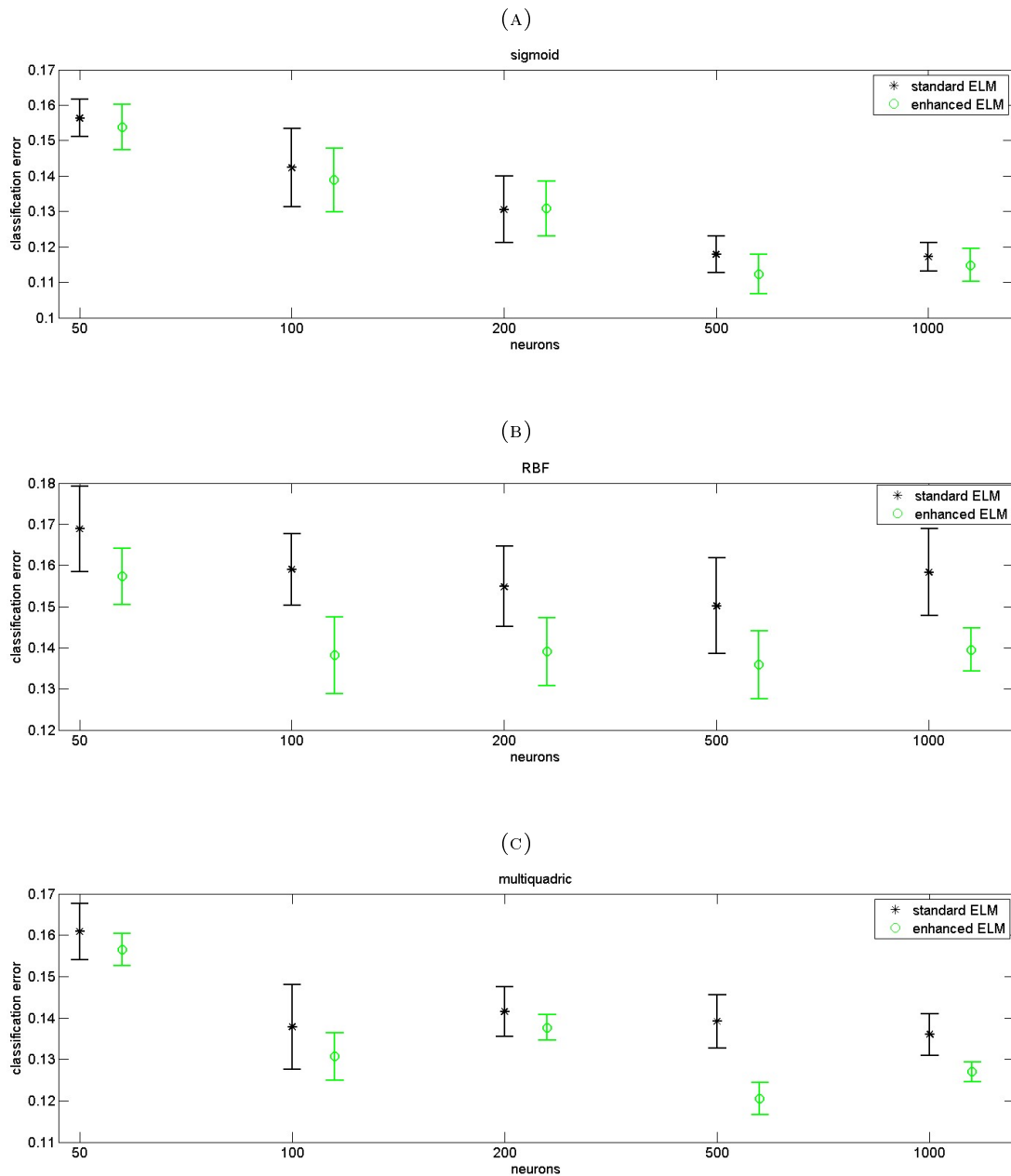


FIGURA 5.5: Risultati degli esperimenti utilizzando il dataset Landsat: a) sigmoide; b) RBF; c) multiquadrica

La Tabella 5.4 fornisce il confronto tra i predittori basati su ELM e quello su SVM. In questo caso, la ELM migliorata non ha raggiunto l'accuratezza di classificazione ottenuto da SVM. Ciononostante, il divario tra il risultato migliore ottenuta dalla ELM (11,2%) e l'errore di classificazione raggiunto da SVM (10,3%) non è grande.

TABELLA 5.4: Confronto tra ELM e SVM per il dataset Landasat.

<b>Funzione</b>	Predittore Migliore	Errore di Classificazione	Configurazione
Sigmoide	ELM Migliorata	11.2	$N = 500$
RBF	ELM Migliorata	13.5	$N = 500$
Multiquadratica	ELM Migliorata	12.0	$N = 500$
SVM		10.3	$(C, \sigma) = (10^{-2}, 1)$

Osservando la Fig. 5.5b si nota che già con  $N = 50$  la ELM migliorata presenta un errore di generalizzazione simile a quello ottenuto con  $N = 500$ .

**Dataset Coverttype** Il dataset Coverttype fornisce un set di training comprendente 59.535 campioni e un set di test che ne include 271.617 aventi 8 dimensioni. Nell'esperimento proposto, sia il set di training che quello di test includono 5000 pattern per classe; tutte le 8 caratteristiche sono state effettivamente normalizzate nell'intervallo  $[-1; 1]$ . Nella sessione sono state coinvolte tre diverse funzioni di attivazione/similarità: funzione sigmoide, RBF e multiquadratica. L'implementazione migliorata della ELM ha impiegato l'Algoritmo 3 con soglia  $\tau = 0.5$ . La Fig. 5.6 mostra i risultati dei tre esperimenti. Il formato adottato per i grafici replica quello utilizzato nella Fig. 5.3. Le figure evidenziano che la ELM migliorata mostra risultati significativamente più soddisfacenti rispetto alla ELM standard solo utilizzando la RBF come funzione di attivazione/similarità. Si ottiene un miglioramento più piccolo quando si adotta la funzione sigmoide.

La Tabella 5.5 riporta il confronto tra i predittori basati su ELM e quello su SVM. In questo caso, la performance raggiunta da SVM (errore di classificazione di 16,8%) supera decisamente la performance ottenuta da ELM.

TABELLA 5.5: Confronto tra ELM e SVM per il dataset Coverttype.

<b>Funzione</b>	Predittore Migliore	Errore di Classificazione	Configurazione
Sigmoide	ELM Migliorata	19.7	$N = 1000$
RBF	ELM Migliorata	20.4	$N = 1000$
Multiquadratica	ELM Migliorata	20.3	$N = 1000$
SVM		16.8	$(C, \sigma) = (10^1, 0.1)$

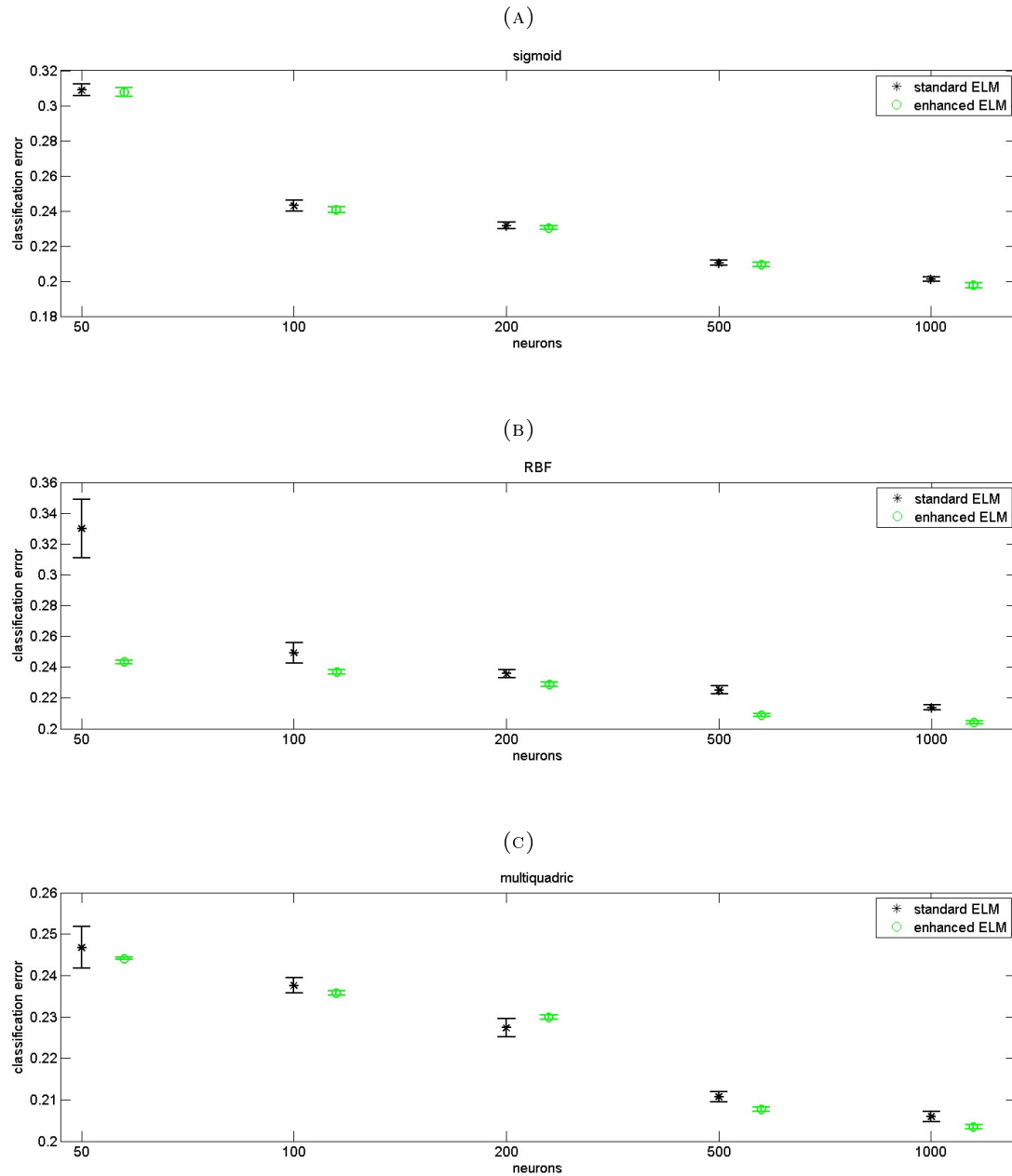


FIGURA 5.6: Risultati degli esperimenti utilizzando il dataset Covertype: a) sigmoide; b) RBF; c) multiquadratica

**Dataset Cod-RNA** Il dataset Cod-RNA fornisce un set di training che comprende 59.535 campioni e un set di test che include 271.617 campioni, estratti da uno spazio a 8 dimensioni. Nell'esperimento proposto, sia il set di training che il set di test includono 5000 pattern per classe; tutte le 8 features sono state normalizzate nell'intervallo  $[-1; 1]$ . Nella sessione sono state coinvolte tre diverse funzioni di attivazione/similarità: funzione sigmoide, RBF e multiquadratica. L'implementazione della ELM migliorata ha impiegato l'Algoritmo 3 con soglia  $\tau = 0,5$ . La Fig. 5.7 illustra i risultati dei tre esperimenti. Il formato adottato per i grafici replica quello utilizzato nella Fig. 5.3. Le figure mostrano

che la ELM migliorata presenta risultati più soddisfacenti rispetto alla ELM standard, in particolare quando si utilizza la RBF come funzione di attivazione/similarità.

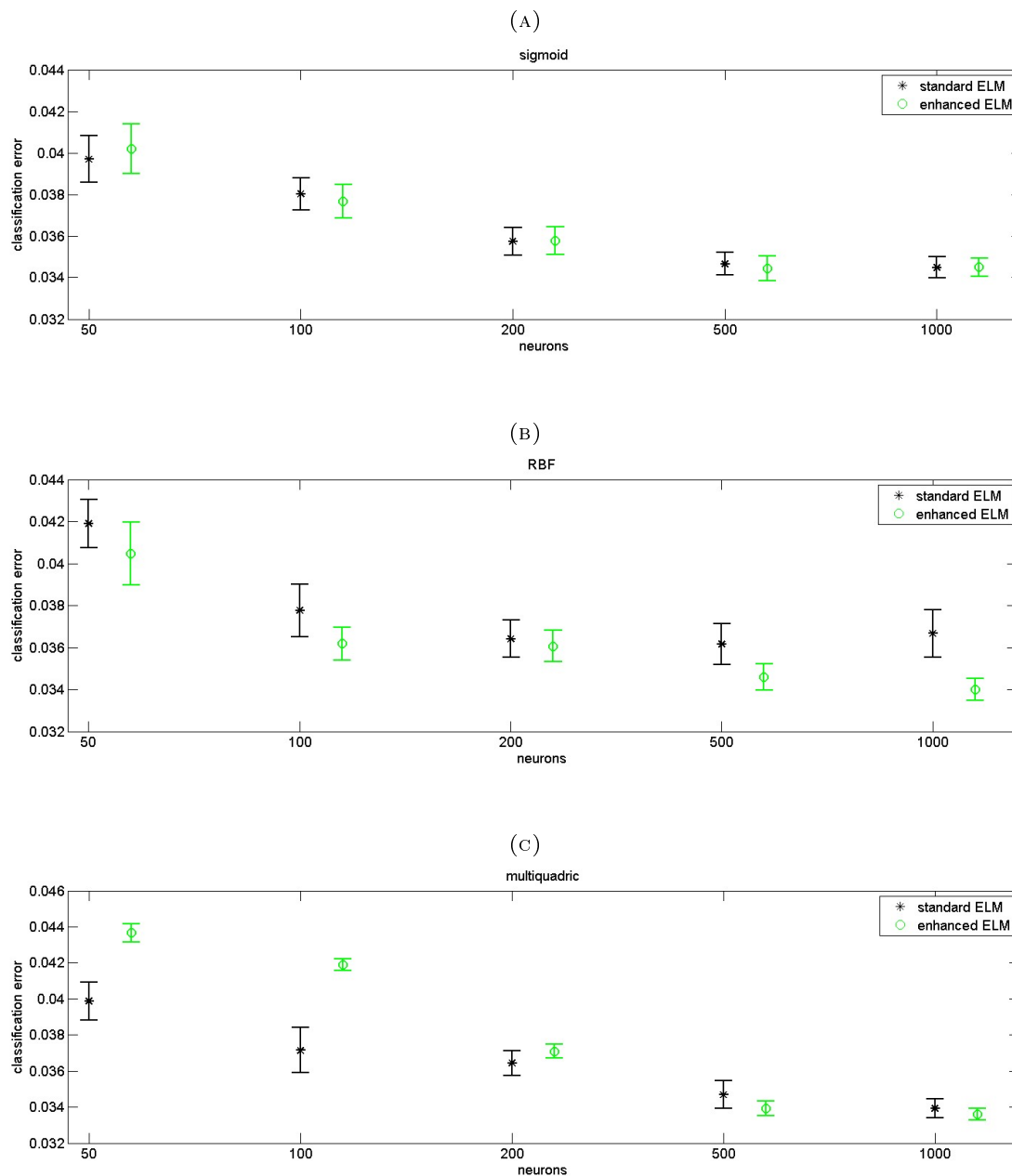


FIGURA 5.7: Risultati degli esperimenti utilizzando il dataset CodRNA: a) sigmoide; b) RBF; c) multiquadratica

La Tabella 5.6 evidenzia il confronto tra i predittori basati su ELM e quello su SVM. I risultati numerici rivelano che i predittori relativi alla ELM migliorata hanno un errore di classificazione più basso rispetto a SVM. Tale risultato conferma che la soluzione proposta può raggiungere risultati molto interessanti.

TABELLA 5.6: Confronto tra ELM e SVM per il dataset Cod-RNA.

Funzione	Predittore Migliore	Errore di Classificazione	Configurazione
Sigmoide	ELM migliorata	3.4	N = 500
RBF	ELM migliorata	3.3	N = 1000
Multiquadratica	ELM migliorata	3.3	N = 1000
SVM		3.7	(C,σ) = (1,1)

### 5.5.1.2 Considerazioni

Le prove empiriche hanno supportato l'algoritmo proposto per il campionamento del fattore di forma che, in generale, consente di ottenere un predittore più efficace rispetto agli approcci convenzionali per il training di ELM, con un incremento minimo del costo della fase di addestramento. In realtà si è osservato che, in alcuni casi, la ELM migliorata è stata in grado raggiungere le prestazioni di un sistema di classificazione come SVM.

## 5.5.2 Algoritmo per il parametro della soglia

Una procedura di apprendimento di una ELM basata su funzioni hard-limiter inizia impostando casualmente le coppie  $\{\hat{\mathbf{r}}, r\}$ , come in (5.19). Tuttavia, considerando il risultato dell'analisi proposto nella Sezione 5.4.2, le peculiarità della specifica funzione di attivazione consentono alcuni miglioramenti.

Il valore scalare  $r$  in (5.19) imposta la soglia della funzione hard-limiter. Proiettando qualsiasi dato di input  $\mathbf{x} \in \mathcal{X}$ , su  $\hat{\mathbf{r}}$  si mette in relazione  $\mathbf{x}$  a  $r$ , cioè alla posizione del landmark  $\mathbf{l}$  corrispondente, come in (5.19). Questo approccio si estende ad una coppia di sottoinsiemi nello spazio di input in base alle relative proiezioni di ciascun pattern rispetto a  $\mathbf{r}$ . I due sottoinsiemi possono essere formalizzati come

$$\Omega_{un} = \{(\mathbf{x}, y)_i | \mathbf{x}^t \hat{\mathbf{r}} < r\}, \quad \Omega_{on} = \{(\mathbf{x}, y)_i | \mathbf{x}^t \hat{\mathbf{r}} \geq r\} \quad (5.24)$$

In pratica la (5.24) mostra che i dati vengono proiettati in due set  $(\Omega_{un}, \Omega_{on})$ . La situazione migliore, ma improbabile, si verifica quando un sottoinsieme include solo i pattern appartenenti a una classe (es.  $y = -1$ ), mentre l'altro contiene solo i campioni appartenenti all'altra classe (es.  $y = +1$ ). In effetti una scelta inadeguata di  $\hat{\mathbf{r}}$  può rendere questo obiettivo irrealizzabile, indipendentemente dal valore assunto da  $r$ : questo è solitamente il caso in cui  $\hat{\mathbf{r}}$  è impostato in modo casuale. Tuttavia dato  $\hat{\mathbf{r}}$ , si può scegliere il valore di  $r$  in modo da ottenere il miglior partizionamento possibile. Questa



è la logica alla base della procedura di training introdotta in questa sezione per generare le coppie  $\{\hat{r}, r\}_n$  con un sovraccarico computazionale minimo.

Per prima cosa le posizioni  $\hat{r}$  sono impostate random. In secondo luogo, per ogni neurone, un gruppo di valori  $C$  di candidati per  $r$  viene estratto casualmente. Infine, il valore finale di  $r_n$  viene selezionato secondo un criterio di ottimalità. Poiché la funzione soglia ha una natura binaria, come criterio è stato scelto il massimo guadagno di entropia [100], dato che coincide con la separazione ottima delle classi. Di conseguenza l'entropia  $H(\Omega)$  di un insieme  $\Omega$  deriva dalla proporzione tra i campioni di entrambe le classi. L'Algoritmo 4 delinea la procedura completa per produrre le coppie  $\{\hat{r}, r\}_n$  da utilizzare nel problema di apprendimento (5.11).

Poiché il criterio di ottimalità è applicato a un pool preselezionato di valori candidati per  $r_n$ , non è previsto che l'algoritmo trovi il valore migliore per la soglia, dato  $\hat{r}_n$ . Tuttavia, cambiando il numero  $C$  di candidati si può bilanciare il costo computazionale dell'algoritmo e le prestazioni di generalizzazione della rete. Come nel caso della ricerca casuale, l'algoritmo evita il ricalcolo dei prodotti scalari che caratterizzano la funzione di attivazione, limitando in tal modo il sovraccarico nella fase di training.

### 5.5.2.1 Risultati sperimentali

I test sperimentali riguardano l'abilità della procedura di apprendimento (Algoritmo 4) nel bilanciare l'accuratezza e la dimensione dello strato hidden. Le performance di generalizzazione sono state valutate confrontando le prestazioni di tre macchine: una ELM standard basata sulla funzione sigmoide (E-S), una ELM standard basata sulla funzione hard-limiter (E-L) e una ELM *hardware-friendly* basata sulla funzione hard-limiter (HE). I primi due predittori sono stati realizzati utilizzando le procedure di apprendimento convenzionali associate al modello ELM (come nella Sezione 5.1). Il terzo predittore, al contrario, ha impiegato un processo di addestramento supportato dall'Algoritmo 4.

Questa sessione utilizza 15 set di dati dal repository UCI [97]: *Breast Cancer Wisconsin (Original, Diagnostic, Prognostic)*, *Blood Transfusion*, *Connectionist Bench, Default of Credit Card Clients*, *Detect Malicious Executable*, *Ionosphere*, *LSVT Voice Rehabilitation*, *MAGIC Gamma Telescope*, *Ozone Level Detection*, *Pima Indians Diabetes*, *Planning Relax*, *QSAR Biodegradation*, *Statlog Australian Credit Approval*. Tutti i dati sono stati normalizzati nell'intervallo  $[0, 1]$ . In ogni esperimento i predittori sono stati confrontati adottando tre configurazioni per il numero di neuroni:  $N = \{25, 100, 500\}$ . La model selection impiega il seguente range per il parametro di regolarizzazione:  $\lambda \in \{2^i, i = -10, -9, \dots, 10\}$ . Nella fase di training di HE il numero

---

**Algoritmo 4** Strategia di mapping migliorata per le funzioni di attivazione con soglia che utilizzano il massimo guadagno di entropia

---

### Input

- Un training set con etichette  $\mathcal{T} = \{(\mathbf{x}, y)_i; i = 1, \dots, Z\}$ .
- Un numero di neuroni  $N$ .
- Un numero di candidati  $C$ .

### 0. Inizializzazione

- **a.** Si generi random un set di landmark  $\mathbf{r}_j, j = 1, \dots, N$ .
- **b.** Per ogni neurone, si calcoli  $r_j$  applicando la routine seguente:
  - si generi un insieme di candidati  $\Delta = \{r_c, c = 1, \dots, C\}$  con  $r_c = \text{unifrnd}(-\sqrt{D}, \sqrt{D})$ ;
  - si costruisca il sottoinsieme  $\Omega_u(r_c), \Omega_o(r_c)$  per ogni  $r_c \in \Delta$ ;
  - si imposti  $r_{c^*} = \text{argmin}_c \{H(\Omega_u(r_c)) + H(\Omega_o(r_c))\}$ .

### 1. Mapping

Si rimappino tutti i pattern  $\mathbf{x} \in \mathcal{T}$  usando la seguente funzione di mapping:

$$\boldsymbol{\psi}(\mathbf{x}) = \{\phi(\mathbf{x}, \mathbf{r}_1, r_{1^*}), \dots, \phi(\mathbf{x}, \mathbf{r}_N, r_{N^*})\}$$

### 2. Apprendimento

Si addestri un predittore lineare nello spazio  $\boldsymbol{\psi} : \mathcal{X} \rightarrow \mathbb{R}^N$ .

---

di candidati per il landmark **I** è stato impostato a  $C = 3$ . Ogni esperimento ha coinvolto cinque *run*, cioè cinque coppie di training-test casuali; il 70% dei pattern sono stati utilizzati come set di training e i rimanenti hanno costituito il set di test. L'accuratezza su ogni run è stata calcolata con la media su 40 estrazioni dei parametri casuali.

La Tabella 5.7 riassume i risultati della prima sessione di esperimenti, in cui il problema dell'apprendimento riguarda il termine di regolarizzazione convenzionale  $l_2$  (5.5). La prima colonna indica il set di dati. La seconda fornisce, rispettivamente, la dimensionalità  $D$  e il numero di campioni nel dataset. La terza riporta l'errore di classificazione medio sul set di test (come percentuale espressa nell'intervallo  $[0, 1]$ ) raggiunto da E-S, per  $N = 25$ . La quarta e la quinta mostrano, rispettivamente, il guadagno/la perdita nell'errore di classificazione raggiunto da E-L e HE rispetto a E-S. Pertanto una quantità negativa indica che il predittore ha ridotto l'errore di classificazione rispetto a E-S. Lo stesso formato è stato applicato per presentare i risultati ottenuti con  $N = 100$  e  $N = 500$ .

I risultati numerici portano chiaramente a tre considerazioni importanti. Innanzitutto, il

TABELLA 5.7: Performance di generalizzazione con la  $l_2$  come termine di regolarizzazione.

Dataset	D / Dim	N = 25			N = 100			N = 500		
		E-S	E-L	HE	E-S	E-L	HE	E-S	E-L	HE
BCW O	10 / 699	0.05	0.01	0.00	0.05	0.01	0.00	0.05	0.01	0.00
BWC D	32 / 569	0.02	0.05	0.01	0.01	0.02	0.00	0.01	0.01	0.00
BWC P	34 / 198	0.26	0.12	0.08	0.23	0.09	0.06	0.23	0.08	0.04
Blood	5 / 748	0.30	0.10	0.04	0.30	0.06	0.02	0.30	0.04	0.03
Conn. B.	60 / 208	0.25	0.05	0.02	0.18	0.04	0.01	0.15	0.01	0.00
Def. CC	24 / 30000	0.23	0.13	0.03	0.21	0.07	0.00	0.20	0.03	-0.01
Mal. Ex.	513 / 373	0.04	0.01	-0.02	0.02	0.00	0.00	0.02	0.00	0.00
Iono	34 / 351	0.14	0.02	0.02	0.11	-0.03	-0.03	0.09	-0.03	-0.03
LSVT	309 / 126	0.25	0.05	0.01	0.21	0.05	0.01	0.15	0.04	0.01
MAGIC	11 / 19020	0.17	0.07	0.04	0.15	0.06	0.02	0.13	0.03	0.00
Ozone	73 / 2536	0.22	0.05	0.03	0.18	0.05	0.04	0.16	0.04	0.02
Pima	8 / 768	0.13	0.06	0.02	0.13	0.03	0.01	0.13	0.02	0.00
Pl. Rel.	13 / 128	0.27	0.05	0.07	0.28	0.03	0.04	0.29	0.02	0.04
QSAR	41 / 1055	0.18	0.12	0.03	0.15	0.06	0.01	0.15	0.03	0.00
StatLog	14 / 690	0.15	0.06	0.02	0.13	0.01	0.00	0.13	0.00	0.00

divario tra la HE proposto e la E-S è molte volte più piccolo dello scarto corrispondente tra E-L ed E-S. Quindi, HE presenta quasi sempre prestazioni migliori di E-L. In secondo luogo, la differenza tra HE ed E-S il più delle volte è inferiore al 4%. Ciò a sua volta significa che l'Algoritmo 4 è in grado di bilanciare l'accuratezza e la dimensione dello strato hidden. In terzo luogo, tale tendenza rimane valida quando  $N = 25$ , cioè quando si desidera impiegare un predittore che può adattarsi ai dispositivi con risorse ridotte.

La seconda sessione sperimentale mira a valutare l'efficacia della procedura di apprendimento che supporta la HE nel limitare configurazioni dei neuroni infruttuose. Questo è un grosso problema quando si mira a un giusto equilibrio tra le prestazioni di generalizzazione e il numero di neuroni. Per completare tale analisi, è stata utilizzata la funzione costo basata sulla regolarizzazione  $l_1$ :

$$\min_{\beta} \{ \|\mathbf{y} - \mathbf{H}\beta\|^2 + \lambda \|\beta\| \} \quad (5.25)$$

In realtà, si prevede che tale procedura di apprendimento produca una soluzione più sparsa eliminando i neuroni inutili (*pruning*), cioè quelli caratterizzati da una configurazione dei parametri  $\{\hat{r}, r\}_n$  non opportuna. Quindi, dato  $N$ , si può rivelare la percentuale di neuroni che svolgono un ruolo attivo nella predizione, che corrispondono a quelli "sopravvissuti" al pruning.

La Tabella 5.8 riassume i risultati di questa seconda sessione di esperimenti che ha coinvolto gli stessi 15 dataset utilizzati nella prima sessione e ha adottato la stessa impostazione. La prima colonna indica il nome del dataset. La seconda, terza, quarta e quinta

TABELLA 5.8: Performance di generalizzazione e sparsità.

Dataset	N = 25				N = 100			
	E-S	HE	% Attivi	% Attivi	E-S	HE	% Attivi	% Attivi
			E-S	HE			E-S	HE
BCW O	0.05	0.00	72.6%	81.9%	0.05	0.00	39.3%	52.7%
BWC D	0.02	0.01	82.2%	88.9%	0.01	0.01	53.8%	73.0%
BWC P	0.26	0.08	75.0%	73.0%	0.23	0.06	30.4%	40.0%
Blood	0.31	0.03	26.0%	61.5%	0.31	0.02	8.3%	33.2%
Conn. B.	0.25	0.01	75.8%	74.0%	0.19	0.02	52.4%	53.1%
Def. CC	0.23	0.03	87.6%	84.6%	0.21	-0.01	57.6%	73.0%
Mal. Ex.	0.05	-0.03	82.2%	87.9%	0.02	0.00	56.7%	64.9%
Iono	0.14	0.02	81.4%	78.5%	0.11	-0.03	50.2%	50.0%
LSVT	0.25	0.02	69.8%	68.5%	0.21	0.02	34.5%	32.0%
MAGIC	0.18	0.03	79.4%	89.4%	0.16	0.01	44.8%	79.5%
Ozone	0.22	0.03	73.7%	69.8%	0.18	0.04	44.8%	39.9%
Pima	0.13	0.02	52.2%	72.4%	0.13	0.00	23.9%	45.1%
Pl. Rel.	0.28	0.04	48.0%	49.2%	0.29	-0.01	17.6%	19.9%
QSAR	0.18	0.03	81.6%	80.7%	0.15	0.01	56.6%	61.0%
StatLog	0.15	0.02	87.5%	86.8%	0.13	0.00	58.9%	65.4%

colonna si riferiscono ai predittori con  $N = 25$ . La seconda colonna riporta l'errore di classificazione medio sul set di test (come percentuale espressa nell'intervallo  $[0, 1]$ ) raggiunto da E-S con il termine di regolarizzazione  $l_1$ . La terza indica il divario in termini di errore di classificazione tra E-S e HE, anch'esso addestrato sfruttando la regolarizzazione  $l_1$  nella funzione costo. L'errore di classificazione raggiunto da E-S fornisce il riferimento; quindi quantità negative indicano che HE è migliorato rispetto a E-S. La quarta colonna indica, in percentuale su  $N$ , la quantità di neuroni sopravvissuti al pruning in E-S. La quinta colonna fornisce la quantità analoga per HE. Lo stesso formato viene utilizzato per presentare le performance ottenute con  $N = 100$ . La tabella non fornisce alcun confronto con E-L poiché la Tabella 5.7 ha già dimostrato che tale predittore è meno efficace di quello proposto. In effetti, la Tabella 5.8 non include la configurazione  $N = 500$ , in quanto la quantità di neuroni inutili cresce in modo asintotico mentre  $N$  cresce.

I risultati numerici mostrano che, in termini di errore di classificazione, il divario tra E-S e HE rimane basso anche quando è coinvolta la regolarizzazione  $l_1$ . In effetti, si nota che le prestazioni di generalizzazione di E-S non cambiano significativamente rispetto agli esperimenti che comportano il termine di regolarizzazione convenzionale  $l_2$  (Tabella 5.7). Inoltre la Tabella 5.8 evidenzia che la quantità di neuroni eliminati dal pruning è spesso più grande in E-S che in HE. In alcuni casi la differenza è in sostanziale. Ciò a sua volta significa che E-S e HE, in generale, includono una diversa quantità di neuroni con configurazioni infruttuose. Come risultato principale questa sessione sperimentale ha confermato che HE può bilanciare efficacemente le prestazioni di generalizzazione e le

dimensioni dello strato hidden. Tale abilità consente a HE di sfruttare al meglio una funzione di attivazione grossolana come la hard-limiter.

### 5.5.3 Analisi Comparativa

L'Algoritmo 3 e l'Algoritmo 4, come accennato nell'introduzione, condividono alcuni punti comuni. Infatti, questi coinvolgono una strategia di selezione supervisionata per l'impostazione del parametro nascosto  $\chi$ . Questa strategia non richiede di ricalcolare la metrica  $M(\mathbf{x}, \mathbf{r})$  che comporta operazioni vettoriali complesse (si veda (5.15)). D'altro canto, ciascuno di questi approcci presenta alcune peculiarità che lo rendono più adatto ad uno scenario applicativo rispetto ad un altro.

L'Algoritmo 4 è stato progettato per gestire le peculiarità della funzione di soglia. Infatti questa ammette solo due valori e, nonostante porti a evidenti vantaggi per un'eventuale implementazione hardware, è più soggetta a configurazioni patologiche rispetto alle attivazioni analogiche standard come SGM o RBF. Per questa ragione l'Algoritmo 3, anche se in linea di principio è applicabile per la selezione dei parametri liberi, risulta meno efficace, data la natura binaria del mapping. Di conseguenza, è stato proposto uno specifico criterio di selezione.

L'Algoritmo 3 è stato progettato per scartare possibili configurazioni patologiche dei parametri random. L'algoritmo impiega qualsiasi funzione di attivazione che sia buona per una porzione del dominio di input. Inoltre questo algoritmo, data la natura randomica nella scelta dei parametri in ciascun neurone, richiede un numero di step non fissato a priori.

La letteratura di ELM fornisce diversi lavori che trattano la costruzione dello strato di mapping. In generale l'obiettivo è definire una strategia in grado di bilanciare la capacità di generalizzazione e il numero di neuroni nel predittore finale. In [89, 90] il problema di ottimizzazione è stato leggermente modificato allo scopo di favorire la soluzione sparsa, rimuovendo così i neuroni inefficaci. In [90] gli autori hanno sfruttato un termine di regolarizzazione  $l_1$  per rimuovere i neuroni inutili: quelli selezionati sono stati impiegati nell'addestramento di una ELM standard. L'ottimizzazione del pruning della ELM [89] fa affidamento su una combinazione di più regressioni sparse e un meccanismo di *leave-one-out* per eliminare i neuroni meno informativi. In entrambi i lavori il processo di pruning richiede uno sforzo computazionale elevato e aggiunge alcuni iper-parametri per l'ottimizzazione del problema.

Molti lavori riguardano l'ottimizzazione di problemi di natura biologica: *self-adaptive evolutionary ELM* [91], *dolphin swarm ELM* [101], *genetic ensemble of ELM* [102], *particle swarm optimization based* [92] e *Artificial Immune System based ELM* [103]. Tutti questi approcci condividono l'idea comune di sfruttare strategie derivate dall'osservazione dei fenomeni naturali per migliorare le prestazioni delle macchine in termini di capacità di generalizzazione. Tali modelli richiedono un elevato sforzo computazionale. Di fatto, la maggior parte di questi comporta problemi di ottimizzazione non convessi.

In [104, 105] l'obiettivo è l'implementazione del predittore su un dispositivo con risorse limitate. In pratica, solo questi due lavori sono comparabili in termini di costi computazionali rispetto a quelli introdotti in questa tesi. La strategia di mapping è stata progettata per soddisfare specifici vincoli sulla funzione di attivazione, rendendo queste soluzioni meno generali di quella proposta.

## Capitolo 6

# Implementazione hardware

Una quantità considerevole di applicazioni si basa su sistemi embedded al fine di fornire predizioni online [90, 106–111]; vale a dire che la funzione di classificazione dovrebbe elaborare in tempo reale ogni nuovo campione in ingresso al sistema. A questo proposito i sistemi embedded a basso consumo e con risorse limitate forniscono uno scenario stimolante per l'implementazione di tali predittori. Le reti neurali feed forward a strato singolo, come i modelli introdotti nel Capitolo 5, rappresentano un'opzione efficace per l'obiettivo di implementazioni su dispositivi a basso costo perché, in generale, raggiungono un compromesso soddisfacente tra accuratezza della classificazione e complessità computazionale. La funzione di previsione alla fine consiste in una somma ponderata di funzioni di attivazione non lineari, in cui il numero di termini è uguale al numero di neuroni.

In realtà i paradigmi dell'apprendimento *deep* possono offrire prestazioni migliori in termini di accuratezza; d'altro canto l'implementazione hardware della funzione decisionale corrispondente non si adatta ancora ai dispositivi con risorse limitate [93, 112, 113].

Dal punto di vista dell'hardware, come introdotto nel Capitolo 5, la configurazione più interessante per le SLFN è quella che sfrutta solo le funzioni di attivazione hard-limiter (soglia) nello strato hidden. Questa configurazione riduce drasticamente la complessità del circuito nell'implementazione digitale dei predittori. Tuttavia esistono due inconvenienti cruciali. Innanzitutto la fase di apprendimento (offline) non può essere supportata da algoritmi convenzionali, poiché la funzione hard-limiter non è differenziabile [114]. In secondo luogo la soglia, in generale, influisce sulla capacità di generalizzazione dell'eventuale SLFN, che potrebbe richiedere un numero elevato di neuroni per ottenere un'accuratezza soddisfacente.

Il framework teorico della Extreme Learning Machine [80, 86], come sottolineato nella Sezione 5.5.2, offre una soluzione praticabile a questi problemi, gestendo appropriatamente la procedura di campionamento. L'iter di apprendimento raggiunge due obiettivi. In primo luogo fornisce uno strumento adatto per addestrare una SLFN basata sulla funzione di attivazione hard-limiter. Inoltre fornisce un algoritmo di training in grado di bilanciare efficacemente le prestazioni di generalizzazione e il numero di neuroni nello strato hidden. Quest'ultimo aspetto rappresenta una soluzione importante quando si vuole realizzare un dispositivo con risorse limitate.

Questo capitolo sfrutta i risultati della ricerca presentata nella Sezione 5.5.2 per concentrarsi sulla progettazione dell'architettura digitale, che può supportare in modo efficiente l'implementazione della SLFN addestrata. L'enfasi è posta su dispositivi a basso costo e a basse performance come CPLD e FPGA di fascia economica. Il presente lavoro analizza due strategie diverse di progettazione che possono adattarsi a tale scenario. La prima strategia riguarda principalmente un utilizzo efficiente delle risorse hardware: per questo motivo tutte le scelte progettuali hanno lo scopo di minimizzare l'allocazione delle risorse. La seconda strategia, al contrario, riduce la latenza (il numero di cicli di clock necessari ad ottenere il risultato) attraverso la parallelizzazione della parte che richiede più tempo nella rete neurale, cioè il calcolo del prodotto scalare  $\mathbf{x} \cdot \hat{\mathbf{r}}$  nella (5.19). L'obiettivo in comune ad entrambe le scelte è quello di realizzare un predittore eliminando la necessità di qualsiasi moltiplicatore nell'architettura finale.

In linea di principio, quando lo strato hidden non è soggetto a criteri di selezione o di pruning, tutte le memorie ROM utilizzate per memorizzare i parametri casuali  $\{\mathbf{r}, \chi\}$  possono essere sostituite da generatori di numeri pseudo casuali (PRNG). Questa scelta progettuale riduce drasticamente i requisiti per l'occupazione di area per l'implementazione dei predittori finali. Sfortunatamente, quasi tutte le strategie allo stato dell'arte adatte a migliorare il trade-off tra complessità computazionale e performance di generalizzazione (riassunte nella Sezione 5.5.3) vengono scartate dai vincoli imposti dall'uso del PRNG. Tuttavia è possibile adattare l'architettura per gestire un sottoinsieme di tecniche di selezione dei parametri con un sovraccarico computazionale minimo.

Gli obiettivi di questa ricerca differiscono dai lavori descritti in [104, 105, 110], in cui l'obiettivo principale è stata l'implementazione VLSI, e da [111], in cui l'architettura generale è stata progettata per il training online. In [104] gli autori impiegano una funzione di attivazione a tre stadi di pipeline e una procedura di pruning per ridurre la complessità del predittore finale nel circuito. Tuttavia, il modello globale è mirato per implementazioni analogiche. I pesi espressi in logica binaria sono stati applicati anche a reti neurali convoluzionali per limitare la complessità computazionale [93], le



reti risultanti determinano approssimazioni inadeguate per le SLFN, poiché queste ultime implicano un numero limitato di parametri.

Recentemente, diversi autori hanno affrontato il problema di salvare efficientemente in memoria i pesi casuali dello strato hidden. Sfruttando la loro natura randomica, in [104, 115], gli autori hanno proposto uno schema di rotazione che consente di memorizzare un piccolo numero di parametri hidden. Questi approcci differiscono da quelli proposti nella tesi perché propongono un design misto analogico-digitale ed i vantaggi di questi metodi diventano meno evidenti in un'implementazione completamente digitale, che resta l'obiettivo principale.

## 6.1 Realizzazione digitale

In questo capitolo si propone la progettazione di un'architettura digitale modulare in grado di realizzare la funzione decisionale espressa in (5.11), per il caso specifico dell'attivazione dei neuroni (5.19):

$$f(\mathbf{x}) = \sum_i^N \beta_n \cdot \text{sign}(\mathbf{x}^t \hat{\mathbf{r}}_n - r_n) = \sum_i^N \beta_n \cdot \text{sign}(I_n) \quad (6.1)$$

Come principale conseguenza del target su dispositivi con risorse limitate, la strategia di progettazione richiede di affrontare alcuni problemi di implementazione con lo scopo di raggiungere un corretto equilibrio tra accuratezza, allocazione delle risorse e complessità computazionale. Di seguito, si presume che sia la features  $i$ -esima di  $x_i$  sia l' $n$ -esimo peso  $\beta_n$  vengano rappresentati come un *fixed point* a  $B$ -bit in complemento a 2.

Il primo problema critico in (6.1) riguarda il calcolo della funzione di attivazione sull'input  $I$ :

$$I(\mathbf{x}, \hat{\mathbf{r}}_n, r_n) = \|\mathbf{r}_n\|(\mathbf{x}^t \hat{\mathbf{r}}_n - r_n) \quad (6.2)$$

Una proprietà importante di combinare  $I$  con l'unità di soglia è che la quantità  $\|\mathbf{r}\|$ , che caratterizza la (5.19), non influisce il segno dell'attivazione  $I$ , riducendo automaticamente il numero di calcoli. L'operazione (6.2) comporta  $D$  moltiplicazioni e  $D$  somme (dove  $D$  è la dimensionalità del dato  $\mathbf{x}$ ); inoltre, dovrebbe essere ripetuta  $N$  volte (una per ogni neurone). In realtà, le moltiplicazioni possono essere evitate imponendo il seguente vincolo sui pesi casuali  $\hat{\mathbf{r}}_{d,n}$ :

$$\hat{\mathbf{r}}_{d,n} = \text{sign}(r_1) 2^{-\lceil r_2 \rceil} = s \cdot 2^{-k} \quad (6.3)$$

dove  $r_1 \in \{-1, 1\}$  e  $r_2 \in N^+$  sono quantità casuali. Si noti che questo vincolo non viola nessuno dei di quelli nella teoria di ELM [81]. Di conseguenza, il calcolo del singolo

termine  $x_d \cdot \hat{r}_{dn}$  può essere completato con un'operazione di *shift* ( $x_d \gg k_{dn}$ ), che dovrebbe essere seguita da un cambiamento di segno quando  $s_{dn} < 0$  (cioè  $\hat{r}_{dn} < 0$ ).

In linea di principio, per applicare la conversione del segno al termine che ha subito l'operazione di shift, è necessario un operatore *not* bit a bit, seguito da una somma con un termine costante, il cui valore è '1 Bit Meno Significativo' (LSB). Tuttavia, quando si utilizza la (6.1) si conosce in anticipo quanti pesi  $r_{dn}$  sono negativi. A tal proposito, sia  $W$  il numero di  $r_{dn}$  negativi. Allora, quando si calcola  $\mathbf{x}^t \hat{\mathbf{r}}_n$ , i singoli termini  $W$  '1 LSB' possono essere sostituiti da un singolo termine cumulativo  $\zeta$  che tiene in conto del numero totale di pesi negativi. Di conseguenza, si ha:

$$\mathbf{x}^t \hat{\mathbf{r}}_n - r_n = \sum_d \bar{x}_d - r_n + \zeta_n \quad (6.4)$$

dove

$$\bar{x}_d \begin{cases} x_d \gg k_{dn}, & \text{se } s_{dn} = 1 \\ \text{NOT}(x_d \gg k_{dn}), & \text{se } s_{dn} = -1 \end{cases}$$

Tale approccio semplifica notevolmente la progettazione del dispositivo digitale.

Il secondo problema critico riguarda il calcolo di  $\phi(I)$ . Questa operazione deve essere completata  $N$  volte (una per ogni neurone). Una scelta standard per la funzione di attivazione  $\phi$  è la sigmoide. In letteratura l'implementazione hardware di tale funzione è ottenuta sfruttando approssimazioni lineari a tratti o affidandosi alle *lookup tables* [116]. Solitamente si preferisce la prima soluzione poiché le lookup tables richiedono memorie molto grandi. L'approssimazione lineare a tratti comporta, però, sempre una moltiplicazione e una somma. Invece la funzione di attivazione basata sulla soglia, in un'implementazione digitale, consiste soltanto nel leggere il bit più significativo.

Inoltre una SLFN basata sulla funzione di attivazione hard-limiter è caratterizzata da un ulteriore vantaggio: il termine  $\beta_n \cdot \phi(I)$  in (6.1) può essere calcolato senza fare affidamento su un moltiplicatore, poiché  $\phi(I)$  imposta solo il segno di  $\beta_n$ .

L'architettura, che supporta l'implementazione digitale della SLFN, può essere organizzata secondo tre moduli principali:

- Il primo modulo, *Input*, memorizza le componenti (features) del campione  $\mathbf{x}$  che deve essere classificato.
- Il secondo modulo, *Neuron*, calcola per l' $n$ -esimo neurone la quantità  $a_n = \text{sign}(\mathbf{x}^t \hat{\mathbf{r}}_n - r_n)$ .

- Il terzo modulo, *Output*, costituisce il livello di output e finalizza in modo incrementale il calcolo di (6.1). Si attiva non appena *Neuron* fornisce  $a_n$  per il neurone  $n$ -esimo.

L'architettura, inoltre, include una macchina a stati finiti che governa l'intero processo di controllo e le risorse di archiviazione per memorizzare le seguenti quantità generate dalla procedura di apprendimento:  $\{(s, k)_d; d = 1, \dots, D\}_n, \zeta_n, l_n, \beta_n; n = 1, \dots, N$ .

La struttura complessiva può portare a due approcci di implementazione diversi. Il primo si basa su un modulo *Neuron* che elabora in sequenza le features  $x_d$ , come in [69]. Nel secondo, il modulo *Neuron* elabora in parallelo tutte le  $D$  componenti dell'input  $\mathbf{x}$ .

### 6.1.1 Implementazione seriale del neurone

L'architettura che implementa l'approccio sequenziale è mostrata in Fig. 6.1.

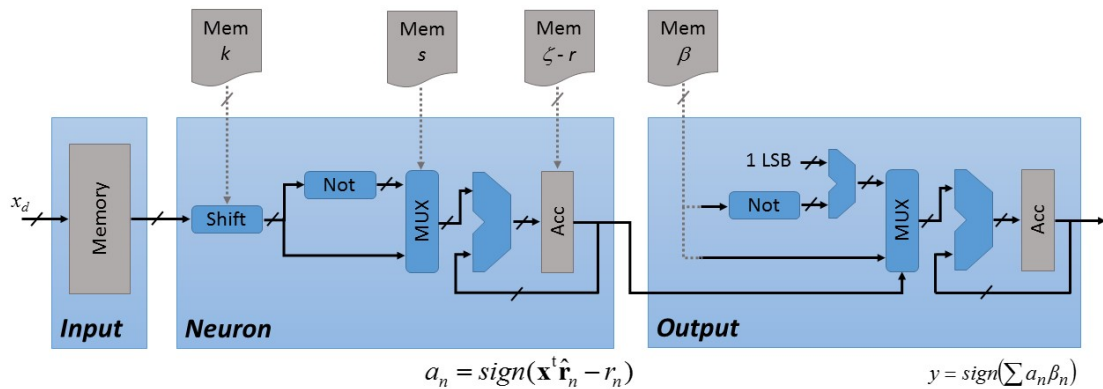


FIGURA 6.1: Architettura dell'implementazione digitale del predittore: approccio sequenziale.

Il design interno dei tre moduli principali è organizzato come segue:

1. **Input**: una memoria che salva i valori delle  $D$  features. Il modulo supporta un input seriale e fornisce come output una singola feature  $x_d$  per ciclo di lettura.
2. **Neuron**: questo modulo calcola la quantità  $a_n$  step-by-step. Per prima cosa, quando una feature  $x_d$  entra nel modulo, un'operazione di shift calcola il termine  $x_d \gg k_{dn}$ , come espresso in (6.3); gli ingressi al modulo di shift provengono da una memoria in cui sono stati salvati i numeri interi  $k_{dn}$ . In secondo luogo, il bit del segno corrispondente  $s_{dn}$  pilota un multiplexer, che seleziona l'output dell'operatore bitwise *not* solo quando  $s = -1$ , cioè quando  $x_d \cdot \hat{r}_{dn}$  comporta una conversione del segno. L'ultimo blocco somma-e-accumula ha il compito di eseguire in modo

incrementale il calcolo del termine finale, mostrato in (6.4). Il risultato si ottiene inizializzando l'accumulatore al valore  $(\zeta_n - r_n)$  prima che inizi l'elaborazione di  $x_1$ ; una memoria supporta tale procedura. Il modulo fornisce come output il bit del segno nell'accumulatore, cioè  $a_n$ .

- 3. Output:** questo modulo è progettato per ottenere la predizione  $y(\mathbf{x})$  step-by-step. Come per  $a_n \in \{-1, 1\}$ , le quantità  $a_n \cdot \beta_n$  vengono calcolate sfruttando un'unità di segno, che supporta la conversione di  $\beta_n$ , se necessario. I pesi  $\beta_n$  sono stati salvati nella memoria corrispondente. Un blocco somma-e-accumula supporta il calcolo incrementale di  $y$ . Il modulo fornisce in uscita il bit di segno nell'accumulatore, cioè la previsione.

La Fig. 6.2 mostra un esempio del flusso di elaborazione per  $D = 3$  e  $N = 4$ , vale a dire tre features per il dato e quattro neuroni nello strato hidden.

clock cycles															
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	$x_1$	$x_2$	$x_3$												
	$I_{11}$	$I_{21}$	$I_{31}$	$I_{12}$	$I_{22}$	$I_{32}$	$I_{12}$	$I_{22}$	$I_{32}$	$I_{13}$	$I_{23}$	$I_{33}$			
		$N_{1(1)}$	$N_{1(2)}$	$N_{1(3)}$	$N_{2(1)}$	$N_{2(2)}$	$N_{2(3)}$	$N_{3(1)}$	$N_{3(2)}$	$N_{3(3)}$	$N_{4(1)}$	$N_{4(2)}$	$N_{4(3)}$		
				$a_1$			$a_2$			$a_3$			$a_4$		
					$O_1$			$O_2$			$O_3$			$O_4$	
															$y$

FIGURA 6.2: Esempio di flusso delle operazioni per l'architettura sequenziale.

Lo schema delinea l'attività funzionale dei seguenti processi:

- $I_{dn}$ : ciclo di input. Il modulo di *Input* fornisce come output la  $d$ -esima feature; l'unità di storage fornisce al datapath i parametri  $\hat{r}_{dn}$ .
- $N_{n(d)}$ : ciclo per il modulo *Neuron*. Il modulo istanzia il neurone  $n$ -esimo ed elabora la feature  $d$ -esima, calcolando in tal modo  $x_d \cdot \hat{r}_{dn}$ .
- $O_n$ : ciclo di elaborazione del modulo *Output*. Il modulo elabora  $a_n$ , calcolando quindi  $a_n \cdot \beta_n$ .

Nell'esempio proposto, le features  $x_1$ ,  $x_2$  e  $x_3$  sono salvate nella memoria realizzata nel modulo *Input* sul fronte di salita dei cicli di clock 2, 3 e 4, rispettivamente. Di conseguenza,  $I_{11}$  può essere completato entro il ciclo di clock 2 e il modulo *Neuron* può avviare il calcolo incrementale di  $a_1$  al ciclo di clock 3 ( $N_{1(1)}$ ). Al fronte di salita del clock

6 l'accumulatore realizzato nel modulo *Neuron* memorizza  $a_1$ , poiché è stato completato  $N_{1(D)}$ . Pertanto, entro il ciclo di clock 6 il modulo *Output* calcola  $a_1 \cdot \beta_1(O_1)$ . Alla fine,  $O_N$  viene eseguito al clock 15 e la previsione finale  $y$  è disponibile nell'accumulatore ospitato dal modulo *Output* al clock 16. Nel complesso, questo approccio garantisce un risultato di previsione in  $(D \cdot N + 3)$  cicli di clock, a partire dalla lettura della prima feature.

### 6.1.2 Realizzazione parallela del neurone

La Fig. 6.3 mostra l'architettura che impiega il modulo *Neuron* parallelo. Il design

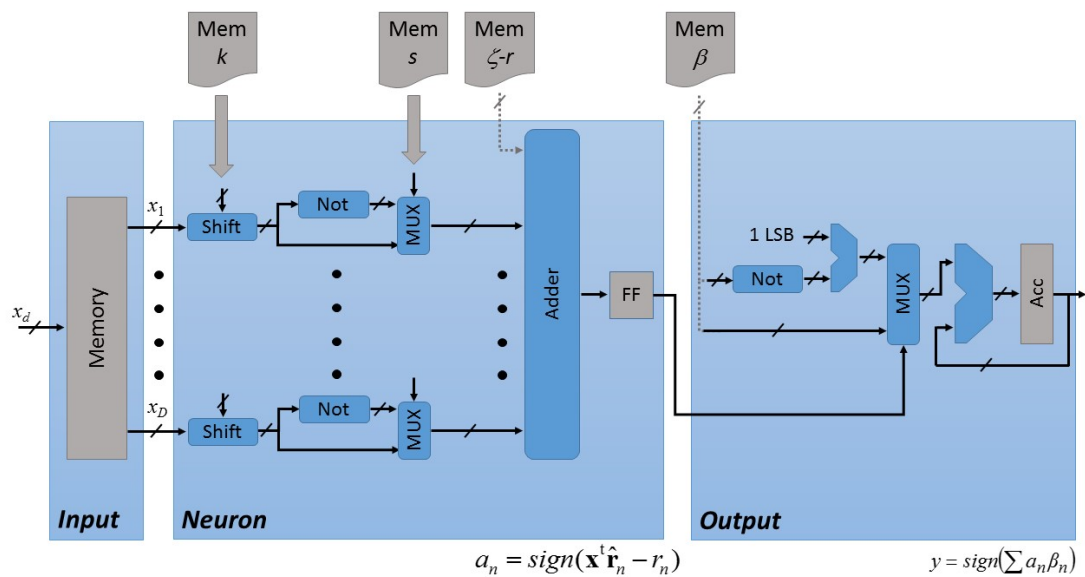


FIGURA 6.3: Architettura per l'implementazione digitale del predittore: approccio parallelo.

interno dei moduli principali cambia come segue:

1. **Input:** il modulo contiene una memoria con input seriale che salva  $D$  features. Tuttavia, in questa architettura, fornisce come output tutte le  $D$  features in parallelo per ciclo di lettura.
2. **Neuron:** questo modulo calcola la quantità  $a_n$  in un singolo ciclo di clock. Di conseguenza, le features  $x_d$  vengono elaborate in parallelo da  $D$  istanze del blocco 'shift-not multiplexer' già descritto precedentemente. Un sommatore con  $D + 1$  input finalizza il calcolo di  $a_n$ ; l'input aggiuntivo si riferisce alla quantità  $(\zeta_n - r_n)$ . È stato aggiunto un flip-flop (FF) per memorizzare il bit di segno in  $a_n$ .
3. **Output:** questo modulo è ancora progettato per ottenere la previsione  $y(\mathbf{x})$  step-by-step.

Tale design supporta un'elaborazione completamente in pipeline, come mostrato in Fig. 6.4.

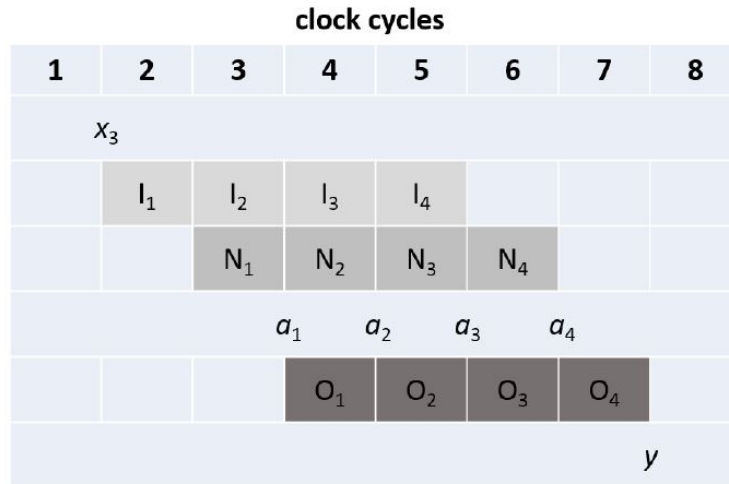


FIGURA 6.4: Esempio di flusso del processo dell'architettura parallela.

La figura, come la Fig. 6.2, si riferisce ad un esempio del flusso di elaborazione per  $D = 3$  e  $N = 4$ . L'operazione di pipeline coinvolge i tre processi seguenti per ogni ciclo singolo di clock:

- $I_n$ : ciclo di input. Il modulo *Input* fornisce come output le  $D$  features in parallelo; le unità di archiviazione forniscono al datapath i parametri  $\{\hat{r}_{n1}, \dots, \hat{r}_{nD}\}$ .
- $N_n$ : ciclo di elaborazione del modulo *Neuron*. Il modulo istanzia il neurone  $n$ , calcolando quindi  $a_n$ .
- $O_n$ : ciclo di elaborazione del modulo *Output*. Il modulo elabora  $a_n$ , calcolando quindi  $a_n \cdot \beta_n$ .

In questo caso, l'elaborazione può iniziare solo quando la feature  $D$ -esima è stata salvata nella memoria realizzata nel modulo *Input*. Di conseguenza, nell'esempio proposto, il flusso di elaborazione viene attivato dalla disponibilità di  $x_3$  alla porta di ingresso (fronte di salita del clock 2). Al fronte di salita del clock 4 il modulo *Neuron* memorizza  $a_1$  nel FF corrispondente; infatti,  $a_N$  è pronto dopo  $(N - 1)$  cicli di clock. La previsione finale  $y$  è disponibile nell'accumulatore, ospitato dal modulo *Output*, al colpo di clock 8. Nel complesso, questo approccio garantisce che la previsione sia calcolata in  $(N + 3)$  cicli di clock, a partire dalla lettura di  $x_D$ .

### 6.1.3 Confronto tra i due design: utilizzo dell'area vs latenza

I due approcci diversi alla progettazione del predittore possono essere confrontati in base ai seguenti attributi:

1. Area occupata: questo parametro dipende in gran parte dal design del modulo *Neuron*. Nell'approccio completamente parallelo (Fig. 6.3), la dimensionalità  $D$  dello spazio di input ha un impatto diretto sulle risorse occupate dall'istanza del neurone  $n$ -esimo. L'implementazione hardware corrispondente supporta l'elaborazione parallela delle  $D$  features; inoltre, il blocco *Adder* deve elaborare  $D + 1$  ingressi. Viceversa, nell'approccio sequenziale di Fig. 6.1 l'utilizzo delle risorse del modulo *Neuron* non è influenzato da  $D$ . Anche il numero di bit  $B$  gioca un ruolo nel determinare l'area occupata dal modulo *Neuron*. In effetti,  $B$  ha un impatto maggiore sul design sequenziale. Nondimeno,  $D$ ,  $B$  e il numero di neuroni  $N$  influenzano, in entrambi gli approcci, la dimensione delle memorie incluse nell'architettura. A questo proposito si dovrebbe considerare che la dimensione della memoria, ospitata nel modulo *Input*, cresce come  $D \times B$ ; inoltre, anche le memorie che salvano i parametri  $s$  e  $k$  crescono come  $D \times N$ .
2. Latenza: allo scopo di valutare questo parametro, siano  $T_s$  e  $T_p$  rispettivamente i tempi di durata dei periodi di clock delle implementazioni hardware risultanti dall'architettura sequenziale (Fig. 6.1) e l'architettura parallela (Fig. 6.3). Di conseguenza, la latenza associata alla predizione quando si adotta il primo approccio è  $(D \cdot N + 3) \cdot T_s$ , mentre l'approccio pipeline produce una previsione in  $(N + 3) \cdot T_p$ . Questo a sua volta significa che, dato  $N$ , si avrebbe bisogno di  $T_s \simeq T_p/D$  per ottenere un predittore, basato sull'architettura sequenziale, che abbia la stessa latenza di un predittore basato sull'architettura parallela. D'altra parte, si dovrebbe considerare che nell'approccio sequenziale l'elaborazione può iniziare quando viene memorizzata la feature  $x_1$ . Nell'approccio completamente pipeline, al contrario, l'elaborazione può iniziare solo quando viene memorizzata l'ultima feature  $x_D$ . Tale aspetto potrebbe diventare rilevante quando  $D \gg N$ .

Nel complesso l'approccio sequenziale si adatta meglio agli scenari in cui l'utilizzo efficace dell'area ha la priorità sulla latenza. Pertanto, tale approccio è efficace quando si vuole massimizzare la quantità  $D \times N$ , dato  $N$ . Viceversa, l'approccio parallelo supporta implementazioni che bilanciano opportunamente  $D \times N$  e latenza.

## 6.2 Architettura basata su generatore di numeri pseudo casuali

La memorizzazione dei parametri dei neuroni hidden è uno dei principali fattori del consumo di area quando si realizza una SLFN su dispositivi. In linea di principio, è necessario memorizzare  $D + 1$  numeri reali per ogni neurone.

Come conseguenza del vincolo (6.3), i parametri hidden  $\hat{r}_{dn}$  sono descritti dalle coppie  $\{k, s\}_{dn}$ , dove  $s$  è un bit e il valore  $k$  è un numero intero sempre inferiore a  $B$ , poiché corrisponde al numero massimo di operazioni di shift in un registro di dimensioni  $B$ . In pratica, ad ogni coppia  $\{k, s\}$  servono solo  $1 + \log_2(B)$  bit. Tuttavia, la dimensione di queste due memorie aumenta come  $D \times N$  e diventa una delle principali fonti di occupazione di area.

È interessante notare che l'unico requisito sulle coppie  $\{k, s\}$  è che vengano generate casualmente; quindi è possibile produrre questi numeri usando un generatore di numeri pseudo-casuale (PRNG) che, data una condizione iniziale fissa, fornisca sempre la stessa sequenza caotica di numeri. In questo modo, il requisito di memorizzare  $N \times D$  parametri diventa quello di implementare un generatore di sequenza per  $N \times D$  valori.

Il *Linear Feedback Shift Register* (LFSR) [117] è una realizzazione semplice e compatta di tali modelli. Dato un registro a scorrimento di dimensione  $m$  e una funzione di feedback basata sull'operatore binario *xor*, che riceve in input alcuni bit del registro, è possibile ottenere una sequenza di numeri non ripetuti con periodo  $2^m - 1$ . Per ogni dimensione  $m$ , le tabelle standard [118] forniscono l'indice del bit che dovrebbe essere usato nella funzione di feedback *xor*.

In generale, l'intero set  $\{\mathbf{K}, \mathbf{s}, r\}$  può essere impostato casualmente; tuttavia questo capitolo fa riferimento all'algoritmo di training proposto nella Sezione 5.5.2 perché fornisce un compromesso migliore tra le prestazioni di generalizzazione e le dimensioni dello strato hidden. Per questo motivo, il parametro  $r$  è soggetto a un criterio di selezione e deve essere memorizzato in una memoria ROM (l'estensione al caso generale è semplice).

Due vincoli di progettazione guidano la dimensione  $m$  del registro a scorrimento: 1)  $m$  dovrebbe essere abbastanza grande da garantire che, ad ogni ciclo di clock, siano forniti abbastanza bit per rappresentare i parametri hidden; 2)  $m$  dovrebbe essere tale da garantire che il periodo della sequenza casuale sia maggiore di  $N \times D$ , cioè la quantità di parametri hidden.

In base all'architettura selezionata il rapporto tra i vincoli cambia. Nell'approccio seriale (Fig. 6.1) la dimensione del generatore casuale è soggetta alle seguenti condizioni:



ad ogni ciclo di clock deve essere generata una coppia  $\{k, s\}$ , per cui sono necessari  $m \geq 1 + \log_2(B)$  bit; allo stesso tempo  $m$  dovrebbe essere tale che risulti  $2^m - 1 \geq N \times D$  per evitare configurazioni identiche. L'approccio parallelo (Fig. 6.3) richiede  $D$  coppie  $\{k, s\}$  ad ogni ciclo di clock, quindi la prima condizione diventa  $m \geq D + D * \log_2(B)$ ; mentre il vincolo sul periodo della sequenza casuale diventa meno stringente  $2^m - 1 \geq N$ . Si noti che la dimensione del registro è, in entrambi i casi, trascurabile rispetto al numero di elementi che dovrebbero essere memorizzati; infatti i primi vincoli corrispondono all'output della ROM in un singolo ciclo di clock, mentre il secondo è soddisfatto da una funzione esponenziale di  $m$ .

In linea di principio, l'uso di un LFSR per sostituire le memorie ROM impedisce l'impiego di tecniche di pruning o selezione perché la sequenza casuale non può essere controllata. D'altro canto queste tecniche sono particolarmente utili nei casi in cui è necessario evitare che la dimensione dello strato hidden sia troppo grande.

Dato uno stato del registro l'LFSR produce sempre la stessa sequenza di numeri. In altre parole, il *seed* di un LFSR corrisponde al suo output. Questa caratteristica permette l'uso di una strategia per l'architettura seriale (Fig. 6.1) per cui viene memorizzata solo la prima coppia  $\{s_{1d}, k_{1d}\}$ , mentre gli altri parametri possono essere generati usando l'LFSR. Considerando le architetture seriali il sovraccarico è trascurabile in termini di consumo di area: per ogni neurone è possibile memorizzare solo il punto di partenza del PRNG, cioè  $\{s_{1d}, k_{1d}\}$ , riducendo il numero di parametri da memorizzare da  $N \times D$  a  $N$ . Sfortunatamente, lo stesso ragionamento non vale per il caso dell'architettura parallela in cui il punto di partenza di ogni sequenza corrisponderebbe alla sequenza stessa poiché ogni neurone ha bisogno di essere calcolato in un singolo ciclo di clock.

### 6.3 Analisi sperimentale

La sezione sperimentale fornisce un confronto dettagliato sull'implementazione dei modelli proposti in dispositivi a basso costo. I test sperimentali sono divisi in tre parti. Nella prima vengono forniti test riguardanti l'implementazione dell'architettura seriale, con particolare attenzione al confronto con altre proposte allo stato dell'arte. Successivamente viene presentato un confronto dettagliato tra le architetture presentate nella Sezione 6.1.1 e nella Sezione 6.1.2. Infine nella Sezione 6.2 sono raffigurati i risultati ottenuti sostituendo le memorie ROM tradizionali con un generatore pseudo-casuale.

### 6.3.1 Implementazione dell'architettura seriale

L'architettura descritta in Fig. 6.1 è stata testata su due dispositivi che dispongono di poche risorse, vale a dire Altera CPLD 5M1270Z e Altera CPLD 5M2210Z<sup>1</sup>. Le due piattaforme differiscono per il numero di macrocelle disponibili: la seconda offre un numero quasi un doppio di macrocelle rispetto alla prima. Il primo test è volto a valutare la capacità dell'architettura proposta di adattarsi a tali CPLD. In pratica l'occupazione dell'area è determinata da  $D$  (dimensionalità dell'input) e  $N$  (dimensione dello strato hidden), in quanto caratterizzano la grandezza delle memorie coinvolte. La Fig. 6.5 mostra  $D$  sull'asse  $x$ , mentre sull'asse  $y$  viene riportata la dimensione  $N$  che corrisponderebbe all'occupazione totale dell'area durante la realizzazione del predittore sul dispositivo. In questa figura le barre nere si riferiscono al 5M2210Z, mentre le barre grigie al 5M1270Z. Tutte le implementazioni hanno adottato una rappresentazione a virgola fissa a 12 bit. In termini di consumo energetico, tali schede hanno richiesto, rispettivamente, circa 75  $mW$  e circa 48  $mW$  (con una frequenza di clock fissata a 33MHz in entrambi). Questi risultati confermano che i dispositivi con risorse limitate possono supportare predittori addestrati su problemi di bassa dimensionalità. In effetti è possibile sfruttare le FPGA quando si affrontano problemi caratterizzati da un maggiore numero di features.

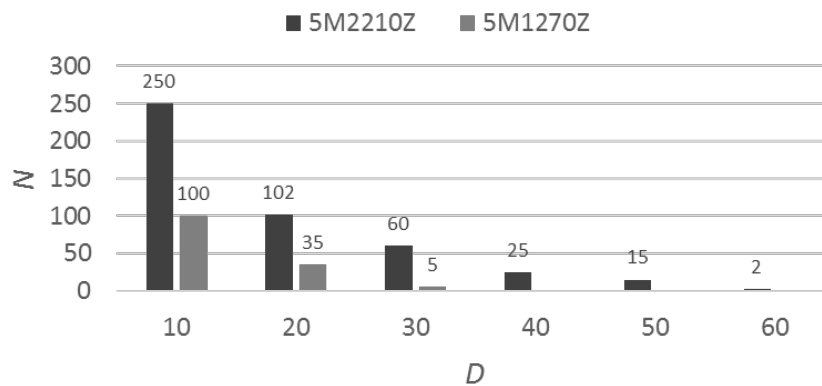


FIGURA 6.5: Configurazioni che corrisponderebbero all'occupazione totale di area quando si realizza un predittore in base all'implementazione seriale del neurone.

Il secondo test mira a fornire un confronto diretto con il design allo stato dell'arte per l'implementazione digitale del classificatore ELM [90]. In [90] il predittore utilizza una ELM standard basata sulla funzione sigmoide; l'addestramento è supportato da una procedura personalizzata che realizza meccanismi di pruning. La realizzazione digitale corrispondente deriva da una strategia di progettazione che combina l'elaborazione parallela e in pipeline, assicurando un risultato di previsione in  $D \times N$  cicli di clock.

<sup>1</sup><https://www.altera.com/products/cpld/max-series/max-v/overview.html>

Per fornire un confronto equo, il test ha coinvolto le implementazioni discusse in [90], realizzate su CPLD 5M1270Z. Le configurazioni del predittore utilizzano tre benchmark, che a loro volta prevedono diverse scelte per  $D$ . Per ogni benchmark sono stati utilizzati due predittori, che differiscono nel numero di neuroni  $N$ : un predittore deriva da una strategia di training convenzionale, mentre l'altro utilizza un meccanismo di pruning, riducendo così la dimensione dello strato hidden.

La Tabella 6.1 riporta i risultati di questo test e confronta, fissati  $D$  e  $N$ , gli utilizzi di area richiesti dal  $HE$  proposto e l'architettura presentata in [90]. La seconda e la terza colonna della tabella si riferiscono ai predittori addestrati su *Breast Cancer Wisconsin* ( $D = 9$ ). La quarta e la quinta sono relative ai predittori addestrati su *Ionosphere* ( $D = 34$ ). Le ultime due riguardano i predittori addestrati su *MNIST* ( $D = 81$ ). L'occupazione di area mostrata non tiene conto del modulo *Input* (come in Fig. 6.1), poiché il classificatore originale progettato in [90] non include tale modulo e adotta la serializzazione per leggere i pattern di input.

TABELLA 6.1: Occupazione di area - CPLD 5M1270Z

$D$	9		34		81	
$N$	4	10	22	80	18	70
$HE$	17%	22%	50%	> 100%	73%	> 100%
[90]	40%	40%	86%	> 100%	95%	> 100%

I risultati numerici confermano l'affidabilità del design proposto. Il predittore  $HE$  occupa sempre un'area inferiore rispetto al corrispondente basato sull'architettura in [90]. In alcuni casi, tuttavia, entrambe le soluzioni adottano un'implementazione che non si adatta all'area disponibile sul CPLD. A questo proposito, è interessante notare che  $HE$  raggiungerebbe l'occupazione totale di area con  $N = 55$  quando  $D = 34$ , e con  $N = 25$  quando  $D = 81$ .

### 6.3.2 Confronto tra l'implementazione seriale e parallela

Le architetture mostrate in Fig. 6.1 e in Fig. 6.3 sono state testate su due dispositivi con poche risorse, ovvero il CPLD Altera 5M2210Z e la FPGA Altera EP4CGX15BF14A<sup>2</sup>, che non include moltiplicatori. La FPGA, di fascia economica, è stata scelta con l'obiettivo di rilassare il vincolo sulle risorse limitate. I test mirano a valutare la capacità delle architetture proposte di adattarsi a tali dispositivi, quando il vincolo sull'occupazione di area è rilassato rispetto alla precedente configurazione sperimentale (Sezione 6.3.1).

La Fig. 6.6 presenta i risultati dei test basati sul CPLD.

<sup>2</sup>Altera FPGAs <https://www.altera.com/products/general/fpga.html>

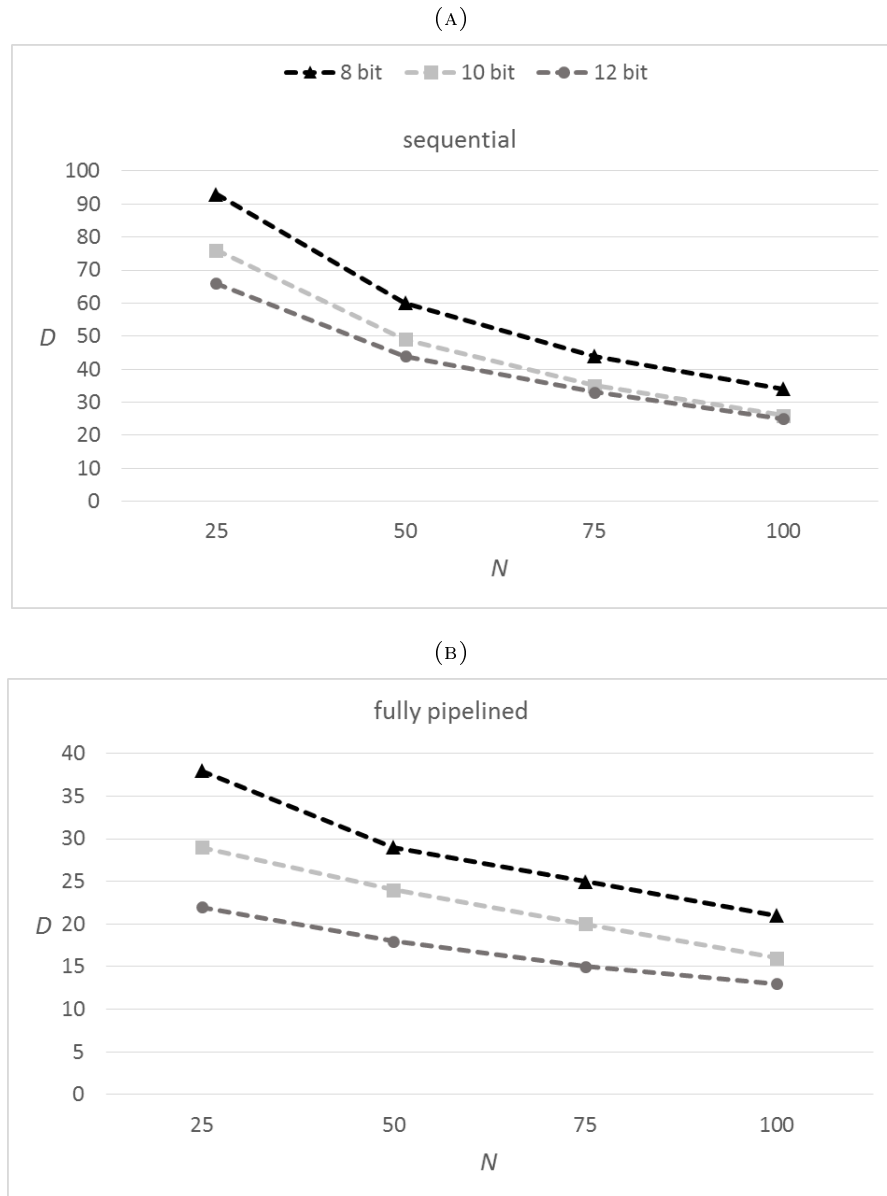


FIGURA 6.6: Test su CPLD: configurazioni che corrisponderebbero all'occupazione totale di area durante la realizzazione hardware del predittore: (a) approccio sequenziale, (b) approccio in pipeline.

La Fig. 6.6a si riferisce all'implementazione hardware dell'approccio sequenziale (Fig. 6.1). La figura presenta il numero di neuroni  $N$  sull'asse  $x$ , mentre l'asse  $y$  mostra la dimensionalità dello spazio di input,  $D$ , che corrisponderebbe all'occupazione totale dell'area sul dispositivo quando si costruisce il predittore. Il test ha coinvolto quattro configurazioni diverse di  $N = \{25, 50, 75, 100\}$ . La figura rappresenta i risultati ottenuti con tre diverse configurazioni per il numero di bit  $B$ : i marcatori neri si riferiscono ai test basati su  $B = 8$ ; quelli grigio chiaro ai test con  $B = 10$ , quelli grigio scuro ai test con  $B = 12$ . Tutte le implementazioni richiedono circa  $130mW$  in termini di consumo energetico, con frequenza di clock fissata a  $55MHz$ . I test hanno mostrato che l'approccio

sequenziale è in grado di supportare un predittore progettato per affrontare un problema di 90 dimensioni, con la configurazione  $\{N = 25, B = 8\}$ . Se si impostasse  $B = 12$  per avere un'accuratezza maggiore,  $D$  si ridurrebbe a 60, considerando sempre  $N = 25$ . Infine, si noti che un problema di 25 dimensioni può essere gestito con la configurazione  $\{N = 100, B = 12\}$ . Questo è un risultato notevole per un'implementazione su un dispositivo di fascia bassa e a basso costo come il CPLD 5M2210Z. La Fig. 6.6b si riferisce all'implementazione hardware dell'approccio in pipeline dei neuroni (Fig. 6.3); la figura adotta lo stesso formato di Fig. 6.6a. Come previsto, fissato  $N$ , l'approccio totalmente in pipeline impone un vincolo più stringente su  $D$  rispetto a quello sequenziale. Ciononostante, la figura mostra che si è in grado di realizzare un predittore progettato per gestire un problema a 37 dimensioni con la configurazione  $\{N = 25, B = 8\}$ ; inoltre un problema con 20 dimensioni può essere supportato con  $\{N = 100, B = 8\}$ . L'approccio parallelo si è dimostrato in grado di ottenere prestazioni interessanti in termini di latenza, poiché la frequenza di clock è stata impostata su  $35MHz$  in tutte le sue realizzazioni. Ciò a sua volta significa che  $T_s = T_p/1.5$  indipendentemente da  $D$ , mentre si avrebbe bisogno di  $T_s = T_p/D$  per ottenere un'implementazione dell'approccio sequenziale che possa competere con la corrispondente realizzazione parallela. Ovviamente, si dovrebbe anche tener conto del trade-off tra latenza e consumo di energia, che nell'approccio completamente pipeline è influenzato da  $D$ . Nelle implementazioni testate, il consumo energetico varia da un minimo di  $150mW$  (con  $D = 13$ ) ad un massimo di  $250mW$  (con  $D = 38$ ).

Complessivamente, è interessante valutare come la quantità  $D \times N$  sia cambiata in funzione di  $N$ , per entrambi gli approcci. A questo scopo sia  $D_N$  il valore assunto da  $D$  quando il numero di neuroni nel predittore è stato impostato su  $N$ . Di conseguenza la Tabella 6.2 fornisce il valore della quantità  $D_{25}/D_N$  per  $N = \{50, 75, 100\}$ . La tabella è organizzata come segue: la prima colonna mostra  $N$ ; la seconda e la terza si riferiscono all'approccio sequenziale e forniscono il valore di  $D_{25}/D_N$  con la configurazione  $B = 8$  e  $B = 12$ , rispettivamente; lo stesso formato è usato per la quarta e quinta colonna, che si riferiscono all'approccio completamente pipeline. In questo caso i risultati numerici mostrano che  $D$ , con l'aumentare di  $N$ , è sceso più lentamente rispetto all'approccio sequenziale. Quindi, l'implementazione sequenziale, fissato  $N$ , può massimizzare la quantità  $D \times N$ , mentre quello parallelo si ridimensiona meglio con la crescita di  $N$ .

TABELLA 6.2: Implementazione su CPLD:  $D_{25}/D_N$  in funzione di  $N$ 

	sequenziale		parallelo	
$N$	B=8	B=12	B=8	B=12
50	1.55	1.50	1.31	1.22
75	2.11	2.00	1.52	1.46
100	2.73	2.64	1.80	1.69

La Fig. 6.7 presenta i risultati dei test basati sulla FPGA.

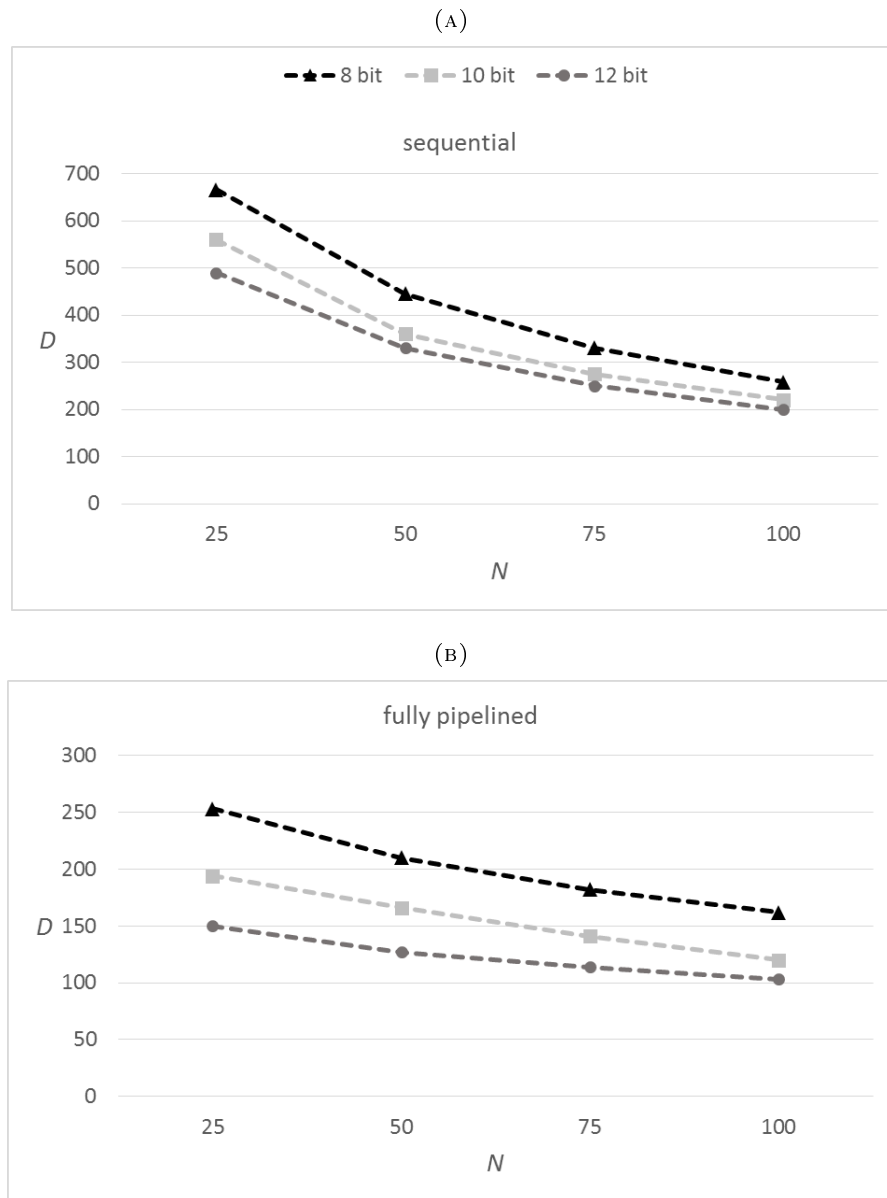


FIGURA 6.7: Test su FPGA: configurazioni che corrisponderebbero all'occupazione totale di area durante la realizzazione hardware del predittore: (a) approccio sequenziale, (b) approccio parallelo.

Con la stessa configurazione della Fig. 6.6, la Fig. 6.7a si riferisce all'implementazione hardware dell'approccio sequenziale, mentre la Fig. 6.7b all'architettura parallela. Entrambe le figure adottano lo stesso formato di Fig. 6.6a. Le realizzazioni su FPGA consentono, a entrambi gli approcci, di supportare predittori che possono gestire  $D > 100$ ; per esempio, la Fig. 6.7a mostra che, adottando l'approccio sequenziale, si potrebbero affrontare problemi a 500 dimensioni. Dunque risulta interessante confrontare, in termini di consumo energetico e latenza, le implementazioni su FPGA con quelle realizzate su CPLD. I predittori supportati dall'architettura sequenziale hanno mostrato un consumo

di energia di circa  $130mW$  sulla FPGA: questo parametro non è cambiato rispetto alle implementazioni su CPLD. Ciononostante la frequenza di clock su FPGA è salita a  $75MHz$ . Le implementazioni basate sull'approccio parallelo richiedono un consumo di energia di  $120mW$  sulla FPGA, con la frequenza di clock fissata a  $50MHz$ . Pertanto, le implementazioni FPGA hanno confermato che  $T_s \simeq T_p/1,5$ .

La Tabella 6.3 riporta il valore della quantità  $D_{25}/D_N$  per  $N = \{50, 75, 100\}$  nel caso delle implementazioni su FPGA. Questa tabella adotta lo stesso formato della Tabella 6.2. I risultati numerici confermano che anche sulle FPGA l'approccio completamente pipeline si adatta meglio al crescere di  $N$ .

TABELLA 6.3: Realizzazione su FPGA:  $D_{25}/D_N$  in funzione di  $N$

$N$	sequenziale		parallelo	
	B=8	B=12	B=8	B=12
50	1.49	1.48	1.20	1.18
75	2.01	1.96	1.39	1.31
100	2.58	2.45	1.56	1.45

### 6.3.3 Confronto tra le implementazioni con ROM e PRNG

Lo scopo di questa fase sperimentale è la quantificazione del guadagno in area introdotto dall'uso di un PRNG per sostituire le memorie ROM, come introdotto nella Sezione 6.2. Le architetture mostrate in Fig. 6.3 e 6.1 sono state testate sui due dispositivi impiegati in Sezione 6.3.2, usando la stessa impostazione della sottosezione 6.3.2. I test mirano a valutare la capacità delle architetture proposte di adattarsi a tale dispositivi.

La Fig. 6.8 presenta i risultati dei test basati sul CPLD.

La Fig. 6.8a si riferisce all'implementazione hardware dell'approccio sequenziale. La figura mostra  $N$  sull'asse  $x$ , mentre l'asse  $y$  rappresenta la dimensionalità dello spazio di input  $D$  che corrisponderebbe all'occupazione totale di area sul dispositivo durante l'implementazione hardware del predittore. Il test ha coinvolto quattro configurazioni diverse di  $N : \{25, 50, 75, 100\}$ . La figura mostra, inoltre, i risultati ottenuti con due impostazioni diverse per il numero di bit  $B$ : le linee nere si riferiscono ai test con  $B = 8$ ; quelle grigio chiaro riguardano i test con  $B = 12$ . I marcatori circolari si riferiscono all'implementazione della ROM standard, mentre i diamanti rappresentano il PRNG. La frequenza di clock è impostata a  $20MHz$  per tutte le realizzazioni. I test hanno mostrato che le configurazioni basate su PRNG permettono sempre, a parità di  $N$ , di risolvere problemi con maggiore dimensionalità rispetto a quelle basate sulla memoria standard. Inoltre all'aumentare di  $N$ , il divario aumenta notevolmente. È interessante notare che

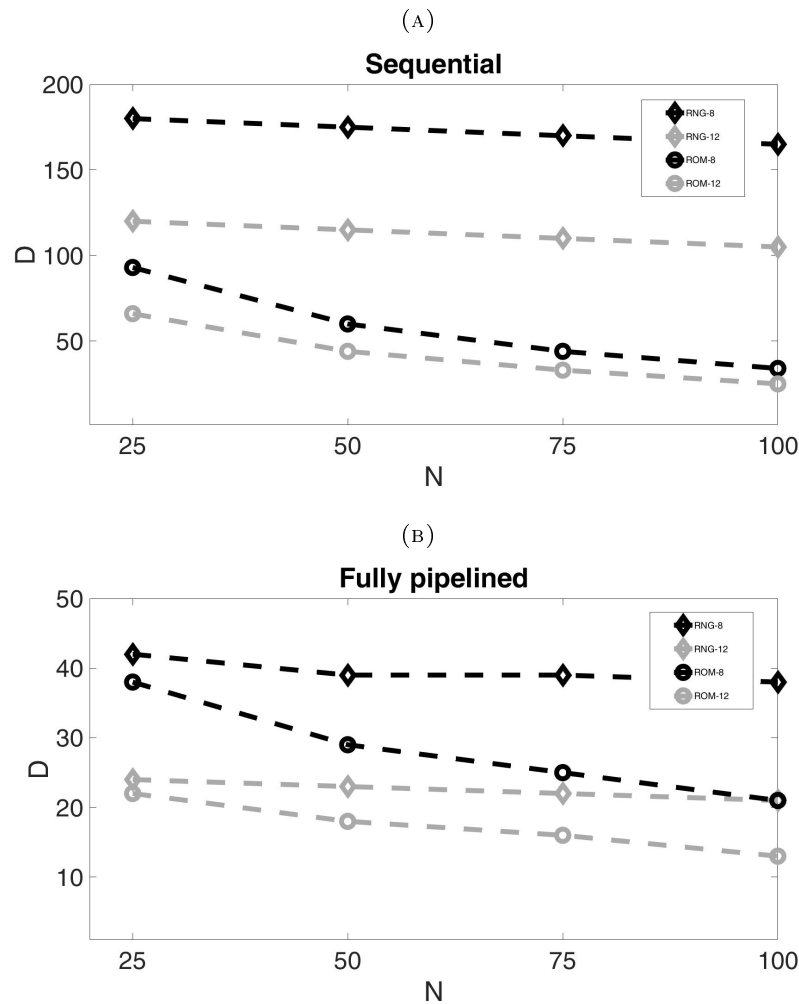


FIGURA 6.8: Test su CPLD: configurazioni che corrisponderebbero all'occupazione totale di area quando si realizza su hardware il predittore con memorie rom standard (diamanti come marcatori) o PRNG (marcatori circolari): (a) approccio sequenziale; (b) approccio parallelo.

la configurazione basata su aritmetica a 8 bit può supportare l'implementazione di una rete con  $N = 100$  e  $D = 165$ .

La Fig. 6.8b si riferisce all'implementazione hardware dell'approccio completamente pipeline; la figura adotta lo stesso formato di Fig. 6.8a. Come previsto, fissato  $N$ , l'implementazione parallela impone vincoli più rigidi su  $D$  rispetto a quella sequenziale. Come nel caso precedente, le soluzioni basate sul PRNG mostrano performance sempre migliori rispetto all'implementazione classica in termini di occupazione dell'area, a parità di  $N$  e  $D$ . Questo vantaggio diventa particolarmente evidente quando  $N = 100$ : per la soluzione basata su ROM il limite è  $D = 21$ , mentre il PRNG può gestire classificatori con  $D = 38$ .

La Fig. 6.9 eredita il formato dalla Fig. 6.8. Il risultato si riferisce all'implementazione delle architetture su FPGA dove la frequenza di clock è  $50MHz$  e le scelte possibili per  $D$  sono  $\{100; 500; 1000\}$ . I risultati confermano la tendenza mostrata nelle precedenti



simulazioni. L'osservazione più importante è che l'implementazione con PRNG consente configurazioni con un elevato numero di neuroni e un'alta dimensionalità dello spazio dei campioni: un predittore basato sull'implementazione del neurone seriale, con una dimensione dello strato hidden pari a  $N = 1000$ , può gestire i dati di input con dimensionalità superiore a 1200. Invece per l'input nell'architettura completamente pipeline è limitata a 200, ma la latenza diventa  $\simeq 200$  volte più piccola.

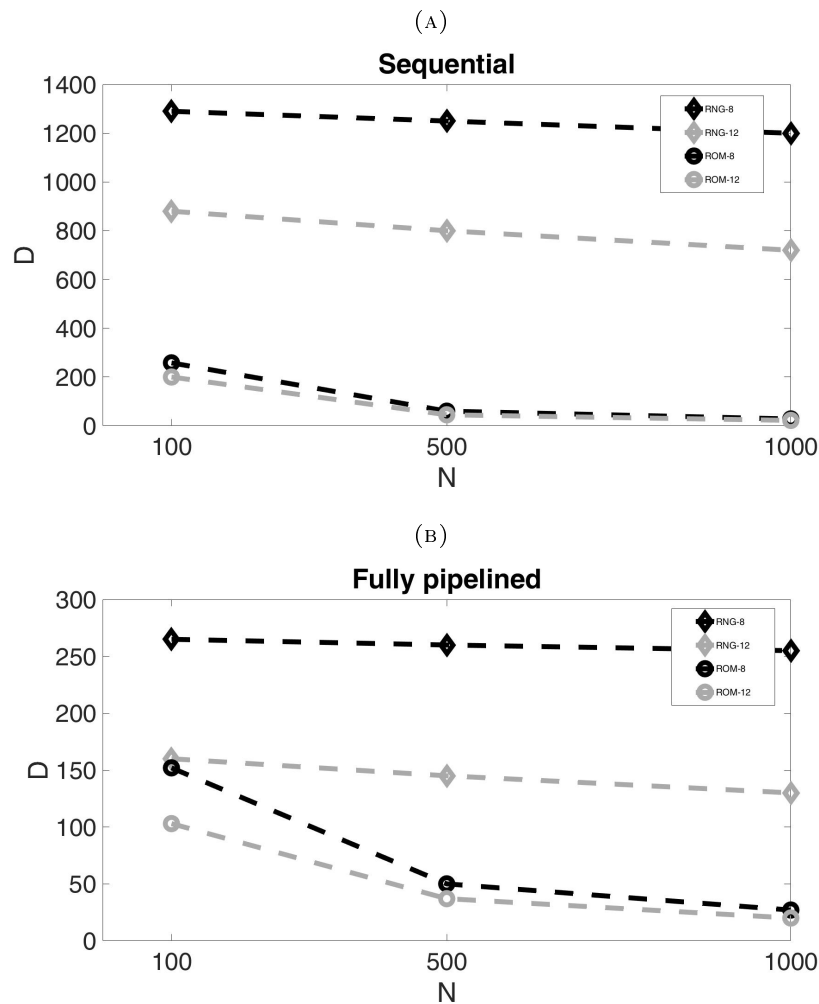


FIGURA 6.9: Test su CPLD: configurazioni che corrisponderebbero all'occupazione totale di area quando si realizza su hardware il predittore con memorie rom standard (diamanti come marcatori) o PRNG (marcatori circolari): (a) approccio sequenziale, (b) approccio parallelo.

## 6.4 Considerazioni teoriche finali

Questo capitolo ha presentato architetture digitali progettate per implementare un classificatore basato su reti neurali casuali su dispositivi a basso costo. La ricerca si è concentrata su realizzazioni che adottano l'hard-limiter come funzione di attivazione.

Questa configurazione può trarre vantaggio da due attributi: 1) può portare a una progettazione *hardware-friendly* e 2) la letteratura fornisce schemi di apprendimento che possono bilanciare adeguatamente le prestazioni di generalizzazione e le dimensioni dello strato hidden. La seconda caratteristica è degna di nota in quanto la quantità di neuroni gioca un ruolo fondamentale nel determinare l'utilizzo delle risorse del predittore finale. Sono state sfruttate le peculiarità delle reti casuali per ridurre i requisiti di memoria. Le due architetture delineate nel capitolo affrontano il trade-off tra l'utilizzo dell'area e la latenza da diversi punti di vista. L'approccio che si basa su un calcolo sequenziale delle attivazioni dei neuroni dà la priorità all'occupazione dell'area. Come risultato principale le risorse utilizzate dal modulo *Neuron* non dipendono da  $D$ . L'architettura che impiega il calcolo parallelo delle features dà la priorità alla latenza. In questa configurazione, l'area occupata da una singola istanza del modulo *Neuron* aumenta con l'aumentare di  $D$ . Verifiche sperimentali hanno dimostrato che la strategia di progettazione proposta supporta la realizzazione di classificatori integrati sia nei dispositivi FPGA di fascia economica sia in quelli a basso costo come i CPLD, ottenendo un notevole guadagno rispetto ai precedenti approcci allo stato dell'arte.

## 6.5 Esempio applicativo sulla classificazione delle scariche parziali

Si è deciso di utilizzare l'architettura seriale (sottosezione 6.1.1) per realizzare un classificatore binario da impiegare nel problema del riconoscimento delle scariche parziali.

Al fine di verificare le prestazioni di questa architettura sono stati creati tre dataset, costituiti da scariche misurate in laboratorio. Per ottenere tre tipi diversi di difetto è stata applicata un'onda sinusoidale a tre provini appositamente creati: filo attorcigliato (tecnicamente conosciuto come *twisted pair*) (TP), elettrodo conduttivo a punta sottile contro piano isolato (ECS) ed elettrodo conduttivo piano contro piano isolato (ECP). Per campionare i segnali delle scariche è stato utilizzato un sistema di acquisizione a 8 bit avente banda nel range  $0 \div 200MHz$  e con massima frequenza di campionamento pari a  $1GSamples/s$ . L'impedenza di misura è un RLC. Al fine di ottenere il PD Pattern il sistema è attivato (*trigger*) da un'onda sinusoidale a  $50Hz$  e i segnali campionati a  $500MSamples/s$ . In Fig. 6.10 viene riportato il sistema di misura.

Per ognuna delle tre sorgenti di scarica sono stati raccolti 100 Pattern PD. Da ciascuno di questi sono state calcolate quattro distribuzioni: numero-ampiezza, numero-fase, carica-fase e carica media-fase [119]. Per il problema della classificazione sono state scelte come features quelle riguardanti la forma delle distribuzioni e altri parametri statistici (Sezione

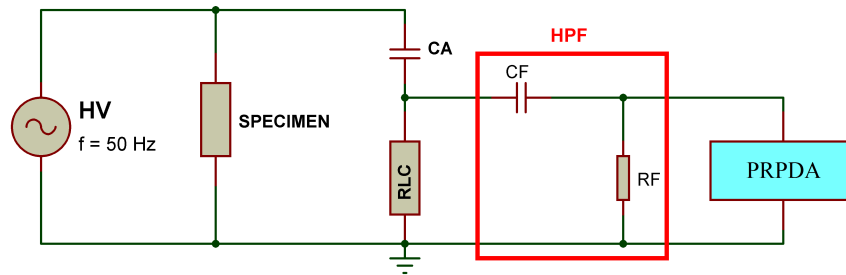


FIGURA 6.10: Sistema di acquisizione dei segnali di scarica.

3.1.1). Le 32 features selezionate sono riepilogate nella Tabella 6.4 ( $\gamma$  è la *Skewness*,  $\kappa$  la *Kurtosi*,  $\beta$  proviene dalla distribuzione di Weibull e *MSE* è l'errore quadratico medio di  $\beta$ ). Sono stati creati tre dataset bi-classe contenenti 200 dati ciascuno combinando i tre siti di scarica precedenti: TP vs ECS, TP vs ECP e ECS vs ECP. Ogni riga della matrice di ciascun dataset rappresenta un'osservazione e le colonne sono le features normalizzate nell'intervallo  $[-1; +1]$ . In questo modo è stato possibile testare le prestazioni dell'architettura seriale.

TABELLA 6.4: Features per ciascuna distribuzione.

Distribuzioni\Cicli alimentazione	Semi-Ciclo Positivo	Semi-Ciclo Negativo
Ampiezza - Numero	$\gamma, \kappa, \beta, MSE$	$\gamma, \kappa, \beta, MSE$
Fase - Numero	$\gamma, \kappa, \text{moda}, \text{media}$	$\gamma, \kappa, \text{moda}, \text{media}$
Fase - Carica	$\gamma, \kappa, \text{moda}, \text{media}$	$\gamma, \kappa, \text{moda}, \text{media}$
Fase - Carica Media	$\gamma, \kappa, \text{moda}, \text{media}$	$\gamma, \kappa, \text{moda}, \text{media}$

### 6.5.1 Test Sperimentali

Ciascun dataset, dopo essere stato etichettato (es. per il dataset TP vs ECP, i dati provenienti dal primo sono stati etichettati come +1 e i secondi come -1), è stato diviso in 10 *cross-fold*: il dataset originale viene randomicamente diviso in 10 parti uguali. Dei 10 sottocampioni generati, uno viene impiegato per la fase di validazione mentre gli altri sono utilizzati per il training e per la fase di test (ovvero la *model selection* per la scelta del parametro ottimo di regolarizzazione). Questo processo viene ripetuto 10 volte in modo tale da utilizzare tutti i 10 sottocampioni per la validazione. I risultati di ciascuna operazione sono stati mediati in modo tale da ottenere un unico errore di classificazione. Il vantaggio di questo metodo è che tutte le osservazioni vengono utilizzate sia per il training, per il test e per la validazione del modello e che vengono impiegate esattamente una volta in fase di validazione.

Per un confronto completo si è deciso di paragonare i risultati provenienti da questa architettura, che sarà chiamata ELM-HW, con quelli provenienti dalla SVM e dalla ELM

Standard (ELM-ST) ovvero che non tiene conto dei numeri espressi con una quantità limitata di bit. Le fasi di training, di test e di validazione della SVM sono state affrontate con l'algoritmo della cross-fold. Per le due macchine ELM sono state scelte quattro configurazioni di neuroni nello strato hidden:  $N = \{50, 100, 250, 500\}$ . Inoltre, per ciascuna cross-fold, i parametri random sono stati estratti 30 volte e i risultati sono stati mediati. I pesi  $\beta$  provenienti dalla fase di test sono stati trasformati in numeri a virgola fissa a 12 bit per essere salvati nell'opportuna memoria dell'architettura seriale (Fig. 6.1). La *model selection* delle due ELM riguarda soltanto il parametro  $\lambda$  (5.7); mentre, per la SVM, la model selection è stata condotta sia sul parametro di regolarizzazione  $\lambda$  sia sul parametro libero della funzione kernel utilizzata (in questo esempio è stata impiegata la RBF per cui il parametro libero rappresenta la larghezza della RBF). I risultati sono stati riportati nella Tabella 6.5.

TABELLA 6.5: Errori percentuali di generalizzazione

Dataset \ Neuroni	ELM-HW				ELM-ST				SVM
	N=50	N=100	N=250	N=500	N=50	N=100	N=250	N=500	
TP vs ECS	11.3	10.9	11.1	11.4	11.6	11.5	11.4	11.7	12.5
TP vs ECP	0.6	0.3	0.1	0.0	0.1	0.3	0.4	0.3	1.0
ECP vs ECS	15.3	15.8	15.0	14.9	19.4	18.7	17.4	16.8	14.0

È importante notare che la ELM-HW raggiunge ottimi risultati, anche migliori di SVM, per quanto riguarda i primi due dataset. Considerando i primi due dataset gli errori di classificazione di ELM-HW e di ELM-ST sono simili, mentre per l'ultimo sono migliori nella ELM-HW. La differenza nel terzo dataset tra la ELM-HW e SVM non è così rilevante. Per cui si può affermare che l'architettura hardware di una rete neurale randomica basata sulla funzione di attivazione hard-limiter può essere utilizzata per la classificazione supervisionata di scariche parziali. Da un punto di vista hardware si può notare che l'errore di classificazione non varia eccessivamente al variare del numero di neuroni, per cui è possibile realizzare un dispositivo hardware che impieghi soltanto 50 neuroni, occupando dunque meno area possibile.

Infine, considerando gli errori percentuali dei tre dataset a confronto, si osserva che nel secondo problema (TP vs ECP) si ottengono risultati nettamente migliori rispetto agli altri due casi. Il motivo potrebbe essere legato alla scelta del tipo di features estratte, che potrebbero rappresentare meglio i dati rispetto agli altri due dataset, permettendo dunque una classificazione con errore minore. Un altro possibile motivo potrebbe essere legato alla distribuzione di probabilità dei dati di ingresso: in fase di training la rete neurale apprende la distribuzione di probabilità da cui vengono estratti i dati; in fase di validazione si estraggono nuovi dati per essere processati dall'algoritmo e dunque classificati. Nel caso di TP vs ECP la rete neurale ha appreso in fase di training una distribuzione rappresentativa dell'intero spazio dei dati, poiché in fase di validazione ha

ottenuto errori molto bassi. Per quanto riguarda gli altri due dataset, la rete neurale ha appreso una buona rappresentazione dello spazio dei dati di ingresso (gli errori sono comunque bassi in percentuale), ma non tale da ottenere risultati comparabili a TP vs ECP. Se si utilizzassero altre features per TP vs ECS e ECP vs ECS si potrebbero ottenere risultati migliori, poiché potrebbero essere più rappresentativi degli spazi in questione.

### 6.5.2 Test delle performance hardware

L'architettura è stata realizzata su due dispositivi dell'Altera: la FPGA EP4CE10E22A7 e il CPLD 5M2210Z. Lo scopo di questo test è valutare l'occupazione di area in funzione della differente configurazione del numero di neuroni. Inoltre vengono riportati altri due parametri importanti: la massima frequenza di clock e il consumo in potenza. Nella Tabella 6.6 e nella Tabella 6.7 vengono riportati i risultati relativi ai due dispositivi.

TABELLA 6.6: Realizzazione su FPGA.

	<b>N=50</b>	<b>N=100</b>	<b>N=250</b>	<b>N=500</b>
Occupazione di Area (%)	16	26	54	98
Consumo in Potenza (mW)	77.4	80	95.5	118.7
Massima Freq Clock (MHz)	86.96	91.21	96.47	104.5

TABELLA 6.7: Realizzazione su CPLD.

	<b>N=50</b>	<b>N=100</b>	<b>N=250</b>	<b>N=500</b>
Occupazione di Area (%)	76	119	~	~
Consumo in Potenza (mW)	107	~	~	~
Massima Freq Clock (MHz)	57.61	~	~	~

Nella Tabella 6.7 si nota che con 100 neuroni il CPLD non dispone di unità logiche sufficienti per implementare l'architettura. Con una simulazione ulteriore si è calcolato che il massimo numero di neuroni che il dispositivo può contenere è  $N = 75$  con un consumo pari a  $129.45mW$  e una frequenza di clock di  $57.35MHz$ .

Nella Tabella 6.5 è stato mostrato che con 50 neuroni si ottiene un buon errore di generalizzazione. Per cui il CPLD, ovvero un dispositivo low-cost, può essere utilizzato per la classificazione delle scariche. Se è necessario disporre di un dispositivo più performante è possibile impiegare la FPGA.

### 6.5.3 Considerazioni finali

Nella Sezione 6.5 è stato mostrato un esempio applicativo di classificazione supervisionata impiegando l'algoritmo della ELM con funzione di attivazione hard-limiter. L'algoritmo è

stato implementato su due dispositivi hardware secondo lo schema proposto nella Sezione 6.1.1. Le prove sperimentali sono state eseguite su tre dataset binari, ottenendo buoni risultati in fase di classificazione anche comparandoli con quelli ottenuti da una kernel SVM.

Questa architettura è in grado di classificare qualsiasi tipologia di scarica purché questa sia stata etichettata e quindi presentata al classificatore in fase di training. Inoltre è possibile trasformare il classificatore da binario a multi-classe aggiungendo delle memorie per i pesi  $\beta$ . Se si chiama  $NC$  il numero di classi totale si devono risolvere  $\frac{NC!}{(NC-2)! \cdot 2!}$  problemi binari e, di conseguenza, il numero di memorie dei pesi  $\beta$  diventa  $\frac{NC!}{(NC-2)! \cdot 2!}$ . Per ultimare il classificatore multi-classe è necessario implementare un arbitro che classifichi le scariche parziali per maggioranza di voti. Questi accorgimenti ovviamente rendono l'architettura molto più pesante in termini di occupazione di area, perciò sarà necessario disporre di un dispositivo più capiente dal punto di vista delle unità logiche.

## Capitolo 7

# Classificazione non supervisionata

Durante il monitoraggio di un isolamento elettrico è possibile che due o più sorgenti di scarica siano attive contemporaneamente. Ne consegue che, durante il processo di acquisizione, vengano salvati segnali nel tempo provenienti da tutte le sorgenti attive. Non sapendo quali e quanti difetti stiano generando impulsi di scarica è necessario adottare tecniche di apprendimento automatico non supervisionate con lo scopo di separare le diverse fonti.

Le varie sorgenti di impulsi presentano morfologie differenti e, inoltre, si trovano a distanze diverse rispetto alla collocazione del sistema di misura. Per questi motivi, i segnali generati dal passaggio di elettroni attraverso il sistema di isolamento vengono condizionati diversamente in funzione del luogo e dal fenomeno per cui vengono originati.

In questo capitolo è stato svolto uno studio delle tecniche esistenti in letteratura. Inoltre sono state effettuate delle prove sperimentali per confrontare l'efficacia degli algoritmi non supervisionati nel separare diverse tipologie di scariche.

### 7.1 Acquisizione sperimentale dei dati

Il sistema di acquisizione delle scariche è riportato in Fig. 7.1. Al fine di campionare i segnali relativi alle scariche è stata applicata una tensione sinusoidale, con una frequenza pari a  $50\text{Hz}$ , ad una bobina con isolamento in nastro micato (Fig. 7.2).

I segnali nel tempo, dopo essere stati filtrati con un filtro passa alto per eliminare i disturbi relativi alla tensione di alimentazione, sono stati acquisiti con il Picoscope 5444D. Questo oscilloscopio permette di configurare il numero di bit con cui vengono rappresentati i dati campionati. Un numero maggiore di bit comporta una diminuzione della frequenza di

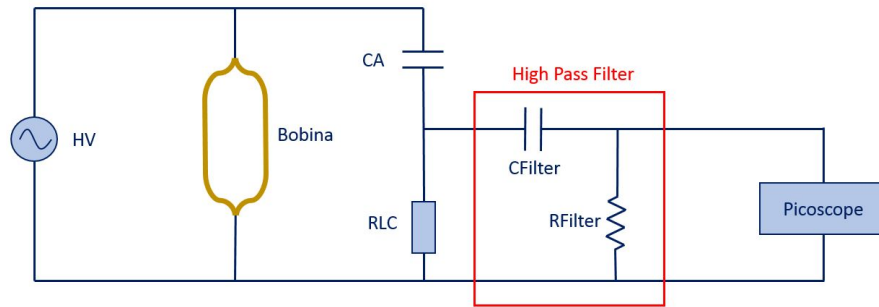


FIGURA 7.1: Sistema di acquisizione.

campionamento. Per questa tipologia di esperimenti è stato scelto un numero di bit pari a 12, con una frequenza di campionamento risultante pari a  $250\text{MSamples/s}$ . L'oscilloscopio è stato configurato per campionare e memorizzare un massimo di 36 schermate da  $20\text{ms}$  ciascuna, cioè pari ad un periodo della tensione di alimentazione.

Per ottenere diverse tipologie di difetto sono state prese in considerazione due bobine con sorgenti di scarica differenti. I segnali relativi alle due sorgenti vengono riportati in Fig. 7.3. In Fig. 7.4 sono mostrati i relativi PD pattern. I dati, successivamente, sono stati processati in modo tale da ottenere un dataset contenente 2000 segnali da 500 campioni ciascuno. I due dataset contengono 1000 osservazioni provenienti dalla prima bobina e 1000 provenienti dalla seconda. Per le prove di riconoscimento automatico si è supposto di non avere alcuna informazione a priori sulle scariche campionate (quanti difetti sono presenti nel sistema di isolamento), quindi i dati sono stati prima elaborati



FIGURA 7.2: Bobina con isolamento in nastro micato.



con tecniche di estrazione delle features introdotte nel Capitolo 3 per poi essere classificati con algoritmi non supervisionati (Sezione 3.2). inoltre in un caso reale è possibile che due o più difetti caratterizzino il sistema di isolamento di una stessa bobina, per cui è necessario impiegare tecniche non supervisionate in fase di classificazione.

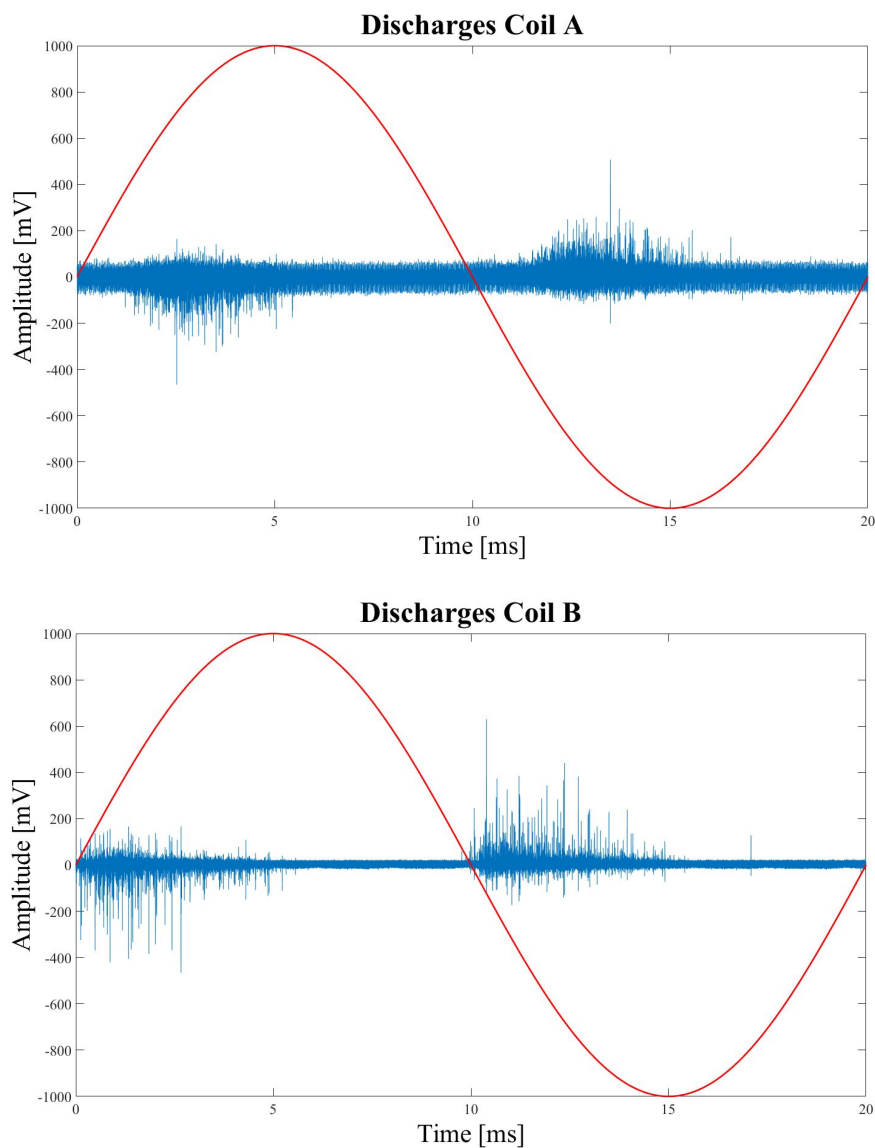


FIGURA 7.3: Segnali provenienti dalle sorgenti di scarica delle due bobine. I segnali riportati sono stati campionati in un periodo della sinusoide a  $50Hz$ .

## 7.2 Estrazione delle features

In queste prove sperimentali sono state impiegate 3 tecniche di estrazione delle features introdotte nella Sezione 3.1. Inoltre questi algoritmi sono stati applicati sia ai segnali originali sia alle funzioni di autocorrelazione calcolate sui segnali nel tempo. Supponendo di non conoscere a priori il numero di difetti presenti nel dataset, si confrontano le tecniche

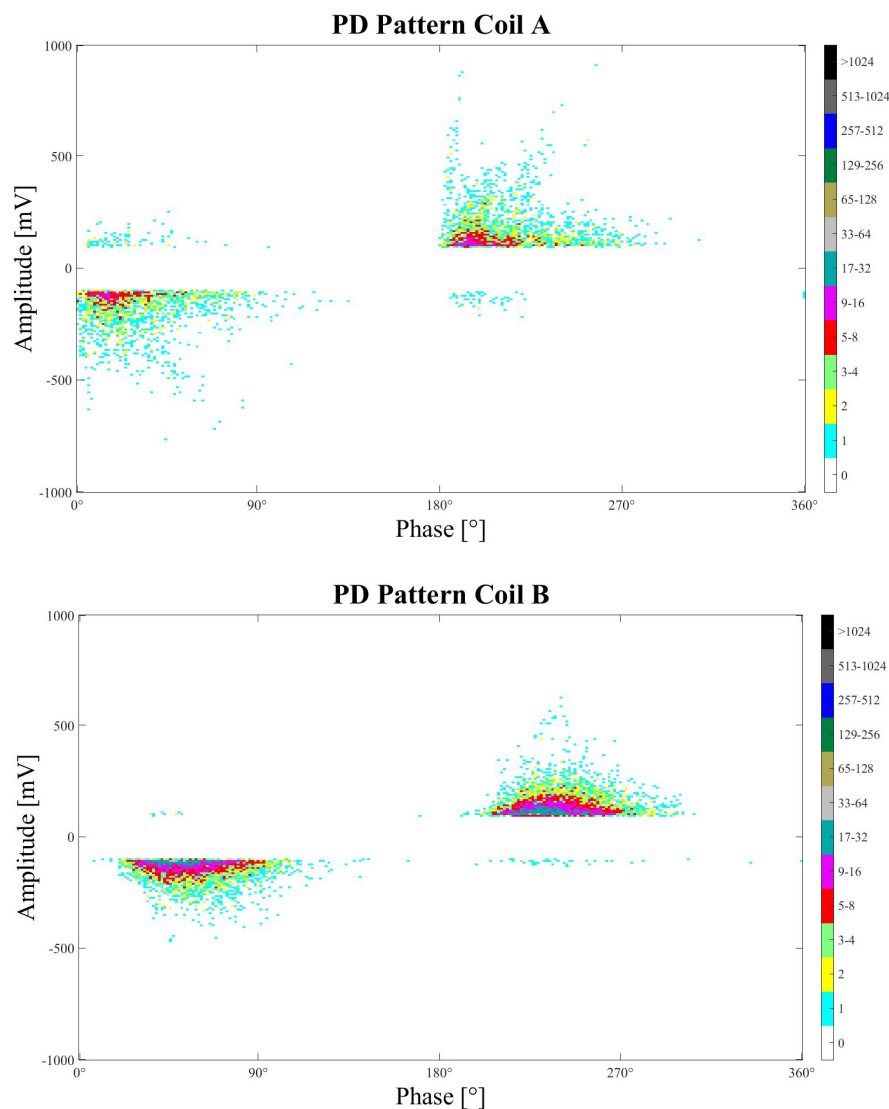


FIGURA 7.4: PD Pattern relativi alle sorgenti di scarica delle due bobine.

di riduzione della dimensionalità del dato originale cercando l'algoritmo che permetta di visualizzare meglio le differenze nello spazio ri-mappato. Ovvero si cercano le tecniche che permettano, in prima battuta, di visualizzare cluster distinti.

Le tre tecniche che vengono confrontate sono: t-SNE (Sezione 3.1.4), PCA (Sezione 3.1.3) e mappa TW (Sezione 4.3.1). Tutti e tre gli algoritmi sono stati utilizzati per mappare i dati a 500 dimensioni (i campioni nel tempo) in uno spazio bi-dimensionale, per permettere una facile visualizzazione.

### 7.2.1 Mapping dei dati con t-SNE

Questo algoritmo è basato sulla minimizzazione di una funzione costo non lineare, la cui soluzione rappresenta un minimo locale dipendente dal punto di partenza dell'algoritmo e

la metrica per la distanza. Nelle prove sperimentali eseguite sui dataset dei dati originali e delle funzioni di autocorrelazione si è deciso, per ciascuna metrica della distanza, di trovare 5 soluzioni differenti per il mapping, tenendo in considerazione quella che presenta la funzione costo più piccola. Le 4 metriche per la distanza sono: *Euclidean*, *Chebychev*, *Cosine* e *Cityblock*.

In Fig. 7.5 viene riportato il mapping relativo ai dati originali, mentre in Fig. 7.6 quello calcolato a partire dalle funzioni di autocorrelazione.

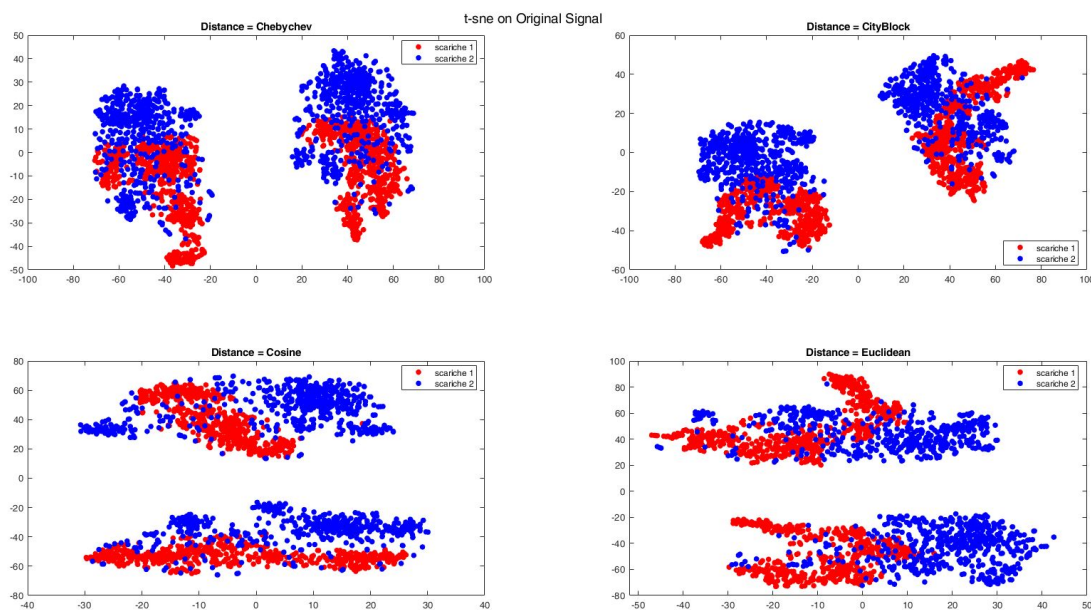


FIGURA 7.5: t-SNE applicato ai dati originali.

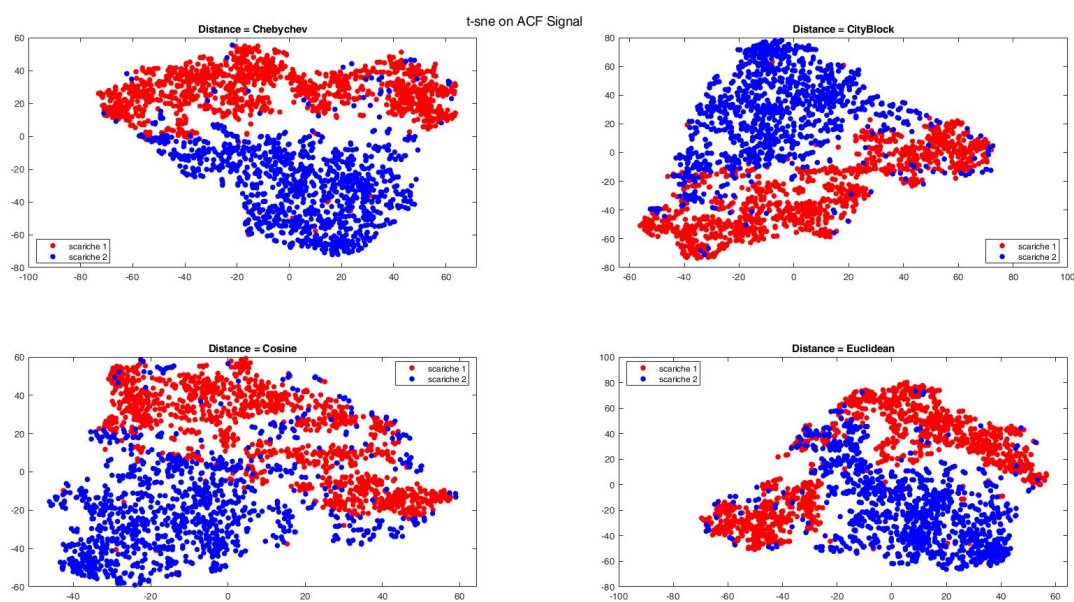


FIGURA 7.6: t-SNE applicato alle funzioni di autocorrelazione.

I due colori differenti servono ad indicare la provenienza dei dati da due fonti di scarica. Per cui, anche se in Fig. 7.5 si osservano, per tutte le metriche, cluster ben distinti, in realtà si otterrebbe una separazione sbagliata delle due classi poiché ciascun cluster è costituito da parte dei dati provenienti dal primo difetto e parte dal secondo. In Fig. 7.6, anche se i cluster risultano meno separabili a prima vista, in realtà il mapping evidenzia una buona (probabile) separabilità delle due sorgenti di scariche. Se non si conoscessero a priori i difetti si sceglierebbe la metrica che ha portato a divisioni dei dati più evidenziabili. Al fine di separare i diversi difetti, è stato deciso, per i motivi sopra elencati, di utilizzare gli algoritmi di clustering nel caso del t-SNE con distanza di Chebychev, applicato alle funzioni di autocorrelazione. Se non si avesse una minima conoscenza della provenienza dei dati di input, l'utilizzo di questo algoritmo porterebbe ad una classificazione completamente sbagliata poiché si sceglierebbe uno dei quattro mapping in Fig. 7.5 che presenta cluster ben distinti.

### 7.2.2 Mapping dei dati con PCA

In Fig. 7.7 e Fig. 7.8 vengono riportati, rispettivamente, i mapping effettuati con l'algoritmo della PCA sui dati originali e sulle funzioni di autocorrelazione.

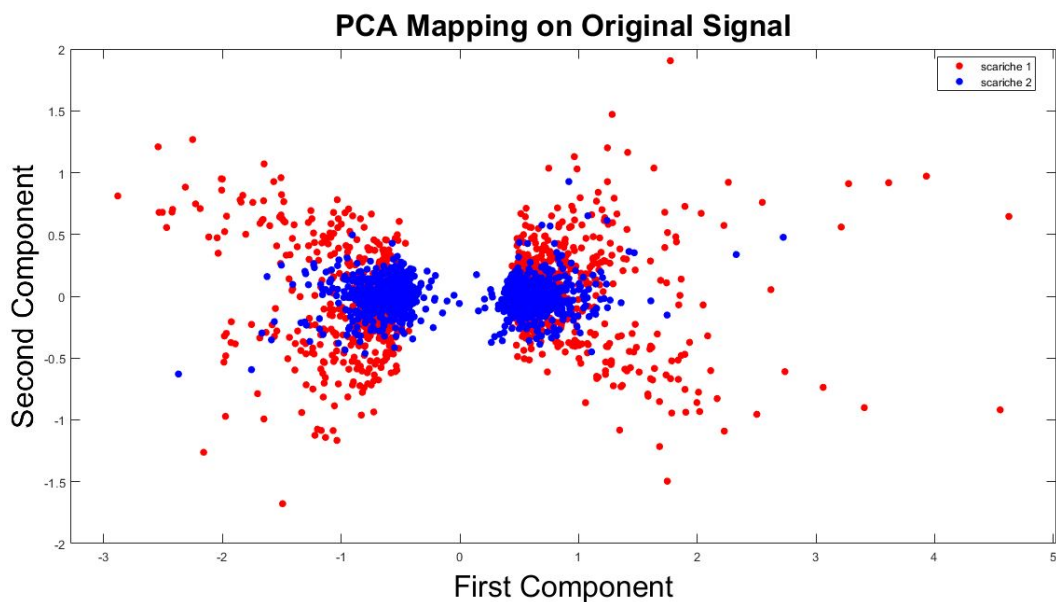


FIGURA 7.7: PCA applicata ai dati originali.

In Fig. 7.7 si formano due cluster facilmente distinguibili ma, come nel caso del t-SNE per i dati originali, questi cluster sono costituiti da punti appartenenti ad entrambe le sorgenti di scarica, per cui la separazione risulterebbe sbagliata. In Fig. 7.8 le funzioni di autocorrelazione sono state proiettate in uno spazio in cui risulta impossibile applicare

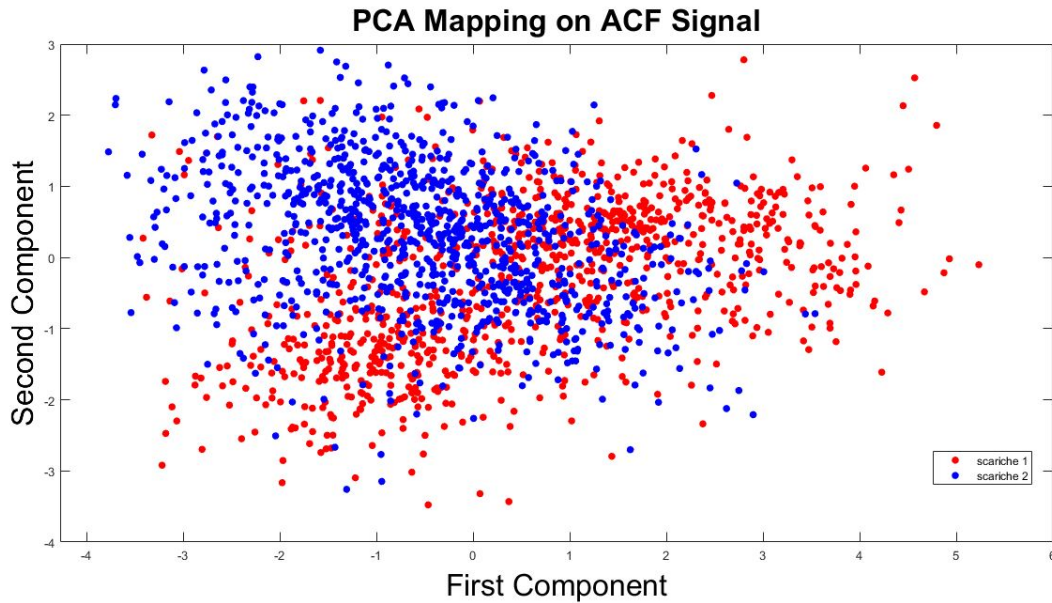


FIGURA 7.8: PCA applicata alle funzioni di autocorrelazione.

un algoritmo di separazione poiché i dati non formano cluster distinguibili. Per cui si è deciso di escludere la PCA per la classificazione dei difetti in queste prove sperimentali.

### 7.2.3 Mapping dei dati con TW

I dati originali e le funzioni di autocorrelazione sono stati proiettati in uno spazio bi-dimensionale utilizzando le equazioni (4.1), (4.2) e (4.3). In Fig. 7.9 e Fig. 7.10 vengono riportati, rispettivamente, i mapping effettuati con il metodo del TW sui dati originali e sulle funzioni di autocorrelazione.

Per entrambi i mapping si possono distinguere due insiemi. Quindi verranno utilizzati algoritmi di clustering per la separazione dei difetti in entrambi gli spazi TW. In Fig. 7.9 si può notare che un cluster è molto allungato, per cui è necessario utilizzare tecniche che riescano a raggruppare queste morfologie.

## 7.3 Classificazione non supervisionata

Per la separazione delle diverse sorgenti di scarica sono utilizzati gli algoritmi descritti nella Sezione 3.2: K-means, DBSCAN e OPTICS. Queste tecniche sono state applicate agli spazi rimappati; in particolare agli spazi bi-dimensionali che presentano già a prima vista una possibile divisione: t-SNE applicato alle funzioni di autocorrelazione con distanza di Chebychev e TW applicato sia ai dati originali che alle funzioni di autocorrelazione.

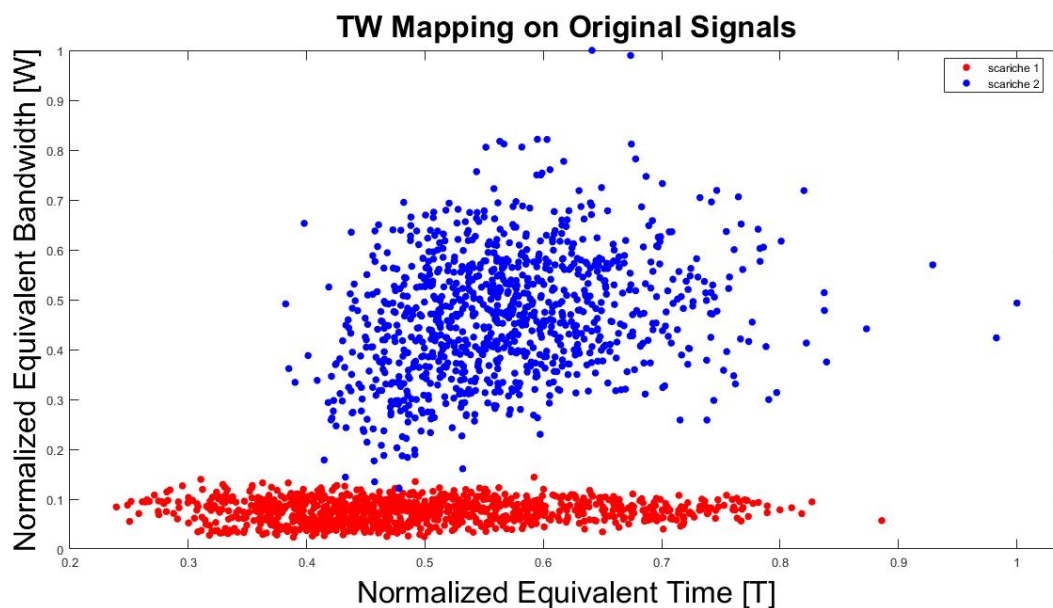


FIGURA 7.9: TW applicato ai dati originali.

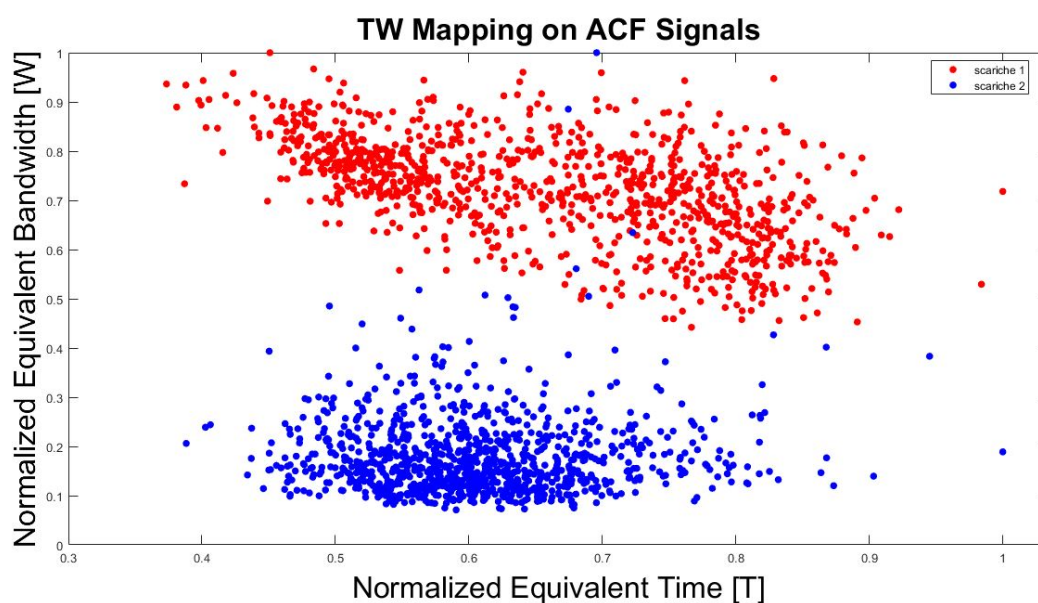


FIGURA 7.10: TW applicato alle funzioni di autocorrelazione.

### 7.3.1 K-means

Siccome si è supposto di non conoscere il numero di difetti a priori, non si può fissare il numero di cluster in cui vengono raggruppati i dati. Normalmente si procede variando il numero dei possibili cluster e viene stabilito scegliendo il punto in cui il grafico della funzione costo, al variare del numero di cluster, presenta un "gomito". La funzione costo è rappresentata dalla somma delle distanze di ciascun cluster dal rispettivo centroide. Inoltre, siccome il K-means fornisce una soluzione sub-ottima, è necessario ripetere l'algoritmo più volte, variando la posizione iniziale dei centroidi, e salvare la soluzione che

presenta il valore più basso per la funzione costo. In Fig. 7.11, Fig. 7.12 e Fig. 7.13 sono raffigurate le distanze dei punti dai rispettivi centroidi, al variare del numero di cluster, per i tre diversi mapping.

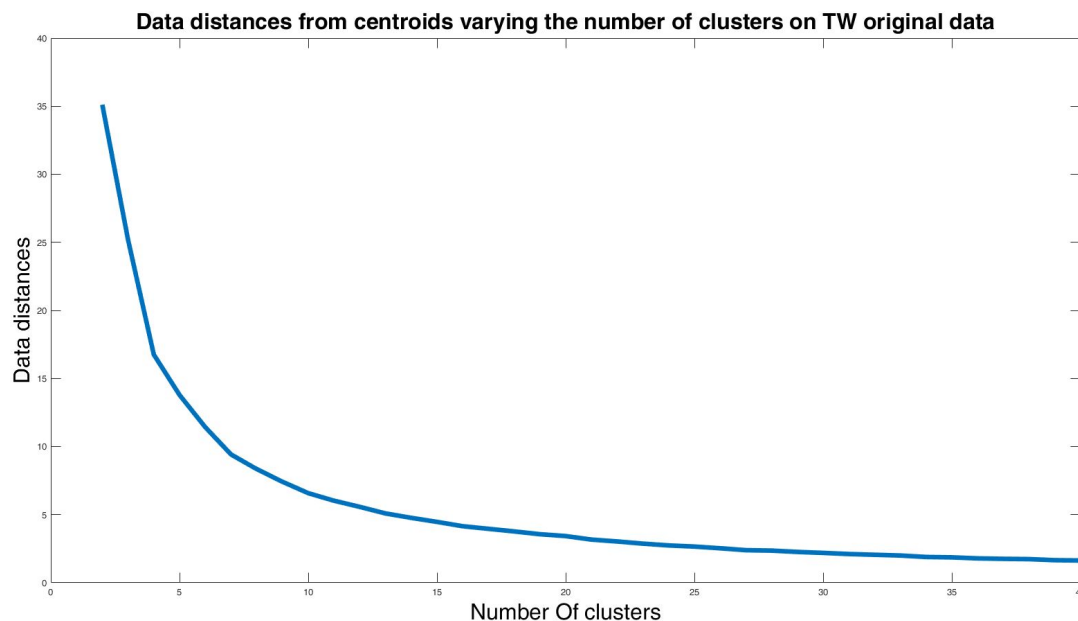


FIGURA 7.11: Distanze dei dati dai rispettivi centroidi al variare del numero di cluster nello spazio TW dei dati originali.

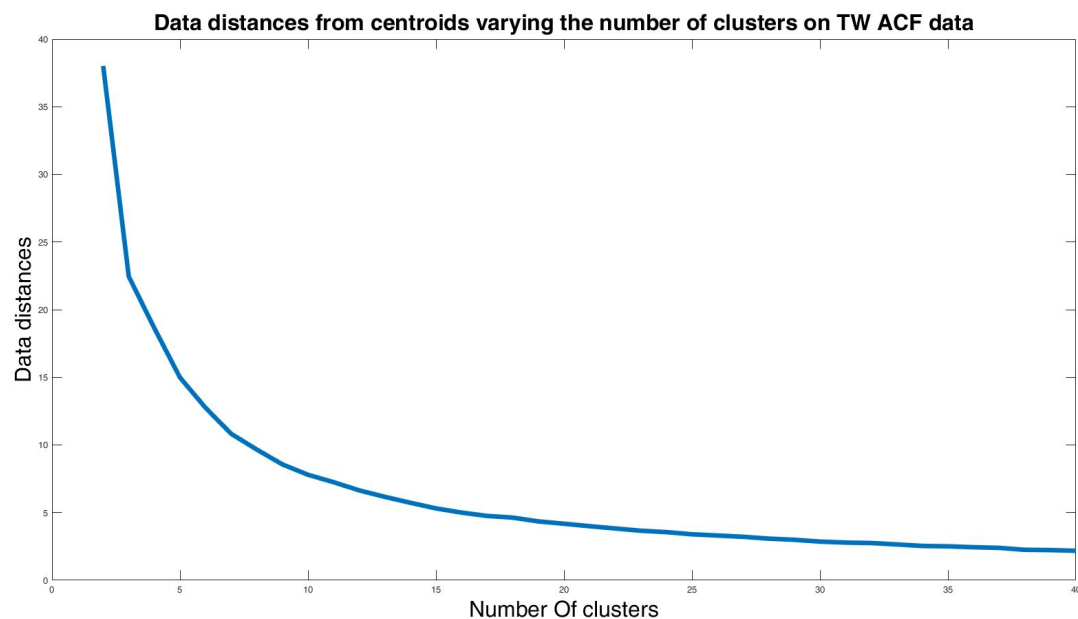


FIGURA 7.12: Distanze dei dati dai rispettivi centroidi al variare del numero di cluster nello spazio TW delle funzioni di autocorrelazione.

Per tutti e tre i grafici si può osservare il "gomito" con un numero di cluster pari a 7. Una volta stabilito è stato possibile raggruppare i dati definitivamente secondo l'algoritmo del K-means con  $K = 7$ .

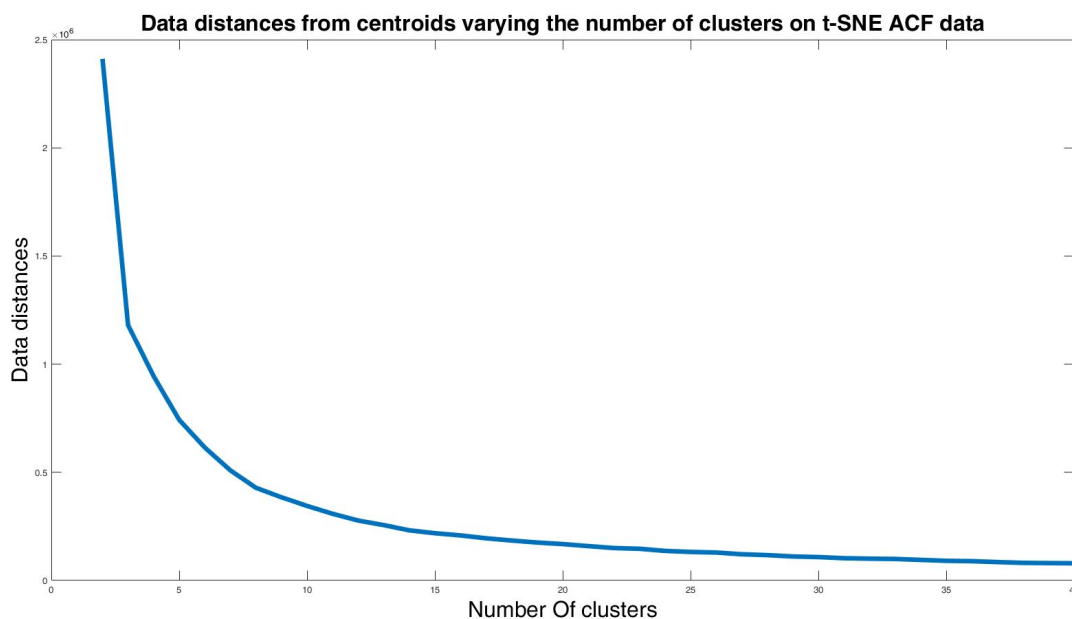


FIGURA 7.13: Distanze dei dati dai rispettivi centroidi al variare del numero di cluster nello spazio t-SNE delle funzioni di autocorrelazione.

In Fig. 7.14, Fig. 7.15 e Fig. 7.16 sono riportati i risultati del clustering per i tre mapping. Le "X" nere rappresentano i centroidi, mentre i diversi colori evidenziano come sono stati raggruppati i dati.

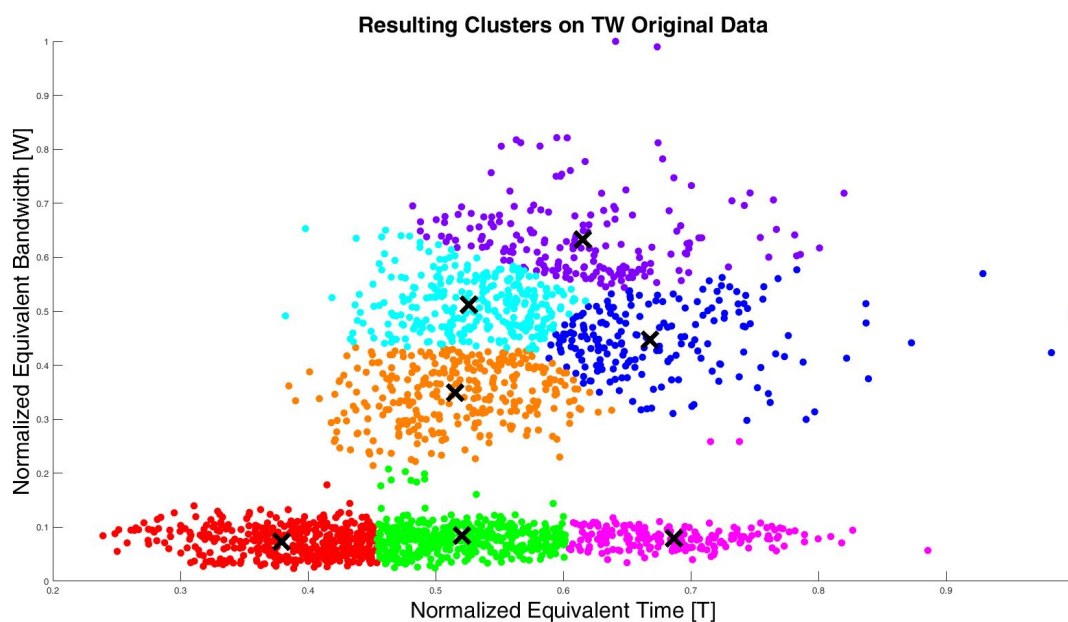


FIGURA 7.14: Risultato del clustering sui dati originali proiettati nello spazio TW.

Poiché tra alcuni cluster non vi è una netta separazione, si può pensare che alcuni di questi potrebbero essere uniti per formare cluster più grandi e ridurre il numero delle possibili classi di difetti presenti. Per fare ciò si sono impiegati gli altri due algoritmi presentati nella Sezione 3.2. Il problema di questi è che devono ricevere due parametri in



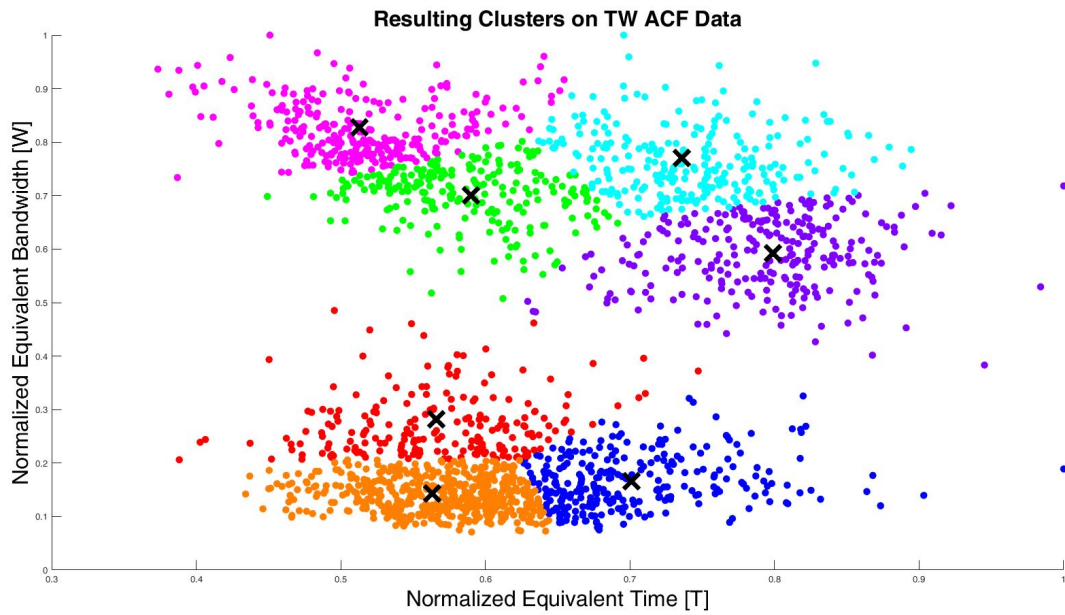


FIGURA 7.15: Risultato del clustering sulle funzioni di autocorrelazione proiettate nello spazio TW.

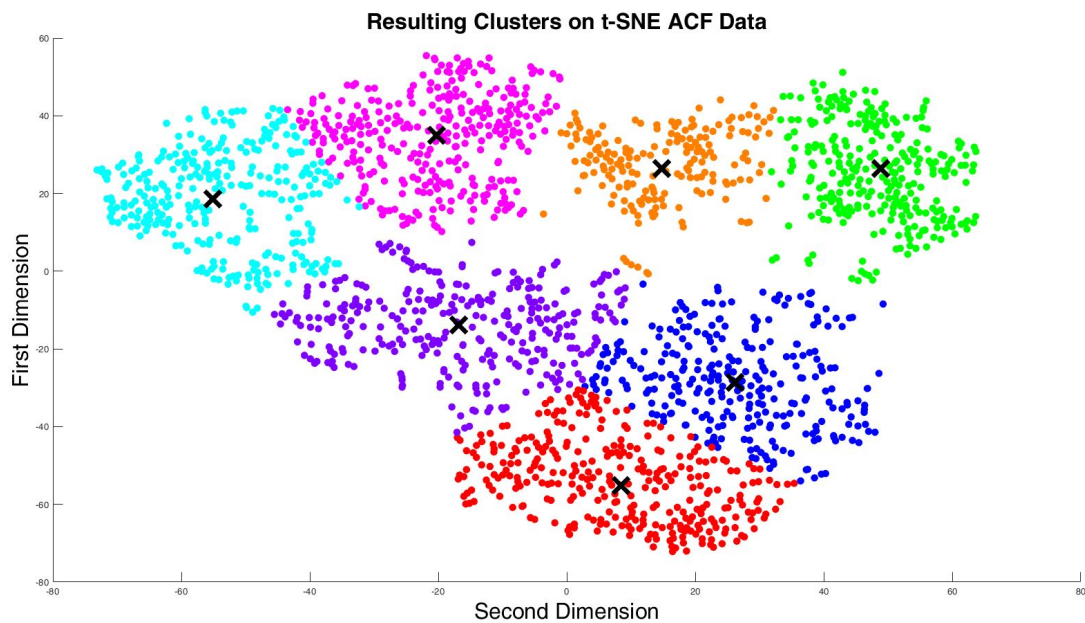


FIGURA 7.16: Risultato del clustering sulle funzioni di autocorrelazione proiettate nello spazio t-SNE con la distanza di Chebychev.

ingresso: il raggio  $\epsilon$  e il numero minimo di punti  $MinPts$ . Siccome sono due parametri liberi, definibili dall'utente, la soluzione potrebbe cambiare molto al variare dei parametri stessi e gli algoritmi potrebbero fornire risultati completamente sbagliati. Si è deciso, dunque, di integrare parte della soluzione del k-means per il calcolo di  $\epsilon$  e di  $MinPts$ .

Per ciascun cluster si definisce  $\epsilon_i$  come in (7.1): ovvero la distanza tra il centroide  $\mathbf{o}_i$  di

quel cluster e il punto, appartenente ad un altro cluster, più vicino al centroide  $\mathbf{o}_i$ .

$$\epsilon_i = \min \text{dist}(\mathbf{x}_j, \mathbf{o}_i), \quad \forall i = 1, \dots, K \text{ e } \forall \mathbf{x}_j \notin \mathbf{C}_i \quad (7.1)$$

dove  $\mathbf{x}_j$  è un punto nello spazio rimappato,  $K$  è il numero di cluster e  $\mathbf{C}_i$  è l'insieme dei punti appartenenti al cluster avente centroide  $\mathbf{o}_i$ .

Una volta stabilito  $\epsilon_i$ , il numero di punti *MinPts*, per ciascun cluster, viene calcolato come in (7.2): ovvero il numero di punti appartenenti ad un cluster aventi distanza  $d < \epsilon_i$  dal proprio centroide  $\mathbf{o}_i$ .

$$\text{MinPts}_i = n^o \text{ punti con } \text{dist}(\mathbf{x}_j, \mathbf{o}_i) < \epsilon_i, \quad \forall i = 1, \dots, K \text{ e } \forall \mathbf{x}_j \in \mathbf{C}_i \quad (7.2)$$

Infine il parametro *MinPts* viene scelto come il più piccolo tra i  $k$  possibili candidati, ed  $\epsilon$  viene scelto corrispondentemente al  $\epsilon_i$  del cluster avente il *MinPts* più piccolo. A questo punto si procede al calcolo dei cluster con DBSCAN e OPTICS.

### 7.3.2 DBSCAN

Una volta che  $\epsilon$  e *MinPts* sono stati inizializzati, il DBSCAN definisce i raggruppamenti. Tutti i punti che non sono stati connessi ai cluster vengono definiti come rumore ed evidenziati con delle "x" nere sui grafici. In Fig. 7.17, Fig. 7.18 e Fig. 7.19 sono riportati i risultati del clustering per i tre mapping.

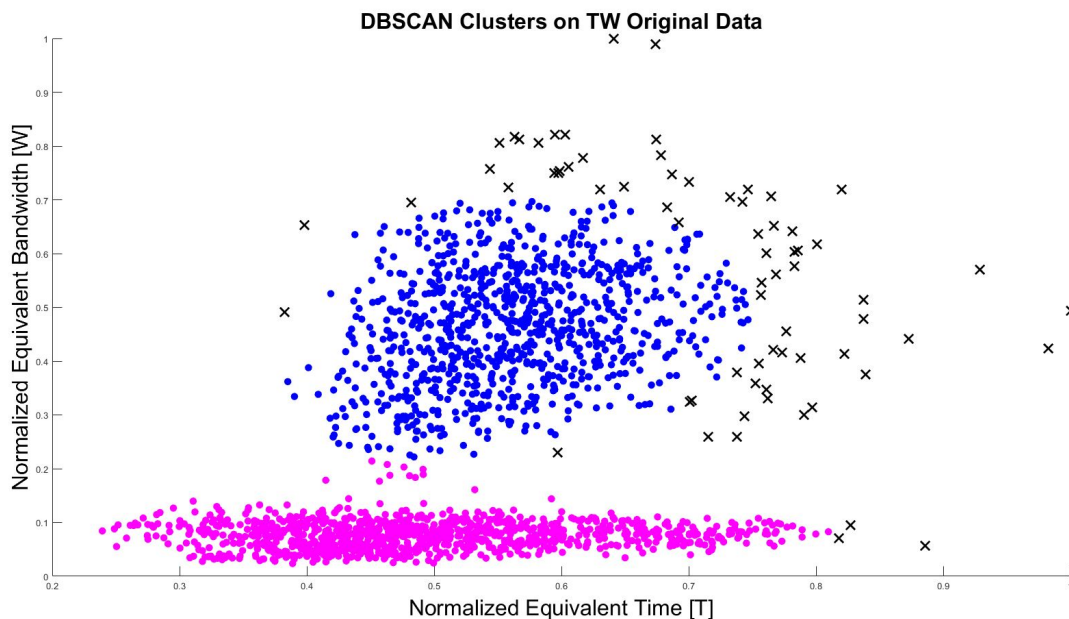


FIGURA 7.17: Risultato del clustering con DBSCAN sui dati originali proiettati nello spazio TW.

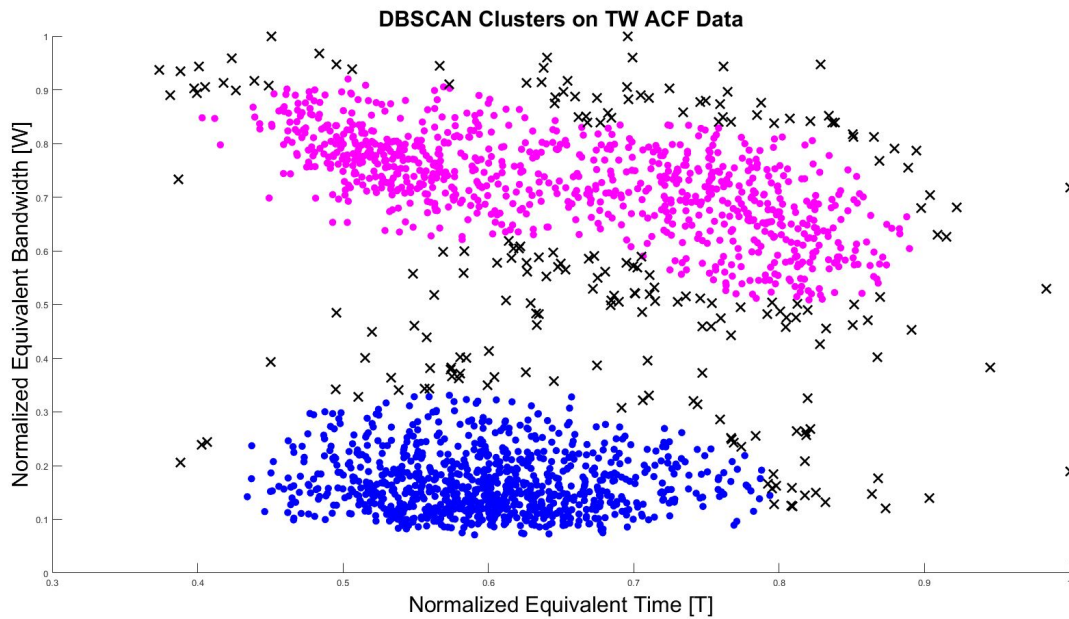


FIGURA 7.18: Risultato del clustering con DBSCAN sulle funzioni di autocorrelazione proiettate nello spazio TW.

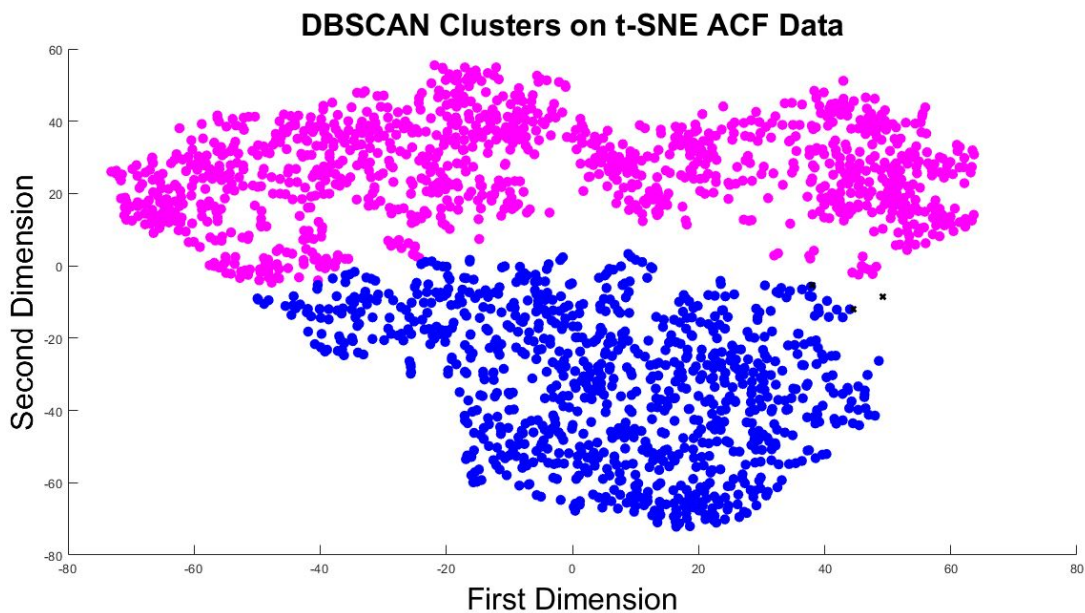


FIGURA 7.19: Risultato del clustering con DBSCAN sulle funzioni di autocorrelazione proiettate nello spazio t-SNE con la distanza di Chebychev.

Si nota subito che per tutti i mapping il DBSCAN raggruppa i dati in 2 cluster che corrispondono ai due tipi di difetti da cui si sono misurati i segnali di scarica durante la fase di acquisizione dei dati. Si osserva che qualche punto viene separato erroneamente, se si confrontano i grafici del clustering con i rispettivi mapping in cui le due sorgenti di scarica sono state rappresentate con i colori blu e rosso (Fig. 7.9, Fig. 7.10 e Fig. 7.6). Inoltre, essendo il mapping del t-SNE più denso anche nei punti periferici, praticamente tutti i dati sono stati assegnati a uno dei due cluster contrariamente ai due mapping

effettuati con il TW. Infatti molti campioni sono stati etichettati come "rumore", poiché non giacciono in un intorno  $\epsilon$  dei punti del cluster.

### 7.3.3 OPTICS

In OPTICS i cluster vengono stabiliti osservando il grafico relativo alle *reachability distances*, in cui i dati vengono ordinati in funzione della loro distanza ai punti vicini. Risulta dunque che i punti appartenenti allo stesso cluster formino delle valli nel grafico. Ciascun cluster è separato da picchi più o meno alti, poiché questi rappresentano punti di transizione tra un cluster e l'altro. In Fig. 7.20, Fig. 7.21 e Fig. 7.22 vengono riportate le *reachability distances* ordinate per i mapping, con *MinPts* fissato secondo i criteri sopra citati.

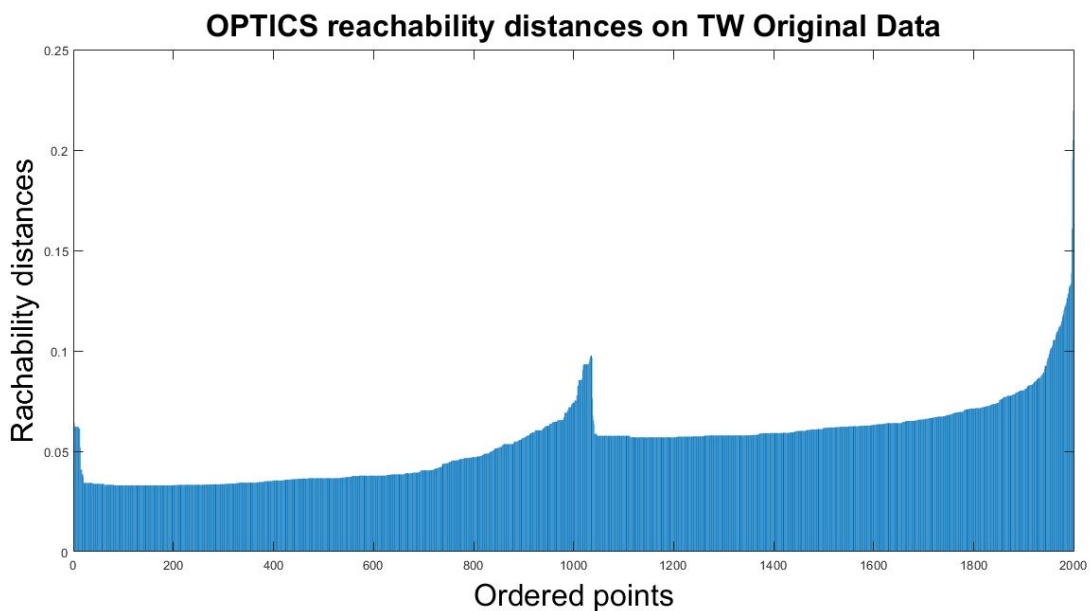


FIGURA 7.20: Reachability distances ordinate dei dati originali proiettati nello spazio TW.

In Fig. 7.20 si possono distinguere chiaramente due avvallamenti. Stabilendo una soglia sull'asse y del grafico è possibile dunque creare i cluster e definire gli eventuali punti che saranno definiti come "rumore". In questo esempio si è deciso di utilizzare  $RD = 0.075$  come valore di soglia, ottenendo il clustering riportato in Fig. 7.23.

Il risultato ottenuto con OPTICS, in questa prova sperimentale, è molto simile a quello ottenuto con il DBSCAN ad eccezione di alcuni punti classificati come rumore.

In Fig. 7.21 ci si accorge che il numero di cluster potrebbe variare in funzione della soglia scelta. I raggruppamenti risultanti, per due scelte effettuate sul valore della soglia, sono riportati in Fig. 7.24.

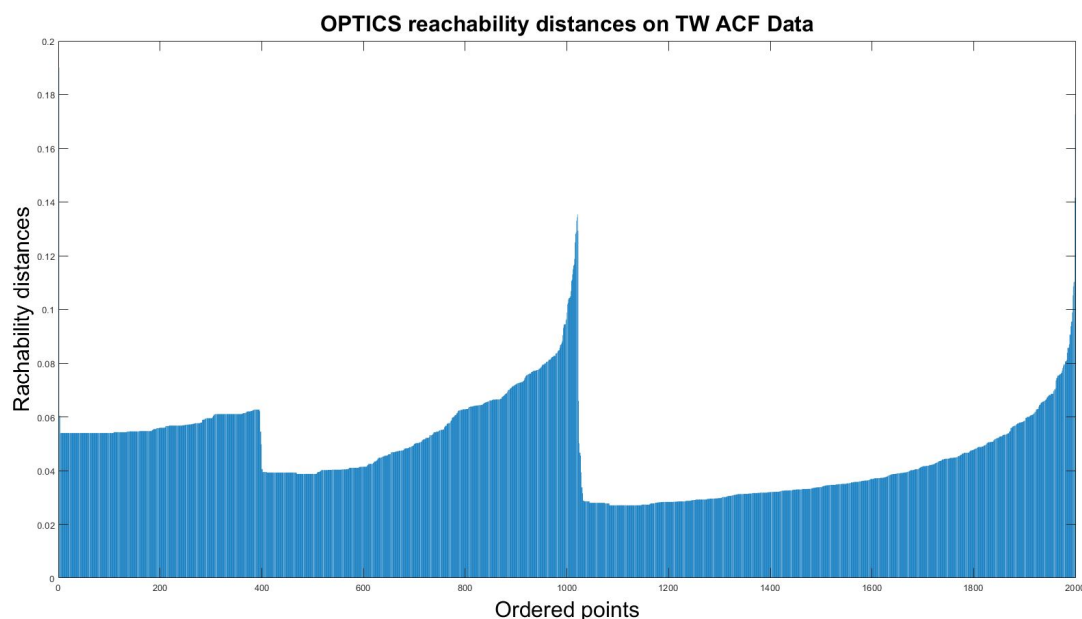


FIGURA 7.21: Reachability distances ordinate delle funzioni di attivazione proiettate nello spazio TW.

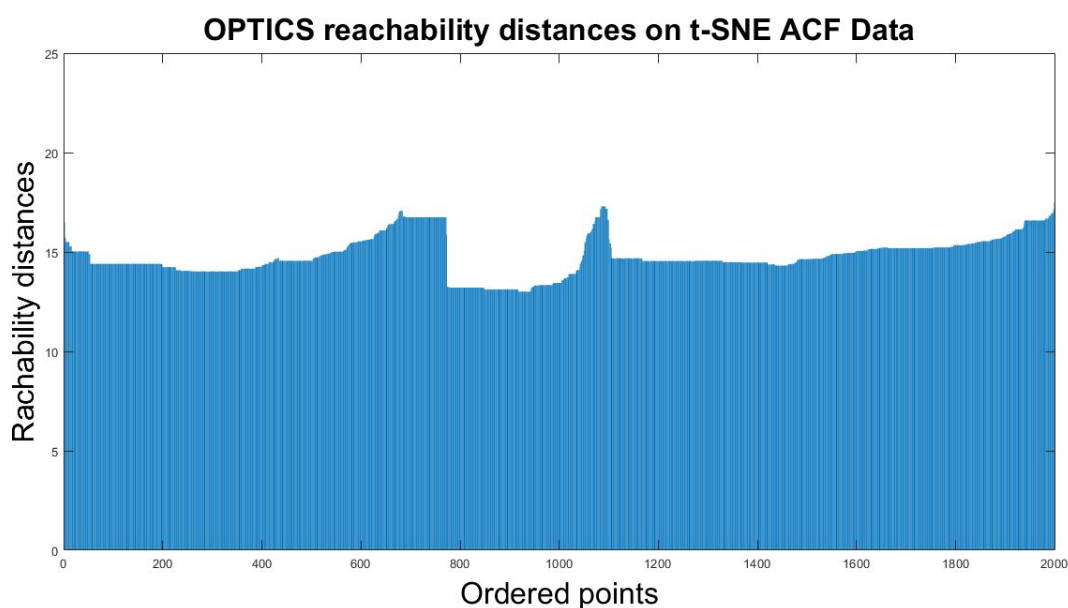


FIGURA 7.22: Reachability distances ordinate delle funzioni di autocorrelazione proiettate nello spazio t-SNE con la distanza di Chebychev.

In Fig. 7.24a, il valore della soglia  $th$  è pari a 0.062. Questa decisione porta alla formazione di 3 cluster con innumerevoli punti classificati come rumore. La seconda opzione ( $th = 0.085$ ), mostra risultati molto simili a quelli riportati dal DBSCAN, con una drastica diminuzione del numero di campioni non associati ad alcun cluster.

Per l'ultimo mapping, osservando la Fig. 7.22, si è deciso di stabilire la soglia a  $th = 16$ . Il risultato del clustering è riportato in Fig. 7.25 ed è molto differente da quello relativo al

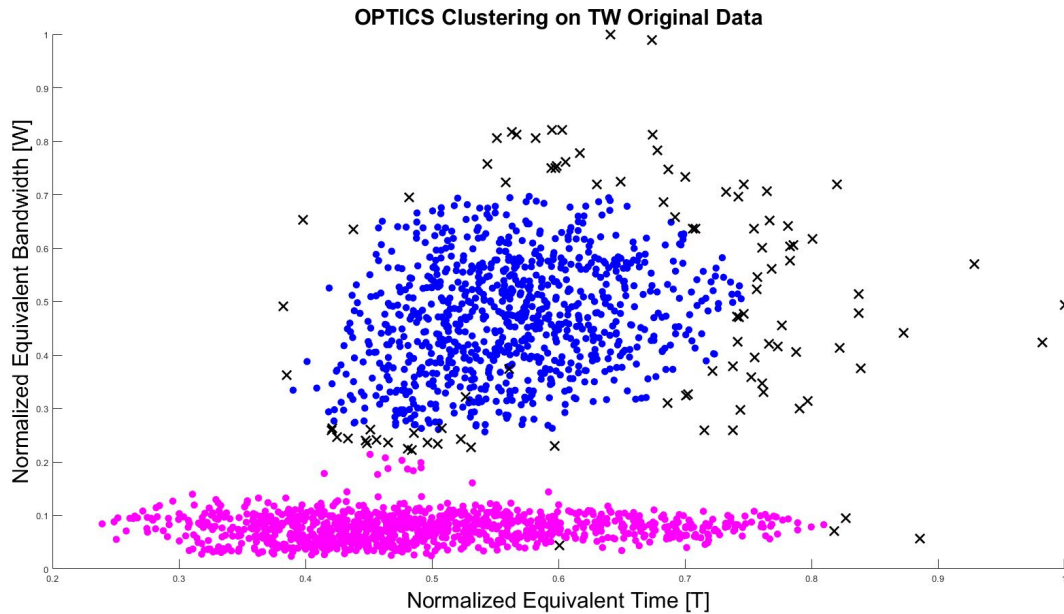


FIGURA 7.23: Risultato del clustering con OPITCS sui dati originali proiettati nello spazio TW, con un valore di soglia pari a 0.075.

DBSCAN (Fig. 7.19). Innanzitutto è cambiato il numero di cluster: da 2 nella soluzione del DBSCAN a 3 in OPTICS. In secondo luogo, nel clustering effettuato con OPTICS vi è un numero elevatissimo di campioni che sono stati classificati come rumore.

## 7.4 Osservazioni

La PCA è stata esclusa a priori per la fase di separazione poiché il nuovo spazio, in un caso, presenta due cluster ciascuno formato da dati di entrambe le sorgenti (Fig. 7.7), nell'altro i campioni formano una nuvola unica in cui sarebbe impossibile applicare un algoritmo di riconoscimento (Fig. 7.8). Nel primo caso, nell'ipotesi in cui non si abbiano informazioni a priori sulla fonte dei segnali acquisiti, si rischierebbe di effettuare un clustering completamente errato. Per cui si rende necessario o avere parziale conoscenza di ciò che si è misurato o utilizzare altre tecniche di estrazione delle features per il problema in questione.

Dalle prove sperimentali effettuate si evince che il t-SNE è molto sensibile alle condizioni iniziali e alla scelta effettuata sulla metrica della distanza. In un caso reale, in cui non si conosce il vero numero di difetti all'interno di un isolamento elettrico, è molto rischioso utilizzare questa tecnica per proiettare i dati. Anche osservando il caso migliore delle prove effettuate sul mapping in un nuovo spazio (Fig. 7.6) con distanza di Chebychev, si nota che alcuni campioni provenienti da una sorgente si trovano nel cluster dell'altra tipologia di difetto (questa informazione non è disponibile in un caso pratico in cui non

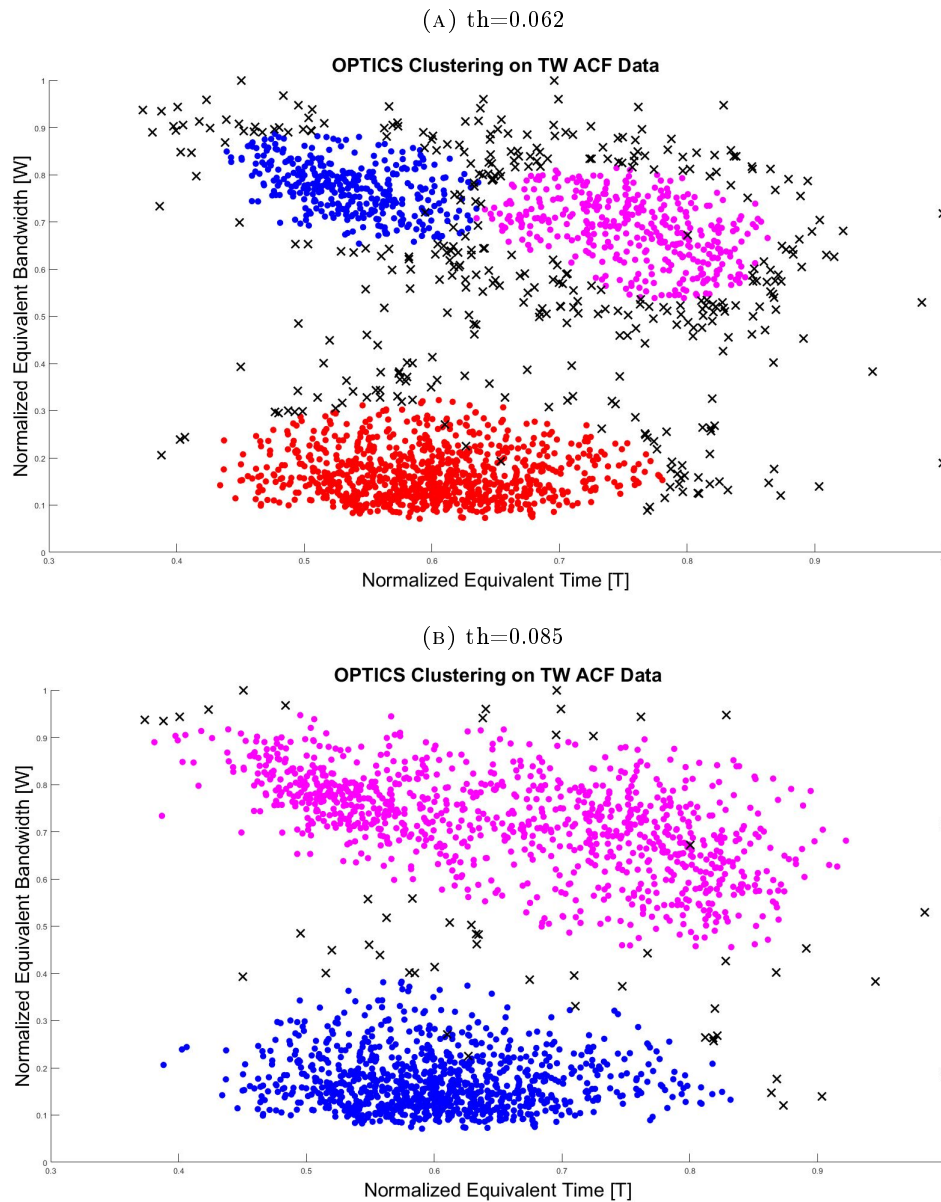


FIGURA 7.24: Risultato del clustering con OPTICS sulle funzioni di autocorrelazione proiettate nello spazio TW, con un valore di soglia pari a: a) 0.062 e b) 0.085.

si conosce a priori il numero di sorgenti di scarica). Oltretutto i risultati riportati in Fig. 7.19 e Fig. 7.25 sono molto differenti tra loro. Anche se il clustering effettuato con il DBSCAN mostra un buon risultato nella separazione delle due sorgenti, si conclude che il t-SNE potrebbe portare a configurazioni degli spazi rimappati (con una scelta sbagliata dei punti iniziali e della metrica per la distanza) per le quali, in fase di clustering, si commetterebbero errori notevoli. Un modo per compensare questi errori è possedere anche una piccola informazione sul dominio in cui si stanno eseguendo i test sperimentali.

La tecnica di estrazione delle features con la mappa TW si è mostrata molto efficace allo scopo di questa tesi. Infatti sia con i dati originali espressi nel dominio tempo sia con le funzioni di autocorrelazione è stato possibile distinguere le due sorgenti di scarica con

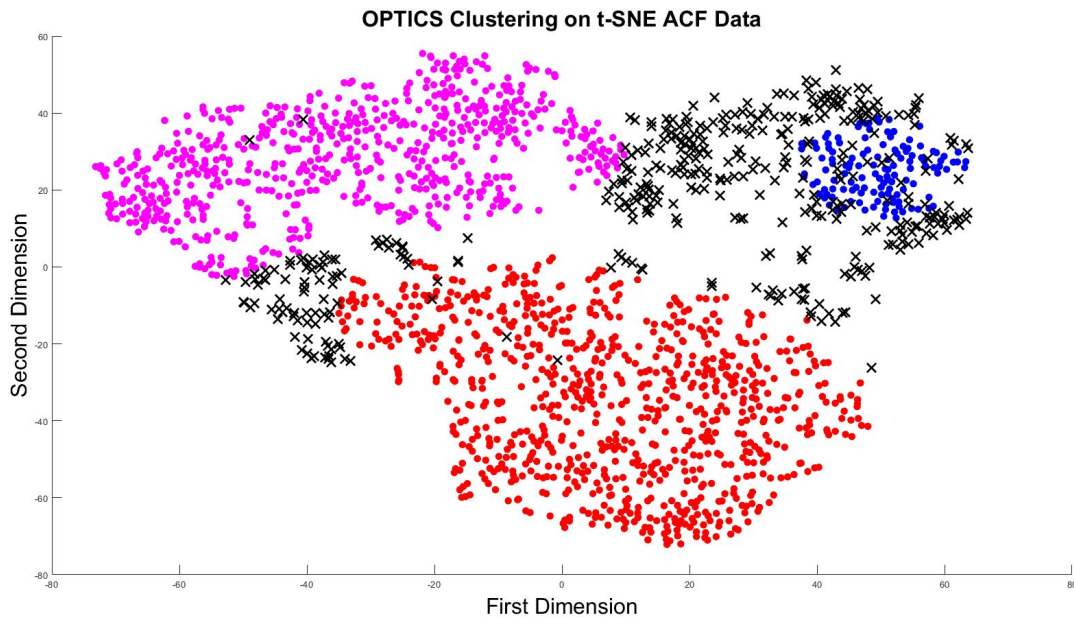


FIGURA 7.25: Risultato del clustering con OPTICS sulle funzioni di autocorrelazione proiettate nello spazio TW, con un valore di soglia pari a 16.

entrambi gli algoritmi di clustering. Si è notato, però, che con OPTICS utilizzato nello spazio TW delle funzioni di autocorrelazione si possono ottenere due soluzioni (Fig. 7.24) al variare del valore della soglia. Osservando la Fig. 7.21 si nota che tra la seconda e terza valle vi è una separazione molto più netta rispetto a quella esistente tra la prima e la seconda. In un caso pratico, in cui non si conosce a priori il numero di difetti, si effettua la divisione con il numero di cluster più elevato. Si procede a costruire il PD pattern relativo alle scariche appartenenti a ciascun cluster per stabilire la natura dei difetti. Se due PD pattern rappresentano lo stesso difetto allora si può rivalutare il valore della soglia in modo tale che i due cluster vengano uniti insieme per formarne uno unico.



## Capitolo 8

# Impianto pilota per misure di scariche parziali su turbine eoliche

I sistemi di misura on-line devono garantire l'integrità dell'apparato sotto osservazione e devono essere il meno invasivi possibile, soprattutto se collocati in ambienti difficili. Per questo motivo si è deciso di utilizzare sensori per la captazione di segnali irradiati a bordo delle navicelle delle turbine eoliche. Data la particolare natura dell'applicazione è stato necessario progettare un sistema di controllo remoto per l'acquisizione e la trasmissione dei segnali [120].

### 8.1 Struttura della turbina eolica Vestas

Il prototipo è stato installato su una turbina eolica prodotta dalla Vestas, una torre da 105 metri di altezza e con una lunghezza delle pale pari a 44 metri, per una potenza di 3MW. Il generatore è collegato direttamente all'albero che trasmette il movimento delle pale attraverso un Gearbox. Il rotore viene alimentato tramite inverter al fine di ottenere la frequenza di rete in uscita dal generatore, senza impiegare variatori di velocità. In questo modo, gestendo solamente una parte della potenza prodotta, è possibile avere la corretta frequenza in uscita dallo statore. La tensione sul rotore è di 1.7kV, mentre quella generata è di 1kV @ 50Hz. Sia rotore che statore sono realizzati con avvolgimenti in piattina. Questi tipi di avvolgimenti vengono impiegati nelle macchine sottoposte a tensioni alternate sinusoidali con valore efficace  $\geq 700V$ ; per questo motivo, essi possono essere soggetti a scariche parziali e pertanto il loro isolamento è impregnato attraverso la tecnica VPI (*Vacuum Pressure Impregnation*) o *ResinRich*, con materiali ad alta rigidità dielettrica quali mica, resine epossidiche e poliesteri. L'isolamento in piattina del rotore e dello statore è dunque sottoposto alle sollecitazioni TEM che ne determinano

l'invecchiamento. Quando l'isolamento viene sottoposto a surriscaldamento per lunghi periodi (ad esempio per riduzione dell'efficienza del sistema di raffreddamento, parziale ostruzione dei canali di raffreddamento, sovraccarichi ecc.) le resine organiche tendono a degradare e gli strati di isolamento si distaccano uno dall'altro (delaminazione). Di conseguenza si generano scariche parziali negli interstizi e nei vuoti e la bobina "allentata" incomincia a "vibrare", con conseguente abrasione meccanica dei nastri micati. Parlando di testate, per contenere le dimensioni, capita che lo spazio tra le spire risulti minimo e abbiano luogo delle scariche parziali, in tali zone, già all'origine. Nell'ottica di effettuare diagnostica per la manutenzione predittiva, sarà necessario analizzare l'andamento dell'attività e monitorare l'evoluzione temporale, per evidenziarne significativi aumenti. Per il collegamento alla rete di distribuzione è utilizzato un trasformatore di tipo cast-resin elevatore, la tensione viene innalzata fino a 24kV. Il trasformatore deve essere esente da scariche, quindi si rende necessario verificare l'effettiva assenza o comunque che l'attività sia inferiore a certi livelli critici, solitamente più piccola di 10pC.

In Fig. 8.1 viene riportato lo schema con la disposizione dei componenti a bordo della navicella.

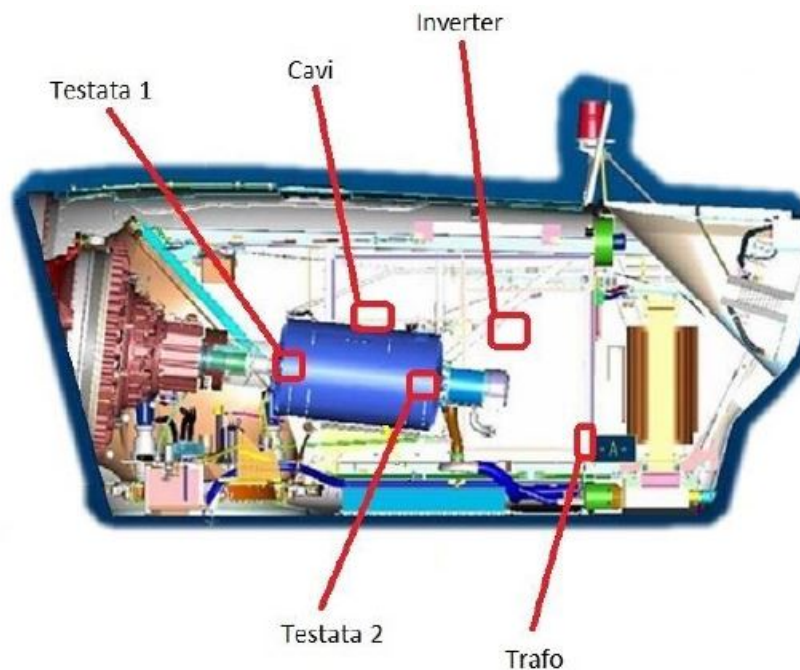


FIGURA 8.1: Schema della navicella della turbina eolica.

Al centro dell'immagine è posizionato il generatore, subito dietro il quadro con l'inverter. Nella parte posteriore (a destra dell'immagine) è collocato il trasformatore, dietro un opportuno pannello. Il vano del trasformatore non è accessibile senza aver sezionato un opportuno pannello.

la tensione di rete e non consente di inserire nulla all'interno. La copertura della navicella è realizzata con vetroresina su un'intelaiatura di metallo, aspetto da tenere in considerazione in fase di misura.

## 8.2 Misure per fini diagnostici

A bordo della turbina eolica è necessario effettuare diagnostica sia sul generatore che sul trasformatore. I due sistemi sono diversi per quanto riguarda il tipo di sistema di isolamento, sia per come è possibile posizionare l'antenna. Per il generatore si possono montare le antenne direttamente sulle due testate, dove non è presente la doppia parete metallica utilizzata per il raffreddamento a liquido e a pochi centimetri dalla carcassa del generatore. Per il trasformatore il discorso è leggermente diverso: non è possibile infatti installare nulla direttamente su di esso per via delle regole sulla sicurezza imposte. L'antenna è stata dunque posizionata a circa 40cm dall'alloggiamento del trasformatore. Essendo la parete separatrice costituita da vetroresina, la distanza non è penalizzante per la misura.

La principale differenza tra il posizionamento delle antenne sul generatore e sul trasformatore riguarda la sensibilità al rumore elettromagnetico ambientale. Le antenne posizionate sul generatore saranno sostanzialmente schermate dalla carcassa, mentre quelle posizionate sul trasformatore, a causa della distanza da esso saranno più soggette ai disturbi elettromagnetici.

I due componenti saranno trattati in modo differente: sul generatore si avrà bisogno monitorare l'andamento temporale delle scariche che avvengono, mentre per il trasformatore sarà sufficiente determinarne la presenza o meno.

## 8.3 Sistema remoto per monitoraggio a lungo termine

Il prototipo per il monitoraggio sul lungo periodo è pensato, non solo per la misura delle scariche irradiate, ma anche per quella tradizionale. Per la prima versione, dunque, è stato concesso di montare anche un sistema per la misura condotta, composto da due ferriti. In questo modo sarà possibile confrontare le misure effettuate con i due metodi, ed avere un riferimento.

In Fig. 8.2 viene mostrato lo schema a blocchi del sistema di monitoraggio remoto.

Il sistema è dotato di una connessione internet, con la quale sarà possibile effettuare misure da remoto e controllare lo stato dei vari componenti. Il sistema si basa su un

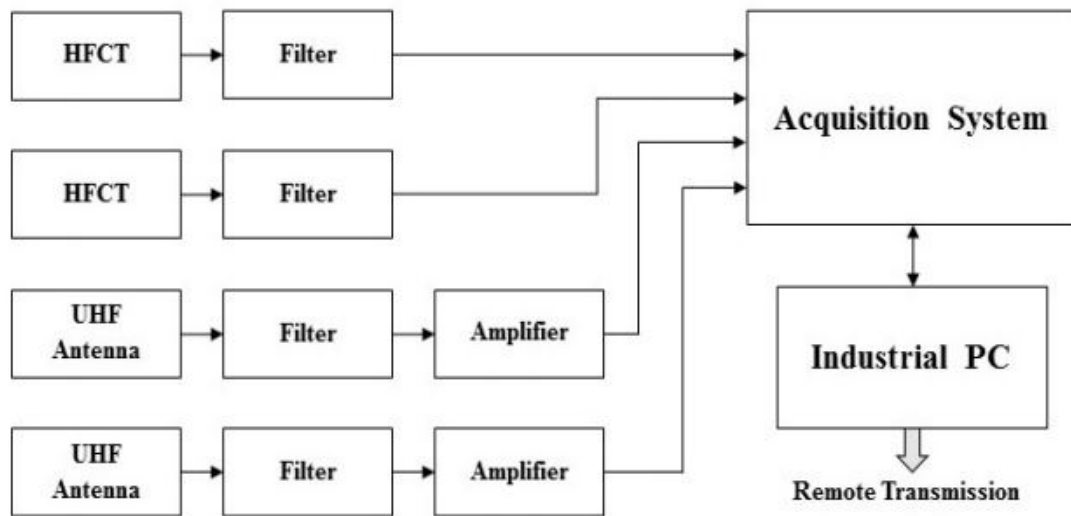


FIGURA 8.2: Schema a blocchi del sistema di monitoraggio.

computer industriale (DA-1000) prodotto dalla Cincoze, in grado di supportare le temperature presenti a bordo della navicella, che in estate possono anche superare i  $50^{\circ}\text{C}$ . Al computer è collegata una scheda di acquisizione per la digitalizzazione dei segnali. Con una banda di  $200\text{MHz}$  e un campionamento massimo di  $1\text{GSample/s}$  è possibile gestire sia i segnali del sistema per la misura condotta, sia per quelli irradiati. Il computer, mediante un software realizzato ad hoc in LabVIEW, è in grado di interagire con la scheda di acquisizione, impostando i canali attivi, la frequenza di campionamento, il numero di campioni acquisiti per canale e il salvataggio dei dati. Inoltre il software è in grado di essere configurato per acquisire e salvare autonomamente i segnali ogni tot di tempo. Tramite la connessione internet è possibile variare i parametri che regolano l'acquisizione e trasmettere i dati salvati su server.

I sensori utilizzati per la captazione dei segnali sono due antenne Solbox<sup>®</sup> (Sezione 4.2.1) per quelli irradiati e due ferriti per i condotti (Sezione 4.2.3). Un'antenna è stata montata sulla testata 2 (sito T2) del generatore, l'altra sul lato superiore del trasformatore (sito K), al di fuori del pannello di separazione.

In Fig. 8.3 vengono riportate le immagini relative al posizionamento delle antenne.

Le due ferriti sono posizionate intorno ai cavi delle fasi in uscita dallo statore (sito C) che portano corrente al trasformatore (Fig. 8.4). La ferrite si comporta come un TA, ma permette di trasdurre segnali ad alte frequenze rispetto alla sola frequenza industriale, per cui è possibile captare gli impulsi dovuti alle scariche parziali. Successivamente vengono convertiti in tensione tramite una resistenza, per permetterne il campionamento con la scheda di acquisizione. Le ferriti, teoricamente, dovrebbero essere collocate nella connessione di terra del centro stella. Data la natura della turbina eolica è impossibile

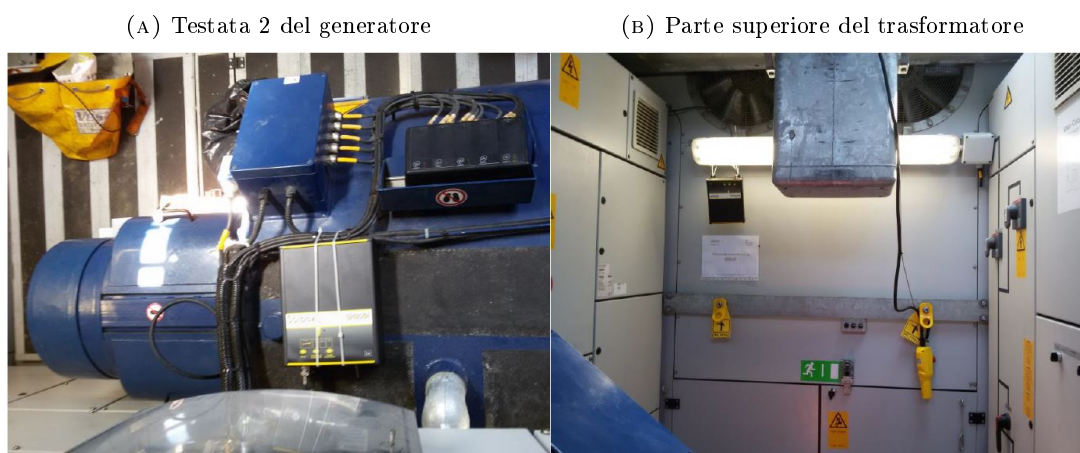


FIGURA 8.3: Posizionamento delle antenne all'interno della navicella: a) testata 2 del generatore, b) parte superiore del trasformatore.

effettuare questa misura, poiché il centro stella non è accessibile. Posizionando i sensori induttivi attorno ai cavi è comunque possibile rilevare i segnali dovuti alle scariche parziali, anche con sensibilità minore.

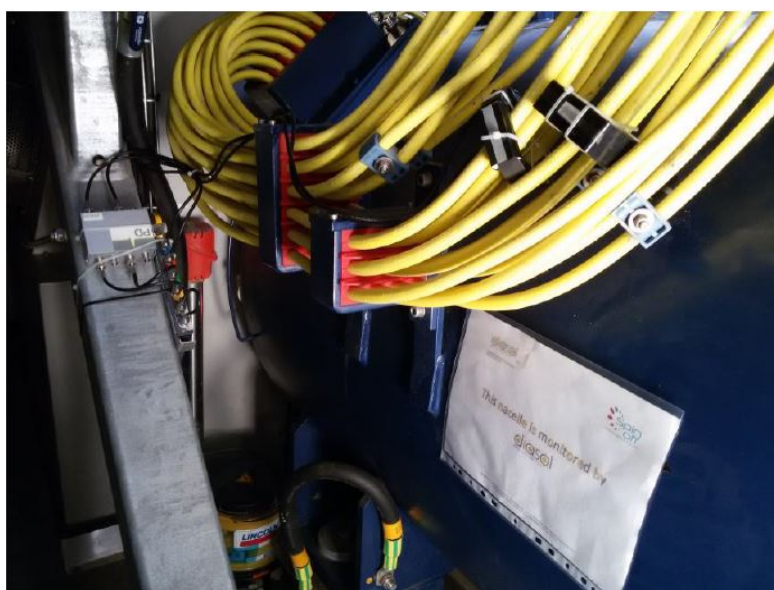


FIGURA 8.4: Posizionamento delle due ferriti attorno ai cavi in uscita dallo statore.

Tutti i segnali captati con i sensori vengono opportunamente filtrati in modo tale da rendere il rapporto segnale-rumore il migliore possibile. I filtri sono stati realizzati dopo una vasta campagna di misure preliminari atte alla caratterizzazione delle frequenze dei disturbi sia in condizioni di riposo della turbina sia durante il normale funzionamento [57].

### 8.3.1 Monitoraggio Temporale

Il monitoraggio è stato effettuato autonomamente per sei mesi. Le misure venivano realizzate ogni ora grazie al software progettato in LabVIEW. In questo modo è stato possibile collezionare dati per un lungo periodo, anche nelle ore notturne, senza l'intervento di un operatore. I dati acquisiti sono stati poi elaborati valutando il palinsesto del funzionamento attivo della turbina. In Fig. 8.5, Fig. 8.6 e Fig. 8.7 sono riportate tre acquisizioni effettuate rispettivamente dopo 2, 4 e 6 mesi. Tutte le figure riportano una scala dei tempi ridotta a  $0.4\mu\text{s}$  in quanto, essendo l'alimentazione del generatore ad inverter, ci si è soffermati negli istanti in cui è possibile avere un *overshoot* della tensione di alimentazione e quindi un fenomeno di scarica più intenso.

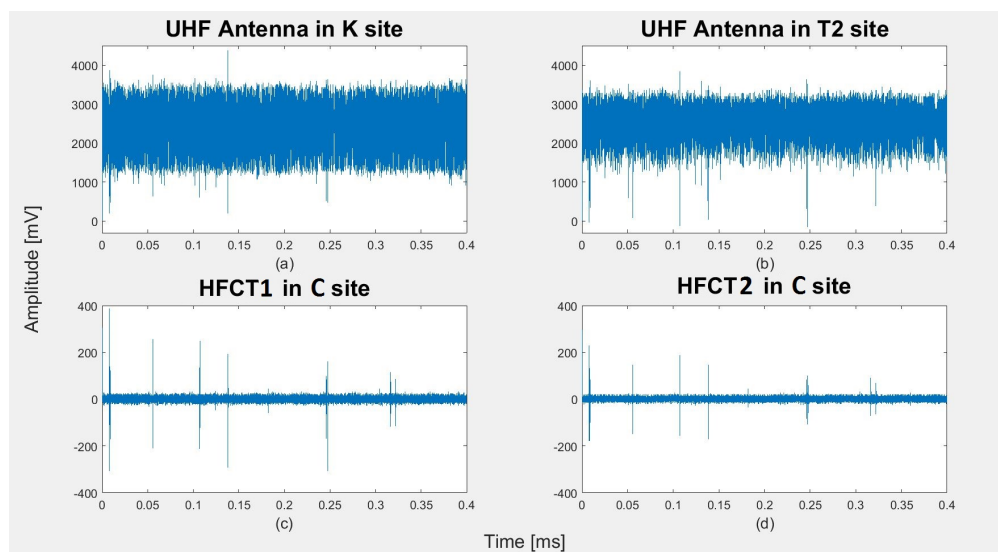


FIGURA 8.5: Acquisizione effettuata dopo 2 mesi.

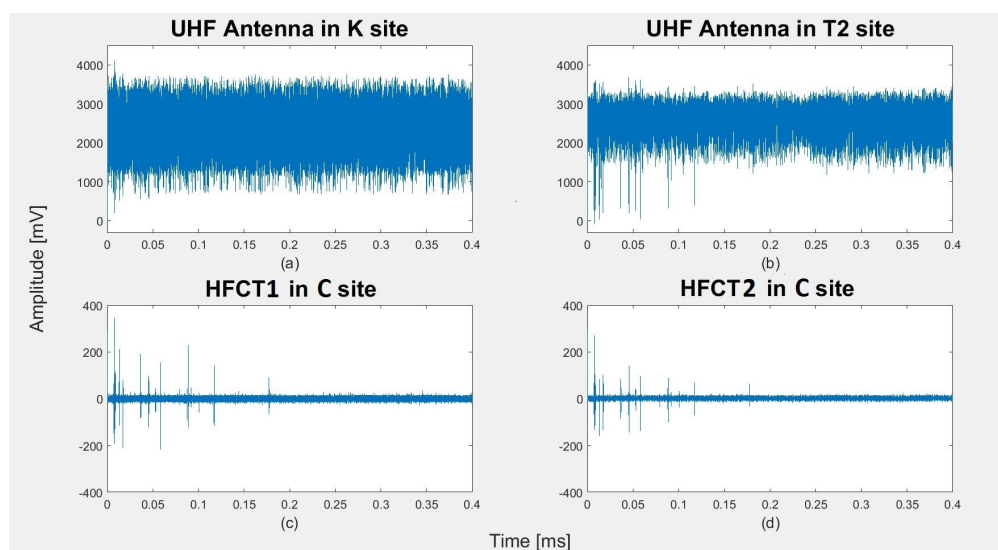


FIGURA 8.6: Acquisizione effettuata dopo 4 mesi.

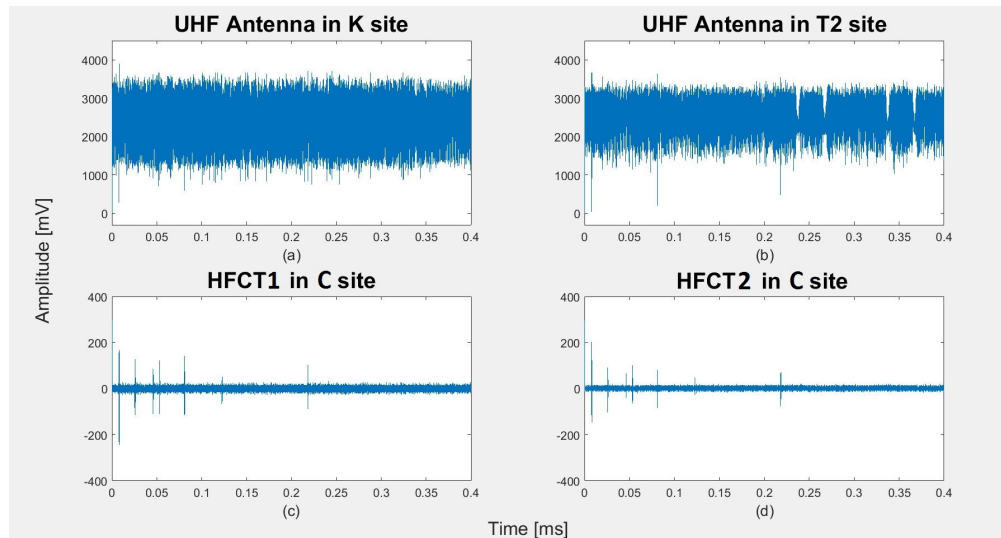


FIGURA 8.7: Acquisizione effettuata dopo 6 mesi.

Per ciascuna figura si osserva che i segnali captati dalle ferriti combaciano per entrambe. Inoltre si nota che l'antenna montata nel sito T2 riesce a misurare alcuni dei segnali rilevati con le ferriti. L'antenna posizionata nel sito K risulta molto più rumorosa rispetto all'altra poiché non è schermata da nessuna struttura metallica ed è collocata più lontano dalla possibile sorgente di scariche rispetto a quella posizionata sul generatore. Per questi motivi non si ha la certezza della provenienza dei segnali, ma si ipotizza che provengano dal generatore, data la corrispondenza delle ferriti con l'antenna situata in T2.

Infine gli impulsi presenti nei grafici potrebbero essere causati o da una sorgente di scarica o dall'inverter. Ciò che comunque si osserva è che, nel corso del tempo, l'ampiezza dei segnali non aumenta e anche la numerosità rimane pressoché costante. Quindi, se i segnali misurati fossero relativi alle scariche parziali, si affermerebbe che il fenomeno non è peggiorato nel corso del tempo.

Per una fase di monitoraggio definitiva deve essere condotto uno studio per l'eliminazione dei segnali non dovuti alle scariche. Le soluzioni potrebbero riguardare sia un miglioramento hardware nella progettazione di nuovi filtri sia uno sviluppo di tecniche software che permettano la distinzione tra scariche e rumore dovuto all'inverter.

## Capitolo 9

# Conclusioni e sviluppi futuri

Questa tesi ha avuto come obiettivo il riconoscimento automatico di difetti, dovuti al fenomeno delle scariche parziali, nell'isolamento di una macchina elettrica. Sono stati utilizzati, sostanzialmente, due approcci: metodi supervisionati e non.

Per quanto riguarda i primi è stato condotto uno studio approfondito sulle reti neurali a singolo strato. Questo ha permesso di reinterpretare lo strato di mapping del modello della ELM introducendo il concetto di funzione di similarità. Così sono stati proposti due algoritmi per la scelta dei parametri liberi delle funzioni di attivazione/similarità, sfruttando la randomicità come principio cardine. Le prove sperimentali hanno mostrato che, in molti casi, la capacità di apprendimento della ELM che impiega gli algoritmi per la selezione dei parametri è migliore rispetto a quella della ELM standard.

In questo quadro teorico sono state proposte diverse implementazioni della ELM su sistemi embedded con risorse limitate. La ricerca si è focalizzata su predittori che sfruttano la hard-limiter come funzione di attivazione non lineare. La tesi ha fornito due architetture per la realizzazione ad-hoc su dispositivi economici e con risorse limitate. Una di queste è stata impiegata come classificatore per il problema supervisionato delle scariche parziali, ottenendo risultati paragonabili o, addirittura, migliori rispetto a SVM.

Per quanto riguarda il riconoscimento non supervisionato sono stati confrontati tre algoritmi di clustering che raggruppano i dati precedentemente proiettati in uno spazio 2D con diverse tecniche di estrazione delle features. I risultati hanno mostrato che la scelta migliore è impiegare la mappa TW come metodo di mapping calcolata sia sui dati campionati come serie temporali, sia sulle funzioni di autocorrelazione. Siccome i pattern nel nuovo spazio non sono rappresentati in nuvole omogenee è stato necessario utilizzare algoritmi di clustering che riuscissero a creare raggruppamenti con queste morfologie (DBSCAN e OPTICS). Il K-means è stato sfruttato per stabilire i parametri di ingresso



degli altri due algoritmi. Sia con il DBSCAN sia con OPTICS si sono ottenuti risultati molto soddisfacenti nella separazione non supervisionata di sorgenti di scariche parziali.

Infine è stato presentato un prototipo di sistema di monitoraggio online in una turbina eolica. Dopo una campagna di misure si è deciso di adottare due sensori induttivi e due elettromagnetici per la captazione dei segnali. Il sistema è in grado di campionare quattro sorgenti di segnali e di trasmettere automaticamente i dati ad un server. Il prototipo è stato impiegato continuativamente per sei mesi per valutare l'intensificazione del fenomeno delle scariche sul generatore e per verificarne la presenza sul trasformatore. I risultati evidenziano che devono essere svolte azioni correttive per l'eliminazione del rumore eccessivo captato dalle antenne e che è necessario distinguere segnali di scarica da quelli generati dall'inverter.

## 9.1 Sviluppi futuri

L'implementazione hardware della ELM richiede dati etichettati per la fase di training. Dunque, per l'utilizzo del predittore, è necessario conoscere esattamente le sorgenti di scarica e fornire, per ogni difetto esistente, dati all'algoritmo in modo tale da creare il classificatore utilizzabile online. Per il riconoscimento di difetti multipli si deve estendere l'architettura alla risoluzione di problemi multi-classe. Per fare ciò è necessario creare, per ogni possibile problema bi-classe, una memoria dei pesi  $\beta$  e realizzare un sistema ad arbitro che prenda una decisione in base alla maggioranza dei voti.

Per quanto riguarda il sistema di monitoraggio online in ambienti ostili è necessario filtrare efficacemente il rumore captato dalle antenne, oltre a distinguere i segnali prodotti dalla commutazione dell'inverter piuttosto che dalle sorgenti di scarica. Inoltre, attualmente, non si dispone ancora di un classificatore di difetti associato al sistema di acquisizione automatico. Per i segnali provenienti dal generatore potrebbe essere realizzato un separatore non supervisionato, non conoscendo a priori quali difetti esistano all'interno dell'isolamento. Per il trasformatore si dovrebbe realizzare un sistema di allarme a soglia per avvisare gli operatori di una possibile criticità al suo interno. Risulta essenziale, soprattutto nel secondo caso, distinguere le sorgenti di rumore dai segnali di scarica in quanto potrebbero generarsi falsi allarmi. Nella fase di classificazione non supervisionata il rumore potrebbe coprire eventuali segnali di scarica, non rendendoli riconoscibili.

# Bibliografia

- [1] High-voltage test techniques - partial discharge measurements. International Standard IEC 60270:2000, International Electrotechnical Commission, Geneva, CH, 2000. URL <https://webstore.iec.ch/publication/1247>.
- [2] AVO International. *Ultrasonic Corona and Leak Detector Instructional Manual*. Instruction Manual AVTM56-9J Rev. A.
- [3] Y Tian, PL Lewin, D Pommerenke, JS Wilkinson, and SJ Sutton. Partial discharge on-line monitoring for hv cable systems using electro-optic modulators. *IEEE Transactions on Dielectrics and Electrical Insulation*, 11(5):861–869, 2004.
- [4] Michel Duval. A review of faults detectable by gas-in-oil analysis in transformers. *IEEE electrical Insulation magazine*, 18(3):8–17, 2002.
- [5] Ljubomir A Kojovic and Martin T Bishop. Comparative characteristics of iron-core current transformers and rogowski coils for applications for protective relaying purposes. In *Protective Relay Engineers, 2009 62nd Annual Conference for*, pages 527–535. IEEE, 2009.
- [6] F Guastavino, E Torello, S Squarcia, A Dardano, M Secci, F Ferraro, and D Pistone. Morphologic analysis and diagnosis of defects inside cast resin medium voltage current transformers insulation by digital partial discharges acquisitions. In *Electrical Insulation and Dielectric Phenomena (CEIDP), 2011 Annual Report Conference on*, pages 493–496. IEEE, 2011.
- [7] Krystian Leonard Chrzan and Federico Moro. Concentrated discharges and dry bands on polluted outdoor insulators. *IEEE Transactions on Power Delivery*, 22(1):466–471, 2007.
- [8] P Mraz, P Treyer, and U Hammer. Evaluation and limitations of corona discharge measurements-an application point of view. In *Condition Monitoring and Diagnosis (CMD), 2016 International Conference on*, pages 278–281. IEEE, 2016.
- [9] GC Stone. Partial discharge. vii. practical techniques for measuring pd in operating equipment. *IEEE Electrical Insulation Magazine*, 7(4):9–19, 1991.

- 
- [10] NC Sahoo, MMA Salama, and R Bartnikas. Trends in partial discharge pattern classification: a survey. *IEEE Transactions on Dielectrics and Electrical Insulation*, 12(2):248–264, 2005.
- [11] R Candela, G Mirelli, and R Schifani. Pd recognition by means of statistical and fractal parameters and a neural network. *IEEE Transactions on Dielectrics and electrical Insulation*, 7(1):87–94, 2000.
- [12] Edward Gulski and A Krivda. Neural networks as a tool for recognition of partial discharges. *IEEE transactions on electrical insulation*, 28(6):984–1001, 1993.
- [13] A Contin and G Rabach. Pd analysis of rotating ac machines. *IEEE transactions on electrical insulation*, 28(6):1033–1042, 1993.
- [14] A Contin, M Cacciari, GC Montanari, and G Rabach. Stochastic approach to the analysis of partial discharge measurement performed in hv generator bars. *Materials Engineering*, 6:345–364, 1996.
- [15] M Cacciari, A Contin, G Rabach, and GC Montanari. Diagnosis of practical insulation systems by partial discharge measurements in the presence of multi-discharge phenomena. In *Electrical Insulation and Dielectric Phenomena, 1993. Annual Report., Conference on*, pages 414–419. IEEE, 1993.
- [16] M Cacciari, A Contin, G Rabach, and GC Montanari. An approach to partial-discharge investigation by height-distribution analysis. *IEE Proceedings-Science, Measurement and Technology*, 142(1):102–108, 1995.
- [17] KX Lai, BT Phung, and TR Blackburn. Application of data mining on partial discharge part i: Predictive modelling classification. *IEEE Transactions on Dielectrics and Electrical Insulation*, 17(3), 2010.
- [18] Hui Ma, Jeffery C Chan, Tapan K Saha, and Chandima Ekanayake. Pattern recognition techniques and their applications for automatic classification of artificial partial discharge sources. *IEEE Transactions on Dielectrics and Electrical Insulation*, 20(2):468–478, 2013.
- [19] L Hao, PL Lewin, and SJ Dodd. Comparison of support vector machine based partial discharge identification parameters. In *Electrical Insulation, 2006. Conference Record of the 2006 IEEE International Symposium on*, pages 110–113. IEEE, 2006.
- [20] L Hao and PL Lewin. Partial discharge source discrimination using a support vector machine. *IEEE Transactions on Dielectrics and electrical Insulation*, 17(1), 2010.

- [21] V Chatpattananan, N Pattanadech, and P Yutthagowith. Partial discharge classification on high voltage equipment with k-means. In *Properties and applications of Dielectric Materials, 2006. 8th International Conference on*, pages 191–194. IEEE, 2006.
- [22] Rui-jin Liao, Li-jun Yang, Jian Li, and Stanislaw Grzybowski. Aging condition assessment of transformer oil-paper insulation model based on partial discharge analysis. *IEEE Transactions on Dielectrics and Electrical Insulation*, 18(1), 2011.
- [23] Sanna Poyhonen, Marco Conti, Andrea Cavallini, Gian C Montanari, and Fiorenzo Filippetti. Insulation defect localization through partial discharge measurements and numerical classification. In *Industrial Electronics, 2004 IEEE International Symposium on*, volume 1, pages 417–422. IEEE, 2004.
- [24] Jian Li, Ruijin Liao, S Grzybowski, and Lijun Yang. Oil-paper aging evaluation by fuzzy clustering and factor analysis to statistical parameters of partial discharges. *IEEE Transactions on Dielectrics and Electrical Insulation*, 17(3), 2010.
- [25] A Contin, A Cavallini, GC Montanari, G Pasini, and F Puletti. Digital detection and fuzzy classification of partial discharge signals. *IEEE Transactions on Dielectrics and Electrical Insulation*, 9(3):335–348, 2002.
- [26] Tadeja Babnik, Raj K Aggarwal, and Philip J Moore. Principal component and hierarchical cluster analyses as applied to transformer partial discharge data with particular reference to transformer condition monitoring. *IEEE transactions on power delivery*, 23(4), 2008.
- [27] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.
- [28] R Liao, Y Fernandess, K Tavernier, GA Taylor, and MR Irving. Recognition of partial discharge patterns. In *Power and Energy Society General Meeting, 2012 IEEE*, pages 1–8. IEEE, 2012.
- [29] L Hao, PL Lewin, JA Hunter, DJ Swaffield, A Contin, C Walton, and M Michel. Discrimination of multiple pd sources using wavelet decomposition and principal component analysis. *IEEE Transactions on Dielectrics and Electrical Insulation*, 18(5), 2011.
- [30] KX Lai, BT Phung, TR Blackburn, and NA Muhamad. Classification of partial discharge using pca and som. In *Power Engineering Conference, 2007. IPEC 2007. International*, pages 1311–1316. IEEE, 2007.

- 
- [31] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.
- [32] PL Lewin, LA Petrov, and L Hao. A feature based method for partial discharge source classification. In *Electrical Insulation (ISEI), Conference Record of the 2012 IEEE International Symposium on*, pages 443–448. IEEE, 2012.
- [33] JA Hunter, L Hao, PL Lewin, C Walton, and M Michel. Partial discharge diagnostics of defective medium voltage three-phase pvc cables. In *Electrical Insulation (ISEI), Conference Record of the 2012 IEEE International Symposium on*, pages 371–375. IEEE, 2012.
- [34] RD Nimmo, G Callender, and PL Lewin. Methods for wavelet-based autonomous discrimination of multiple partial discharge sources. *IEEE Transactions on Dielectrics and Electrical Insulation*, 24(2):1131–1140, 2017.
- [35] Wensheng Gao, Dengwei Ding, and Weidong Liu. Research on the typical partial discharge using the uhf detection method for gis. *IEEE transactions on power delivery*, 26(4):2621–2629, 2011.
- [36] Yu-Hsun Lin. Using k-means clustering and parameter weighting for partial-discharge noise suppression. *IEEE transactions on power delivery*, 26(4):2380–2390, 2011.
- [37] Alfredo Contin and Stefano Pastore. Classification and separation of partial discharge signals by means of their auto-correlation function evaluation. *IEEE Transactions on Dielectrics and Electrical Insulation*, 16(6), 2009.
- [38] Xiaosheng Peng, Chengke Zhou, Donald M Hepburn, Martin D Judd, and Wah Hoon Siew. Application of k-means method to pattern recognition in on-line cable partial discharge monitoring. *IEEE Transactions on Dielectrics and Electrical Insulation*, 20(3):754–761, 2013.
- [39] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.
- [40] Alfredo Contin and Stefano Pastore. Automatic separation of multiple pd sources using an amplitude-autocorrelation relation diagram. In *2012 IEEE International Symposium on Electrical Insulation*, pages 434–438. IEEE, 2012.

- [41] Ming-Xiao Zhu, Jia-Ning Zhang, Yuan Li, Yan-Hui Wei, Jian-Yi Xue, Jun-Bo Deng, Hai-Bao Mu, Guan-Jun Zhang, and Xian-Jun Shao. Partial discharge signals separation using cumulative energy function and mathematical morphology gradient. *IEEE Transactions on Dielectrics and Electrical Insulation*, 23(1):482–493, 2016.
- [42] Ming-Xiao Zhu, Qing Liu, Jian-Yi Xue, Jun-Bo Deng, Guan-Jun Zhang, Xian-Jun Shao, Wen-Lin He, An-Xiang Guo, and Xiao-Wei Liu. Self-adaptive separation of multiple partial discharge sources based on optimized feature extraction of cumulative energy function. *IEEE Transactions on Dielectrics and Electrical Insulation*, 24(1):246–258, 2017.
- [43] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: ordering points to identify the clustering structure. In *ACM Sigmod record*, volume 28, pages 49–60. ACM, 1999.
- [44] Alfredo Contin, Stefano Pastore, and Raffaele Paganin. Evaluation of spaces for the separation of signals due to multiple pd sources. In *Electrical Insulation Conference (EIC), 2015 IEEE*, pages 209–213. IEEE, 2015.
- [45] Nello Cristianini, John Shawe-Taylor, et al. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.
- [46] RM Sharkawy, RS Mangoubi, TK Abdel-Galil, MMA Salama, and R Bartnikas. Svm classification of contaminating particles in liquid dielectrics using higher order statistics of electrical and acoustic pd measurements. *IEEE Transactions on Dielectrics and Electrical Insulation*, 14(3), 2007.
- [47] Guillermo Robles, Emilio Parrado-Hernández, Jorge Ardila-Rey, and Juan Manuel Martínez-Tarifa. Multiple partial discharge source discrimination with multiclass support vector machines. *Expert Systems with Applications*, 55:417–428, 2016.
- [48] Si Wenrong, Li Junhao, Yuan Peng, and Li Yanming. Digital detection, grouping and classification of partial discharge signals at dc voltage. *IEEE Transactions on Dielectrics and Electrical Insulation*, 15(6), 2008.
- [49] MG Danikas, N Gao, and M Aro. Partial discharge recognition using neural networks: a review. *Electrical Engineering*, 85(2):87–93, 2003.
- [50] CS Chang, J Jin, C Chang, Toshihiro Hoshino, Masahiro Hanai, and Nobumitsu Kobayashi. Separation of corona using wavelet packet transform and neural network for detection of partial discharge in gas-insulated substations. *IEEE Transactions on Power Delivery*, 20(2):1363–1369, 2005.

- [51] Demetres Evagorou, A Kyprianou, PL Lewin, A Stavrou, V Efthymiou, AC Metaxas, and GE Georghiou. Feature extraction of partial discharge signals using the wavelet packet transform and classification with a probabilistic neural network. *IET Science, Measurement & Technology*, 4(3):177–192, 2010.
- [52] Adriana Rosa Garcez Castro and Vladimiro Miranda. Knowledge discovery in neural networks with application to transformer failure diagnosis. *IEEE Transactions on Power Systems*, 20(2):717–724, 2005.
- [53] Mang-Hui Wang. Partial discharge pattern recognition of current transformers using an enn. *IEEE Transactions on Power Delivery*, 20(3):1984–1990, 2005.
- [54] T Boczar, S Borucki, A Cichon, and D Zmarzly. Application possibilities of artificial neural networks for recognizing partial discharges measured by the acoustic emission method. *IEEE Transactions on Dielectrics and Electrical Insulation*, 16(1):214–223, 2009.
- [55] A Abubakar Mas’Ud, BG Stewart, and SG McMeekin. Application of an ensemble neural network for classifying partial discharge patterns. *Electric Power Systems Research*, 110:154–162, 2014.
- [56] Abdullahi Mas’ud, Ricardo Albarracín, Jorge Ardila-Rey, Firdaus Muhammad-Sukki, Hazlee Illias, Nurul Bani, and Abu Munir. Artificial neural network application for partial discharge recognition: Survey and future directions. *Energies*, 9(8):574, 2016.
- [57] Fabio Rossi. *Progettazione ed ottimizzazione di sistemi di misura delle scariche parziali per segnali condotti e irradiati*. PhD thesis, Università degli studi di Genova, Facoltà di Ingegneria, 2018.
- [58] Winding wires - test methods - part 5: Electrical properties. International Standard IEC 60851-5:2008, International Electrotechnical Commission, Geneva, CH, 2008. URL <https://webstore.iec.ch/publication/3699>.
- [59] F Guastavino, D Cordano, C Gianoglio, F Rossi, and E Torello. Comparison of conducted and irradiated pd acquisition systems. In *Electrical Insulation and Dielectric Phenomenon (CEIDP), 2017 IEEE Conference on*, pages 365–368. IEEE, 2017.
- [60] F Guastavino, D Cordano, F Rossi, C Gianoglio, and E Torello. Relevance of quantization in pds activity measurements in presence of environmental noise. In *Electrical Insulation and Dielectric Phenomenon (CEIDP), 2017 IEEE Conference on*, pages 373–376. IEEE, 2017.

- [61] F Guastavino, C Gianoglio, E Torello, D Cordano, and A Bruzzone. Comparison between pd acquisition system measurements using different number of bits for the quantization. In *2018 IEEE 2nd International Conference on Dielectrics (ICD)*, pages 1–4. IEEE, 2018.
- [62] Vladimir Naumovich Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.
- [63] Gao Huang, Guang-Bin Huang, Shiji Song, and Keyou You. Trends in extreme learning machines: A review. *Neural Networks*, 61:32–48, 2015.
- [64] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.
- [65] Gaoyang Li, Xiaohua Wang, Xi Li, Aijun Yang, and Mingzhe Rong. Partial discharge recognition with a multi-resolution convolutional neural network. *Sensors*, 18(10):3512, 2018.
- [66] Gaoyang Li, Mingzhe Rong, Xiaohua Wang, Xi Li, and Yunjia Li. Partial discharge patterns recognition with deep convolutional neural networks. In *Condition Monitoring and Diagnosis (CMD), 2016 International Conference on*, pages 324–327. IEEE, 2016.
- [67] VM Catterson and B Sheng. Deep neural networks for understanding and diagnosing partial discharge data. In *Electrical Insulation Conference (EIC), 2015 IEEE*, pages 218–221. IEEE, 2015.
- [68] Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*, 2016.
- [69] Edoardo Ragusa, Christian Gianoglio, Paolo Gastaldo, and Rodolfo Zunino. A digital implementation of extreme learning machines for resource-constrained devices. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2018.
- [70] David Lowe. Adaptive radial basis function nonlinearities, and the problem of generalisation. In *Artificial Neural Networks, 1989., First IEE International Conference on (Conf. Publ. No. 313)*, pages 171–175. IET, 1989.
- [71] Yoh-Han Pao, Gwang-Hoon Park, and Dejan J Sobajic. Learning and generalization characteristics of the random vector functional-link net. *Neurocomputing*, 6(2):163–180, 1994.



- [72] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 2, pages 985–990. IEEE, 2004.
- [73] Paolo Gastaldo, Rodolfo Zunino, Erik Cambria, and Sergio Decherchi. Combining elm with random projections. *IEEE intelligent systems*, 28(6):46–48, 2013.
- [74] Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in neural information processing systems*, pages 1313–1320, 2009.
- [75] Maria-Florina Balcan, Avrim Blum, and Nathan Srebro. A theory of learning with similarity functions. *Machine Learning*, 72(1-2):89–112, 2008.
- [76] Frank Rosenblatt. *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory, 1957.
- [77] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533, 1986.
- [78] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151, 1999.
- [79] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.
- [80] Guang-Bin Huang, Lei Chen, Chee Kheong Siew, et al. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Trans. Neural Networks*, 17(4):879–892, 2006.
- [81] Guang-Bin Huang, Hongming Zhou, Xiaojian Ding, and Rui Zhang. Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2):513–529, 2012.
- [82] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.
- [83] Luca Oneto, Federica Bisio, Erik Cambria, and Davide Anguita. Statistical learning theory and elm for big social data analysis. *ieee CompUTATionAl inTelliGenCe mAGazine*, 11(3):45–55, 2016.
- [84] John Shawe-Taylor, Nello Cristianini, et al. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

- [85] Paolo Gastaldo, Federica Bisio, Sergio Decherchi, and Rodolfo Zunino. Sim-elm: Connecting the elm model with similarity-function learning. *Neural Networks*, 74: 22–34, 2016.
- [86] P Gastaldo, F Bisio, C Gianoglio, E Ragusa, and R Zunino. Learning with similarity functions: a novel design for the extreme learning machine. *Neurocomputing*, 261:37–49, 2017.
- [87] Xia Liu, Shaobo Lin, Jian Fang, and Zongben Xu. Is extreme learning machine feasible? a theoretical assessment (part i). *IEEE Transactions on Neural Networks and Learning Systems*, 26(1):7–20, 2015.
- [88] Shaobo Lin, Xia Liu, Jian Fang, and Zongben Xu. Is extreme learning machine feasible? a theoretical assessment (part ii). *IEEE Transactions on Neural Networks and Learning Systems*, 26(1):21–34, 2015.
- [89] Yoan Miche, Antti Sorjamaa, Patrick Bas, Olli Simula, Christian Jutten, and Amaury Lendasse. Op-elm: optimally pruned extreme learning machine. *IEEE transactions on neural networks*, 21(1):158–162, 2010.
- [90] Sergio Decherchi, Paolo Gastaldo, Alessio Leoncini, and Rodolfo Zunino. Efficient digital implementation of extreme learning machines for classification. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 59(8):496–500, 2012.
- [91] Jiuwen Cao, Zhiping Lin, and Guang-Bin Huang. Self-adaptive evolutionary extreme learning machine. *Neural processing letters*, 36(3):285–305, 2012.
- [92] You Xu and Yang Shu. Evolutionary extreme learning machine–based on particle swarm optimization. In *International Symposium on Neural Networks*, pages 644–652. Springer, 2006.
- [93] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnornet: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision*, pages 525–542. Springer, 2016.
- [94] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *Journal of Machine Learning Research*, 18:187–1, 2017.
- [95] Guang-Bin Huang, Qin-Yu Zhu, KZ Mao, Chee-Kheong Siew, Paramasivan Saratchandran, and Narasimhan Sundararajan. Can threshold networks be trained directly? *IEEE Transactions on Circuits and Systems II: Express Briefs*, 53(3): 187–191, 2006.

- 
- [96] George Casella and Roger L Berger. *Statistical inference*, volume 2. Duxbury Pacific Grove, CA, 2002.
- [97] Uci datasets [online] available. URL <https://archive.ics.uci.edu/ml/datasets.html>.
- [98] Davide Anguita, Alessandro Ghio, Luca Oneto, and Sandro Ridella. In-sample model selection for trimmed hinge loss support vector machine. *Neural processing letters*, 36(3):275–283, 2012.
- [99] Davide Anguita, Alessandro Ghio, Luca Oneto, and Sandro Ridella. In-sample and out-of-sample model selection and error estimation for support vector machines. *IEEE Transactions on Neural Networks and Learning Systems*, 23(9):1390–1406, 2012.
- [100] Leo Breiman. Some properties of splitting criteria. *Machine Learning*, 24(1):41–47, 1996.
- [101] Tianqi Wu, Min Yao, and Jianhua Yang. Dolphin swarm extreme learning machine. *Cognitive Computation*, 9(2):275–284, 2017.
- [102] Xiaowei Xue, Min Yao, Zhaohui Wu, and Jianhua Yang. Genetic ensemble of extreme learning machine. *Neurocomputing*, 129:175–184, 2014.
- [103] Hui-yuan Tian, Shi-jian Li, Tian-qi Wu, and Min Yao. An extreme learning machine based on artificial immune system. In *The 8th International Conference on Extreme Learning Machines (ELM2017)*, Yantai, China, 2017.
- [104] Aakash Patil, Shanlan Shen, Enyi Yao, and Arindam Basu. Hardware architecture for large parallel array of random feature extractors applied to image recognition. *Neurocomputing*, 261:193–203, 2017.
- [105] Yi Chen, Enyi Yao, and Arindam Basu. A 128 channel 290 gmacs/w machine learning based co-processor for intention decoding in brain machine interfaces. In *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on*, pages 3004–3007. IEEE, 2015.
- [106] Rahat Iqbal, Faiyaz Doctor, Brian More, Shahid Mahmud, and Usman Yousuf. Big data analytics and computational intelligence for cyber–physical systems: Recent trends and state of the art applications. *Future Generation Computer Systems*, 2017.
- [107] Chin-Teng Lin, Yu-Ting Liu, Shang-Lin Wu, Zehong Cao, Yu-Kai Wang, Chih-Sheng Huang, Jung-Tai King, Shi-An Chen, Shao-Wei Lu, and Chun-Hsiang

- Chuang. Eeg-based brain-computer interfaces: A novel neurotechnology and computational intelligence method. *IEEE Systems, Man, and Cybernetics Magazine*, 3(4):16–26, 2017.
- [108] Paolo Gastaldo, Luigi Pinna, Lucia Seminara, Maurizio Valle, and Rodolfo Zunino. A tensor-based approach to touch modality classification by using machine learning. *Robotics and Autonomous Systems*, 63:268–278, 2015.
- [109] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L Reyes-Ortiz. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *International workshop on ambient assisted living*, pages 216–223. Springer, 2012.
- [110] Arindam Basu, Sun Shuo, Hongming Zhou, Meng Hiot Lim, and Guang-Bin Huang. Silicon spiking neurons for hardware implementation of extreme learning machines. *Neurocomputing*, 102:125–134, 2013.
- [111] Jose V Frances-Villora, A Rosado-Muñoz, José M Martínez-Villena, Manuel Bataller-Mompean, Juan Fco Guerrero, and Marek Wegrzyn. Hardware implementation of real-time extreme learning machine in fpga: analysis of precision, resource occupation and performance. *Computers & Electrical Engineering*, 51:139–156, 2016.
- [112] Alfredo Canziani, Eugenio Culurciello, and Adam Paszke. Evaluation of neural network architectures for embedded systems. In *Circuits and Systems (ISCAS), 2017 IEEE International Symposium on*, pages 1–4. IEEE, 2017.
- [113] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in neural information processing systems*, pages 3123–3131, 2015.
- [114] Robert Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural networks for perception*, pages 65–93. Elsevier, 1992.
- [115] Enyi Yao and Arindam Basu. Vlsi extreme learning machine: A design space exploration. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 25(1):60–74, 2017.
- [116] Chang-Hung Tsai, Yu-Ting Chih, Wing Hung Wong, and Chen-Yi Lee. A hardware-efficient sigmoid function with adjustable precision for a neural network system. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 62(11):1073–1077, 2015.

- 
- [117] Makoto Murase. Linear feedback shift register, February 18 1992. US Patent 5,090,035.
- [118] Linear feedback shift register maximal length table. URL [https://www.xilinx.com/support/documentation/application\\_notes/xapp052.pdf](https://www.xilinx.com/support/documentation/application_notes/xapp052.pdf).
- [119] Edward Gulski, Peter HF Morshuis, and Frederik H Kreuger. Automated recognition of partial discharges in cavities. *Japanese journal of applied physics*, 29(7R): 1329, 1990.
- [120] F Guastavino, C Gianoglio, E Torello, D Cordano, and A Bruzzone. A predictive maintenance remote system based on partial discharges measurements on wind turbines. In *2018 IEEE 2nd International Conference on Dielectrics (ICD)*, pages 1–3. IEEE, 2018.