

# **POLITECNICO DI MILANO**

Scuola di Ingegneria Industriale e dell'Informazione

Corso di Laurea Magistrale in  
Ingegneria Gestionale



Quality control via sensor data: design of nonparametric control charts based on Self-Organizing Map

Relatore: Chiar.ma Prof.ssa Bianca Maria COLOSIMO

Correlatori: Ing. Marco GRASSO  
Ing. Matteo Francesco MAGGIONI

Tesi di Laurea Magistrale di:

Massimiliano DI MAURO Matr. 787394

Anno Accademico 2013 - 2014

# Ringraziamenti

*Desidero ringraziare prima di tutti la Prof.ssa Bianca Maria Colosimo per avermi dato la possibilità di svolgere questa tesi all'interno del suo gruppo di ricerca.*

*Ringrazio l'Ing. Marco Grasso e l'Ing. Matteo Francesco Maggioni per il prezioso aiuto fornitomi durante lo sviluppo della tesi.*

*Inoltre ringrazio i miei amici che mi hanno sostenuto durante questa esperienza.*

*Infine, ma non per importanza, ringrazio la mia famiglia che nonostante la lontananza mi sono stati sempre vicini in ogni momento del mio percorso di studi.*

# Indice

<b>Abstract</b>	<b>10</b>
<b>1 Introduzione</b>	<b>11</b>
<b>2 Reti Neurali</b>	<b>16</b>
2.1 Framework Teorico . . . . .	16
2.1.1 Paradigmi di apprendimento . . . . .	18
2.1.2 Algoritmi di apprendimento . . . . .	20
2.1.3 Potenzialità e limiti applicativi . . . . .	21
2.2 Stato dell'arte . . . . .	23
2.2.1 Reti neurali in Statistical Process Control . . . . .	23
2.2.2 Ambiti applicativi . . . . .	24
<b>3 Self-Organizing Map</b>	<b>30</b>
3.1 Motivazioni a supporto . . . . .	30
3.2 Il paradigma della Som . . . . .	33
3.2.1 Tipologie di learning . . . . .	34
3.2.2 Parametri decisionali . . . . .	39
3.3 Misure di qualità della Som . . . . .	41
3.3.1 Errore di quantizzazione . . . . .	42
3.3.2 Errore topografico . . . . .	43
3.4 La Som come classificatore unicategorico . . . . .	43
<b>4 Analisi retrospettiva di fase I</b>	<b>45</b>
4.1 Metodi di decontaminazione basati su Self Organizing Map . . . . .	46

<i>INDICE</i>	5
4.1.1 Approccio di Vesanto . . . . .	46
4.1.2 Approccio di Xuefeng . . . . .	49
4.1.3 Approccio proposto . . . . .	50
4.2 Metodi di decontaminazione basati su clustering . . . . .	58
4.2.1 Approccio di Chen-Birch . . . . .	58
<b>5 Analisi delle Prestazioni: simulation study</b>	<b>62</b>
5.1 Definizione dataset di riferimento . . . . .	62
5.2 Effetto dei parametri decisionali sulle prestazioni della Som . . . . .	66
5.2.1 Effetto del numero dei nodi . . . . .	67
5.2.2 Effetto del numero di iterazioni . . . . .	68
5.2.3 Effetto della numerosità del dataset . . . . .	73
5.3 Effetto della percentuale di contaminazione . . . . .	73
5.4 Effetto del numero di cluster . . . . .	76
5.5 Confronto tra approcci di decontaminazione . . . . .	84
<b>6 Conclusioni e Sviluppi Futuri</b>	<b>88</b>
6.1 Conclusioni . . . . .	88
6.2 Sviluppi futuri . . . . .	90
<b>Acronimi</b>	<b>91</b>
<b>Bibliografia</b>	<b>98</b>
<b>A Risultati Simulazioni</b>	<b>99</b>

# Elenco delle figure

1.1	Esempio di segnale di forza in fresatura [1] . . . . .	12
2.1	Neurone Artificiale [2] . . . . .	17
2.2	Tassonomia reti neurali feed-forward e recurrent [3] . . . . .	18
3.1	Competitive Learning [4] . . . . .	31
3.2	Cappello messicano [5] . . . . .	32
3.3	Rete di Kohonen [6] . . . . .	33
3.4	Applicazione Som:esempio bidimensionale . . . . .	42
4.1	Confronto statistiche Scenario I . . . . .	59
4.2	Confronto statistiche Scenario II . . . . .	59
4.3	Confronto statistiche Scenario II . . . . .	59
5.1	Scenario I . . . . .	63
5.2	Scenario II . . . . .	64
5.3	Scenario III . . . . .	65
5.4	Falsi negativi Scenario I valutazione nodi . . . . .	69
5.5	Falsi negativi Scenario II valutazione nodi . . . . .	69
5.6	Falsi negativi Scenario III valutazione nodi . . . . .	69
5.7	Falsi positivi Scenario I valutazione nodi . . . . .	70
5.8	Falsi positivi Scenario II valutazione nodi . . . . .	70
5.9	Falsi positivi Scenario III valutazione nodi . . . . .	70
5.10	FCC Scenario I valutazione nodi . . . . .	71
5.11	FCC Scenario II valutazione nodi . . . . .	71
5.12	FCC Scenario III valutazione nodi . . . . .	71

5.13	Andamento QE fase di ordinamento . . . . .	74
5.14	Andamento aggiornamento pesi fase di ordinamento . . . . .	74
5.15	Andamento QE fase di convergenza . . . . .	75
5.16	Andamento aggiornamento pesi fase di convergenza . . . . .	75
5.17	Dipendenza numerosità training set . . . . .	77
5.18	Falsi positivi Scenario I valutazione contaminazione . . . . .	78
5.19	Falsi positivi Scenario II valutazione contaminazione . . . . .	78
5.20	Falsi positivi Scenario III valutazione contaminazione . . . . .	78
5.21	FCC Scenario I valutazione contaminazione . . . . .	79
5.22	FCC Scenario II valutazione contaminazione . . . . .	79
5.23	FCC Scenario III valutazione contaminazione . . . . .	79
5.24	Falsi negativi Scenario I valutazione contaminazione . . . . .	80
5.25	Falsi negativi Scenario II valutazione contaminazione . . . . .	80
5.26	Falsi negativi Scenario III valutazione contaminazione . . . . .	80
5.27	Confronto Falsi positivi tra k automatico e k=2 . . . . .	81
5.28	Confronto FCC tra k automatico e k=2 . . . . .	81
5.29	Confronto Falsi negativi tra k automatico e k=2 . . . . .	81
5.30	Confronto falsi negativi Chen Birch e Correlazione Scenario I . . . .	87
5.31	Confronto falsi negativi Chen Birch e Correlazione Scenario II . . . .	87
5.32	Confronto falsi negativi Chen Birch e Correlazione Scenario III . . . .	87

# Elenco delle tabelle

2.1	Tabella Letteratura TCM . . . . .	25
2.2	Tabella Letteratura MCM . . . . .	27
3.1	Tabella parametri di simulazione scelti Som 2D . . . . .	41
4.1	Tabella falsi positivi con IC batch means con $\alpha = 5\%$ . . . . .	58
5.1	Tabella parametri di scostamento Scenario 1 . . . . .	63
5.2	Tabella parametri di scostamento Scenario 2 . . . . .	64
5.3	Tabella parametri di scostamento Scenario 3 . . . . .	65
5.4	Tabella tipologia di classificazione . . . . .	66
5.5	Tabella parametri di simulazione scelti analisi nodi . . . . .	68
5.6	Tabella parametri di simulazione scelti analisi iterazioni . . . . .	72
5.7	Tabella parametri di simulazione scelti analisi numerosità dataset . . . . .	76
5.8	Tabella parametri di simulazione scelti analisi contaminazione . . . . .	82
5.9	Tabella parametri di simulazione scelti analisi numero di cluster . . . . .	83
5.10	Tabella parametri di simulazione scelti confronto metodi . . . . .	85
5.11	Tabella falsi positivi con IC batch means con $\alpha = 5\%$ . . . . .	86
A.1	Tabella K scelti automaticamente . . . . .	99

# Abstract

In industrial practice, quality control traditionally occurred downstream to the output. In order to decrease costs linked to industrial wastes and to ensure end products of better quality, the control step is being anticipated through the analysis of the behavior of signals characterizing the process signature. In science, this paradigm shift in the monitoring approach is made possible by the availability of increasingly cheaper sensors for industrial use. Hence, the object of analysis is no longer the end product, but the process needed to realize it, that is described by the signals acquired during the processing. However, this shift of approach involves several challenges related with the complexity of the signals and the violation of traditional assumptions for quality control. Nonparametric and machine learning techniques (i.e., neural networks) are suitable to deal with distributional assumption violations and to cope with complex data patterns. Recent studies showed that those methods can be used in quality control problem, by exploiting only in-control data for training. Nevertheless, two critical issues include: the need for large amounts of training samples and the absence of out-of-control data in training set. The former issue is not a problem when working with signal data, due to the very high sampling. Literature lacks approaches that allow neural networks to recognize anomalous observations in the training phase so that they can be autonomously removed from the dataset, in order to increase reliability and effectiveness in the estimation of the technological signature. This thesis aims to search an innovative method for the identification of anomalies in the training phase of a specific type of neural network: the Self-Organizing Map. As far as Statistical Process Control is concerned, this work belongs to the research field of Profile Monitoring, that is the analysis of stability over time of a functional relationship between the process signal and the product quality. The method presented will be tested on a dataset proposed in the literature, and compared with the current state of the art.



# Sommario

In ambito industriale tradizionalmente il controllo di qualità è sempre avvenuto a valle della realizzazione dell'output. Al fine di ridurre i costi legati agli scarti e garantire una migliore qualità dei prodotti finiti, oggi si sta cercando di anticipare questa fase di controllo analizzando il comportamento dei segnali che caratterizzano la firma di processo. In ambito scientifico, questo cambio di approccio è meglio noto come *paradigm shift* del monitoraggio ed è reso possibile grazie alla disponibilità di sensoristica a costi sempre più ridotti. L'oggetto dell'analisi quindi non è più il prodotto finale ma il processo produttivo che lo realizza descritto dai segnali acquisiti durante la lavorazione. Questo cambio di approccio coinvolge numerose sfide connesse con la complessità dei segnali e con la violazione delle ipotesi tradizionali per il controllo qualità. Le tecniche non parametriche derivanti dal campo del machine learning (i.e., neural network) ben si adattano a trattare la violazione delle ipotesi distribuzionali e far fronte a complessi pattern dei dati. Studi recenti hanno mostrato come questi metodi possano essere usati nei problemi di controllo qualità, sfruttando solo dati in controllo per l'addestramento. Tuttavia, due criticità sono: la necessità di avere una grande quantità di campioni di training e l'assenza di dati fuori controllo tra questi. Il primo non sarà un problema quando si lavora con i dati provenienti da segnali, a causa dell'elevato campionamento, il secondo sarà invece un aspetto critico. In letteratura mancano degli approcci che consentano alle reti neurali di riconoscere osservazioni anomale in fase di training ed eliminarle in maniera autonoma dal proprio dataset al fine di incrementare affidabilità ed efficacia nella stima della firma tecnologica. L'obiettivo di questo lavoro di tesi consiste quindi nella ricerca di un metodo innovativo per l'identificazione di anomalie in fase di addestramento di una specifica tipologia di rete neurale: la Self-Organizing Map. In ambito di *Statistical Process Control* questo lavoro rientra nel filone di ricerca del *Profile Monitoring*, ovvero l'analisi della stabilità nel tempo di una relazione che lega il segnale di processo con la qualità di prodotto. Il metodo presentato verrà testato su un dataset già presente in letteratura e confrontato con l'attuale stato dell'arte.

# Capitolo 1

## Introduzione

L'approccio tradizionale nell'ambito del controllo industriale della qualità prevede la misura a campione sul prodotto finito di una o più grandezze di interesse al fine di monitorarne l'andamento temporale e verificare la stabilità del processo. Questa procedura, denominata controllo "post-process", presenta il limite di consentire l'individuazione di un'eventuale anomalia soltanto a valle del campionamento delle parti prodotte. Ciò si traduce in un inevitabile ritardo nell'intervento di correzione del processo che nel frattempo continuerà a produrre nuove componenti che potrebbero risultare non conformi, comportando quindi un aumento degli scarti e un conseguente incremento dei costi di produzione oltreché una notevole perdita di tempo. La tecnologia odierna permette di acquisire grandi quantità di informazioni durante il processo utilizzando sensori installati nelle macchine utensili e nei sistemi produttivi. L'uso in-process di tali informazioni invece delle (o in aggiunta alle) misure post-process, rappresenta uno shift di paradigma nel controllo statico di processo.

Questo lavoro di tesi si concentra su una particolare tipologia di processi manifatturieri caratterizzati da pattern di segnali che si ripetono ciclicamente nel tempo, denominati "profili".

Il *Profile Monitoring* è un campo di ricerca recente nello Statistical Process Control, che sta attirando l'interesse in vari settori industriali [7]. Questo approccio di analisi viene principalmente utilizzato quando, appunto, i dati di processo seguono un profilo e la stabilità di questa relazione funzionale viene controllata

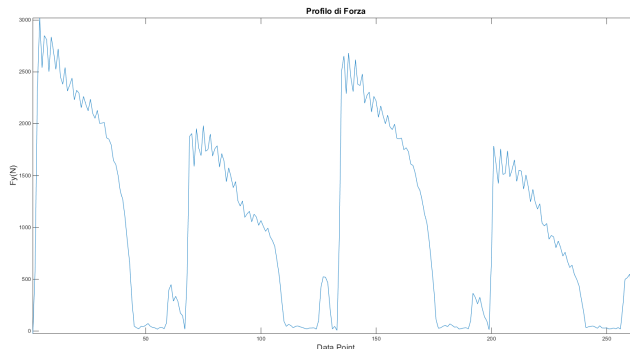


Figura 1.1: Esempio di segnale di forza in fresatura [1]

nel tempo. In molte applicazioni industriali, il processo o la qualità del prodotto non possono essere semplicemente modellati come una variabile casuale ma vengono invece ricondotte ad una forma funzionale, cioè ad una funzione che riguarda una risposta ad una o più variabili di posizione[8]. Esempi di dati funzionali comprendono studi di calibrazione, specifiche geometriche e monitoraggio dei segnali di processo, in figura 1.1 viene illustrato un esempio di profilo del segnale della componente  $y$  della forza di taglio in fresatura. Se si suppone che i dati funzionali sono legati alla qualità del prodotto o del processo, allora possiamo monitorare la stabilità di questo profilo nel tempo. Il *Profile Monitoring* prevede l'identificazione di deviazioni del profilo da un modello nominale o in-controllo al fine di individuare il momento in cui il processo devia verso condizioni di fuori controllo.

Così come l'SPC in generale, anche il *Profile Monitoring* viene condotto in due fasi, denominate fase I e fase II. Nella fase I del *Profile Monitoring* l'obiettivo è quello di separare i dati di processo in controllo dai dati di processo fuori controllo nel dataset di dati storici acquisiti. Si esegue un'analisi preliminare del processo in cui si raccoglie e si analizza un insieme retrospettivo di dati del processo al fine di determinare se rappresentano una situazione in stato di controllo e costruire quindi limiti di controllo attendibili per monitorare correttamente la produzione futura. Le prestazioni dell'analisi di fase I si possono misurare valutando se si riescono ad identificare correttamente i dati di processo fuori controllo nel dataset di addestramento. In questo caso un processo fuori controllo è rappresentato da una condizione in cui a un certo punto avviene un cambiamento nella relazione funzio-

nale tra la risposta e le variabili esplicative. La fase I è di importanza strategica perché influenza direttamente le prestazioni del sistema di monitoraggio quando, in fase II, ci si concentra sull'analisi della stabilità di nuove realizzazioni del processo produttivo. Un addestramento di fase I su un insieme di dati che al suo interno include informazioni anomale sul processo, distorce la stima della firma di processo condizionando la capacità del sistema di monitoraggio nell'individuare condizioni degradate del processo.

L'obiettivo principale delle applicazioni di fase I consiste quindi nel:

1. Capire la variazione naturale del processo nel tempo, distinguendo tra la variabilità interna al profilo e quella presente tra profili.
2. Valutare la stabilità del processo, con la separazione dei profili in controllo da quelli fuori controllo.
3. Modellare il processo nella condizione di "in controllo".

Nella fase I avverrà l'individuazione e la rimozione di eventuali cause assegnabili nei dati ovvero variazioni corrispondenti ad eventi e anomalie del processo eliminabili. Per l'identificazione di queste cause assegnabili è però richiesta una accurata modellazione della firma di processo, che può essere effettuata con diverse tecniche.

Nell'ambito del *Profile Monitoring* esistono diversi approcci di modellazione, tra cui Simple Linear Regression, Multiple and Polynomial Regression, Mixed Models, Nonlinear Regression Models, Wavelets [7]. Queste tecniche di *Profile Monitoring* sono state ampiamente applicate nel campo del "Process Monitoring" con l'obiettivo, ad esempio, di valutare lo stato di salute di una macchina utensile o di valutare la qualità del processo produttivo di un prodotto, attraverso l'analisi dei segnali rilevati dai sensori disponibili a bordo macchina. A tal fine una variabile o un insieme di esse (e.g., forza di taglio, assorbimento di corrente, vibrazioni) vengono acquisite mediante sensori e processate per poi essere comparate con un valore atteso. Ogni deviazione da questo rappresenterà una condizione di anomalia. La modellazione di questi segnali spesso è molto complessa e, specialmente in campo industriale, non si può ricorrere a tecniche troppo sofisticate nella trattazione di questa mole di dati. Per questo motivo stanno riscuotendo il favore dell'industria i

metodi derivanti dall'intelligenza artificiale, detti anche “data driven”, ovvero quell'insieme di tecniche che senza un modello parametrico sono in grado di apprendere dai dati la firma di processo, senza la necessità di fare ipotesi a priori.

Tra le tecniche derivanti dall'intelligenza artificiale hanno avuto grande rilevanza le reti neurali, infatti in diversi lavori è stato dimostrato come esse siano uno strumento promettente per il monitoraggio di complessi processi manifatturieri per via della loro elevata capacità di apprendimento e robustezza al rumore. Tuttavia l'utilità di questi metodi è limitata sia dalla richiesta di dataset di grosse dimensioni per il training, sia dalla qualità dei dati che devono essere forniti per il corretto addestramento. Nonostante queste criticità, che ne limitano la reale applicazione industriale, la gran parte della letteratura riguardante l'applicazione di reti neurali per il monitoraggio del processo trascura questi aspetti e parte dal presupposto che si abbia a disposizione un ampio dataset di training per l'addestramento della rete composto da dati provenienti con certezza da condizioni corrispondenti al funzionamento normale della macchina utensile. Questa condizione è in realtà scarsamente ricreabile dato che il processo evolve dinamicamente nel corso del tempo e spesso intervengono diversi fattori difficilmente individuabili e catalogabili che ne influenzano la stabilità. Pertanto l'insieme dei segnali acquisiti potrebbe contenere al suo interno anche profili derivanti da condizioni fuori controllo o anomale. Una contaminazione di questo tipo nel dataset di addestramento della rete potrebbe avere un forte impatto nelle prestazioni del sistema di monitoraggio portando a risultati pessimi. Queste motivazioni hanno quindi penalizzato fino ad ora l'utilizzo di questo strumento per il monitoraggio, poiché non consente di effettuare un'analisi retrospettiva del dataset di addestramento.

In questo studio si è quindi cercato di determinare un metodo in grado di svolgere l'analisi retrospettiva dei dati di addestramento partendo da uno specifico strumento, ovvero la rete neurale *Self Organizing Map*. Lo scopo del lavoro consiste nel realizzare un sistema di monitoraggio che, nonostante venga addestrato con dati rumorosi e parzialmente contaminati, consenta di riconoscere e separare quei profili che si discostano dalla condizione normale operativa, effettuando quindi una stima della reale firma del processo. A tal scopo verrà utilizzato un dataset di

benchmark discusso già in letteratura e su di esso verrà articolato e sviluppato il metodo proposto evidenziandone i punti di forza, di debolezza e gli sviluppi futuri che questo ramo di ricerca potrà seguire.

Il lavoro è stato strutturato quindi secondo il seguente schema:

**Capitolo 2:** Presentazione sintetica dello stato dell'arte relativo all'uso delle reti neurali in ambito SPC.

**Capitolo 3:** Illustrazione dettagliata della rete Self-Organizing Map

**Capitolo 4:** Introduzione all'analisi retrospettiva basata su Self-Organizing Map.

**Capitolo 5:** Presentazione dei risultati ottenuti e confronto di prestazioni con metodi alternativi.

**Capitolo 6:** Conclusione del lavoro svolto e possibili sviluppi futuri.

# Capitolo 2

## Reti Neurali

### 2.1 Framework Teorico

Kohonen in [5] ha definito le reti neurali in questo modo “*Artificial neural networks are massively parallel interconnected networks of simple (usually adaptive) elements and their hierarchical organizations which are intended to interact with the objects of the real world in the same way as biological nervous system do*”. Una rete neurale è un modello matematico\informatico di calcolo che trae ispirazione dal funzionamento del sistema nervoso biologico. Tale modello risulta composto da un fitto insieme di interconnessioni tra neuroni, che costituiscono delle vere e proprie unità di elaborazione. Nella maggior parte dei casi una rete neurale artificiale è un sistema adattativo che cambia la sua struttura, basata su informazioni esterne o interne che scorrono attraverso la rete, durante la fase di apprendimento. Una rete riceve i segnali esterni su uno strato di nodi d'ingresso ciascuno dei quali è collegato con numerosi nodi interni, organizzati in più livelli. Ciascuna unità svolge un'attività molto semplice che consiste nel diventare attiva nel caso in cui la quantità totale di segnale ricevuta supera la soglia di attivazione. Divenendo attiva l'unità invierà un segnale lungo i canali di comunicazione verso le altre unità a cui è connessa; ciascun arco di collegamento agisce da filtro aumentando o diminuendo l'intensità del segnale a seconda delle sue caratteristiche. Queste connessioni simulano le sinapsi biologiche, da cui prendono il nome; poichè il loro ruolo consiste nel pesare l'intensità dei segnali trasmessi, essi vengono comunemente definiti pesi

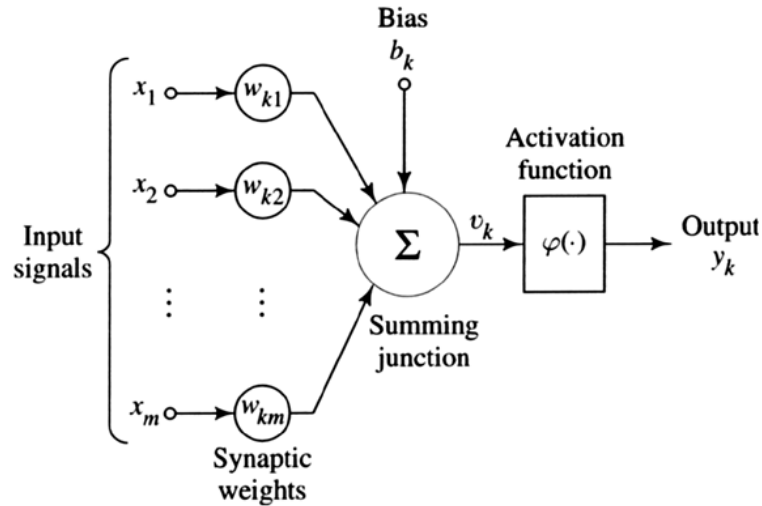


Figura 2.1: Neurone Artificiale [2]

sinaptici o semplicemente pesi. In figura 2.1 si ha una rappresentazione schematica di un neurone artificiale, dove  $w_{i,k}$  è il peso della connessione dal neurone  $i$  a quello  $k$  e  $s_k$  è l'attivazione o l'output del neurone.

Sulla base del pattern di connessione, le reti neurali possono a loro volta essere suddivise in due macro categorie che sono:

- Feed-forward network, nella quale il grafo non avrà loop.
- Recurrent (o feedback) network, nella quale sono presenti dei loop di feedback.

Nella tipologia più comune delle reti feed-forward, chiamata multilayer perceptron (MLP), i neuroni sono organizzati in strati che hanno connessioni unidirezionali tra loro. Connessioni diverse conducono a comportamenti differenti della rete. Le reti feed-forward sono generalmente statiche in quanto producono solo un insieme di valori di output piuttosto che una sequenza di valori per un dato input. Inoltre esse sono senza memoria nel senso che la loro risposta ad un input è indipendente dagli stati precedenti della rete. Le reti ricorrenti, al contrario, sono sistemi dinamici. Quello che le distingue dalle feed-forward è che avranno in un nodo almeno un loop di feedback, ovvero l'output del nodo verrà ripresentato alla rete come un nuovo input; questo porterà la rete in un nuovo stato. Tra le principali reti feed-forward riscontriamo la rete Single layer Perceptron e la Multilayer Perceptron (MLP) che è



un'estensione della precedente. Tra le reti ricorrenti quelle maggiormente impiegate sono la Kohonen denominata anche lattice dove i nodi di output sono disposti in un array mono o bi dimensionale, la ART e la Hopfield che è caratterizzata dall'aver pesi simmetrici.

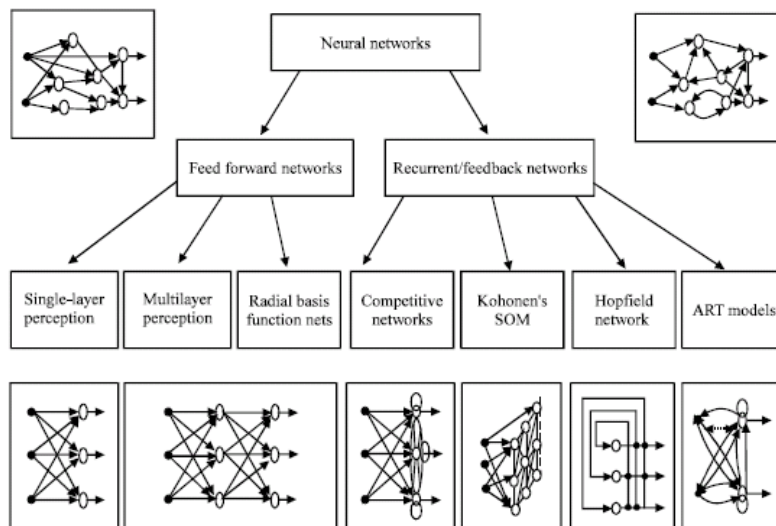


Figura 2.2: Tassonomia reti neurali feed-forward e recurrent [3]

In figura 2.2 è presente la tassonomia principale relativa alle diverse architetture di rete di tipo feed-forward e recurrent. Per capire o progettare il processo di apprendimento della rete occorre innanzitutto avere un modello dell'ambiente nella quale opera, ovvero conoscere quale informazione renderle disponibile al momento dell'addestramento. Inoltre servirà sapere come avviene l'aggiornamento dei pesi della rete, vale a dire l'algoritmo di apprendimento attraverso cui avviene l'update della rete. Questi due aspetti verranno trattati nei paragrafi successivi, che presenteranno i paradigmi e gli algoritmi di apprendimento principali utilizzati nelle varie architetture di rete.

### 2.1.1 Paradigmi di apprendimento

Per paradigma di apprendimento si intende il modello dell'ambiente in cui opera la rete neurale, vi sono due categorie principali di questi paradigmi che sono quello supervisionato e non supervisionato che verranno brevemente illustrati di seguito.

### **Apprendimento supervisionato**

Questa tipologia di apprendimento si basa sulla disponibilità di una collezione di coppie, costituita dal dato del problema con la sua soluzione, che viene generalmente ripartita in due insiemi: un training set, utilizzato per l'apprendimento, e un test o validation set per verificare, ad addestramento concluso, che la rete non si limiti a memorizzare i casi del training set ma possa anche trovare una soluzione appropriata per casi analoghi ad essa ignoti, manifestando così l'importante capacità di generalizzazione. La dicitura "supervisionata" deriva dal fatto che dietro alla rete che impara esiste un "insegnante" il quale premia o punisce la rete in funzione delle sue prestazioni di apprendimento. Questa tipologia di training è utilizzata in particolare nell'algoritmo di back propagation. Durante l'apprendimento la rete riceve in input i dati  $X$  con il rispettivo output desiderato e restituisce un output effettivo  $Y$ , generalmente diverso da quello desiderato nelle fasi iniziali dell'algoritmo. Ne consegue, quindi, un errore quadratico medio su tutti gli esempi del training set e, pertanto, l'algoritmo dovrà cercare di minimizzare questo errore apportando modifiche ai pesi sinaptici.

### **Apprendimento non supervisionato**

L'apprendimento senza supervisione, invece, non fa riferimento a una casistica precostituita di esempi. La rete impara a rispondere in modo ordinato alla statistica degli stimoli esterni che le vengono sottoposti auto-organizzando la propria struttura in modo tale che stimoli simili attivino neuroni vicini e stimoli tra loro diversi attivino neuroni lontani. Il mondo degli input esterni viene così proiettato in una mappa topologica bidimensionale, nella quale input simili attivano zone vicine e input differenti attivano zone lontane. Ne consegue una classificazione automatica dei vari tipi di input. Generalmente la proiezione avviene con una riduzione di dimensionalità. L'apprendimento senza supervisione è, inoltre, di tipo competitivo, nel senso che per ogni input vengono attivati più neuroni ma un solo di essi vince la competizione e viene premiato con una modifica dei suoi pesi sinaptici tale da sintonizzarlo maggiormente con quell'input. Questo tipo di apprendimento è utilizzato in particolare dalla rete di Kohonen o SOM.

### 2.1.2 Algoritmi di apprendimento

Una rete neurale durante la fase di apprendimento modifica i suoi pesi in funzione dell'esperienza aggiornandosi progressivamente. Un aspetto da tenere in considerazione è la legge di apprendimento, infatti durante il training i pesi sinaptici  $W$  vengono modificati secondo un opportuno algoritmo:

$$W(t + 1) = W(t) + \Delta W \quad (2.1)$$

I principali algoritmi di apprendimento sono:

1. Legge di Hebb

$$\Delta W_{i,j} = \eta S_i S_j \quad (2.2)$$

L'idea alla base di questa legge introdotta in [9] è quella di rafforzare i legami tra unità della stessa tipologia. Ad esempio se un'unità  $u_j$  riceve un input da un'altra unità  $u_i$  e se entrambe hanno alti livelli di attivazione il legame si rinforzerà e quindi il peso  $W_{i,j}$  verrà incrementato. Dove  $\eta$  è un coefficiente compreso tra zero e uno, denominato tasso di apprendimento.

2. Legge del gradiente

$$\Delta W_{i,j} = -\eta \frac{dE}{dW_{i,j}} \quad (2.3)$$

dove  $E$  rappresenta un'opportuna metrica dell'errore, relativo a una casistica di esempi che forniscono per ogni input l'output corretto o desiderato  $D_i$  ( $Y_i$  è invece l'output calcolato della rete), consistente in genere nell'errore quadratico medio:

$$E = \frac{1}{m} \sum_i^M (Y_i - D_i)^2 \quad i = 1, \dots, M \quad (2.4)$$

Tale algoritmo introdotto in [10], aumenta i pesi sinaptici se la derivata dell'errore è negativa e li diminuisce in caso contrario. Il coefficiente  $\eta$  è ancora il learning rate compreso tra zero e uno, mentre  $M$  è la numerosità del dataset di addestramento. Questa regola equivale a una discesa, lungo la superficie dell'errore secondo linee di massima pendenza, cercando euristicamente un minimo di  $E$ .

### 3. Legge di Kohonen

$$\Delta W_i = \eta(X - W_i) \quad (2.5)$$

dove  $X$ ,  $W_i$  sono due vettori le cui componenti sono rispettivamente gli  $N$  input  $X_j$  e gli  $N$  pesi sinaptici  $W_{i,j}$  del neurone  $i$ . Tale algoritmo presentato in [6] equivale a una piccola rotazione ( $\eta < 1$ ) del vettore peso verso quello di input. Il suo scopo infatti è organizzare una selva inizialmente casuale di vettori peso (punti  $W_i$ ) in modo tale che ciascuno emigri verso il centroide di un gruppo omogeneo di punti  $X_i$ .

## 2.1.3 Potenzialità e limiti applicativi

### Potenzialità delle reti neurali

La rete neurale trae la sua efficacia, dal punto di vista computazionale, dalla struttura parallela, mentre dal punto di vista dell'apprendimento il suo punto di forza è sicuramente la capacità di generalizzazione dei casi. Tale generalizzazione consiste nella capacità della rete di produrre output ragionevoli in corrispondenza di ingressi mai incontrati nella sessione di training. Questi due aspetti rendono possibile la risoluzione di problemi complessi che sono correntemente intrattabili. Gli aspetti principali che ne fanno uno strumento così versatile sono [4]:

#### 1. Robustezza

Una rete neurale è resistente al rumore, ovvero è in grado di continuare a dare una risposta corretta anche se alcune delle sue connessioni vengono eliminate o lesionate o se viene aggiunto del rumore al segnale d'ingresso, ai canali di trasmissione o alla funzione di attivazione dei nodi. Tale proprietà è comune anche ai sistemi nervosi biologici dove la capacità di apprendere e ricordare non viene alterata in modo sostanziale dalla perdita continua di neuroni. In questo caso le reti lesionate possono essere addestrate nuovamente per riacquistare le abilità perse.

## 2. Flessibilità

Un modello neurale può essere impiegato per un grande numero di finalità diverse: esso non ha bisogno di conoscere le proprietà del dominio specifico di applicazione perché le apprende in base all'esperienza. Questo non significa che un qualsiasi modello neurale possa essere utilizzato per tutti i tipi di compiti, ma implica che l'utente non deve necessariamente conoscere le soluzioni dettagliate e analitiche che caratterizzano il problema sotto indagine. In generale l'utente di una rete neurale deve essere in grado di individuare precisamente le finalità del progetto, il tipo di compito e una serie di vincoli al fine di valutare qual è il modello neurale che risulta più appropriato.

## 3. Generalizzazione

Una rete neurale che è stata addestrata su un numero limitato di esempi è in grado di produrre una risposta adeguata a dei pattern d'ingresso che non ha mai visto in precedenza, ma che presentano tuttavia qualche somiglianza con gli esempi presentati durante la fase di addestramento. Questa proprietà deriva in parte dal fatto che molti modelli neurali rappresentano internamente un numero di associazioni stimolo-risposta più grande del numero di sinapsi disponibili; nel far questo la rete neurale tende a estrarre le caratteristiche invarianti dei pattern d'ingresso piuttosto che memorizzare ciascun singolo pattern. La capacità di generalizzare a nuovi stimoli è una caratteristica molto apprezzata nei tipici campi di applicazione delle reti neurali dove spesso è impossibile ottenere una collezione esaustiva di tutti i dati su cui la rete neurale dovrà operare.

### Limiti delle reti neurali

Rispetto a modelli analitici tradizionali le reti neurali hanno lo svantaggio di non consentire la comprensione del fenomeno, funzionando come delle "black box"; infatti dall'output del sistema non c'è modo di risalire al perché di tale risposta. Altra criticità riguarda la complessità della fase di addestramento che richiede solitamente la definizione di una serie di parametri di proget-

tazione la cui determinazione è affidata molte volte all'esperienza di chi la effettua. Inoltre da questa fase dipendono fortemente le prestazioni successive del metodo. Il dataset di addestramento è, pertanto, un fattore estremamente critico, dato che una non completa rappresentazione degli stati del sistema potrebbe portare a scarsa affidabilità. Al contrario una sovraesposizione della rete a determinate condizioni può condurre al fenomeno dell'overfitting, ovvero la condizione per la quale la rete si specializza eccessivamente su alcune condizioni fornendo risultati precisi nei casi di addestramento, ma scarsamente affidabili o errati discostandosi da esso. Tale fenomeno annullerà uno dei vantaggi del metodo stesso che consiste nella capacità di generalizzazione.

## 2.2 Stato dell'arte

Le reti neurali hanno delle caratteristiche intrinseche che le rendono adattabili a svariati domini di applicazione che vanno dall'informatica alla meccanica, dalla biomedica alla finanza, dalla matematica alla chimica. È possibile utilizzare una rete neurale artificiale per l'approssimazione di funzioni non lineari, per effettuare regressioni e predizioni, per la classificazione in categorie (clustering) e per il data mining. Le caratteristiche principali che le rendono uno strumento così versatile sono sicuramente la capacità di generalizzazione, la robustezza alla presenza di rumore nei dati e la flessibilità. In ambito SPC le reti neurali sono state utilizzate in svariate applicazioni e di seguito si fornirà un breve resoconto sui settori dove sono state adottate.

### 2.2.1 Reti neurali in Statistical Process Control

Per l'applicazione delle reti neurali in ambito SPC si possono distinguere fondamentalmente tre classi principali:

- Identificazione del cambiamento strutturale di un processo attraverso una simulazione di una carta di controllo.

- Identificazione di un preciso pattern caratterizzante un segnale speciale di disturbo su i dati del processo.
- Previsione degli andamenti futuri delle osservazioni del processo.

Nel campo della *Pattern Recognition*, sta avendo maggiore considerazione l'analisi di profili, infatti in molte applicazioni il processo o la qualità del prodotto non possono essere espresse in termini di una variabile casuale ma come una relazione funzionale rispetto a uno o più variabili esplicative. La crescente sensorizzazione degli ambienti industriali ha consentito di mettere a disposizione una notevole quantità di informazione sul processo in tempo reale e, a tale riguardo sarà di fondamentale importanza avere strumenti in grado di segnalare lo stato del processo durante la lavorazione. In questo contesto il binomio rete neurale e Controllo Statistico di Processo ha fornito risultati importanti nel campo del monitoraggio dell'utensile e delle condizioni di macchina. Di seguito verranno illustrati in dettaglio questi contesti applicativi sottolineando la letteratura di riferimento, senza la pretesa di essere esaustivi.

### 2.2.2 Ambiti applicativi

#### Tool Condition Monitoring

Il Tool Condition Monitoring (TCM), sta divenendo essenziale nell'automazione manifatturiera per rilevare la presenza di anomalie dello stesso in modo tale da poter intervenire per prevenire danni alla macchina. Un' anomalia può essere un fenomeno graduale come l'usura dell'utensile o improvvisa come la rottura dello stesso. Riconoscere la presenza di queste alterazioni è fondamentale per poter azionare degli interventi correttivi quali il cambio degli utensili in base alla stima effettuata del livello di usura o per salvare il pezzo in lavorazione e/o la macchina in caso di avvenuta rottura. Una delle maggiori difficoltà che si riscontrano proviene principalmente dalla limitata capacità di sensorizzazione dell'ambiente manifatturiero. I sensori raramente sono posizionati in punti di interesse e, di conseguenza, quando gli stessi risultano collocati in posizioni remote non forniscono la chiarezza di misura necessaria per un sistema di monitoraggio affidabile. Per attenuare

questa limitazione spesso vengono impiegati più sensori al fine di aumentare la robustezza e l'affidabilità del sistema. Un'altra difficoltà consiste nell'assenza di un modello analitico accurato che possa tenere conto dei cambiamenti nelle variabili di misura causate dalla variazione delle condizioni di taglio; infatti tali cambiamenti sono spesso erroneamente attribuiti ad anomalie di processo, provocando falsi allarmi.

Apprendimento	Applicazione	Tipologia di rete	Rif.bibliografico
Supervisionato	TCM	MLP(BP)	[11], [12], [13], [14], [15].
Non Supervisionato	TCM	ART	[11], [16], [14].
	TCM	ART-2	[17], [18], [19], [20].
	TCM	SOM	[21].

Tabella 2.1: Tabella Letteratura TCM

Il monitoraggio di processo viene generalmente effettuato attraverso l'analisi delle misure rilevate; a tal fine una variabile o un insieme di esse (e.g., forza, potenza emissione acustica, corrente di alimentazione del motore) vengono acquisite e processate per poi essere comparate con un valore atteso. Ogni deviazione da questo rappresenterà una condizione anomala. L'utilizzo di reti neurali per modellare il legame dei segnali provenienti dal processo con l'usura dell'utensile è stato studiato approfonditamente in passato sia per quanto riguarda la classificazione dello stato del tagliente nelle varie condizioni sia per effettuare la stima dei livelli di usura, in tabella 2.1 vi sono i principali riferimenti bibliografici nel campo. La classificazione ha riguardato inoltre i possibili stati dell'utensile successivi ad eventi improvvisi quali la frattura ad esempio in [20] o la scheggiatura oppure graduali come l'usura studiata in [15] o in [21]. Per quanto riguarda la stima dell'usura vengono usati prevalentemente metodi di natura supervisionata il Single Layer Perceptron o il Multilayer Perceptron, con quest'ultimo che rappresenta largamente la tipologia di rete più utilizzata in tale ambito dato le migliori performance della stessa. Per compiti di classificazione di usura e rottura dell'utensile oltre il MLP vengono impiegate le reti non supervisionate fra le quali le più utilizzate sono quelle che rientrano nella Adaptive Resonance Theory (ART), le Kohonen Feature Map



e i sistemi neuro-fuzzy non supervisionati. Teoricamente un sistema di rilevazione efficace dovrebbe essere in grado di individuare la frattura rapidamente per prevenire danneggiamenti del pezzo in lavorazione ed eliminare fermate non necessarie del processo causate da falsi allarmi. Diverse misure di grandezze fisiche si sono rivelate buoni indicatori, tra queste la forza di taglio, l'emissione acustica, la corrente di alimentazione del motore, la corrente del mandrino e la vibrazione che sono state a lungo investigate per la loro elevata sensibilità al fenomeno. In generale per poter utilizzare una misura occorre soddisfare due requisiti: la misura deve riflettere la presenza di rottura in diverse condizioni di taglio e l'effetto della stessa sulla misurazione deve essere univocamente distinguibile, in modo tale che altre irregolarità nel processo non vengano confuse con essa. Solitamente nella firma del processo l'avvenuta frattura si manifesta attraverso un cambiamento impulsivo della grandezza che, pertanto, supera una determinata soglia. Le reti sono solitamente addestrate mediante feature estratte dai segnali acquisiti. Vengono adottate diverse metodologie di estrazione delle feature quali: la modellazione mediante modelli analitici, l'uso di trasformate applicate ai segnali come quella di Fourier o Wavelet da cui tramite analisi statistiche si ricavano delle feature esplicative del fenomeno (media, curtosi, asimmetria, R.M.S.) [22] o tecniche di intelligenza artificiale. Diversi autori hanno fatto uso dei coefficienti del modello (AR, ARIMA, MA) impiegato per fittare il segnale di interesse. Un esempio si ha in [13], dove vengono usati in ingresso alla rete i coefficienti di un modello autoregressivo della forza di taglio stimati mediante i minimi quadrati ricorsivi. In alcuni casi per aumentare la capacità di generalizzazione della rete sono stati inseriti anche i parametri di taglio della macchina utensile in [17] e in [18]. Le criticità principali che si riscontrano sono spesso dovute alla difficoltà di generalizzare il metodo, dato che in presenza di una modifica dei parametri del processo la prestazione possono risentirne fortemente. Altro tema fondamentale è la necessità di avere a disposizione un dataset sufficientemente ampio e pulito per effettuare l'addestramento.

Apprendimento	Applicazione	Tipologia di rete	Rif.bibliografico		
Supervisionato	MCM	MLP(BP)	[23], [24], [25], [26].		
	MCM	Fuzzy ARTMAP	[27].		
Non Supervisionato	MCM	SOM	[28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43].		
			MCM	ART-2	[44].
			MCM	Fuzzy-ART	[45].

Tabella 2.2: Tabella Letteratura MCM

### Machine Condition Monitoring

I componenti strutturali delle macchine utensili come le guide e i cuscinetti a sfera sono anch'essi soggetti a degradazione ed usura. La fase di test su base regolare è estremamente importante per ridurre il rischio di guasti improvvisi e catastrofici. La precisione di posizionamento delle macchine utensili dipende dalla funzione critica svolta da componenti quali le guide, le viti a sfera, i cuscinetti e il mandrino. Qualsiasi deterioramento di queste componenti aumenta il rischio di produrre scarti e guasti nel tempo. Tipici indicatori di scarso rendimento sono:

1. L'aumento della temperatura nell'alloggiamento del cuscinetto per l'usura dei cuscinetti dei mandrini.
2. Aumento di potenza richiesta.
3. Asimmetria rotazionale causata dal disallineamento.
4. Aumento significativo dell'ampiezza delle vibrazioni.

Per aumentare la disponibilità all'utilizzo delle macchine utensili occorre un sistema di monitoraggio delle condizioni di macchina capace di fornire in process l'indicazione dell'avvenuto guasto.

Il Machine Condition Monitoring (MCM) viene giudicato in relazione ai sintomi e segni correlati in genere ai parametri di funzionamento. La variazione nel tempo di questi parametri è un indicatore della progressione del guasto e può essere usata per prevederne il progressivo sviluppo oltre a servire come base per la

generazione di allarmi. Tra i vari sintomi le vibrazioni della macchina sono state usate come indicatore pratico di guasto. Molte attrezzature della macchina consistono in cuscinetti, anelli, motori, alberi ed elementi rotanti e le vibrazioni causate dalla presenza di guasti strutturali in queste componenti fornisce una sorgente di informazione delle condizioni di salute della macchina, dato che il profilo della vibrazione cambia con lo sviluppo dei guasti. Questo cambiamento si potrà riflettere in un incremento del livello di vibrazione delle frequenze caratteristiche. Altri metodi utilizzati hanno incluso le emissioni acustiche (AE), segnale associato alle onde elastiche transitorie generate da improvviso rilascio di energia di deformazione. Tale rilascio di energia è sostanzialmente dovuta a concentrazioni di stress, che possono essere causati dalla presenza di difetti strutturali quali crepe. Difficoltà di carattere generale con misurazione basata su AE riguardano la quantificazione di basse magnitudini del segnale di AE e la contaminazione rumore da altre strutture della macchina. I problemi fondamentali nel condition monitoring sono:

1. Identificazione dei pattern di guasto
2. Quantificazione dello sviluppo del guasto

Il dataset misurato, che è rappresentativo del particolare guasto, viene estratto per caratteristiche attraverso tecniche idonee di processamento del segnale. Errori e incertezze nella classificazione dei guasti possono portare a falsi allarmi, che motivano la ricerca di migliori sistemi di condition monitoring più robusti e affidabili. In tale ambito sono numerosi i lavori relativi all'individuazione di condizioni di guasto con l'ausilio di sensori e strumenti data driven per il supporto delle decisioni. Per avere un sistema di rilevazione dei guasti che sia automatico e affidabile occorre identificare i segnali transitori legati alle anomalie in presenza di una forte rumorosità dei dati tempo varianti e ad altre cause di interferenza. Data la non linearità e la non stazionarietà dei segnali in esame le reti neurali ben si prestano alla modellazione di questi comportamenti, in tabella 2.2 vi saranno i lavori principali nell'ambito. La Som rappresenta una delle rete che ha avuto maggiori applicazioni nel campo, spesso in combinazione con tecniche per la riduzione dimensionale dei segnali pre-processati come la PCA ad esempio in [28]. Relativamente alla Som

le modalità con le quali è possibile riconoscere uno stato anomalo derivano nella maggior parte dei casi dal calcolo di misure di distanza dai nodi della mappa e dalla determinazione delle condizioni di fuori controllo mediante delle soglie calcolate su di esse, esempi si hanno in [29], [30], [33], [42] e [32]. Tuttavia questa tipologia di applicazione, senza l'utilizzo di accorgimenti preliminari, può portare a riscontrare un notevole numero di falsi allarmi, cosa non auspicabile in campo manifatturiero dato che tale circostanza costringerebbe ad un numero elevato di fermi macchina e conseguente riduzione della disponibilità. Altri metodi, invece, inseriscono anche dati relativi al guasto. Infatti, dato che le anomalie più frequenti si presentano nelle stesse modalità, è possibile utilizzare anche l'informazione proveniente dalle traiettorie percorse dalle bmu dei dati da testare per dare una valutazione del processo. Un esempio di questa tecnica si ha in [38], tuttavia la medesima rimane strettamente legata alle condizioni addestrate e quindi si presta meno a generalizzazioni del sistema di controllo. Nonostante il monitoraggio di processo attraverso rete neurale sia stato studiato approfonditamente, la fase di analisi retrospettiva mediante la stessa rete è stata fino a questo momento trascurata.

# Capitolo 3

## Self-Organizing Map

La Self-Organizing Map è una rete neurale ad apprendimento non supervisionato che consente di ottenere attraverso l'algoritmo di training sviluppato da Kohonen in [6] un' approssimazione della distribuzione dei dati di input. Rispetto agli approcci supervisionati non richiede dati etichettati, facilitando la sua applicazione in una realtà industriale, inoltre diversi lavori che verranno presentati successivamente dimostrano come consenta di ottenere ottime prestazioni come classificatore uni-categorico. Infatti questa tipologia di rete per sua stessa natura effettua una prima classificazione dei dati di addestramento nei suoi stessi nodi. Attraverso questa sua prerogativa sarà possibile giungere più facilmente alla fase di decontaminazione vera e propria, che è l'obiettivo di questa tesi. Infine la semplicità e gli strumenti di visualizzazione che fornisce a supporto semplificando le fasi di esplorazione a analisi dei dati ne fanno un valido strumento per il trattamento di dati ad elevata dimensionalità. Di seguito verrà introdotta la rete dai suoi fondamenti teorici fino alle sue applicazioni principali.

### 3.1 Motivazioni a supporto

Le reti di Kohonen, denominate anche SOM(Self-Organizing Maps), hanno destato notevole interesse, nonostante la semplicità architetture e operativa, sia perché sono applicabili in molti problemi pratici [4] sia perché sono dotate di due caratteristiche peculiari e rilevanti che sono l'analogia con certe strutture biolo-

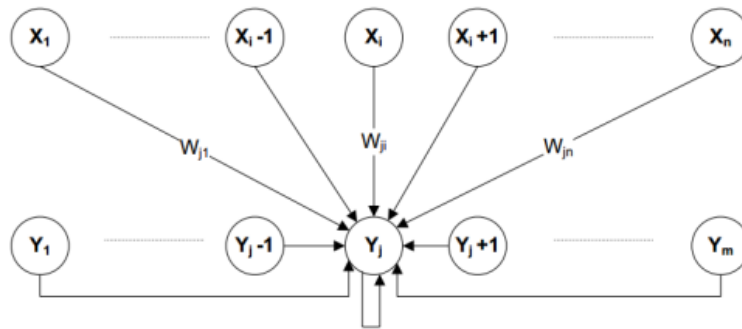


Figura 3.1: Competitive Learning [4]

giche e la capacità di auto-organizzazione. Per quanto riguarda l'analogia neurobiologica rileviamo che sulla corteccia del cervello si vengono a formare, in base all'auto-apprendimento, mappe corticali tali che neuroni vicini sono attivati da stimoli simili. In modo analogo, le reti SOM proiettano un input multidimensionale su una superficie di neuroni artificiali rappresentandolo in due dimensioni ed effettuando, quindi, una notevole compressione pur conservando le similarità originali. Inoltre la capacità di scoprire in modo autonomo proprietà interessanti di un input multidimensionale accomuna le reti SOM alla capacità degli animali di adattarsi all'ambiente senza la necessità di una guida esterna. Per quanto riguarda l'auto-organizzazione le reti SOM sono in grado di apprendere senza la necessità di supervisione e, quindi, senza richiedere un training set di esempi noti. Esse modificano autonomamente i propri pesi sinaptici in funzione degli stimoli esterni. Partendo da una situazione iniziale di non organizzazione esse si auto-organizzano nel senso che, grazie a interazioni semplici e puramente locali, evolvono verso l'emergenza di un ordine globale. Le reti SOM derivano dalle reti ricorrenti a connessioni laterali. Un esempio di queste ultime è presente in figura 3.1, dove si hanno  $M$  neuroni  $j$  (con attivazione  $M_j$ ) connessi a  $n$  input  $X_i$ , tramite pesi sinaptici  $w_{i,j}$ . Inoltre ogni neurone  $j$  è connesso con tutti i neuroni  $k$  del suo strato (incluso se stesso) tramite i pesi sinaptici  $v_{j,k}$ . I neuroni  $k$  vicini al neurone  $j$  lo eccitano ( $v_{j,k} > 0$ ), mentre quelli lontani lo inibiscono ( $v_{j,k} < 0$ ). Il campo di eccitazione di ogni neurone  $j$  viene così ad assumere una forma a "cappello messicano" come rappresentato in figura: 3.2, la dinamica della rete è espressa da:

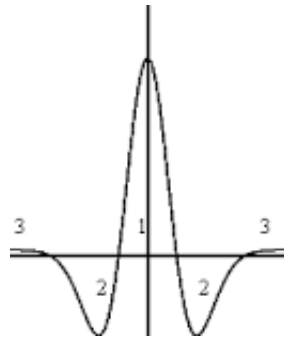


Figura 3.2: Cappello messicano [5]

$$Y_j(t+1) = F\left[\sum_r (w_{j,r} X_r) + \beta \sum_k v_{j,k} M_{j+k}(t)\right] \quad (3.1)$$

dove, se  $V$  è il vicinato,  $(j+k)$  varia tra  $(j-V)$  e  $(j+V)$ ;  $F$  è una funzione di trasferimento non lineare con saturazione (per limitare il valore di  $M_j$ ) e  $\beta$  è un fattore di feed-back, compreso tra zero e uno. Il funzionamento generale della rete, che costituisce un sistema a feed-back positivi e negativi, è il seguente. Presentando un input al tempo  $t=0$ , un vettore  $X$  di  $n$  componenti  $X_i$ , verrà attivato maggiormente un certo neurone  $j$  ( $M_j(0) > M_k(0)$ , per  $k \neq j$ ) in base alla prima sommatoria dell'equazione precedente. Negli istanti di tempo successivi entra in azione la seconda sommatoria e i neuroni in un intorno di  $j$  si rafforzano mutuamente e indeboliscono i neuroni lontani. Si forma così una bolla di attivazione, centrata su  $j$ , con una certa estensione "orizzontale" e una certa ampiezza "verticale":

- l'estensione dipende dal rapporto tra feed-back positivi e negativi; con la prevalenza dei primi si ha l'allargamento della bolla di attivazione, viceversa prevalendo i secondi si ha il restringimento;
- l'ampiezza dipende dal fattore di feed-back  $\beta$  e cresce con esso;

In conclusione:

- ogni input attiverà la sua bolla (dopo un certo transitorio);
- input simili attiveranno bolle vicine e input diversi bolle lontane.

Si può ottenere una notevole semplificazione strutturale, eliminando le connessioni laterali  $v_{j,k}$  a “cappello messicano”, se si definisce un vicinato di neuroni contemporaneamente attivi. In tal caso la bolla si forma per definizione, dato che l’attivazione iniziale di un neurone  $j$  trascina automaticamente quella di tutti i neuroni del suo vicinato. Inoltre tale vicinato può cambiare nel tempo: se si allarga viene simulata la presenza di feed-back positivi, se si restringe quella di feed-back negativi. Estendendo quanto ora detto, da una stringa unidimensionale ad una matrice bidimensionale, otteniamo la rete di Kohonen. Una Som è appunto costituita da una matrice di  $M$  neuroni  $j$  come in figura 3.3, ciascuno collegato ad  $n$  input  $i$  tramite pesi sinaptici  $w_{ji}$ . Non si hanno connessioni laterali, ma le si simula adottando un vicinato topologico. Il vicinato è normalmente un quadrato ma potrà essere esagonale, gaussiano ecc.

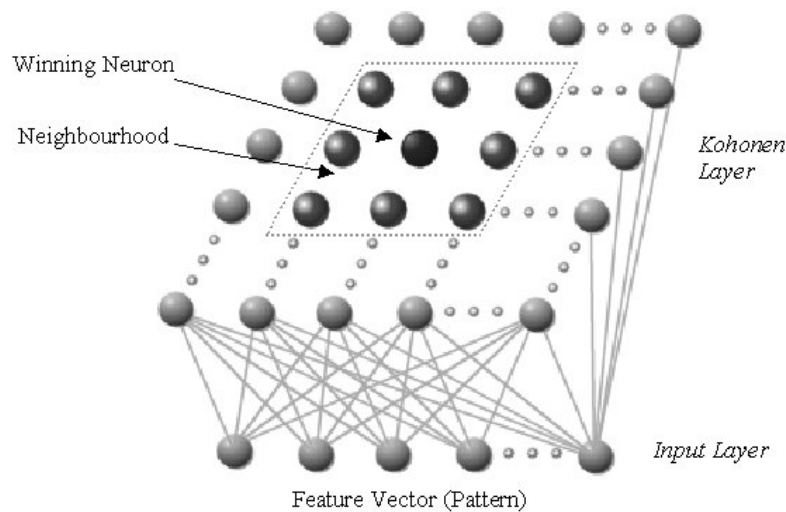


Figura 3.3: Rete di Kohonen [6]

## 3.2 Il paradigma della Som

I dati di training della Self-organizing map [5] vengono eseguiti iterativamente attraverso un algoritmo di addestramento. L’algoritmo individua per ogni dato di training il vettore che meglio lo approssima e agisce su di esso e sul suo vicinato attraverso una neighborhood function al fine di preservare la topologia generale o la struttura dello spazio di input. Il risultato dell’algoritmo è una mappatura non



lineare e ordinata di una raccolta di dati multidimensionali negli elementi di una array regolare a bassa dimensionalità. L'algoritmo si articola nelle seguenti fasi consistenti in:

1. Definire l'insieme delle variabili di input  $x_j$  come vettori  $x = [x_1, x_2, \dots, x_n]^T$   $x \in \mathfrak{R}^n$ .
2. Associare ad ogni elemento nell'array della Som un vettore parametrico reale  $m_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T w \in \mathfrak{R}^n$ .
3. Stabilire una misura di distanza tra  $x$  e  $m_i$  indicata come  $d(x, m_i)$ .

L'immagine di un vettore di input nella mappa viene definita attraverso l'elemento dell'array che è più vicino ad  $x$ , cioè quello per la quale è minima la distanza  $d(x, m_i)$ .

L'obiettivo principale è definire  $m_i$  in modo tale che il risultato dell'addestramento della rete sia una mappatura ordinata e che rappresenti la distribuzione dei dati originali; questo risultato verrà ottenuto attraverso l'algoritmo della SOM. Questo processo è in grado di produrre valori dei modelli ( $m_i$ ) asintoticamente convergenti. L'insieme di questi modelli a sua volta approssimerà la distribuzione dei campioni di input anche in modo ordinato.

### 3.2.1 Tipologie di learning

#### Algoritmo incrementale

L'algoritmo originariamente introdotto da Kohonen può essere visto come un modello di regressione ricorsiva, nella quale solo un sottoinsieme di vettori di riferimento viene processato ad ogni step. Questo determina una mappatura dallo spazio n-dimensionale in una array bidimensionale di nodi, chiamato lattice. Ad ogni nodo  $i$  viene associato un vettore modello, chiamato anche vettore di riferimento  $m_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T w \in \mathfrak{R}^n$ . Prima del processamento ricorsivo il vettore  $m_i$  andrà inizializzato. Possono essere usati valori casuali per le sue componenti inizializzate, ma se saranno scelte con cura la convergenza può essere ottenuta più rapidamente. Le tipologie di reticolo dell'array sono quello rettangolare, esagonale o anche irregolare.

Sia  $x = [x_1, x_2, \dots, x_n]^T$   $x \in \mathfrak{R}^n$  il vettore di input connesso con tutti i neuroni in parallelo attraverso i pesi che sono delle variabili scalari.

La Som effettua una proiezione non lineare della funzione di densità di probabilità dei dati di input in alta dimensionalità in una matrice solitamente bidimensionale. Il risultato dell'organizzazione è che se i pattern di input hanno una funzione di densità di probabilità ben definita allora i vettori di pesi associati ai modelli  $m_i$  dovrebbero cercare di imitarla.

### Selezione best matching unit

Il vettore  $x$  può essere comparato con tutti i vettori di riferimento mediante qualsiasi metrica; la più utilizzata è la distanza euclidea. In tal caso definiamo la best matching unit (bmu) attraverso la più piccola distanza euclidea  $\|x - m_i\|$ , identificata con la seguente quantità  $c$ :

$$c = \arg \min_i \|x - m_i\| \quad (3.2)$$

### Aggiornamento dei pesi

Durante l'apprendimento quei nodi che sono topograficamente vicini all'interno dell'array, fino ad una certa distanza geometrica dal nodo vincitore, si attiveranno l'un l'altro per imparare qualcosa dall'input  $x$ . Questo comporterà nel vicinato un effetto di smoothing nei vettori di pesi, che continuando con l'apprendimento conduce ad un ordine globale. Si riuscirà a raggiungere questo risultato attraverso la definizione di un vicinato  $N_c$  attorno al nodo ad ogni passo dell'algoritmo, aggiornando solo quei nodi che sono all'interno  $N_c$  e tenendo il resto di loro intatti. La larghezza o il raggio definisce i nodi che sono all'interno di  $N_c$ , tale valore può essere tempo variante. I risultati migliori in termini di ordinamento globale sono stati ottenuti quando  $N_c$  è molto largo all'inizio della fase di learning e cresce monotonamente con il tempo. Questo perché all'inizio viene effettuato un ordinamento grezzo tra i valori di  $m_i$  e, successivamente, restringendo  $N_c$  la risoluzione spaziale della mappa risulta migliorata. Viene usato il seguente processo di apprendimento:

$$m_i(t+1) = \begin{cases} m_i + \alpha(t) [x(t) - m_i] & \text{se } i \in N_c(t) \\ m_i & \text{se } i \notin N_c(t) \end{cases} \quad (3.3)$$

dove  $t = 0, 1, \dots, n$  è la coordinata discreta del tempo e  $\alpha(t)$  è il valore scalare del learning rate che decresce nel tempo con  $0 < \alpha(t) < 1$ .

Tuttavia, invece che usare un semplice insieme di vicinato nel quale ciascun nodo dell'insieme è influenzato allo stesso modo, è possibile introdurre come funzione di vicinato sui nodi del lattice una funzione scalare "kernel"  $h_{ci} = h_{ci}(t)$  che aggiorni i pesi nel modo seguente:

$$m_i(t+1) = m_i + h_{ci}(t) [x(t) - m_i(t)] \quad (3.4)$$

Per la convergenza è necessario che  $h_{ci} \rightarrow 0$  quando  $t \rightarrow \infty$ . Solitamente  $h_{ci}(t)$  è una funzione della distanza tra i nodi  $c$  e  $i$  nell'array, i.e.  $h_{ci} = h(\|r_c - r_i\|, t)$ , dove  $r_c$  e  $r_i$  denotano rispettivamente le coordinate del nodo  $c$  e  $i$  nell'array. Con l'incremento di  $\|r_c - r_i\|$  si ha che  $h_{ci} \rightarrow 0$ . Due scelte  $h_{ci}$  sono più frequenti, la più semplice delle due è quella con il vicinato impostato come funzione del tempo  $N_c = N_c(t)$ , e per cui  $h_{ci} = \alpha(t)$  se  $i \in N_c$  e  $h_{ci} = 0$  se  $i \notin N_c(t)$ . Sia  $\alpha(t)$  che il raggio di  $N_c$  generalmente decrescono monotonicamente con il tempo. L'altra scelta è una funzione di vicinato kernel, nota come funzione gaussiana:

$$h_{ci} = \alpha(t) e^{-\frac{\|r_c - r_i\|^2}{2\sigma^2(t)}} \quad (3.5)$$

dove  $\alpha(t)$  è un valore scalare del learning rate e il parametro  $\sigma$  definisce la larghezza del kernel, quest'ultimo corrisponde al raggio di  $N_c$ . Sia  $\alpha(t)$  che  $\sigma$  sono funzioni monotonicamente decrescenti con il tempo.

In sintesi l'algoritmo incrementale può essere ricondotto a due fasi consistenti in:

1. Localizzare la bmu
2. Incrementare la somiglianza in questo nodo e nei suoi vicini topologici.

Questi due step sono usualmente svolti come segue:

1. Inizializzazione della rete: Per ciascun nodo  $i$  inizializzare il corrispondente vettore di riferimento  $m_i$  casualmente. Definire il valore iniziale del learning rate vicino all'unità e il raggio  $s$  pari almeno alla metà del diametro del lattice.
2. Presentazione input: Presentare il vettore di input  $x(t)$  al tempo  $t$  a tutti i nodi del lattice contemporaneamente.
3. Determinazione similarità: Trovare la bmu  $c$  al tempo  $t$  minimizzando la distanza euclidea:

$$c = \arg \min_i \|x - m_i\| \quad i = 1, \dots, N \quad (3.6)$$

dove  $N$  è il numero di neuroni nel lattice.

4. Aggiornamento: Vengono modificati i vettori di riferimento  $m_c$  e tutti quelli all'interno del vicinato del nodo vincente usando la legge seguente:

$$m_i(t+1) = m_i + h_{ci}(t) [x(t) - m_i(t)] \quad \text{con} \quad h_{ci} = a(t) e^{-\frac{\|r_c - r_i\|^2}{2s^2(t)}} \quad (3.7)$$

5. Ripetizione: Far decrescere il learning rate e la larghezza del vicinato. Ripetere il passo 2 scegliendo un nuovo vettore di input  $x(t+1)$  finché un criterio predefinito di stop non venga raggiunto.

### Algoritmo batch

Un secondo tipo di algoritmo della Som è quello batch [46], anche esso è un algoritmo iterativo, ma invece di usare un singolo vettore al tempo  $t$ , l'intero dataset viene presentato alla mappa prima che avvenga la modifica dei pesi, da questo deriva il nome batch. I passi dell'algoritmo vengono riportati di seguito:

1. Selezione della bmu: considerare un insieme di campioni di input  $X$  e un lattice bi-dimensionale dove ad ogni nodo  $i$  è associato un vettore di riferimento  $m_i$ . I valori iniziali di  $m_i$  possono essere selezionati in modo casuale dal dominio dei campioni di input. Attraverso la misura di distanza  $d(x, m_i)$ , il nodo più vicino all'input  $x \in X$  viene individuato e copiato all'interno della sotto-lista associata al nodo.

2. Adattamento (aggiornamento dei vettori di riferimento): una volta che tutti i vettori di  $x \in X$  sono stati distribuiti tra le sotto-liste dei nodi i vettori di riferimento possono essere aggiornati. Questo viene fatto trovando il campione centrale  $\bar{x}_i$  o  $\bar{x}'_i$  per ciascun nodo  $i$  per sostituire il vecchio valore di  $m_i$  con  $\bar{x}_i$  o  $\bar{x}'_i$ , considerando inizialmente l'insieme di vicinato  $N_i$  attorno al vettore di riferimento  $m_i$ . Il vicinato consiste in tutti i nodi all'interno di un certo raggio dal nodo  $i$ . Nell'unione di tutte le sotto-liste in  $N_i$ , si trova il campione  $\bar{x}_i$  che abbia la più piccola somma delle distanze da tutti i campioni  $x(t)$  dove  $t \in N_i$ . Il vettore centrale può essere di due tipologie:
  1. Insieme mediano generalizzato: se il campione  $\bar{x}_i$  è parte dei campioni di input  $x \in X$ .
  2. Mediana generalizzata: Se il campione  $\bar{x}_i$  non è parte dei campioni di input  $x \in X$ . Dato che stiamo usando solo un campione di ingressi  $x \in X$  è possibile trovare un valore di  $\bar{x}_i$  che ha anche la più piccola somma delle distanze da ogni altro  $x \in N_i$  attorno a ciascun nodo  $i$ .

Il campione centrale  $\bar{x}_i$  o  $\bar{x}'_i$  determinato viene usato per sostituire il modello precedente  $m_i$ ; questo viene fatto per ogni nodo  $i$  considerando sempre l'insieme di vicinato  $N_i$  attorno ad ogni nodo  $i$  e sostituendo ciascun vecchio valore di  $m_i$  con  $\bar{x}_i$  o  $\bar{x}'_i$  in operazioni simultanee. La procedura vista sopra dovrebbe essere iterata i.e. tutti i campioni  $x \in X$  vengono nuovamente distribuiti nelle sotto-liste (che saranno adesso molto probabilmente cambiate dato che ciascun  $m_i$  è stato aggiornato nell'iterazione precedente) e nuove mediane vengono calcolate.

Formalmente il processo sopra riportato quando forma gli insiemi di vicinato usa un concetto noto come Voronoi tassellation. Questo è un modo di partizionare un insieme di vettori nelle regioni, confinati dalle linee, così che ogni partizione contenga un vettore di riferimento  $m_i$  che è il vicino più prossimo ad ogni vettore  $x$  all'interno della stessa partizione. Queste linee insieme costituiscono la Voronoi tassellation. Tutti i vettori che hanno un particolare vettore di riferimento come vicino più prossimo costituiscono un insieme di Voronoi  $V$ . Cioè il vettore di riferimento  $m_i$  e la ricerca della bmu definisce una tassellazione dello spazio di input

in un gruppo di insiemi di Voronoi definito come:

$$V_i = \{x : \|x - m_i\| < \|x - m_n\| \quad \forall i \neq n\} \text{ dove } x \in X \quad (3.8)$$

In ogni step di training, le unità della mappa associate con ogni insieme di Voronoi i.e. la sotto-lista di ogni vettore di riferimento  $m_i$  contenente i vettori di dati  $x$  che hanno essa come bmu, costituiscono un insieme di Voronoi. Se inoltre viene usata una funzione di vicinato generale  $h_{ci}$ , allora la formula di aggiornamento dei vettori di riferimento diventa:

$$m_i = \frac{\sum_{x \in X} h_{ci}(t)x}{\sum_{x \in X} h_{ci}} \quad (3.9)$$

dove  $c$  è la bmu per il campione  $x$  corrente.

Questo algoritmo è particolarmente efficace se i valori iniziali dei vettori di riferimento sono ordinati in modo approssimativo. Questo algoritmo non contiene un parametro di apprendimento e pertanto esso sembra raggiungere un risultato asintotico più stabile dei valori di pesi  $m_i$  se comparati con quelli raggiunti dall'algoritmo della Som originale.

### 3.2.2 Parametri decisionali

Il processo di apprendimento coinvolto nella creazione della feature map è di natura stocastica, ciò vorrà dire che l'accuratezza della mappa dipenderà dal numero di iterazione dell'algoritmo. All'interno dell'algoritmo vi saranno una serie di parametri da definire, i cui valori differenti condurranno a SOM diverse. I principali parametri sono:

- La dimensione della SOM: essa determina il grado di generalizzazione che viene prodotto dall'algoritmo, ad un numero maggiore corrisponderà una rappresentazione più dettagliata dello spazio di input del training set.
- Il numero di iterazioni: devono essere in numero sufficiente, dato che l'apprendimento è di natura stocastica.
- Il parametro di apprendimento: esso solitamente nella prima fase di training decresce a partira dal valore 1 fino a valori prossimi allo zero, la legge che

regola tale diminuzione, non sarà un aspetto critico e potrà essere lineare, esponenziale o inversamente proporzionale a  $t$ . Nella seconda fase di training dovrà a partire da valori molto bassi, generalmente 0.01, convergere a zero.

- La funzione di vicinato: La funzione di vicinato potrà essere scelta come l'insieme individuato semplicemente da  $h_{ci}(t)$  se la dimensione del lattice non è molto grande e.g. poche centinaia di nodi. Per dimensioni più grandi si potrà utilizzare la funzione gaussiana. Durante i primi step, quando si effettua la fase di ordinamento, il raggio di  $N_c$  può diminuire linearmente di una unità, durante la fase di convergenza  $N_c$  potrà ancora contenere i vicini più prossimi del nodo  $c$ .
- La dimensione del vicinato: deve essere scelto all'inizio abbastanza largo così che la mappa possa venire ordinata globalmente. Se lo stesso all'inizio è troppo piccolo si riscontrano varie tipologie di parcellizzazioni della mappa, tra le quali le direzioni di ordinamento cambiano in maniera discontinua. Questo fenomeno può essere evitato partendo con un vicinato molto largo per poi lasciarlo decrescere con il tempo. Il raggio iniziale può anche essere superiore della metà del diametro della rete.

### Esempio di addestramento: caso bidimensionale

Al fine di comprendere meglio il funzionamento della Self-Organizing Map, risulta utile vederla applicata ad un caso semplice bidimensionale. Consideriamo i dati provenienti da una distribuzione normale bivariata:

$$X \sim N(\mu, \Sigma) \quad (3.10)$$

$$\mu = [0, 0] \quad (3.11)$$

$$\Sigma = \begin{bmatrix} 1 & 0.6 \\ 0.6 & 1 \end{bmatrix} \quad (3.12)$$

I parametri scelti per l'addestramento del dataset sono tabellati in 3.1. In corrispondenza di un numero elevato di nodi, si riesce a dettagliare meglio lo spazio dei dati di training con una maggiore livello di dettaglio. I dati generati, insieme ai

Lunghezza training fase di ordinamento	50
Raggio iniziale e finale fase di ordinamento	10-1
Lunghezza training fase di convergenza	20
Raggio iniziale e finale fase di convergenza	1-1
Numerosità training set	500
Numero Nodi	150
Funzione di vicinato	Gaussiana
Lattice	Esagonale

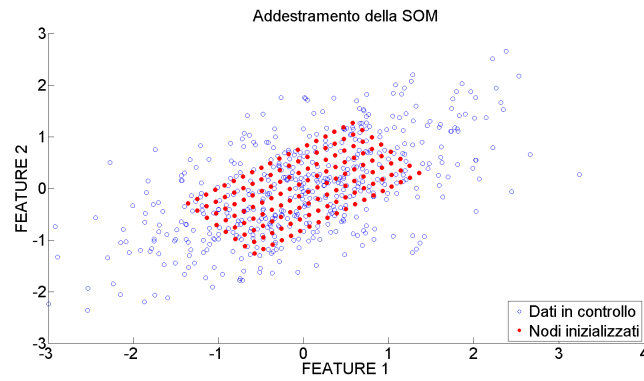
Tabella 3.1: Tabella parametri di simulazione scelti Som 2D

nodi sono rappresentati nella figura 3.4. È possibile vedere come i nodi, a seguito del processo di apprendimento tendano a organizzarsi in modo da seguire e fittare i dati di addestramento. In questo esempio è stato utilizzato l'algoritmo batch della rete, con inizializzazione lineare dei pesi. L'inizializzazione lineare consisterà nel determinare le due direzioni di massima variabilità dei dati di training lungo le quali disporre i nodi della rete.

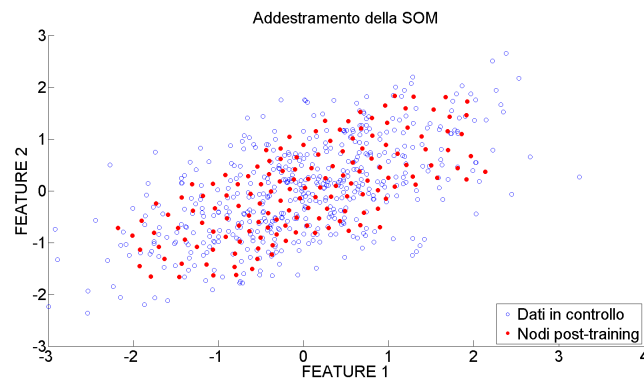
### 3.3 Misure di qualità della Som

Sebbene esistono sempre delle mappe ottime per i dati di input, scegliere i giusti parametri dall'inizio è un compito arduo. Dato che differenti parametri e la loro inizializzazione dà luogo a risultati differenti, è importante pertanto sapere se la mappa ha effettivamente e correttamente adattato se stessa ai dati di training. Questo viene fatto solitamente attraverso una funzione di costo che definisca esplicitamente la soluzione ottima ma dato che è stato mostrato come l'algoritmo della Som non segua il gradiente di nessuna funzione, in generale occorrerà affidarsi a delle misure di qualità [47]. Le due più comunemente usate che consentono di determinare la qualità della mappa e aiutare nella scelta dei parametri sono l'errore di quantizzazione e quello topografico.





(a) Nodi inizializzati



(b) Nodi dopo addestramento

Figura 3.4: Applicazione Som:esempio bidimensionale

### 3.3.1 Errore di quantizzazione

L'errore di quantizzazione medio è una misura di quanto bene la mappa segue i dati di input, la mappa migliore ci si attende che ottenga il più piccolo valore di distanza tra le bmu e i vettori di input. La media di  $\|x - m_i\|$ , ottenuta fornendo in ingresso alla rete addestrata tutti i dati di input viene usata per il calcolo dell'errore con la seguente formula:

$$E_q = \frac{1}{N} \sum_{i=1}^N \|x - m_i\| \quad (3.13)$$

dove  $N$  è il numero di vettori di input usati per l'addestramento. Una Som con un valore più basso di errore di quantizzazione è più accurata nel rappresentare i dati in ingresso di una con un valore più alto.

### 3.3.2 Errore topografico

L'errore topografico misura quanto fedelmente la mappa preserva la topologia dei dati originali. A differenza dell'errore di quantizzazione medio, esso considera la struttura della mappa. Per ogni vettore di input viene calcolata la distanza tra esso e le prime due bmu, se questi due nodi non sono a loro volta vicini nel lattice, allora la topologia non è stata preservata. L'errore viene calcolato con il metodo seguente:

$$E_t = \frac{1}{N} \sum_{k=1} N u(x_k) \quad (3.14)$$

dove  $N$  è il numero di vettori di input del training set e  $u(x_k)$  è una funzione che assume valore pari a uno se la prima e la seconda BMU sono nodi vicini e zero altrimenti.

## 3.4 La Som come classificatore unicategorico

La Som è stata spesso applicata nel campo del "Process Monitoring". L'utilizzo che solitamente ne viene fatto prevede, data la difficoltà di rappresentare tutte le possibili condizioni di guasto, di addestrare la rete esclusivamente con dati provenienti dallo stato in controllo del sistema; questa metodologia d'uso prende il nome di "novelty detection" o "one classifier" e l'obiettivo principale della stessa è quello di determinare le anomalie come deviazioni dalle situazioni note. Questo tipo di approccio è stato utilizzato anche in lavori in cui si fa uso di un'altra tipologia di rete non supervisionata, in particolare in [48] e [49], nella quale tramite la Fuzzy-ART, rete che trae il suo fondamento nella teoria "Adaptive Resonance Theory", viene realizzato uno strumento di monitoraggio in grado di segnalare deviazioni dalla condizione stabile. Un altro contributo è stato fornito da Tax e Duin in [43] che hanno tentato di risolvere il problema della novelty detection utilizzando la Support vector data description. Per quanto riguarda la Som, utilizzata come classificatore unicategorico rispetto alla condizione operativa normale, uno dei primi contributi si ha in [33]. La rete dopo la fase di training fornisce una mappa bidimensionale dei neuroni ciascuno dei quali è caratterizzato da un vettore di pesi di riferimento dei dati di input. Questi vettori sono ottimizzati al fine di rappresen-

tare con accuratezza la densità e la posizione dei dati di input. In fase di test la distanza tra l'ingresso e i pesi può essere impiegata come misura di comportamento anomalo. Lavori più recenti nell'ambito sono [39] e [40], nei quali si addestra la rete con dati provenienti da condizioni normali e successivamente si costruisce una carta di controllo basata sui valori dell'errore di quantizzazione minimo, ovvero la distanza tra il dato e il nodo più vicino. Anziché usare il vettore di dati grezzi sono state calcolate su essi una serie di feature statistiche come media, deviazione standard, asimmetria e autocorrelazione utilizzate come input alla rete. Per calcolare le feature è stata applicata una finestra mobile sul flusso dei dati grezzi; un parametro fondamentale è dunque l'ampiezza di tale finestra che influenza  $\alpha$  e  $\beta$ . Infatti, finestre più grandi evitano falsi allarmi ma richiedono maggiore tempo per determinare il fuori controllo mentre finestre piccole individuano rapidamente i fuori controllo ma hanno alti tassi di falso allarme. L'ipotesi alla base di questi lavori è quella di avere a disposizione dei dataset ampi di dati in condizioni normali, nel caso in cui siano presenti dati corrotti in fase di addestramento le prestazioni di questi diverrebbero molto peggiori. Nell'eventualità in cui la SOM avesse un dataset contaminato la stessa andrebbe ad apprendere, attraverso i nodi del reticolo bidimensionale, anche le condizioni fuori controllo; in questo caso la rete non sarebbe in grado di riconoscere come deviazioni tali anomalie e, pertanto, non segnalerebbe nessun allarme. In questo contesto si rende necessario il presente lavoro di tesi con l'obiettivo di determinare un metodo che sia in grado, attraverso la rete stessa, di effettuare un'analisi retrospettiva dei dati di training per rendere efficace lo strumento di monitoraggio.

# Capitolo 4

## Analisi retrospettiva di fase I

Nella realtà industriale è diventato sempre più semplice ed economico installare sensori e acquisire, così, notevoli moli di dati. Questo, ovviamente, ha portato ad avere a disposizione molta informazione di cui però non si conosce la provenienza. Utilizzare questi dati per addestrare una rete neurale non supervisionata può essere molto pericoloso. Infatti i dati che verranno forniti in ingresso, per un utilizzo della rete come classificatore uni-categorico, saranno considerati come dati provenienti da condizioni normali del processo. Gli outlier possono disturbare la struttura di correlazione del modello e avere un impatto negativo sulla loro accuratezza e affidabilità. Fino ad oggi è stata posta un'attenzione inadeguata alle potenzialità della Self-Organizing Map come metodo per di clustering, probabilmente a causa della carenza di procedure utili per generare raggruppamenti significativi dell'output della Som [50]. Infatti quando non si conosce a priori il numero di cluster presenti nei dati o quando il numero di nodi è differente dal valore di raggruppamenti atteso, si rendono necessari ulteriori step di analisi per giungere al risultato di interesse. Solitamente i raggruppamenti in questa fase vengono determinati manualmente attraverso gli strumenti di visualizzazione che mette a disposizione la rete stessa come l'illustrazione della U-matrix che sostanzialmente mostra la distanza tra i nodi del reticolo e quella rispetto ai vicini [51]. Tuttavia quando la popolazione è densamente popolata sarà difficoltoso riconoscere i diversi raggruppamenti. Il clustering dei nodi della rete è un passo fondamentale per riconoscere i gruppi presenti all'interno dei dati. Dai lavori seguenti è stato preso spunto per

realizzare la metodologia obiettivo della tesi.

## 4.1 Metodi di decontaminazione basati su Self Organizing Map

L'analisi retrospettiva dei dati di addestramento potrà essere svolta effettuando una clusterizzazione di secondo stadio sui nodi della rete. Per tale ragione si è preso in considerazione due approcci presenti in letteratura che verranno illustrati di seguito per selezionare i nodi di interesse della rete.

### 4.1.1 Approccio di Vesanto

Vesanto e Alhoniemi [52] hanno proposto un modo per clusterizzare l'output della Som. Il processo inizia con un'ispezione visiva del risultato della Som; successivamente vengono applicati sia una tecnica di clustering partitivo che agglomerativo senza utilizzare le relazioni di vicinato presenti nel reticolo della rete. Sono stati proposti clustering single, average e complete linkage per quello gerarchico agglomerativo e l'algoritmo K-means per quello partitivo. Altri approcci sono quelli di Murtagh [53] che ha in alternativa proposto un metodo di clustering gerarchico basato su vincoli di continuità nel calcolo della distanza minima fra raggruppamenti.

La Som è un eccellente strumento di data mining nella fase esplorativa dei dati; essa, infatti, li proietta dallo spazio di input nei pesi dei nodi di una griglia regolare a bassa dimensionalità che può essere impiegata in maniera efficace per visualizzarli e esplorarne le proprietà. Quando il numero dei neuroni è grande per facilitare l'analisi quantitativa della mappa e dei dati bisogna raggruppare le unità simili. Pertanto in [52] viene utilizzato un approccio a due stadi: in un primo momento la SOM viene addestrata per produrre i vettori di pesi e in un secondo momento questi vengono clusterizzati. Questa tecnica ha dato risultati comparabili con altre tecniche di clustering e, inoltre, riduce il tempo computazionale necessario non lavorando sul dataset originale. La Som crea un insieme di vettori di riferimento che rappresentano il dataset e realizza una proiezione degli stessi dallo spazio

d-dimensionale di input in una griglia a bassa dimensionalità, generalmente a due dimensioni, in grado di preservarne la topologia, ovvero le distanze reciproche tra i vettori. Questa griglia può essere utilizzata come superficie di visualizzazione per mostrare differenti caratteristiche della Som e quindi dei dati. Tuttavia per produrre una descrizione quantitativa delle proprietà dei dati occorre selezionare dalla SOM i gruppi interessanti di neuroni. Il raggruppamento più ovvio è l'intera mappa, ma nonostante le sue proprietà siano interessanti potrebbe essere ancora più utile indagare separatamente le aree a maggiore concentrazione di dati. Un'altra opzione sarebbe quella di considerare separatamente le singole unità, ma in caso di mappa molto grande questo può comportare di effettuare troppe sintesi. Pertanto per riuscire ad utilizzare efficacemente l'informazione fornita dalla Som sono necessari dei metodi per raggruppare le unità di mappa. Una definizione largamente usata di cluster ottimo prevede di creare quei raggruppamenti che minimizzano le distanze interne e massimizzano quelle tra cluster. Comunque questa lascia molto spazio a differenti interpretazioni; infatti le distanze interne e esterne possono essere definite in modo differente così come il tipo di distanza e il criterio scelto. Molti algoritmi impiegano criteri locali (tipo quelli basati su nearest neighbor) nel clusterizzare i dati ma questi sono fortemente sensibili agli outlier e al rumore. Una soluzione al problema potrebbe essere quella di considerare più dati che siano vicini o fare una somma pesata delle distanze; è stato dimostrato come la SOM implicitamente usi una misura del genere. Il clustering può essere suddiviso in due macro-classi gerarchico e partitivo. Quello gerarchico a sua volta si divide in agglomerativo e divisivo a seconda che parta dall'assegnazione di un cluster a ciascun dato nel primo caso o che parta da un cluster contenente tutti i dati nel secondo. Il dendrogramma è lo strumento che ci consente di visualizzare i cluster che si vengono a formare, esso non ci fornirà un unico clustering ma si potrà scegliere di tagliarlo ad un livello ritenuto idoneo e opportuno. Il clustering partitivo prevede di definire a monte il numero di cluster, successivamente saranno inizializzati i centroidi di ognuno di essi e dopo calcolate le distanze dei dati dai centroidi e questi stessi verranno aggiornati; se la suddivisione non cambia l'algoritmo si arresta. Se il numero di cluster non è noto a priori l'algoritmo può essere

ripetuto per diversi valori di  $N$  per poi adottare un criterio per stabilire quale configurazione sia la migliore. L'algoritmo più noto di questa tipologia è il K-means. I metodi partitivi hanno il vantaggio di non dipendere da cluster determinati in precedenza ma hanno il difetto di fare delle ipotesi preliminari circa la forma dei raggruppamenti ad esempio il K-Means per la sua natura di creare le regioni attorno al centroide individuerà zone sferiche. Esistono diverse indici per la valutazione della bontà dei raggruppamenti ottenuti. Vesanto utilizza quello di Davies-Bouldin che impiega la distanza dal centroide come distanza interna al cluster e quella tra centroidi come distanza tra cluster. La formula usata sarà la seguente:

$$\frac{1}{C} \sum_{k=1}^C \max_{\max l \neq k} \frac{S_c(Q_k) + S_l(Q_k)}{d_{ce}(Q_k, Q_l)} \quad (4.1)$$

dove  $S_c(Q_k)$  e  $S_l(Q_k)$  sono le distanze interne ai cluster e  $d_{ce}(Q_k, Q_l)$  è la distanza tra cluster. Oltre al vantaggio di natura computazionale riscontrabile in particolar modo nell'approccio gerarchico il metodo consente di ridurre il rumore presente nei dati. Infatti i vettori di riferimento sono medie locali dei dati e, pertanto, risultano meno variabili dei dati originali. Il numero delle unità di mappa determina l'accuratezza e la capacità di generalizzazione della SOM. Durante il training la rete forma una rete elastica che ripiega sulla nuvola dei dati conservandone la topologia. L'algoritmo della Som può essere applicato ad ampi dataset dato che la complessità computazionale scala linearmente con il numero di campioni. D'altro canto la complessità computazionale cresce in maniera quadratica al crescere delle unità di mappa. Ovviamente l'approccio a due stadi può funzionare solo nel momento in cui i prototipi riflettano le proprietà dei dati. Un'idea iniziale di quelli che possono essere i cluster nella SOM si può ottenere visualizzando il numero di input che appartengono ad una unità di mappa o attraverso la visualizzazione della U-matrix. La rete determinerà dei nodi interpolanti, a cui non verranno assegnati dati, i quali consentiranno di separare i diversi gruppi dato che verranno visualizzati all'interno della U-matrix come delle valli, tra i diversi cluster presentati come rilievi. Infine le unioni interessanti tra gruppi possono essere definite in base agli indici di Davies-Bouldin [54] e Gap statistico tra cluster [55].

### 4.1.2 Approccio di Xuefeng

In [56] viene presentato un metodo per la visualizzazione e l'individuazione di outlier in complessi dati multi-dimensionali nel campo chimico. Il metodo impiegato integra la Self-organizing map con una adaptive non linear map per la determinazione di outlier multivariati. Gli outlier possono disturbare la struttura di correlazione del modello e avere un impatto negativo sulla sua accuratezza e affidabilità. Per ridurre l'effetto dovuto alla presenza di outlier multipli nella stima della media e della deviazione standard, i dati vengono pre-processati attraverso il robust scaling; successivamente la rete viene addestrata con i dati processati per ottenere un'approssimazione dei dati multidimensionali, poi attraverso la ANLM (adaptive non linear map) viene effettuata una proiezione bidimensionale del piano della SOM nella quale è possibile visualizzare facilmente la distribuzione dei dati. Infine viene proposta una regola detta del quasi 3-delta derivante dal lavoro di [57] per distinguere tra dati normali e outlier. La procedura richiede di seguire i seguenti passi:

1. Si addestra la Som con il dataset originale, ottenendo in output i nodi all'interno del reticolo bidimensionale. Ad ogni nodo sarà associato un vettore di pesi.
2. Supponiamo sia  $w_i$  con  $i = 1 \dots, RxC$  l' $i$ -esimo peso del nodo  $i$  della SOM, il vettore mediano di tutti i pesi, sarà il seguente:

$$w_{j,median} = median(w_{j,1}, \dots, w_{j,RxC}), j = 1, 2, \dots, l \quad (4.2)$$

3. Si valuta la distanza tra ogni vettore di pesi con il nodo determinato al passo precedente attraverso la formula:

$$d_i = \left[ \sum_{k=1}^l (w_{k,i} - w_{k,median})^2 \right]^{\frac{1}{2}}, (i = 1, 2, \dots, RxC) \quad (4.3)$$

4. Si determina il valore mediano e la stima della deviazione standard delle distanze determinate:



$$\begin{cases} d_{median} = median(d_1, \dots, d_{RxC}) \\ d_{MAD} = 1.4826 median(|d_1 - d_{median}|, \dots, |d_{RxC} - d_{median}|) \end{cases} \quad (4.4)$$

5. I nodi verranno classificati come outlier o meno in base alla relazione seguente:

$$\begin{cases} outlier\ unit, & se\ R_i = \left| \frac{d_i - d_{median}}{d_{MAD}} \right| > 3 \\ normal\ unit, & se\ R_i = \left| \frac{d_i - d_{median}}{d_{MAD}} \right| < 3 \end{cases}, (i = 1, 2, \dots, RxC). \quad (4.5)$$

6. I dati che avranno come bmu i nodi classificati precedentemente come outlier saranno i dati fuori controllo da eliminare dal dataset originale.

### 4.1.3 Approccio proposto

Il metodo sviluppato prende spunto dai lavori precedentemente presentati per sfruttarne i punti di forza e superare le debolezze di ciascuno di essi. Infatti il metodo di Vesanto non potrà essere utilizzato direttamente per effettuare la decontaminazione, dato che affidarsi unicamente all' algoritmo di clustering fissando a priori un valore di K delle classi nel caso in cui non siano presenti dei raggruppamenti, porta ad aumentare i falsi positivi in maniera inaccettabile. L' algoritmo sarà di conseguenza forzato a suddividere i dati nelle classi prestabilite con il risultato di ridurre il dataset ed eliminare condizioni che sono presenti nella condizione normale, sottostimando la variabilità presente. Il metodo di Xuefeng, invece, ha la forte debolezza di non consentire il controllo dei valori di falsi positivi, discostandosi da considerazioni distribuzionali a causa della presenza degli outlier e, pertanto, non potrà essere considerato un metodo rigoroso di SPC. Sulla base di questi presupposti è stato introdotto una nuova metodologia che fa uso dell' algoritmo K-Means e della stima Kernel che verranno brevemente illustrati di seguito.

### Algoritmo K-means

Il primo metodo applicato è l'algoritmo K-means [58] che è un algoritmo di clustering partitivo ed iterativo. L'algoritmo inizia selezionando a caso un certo numero  $K$  di centroidi. L'oggetto dell'analisi è un insieme di oggetti  $x_1, \dots, x_I$  ciascuno descritto da  $F$  feature. Questo significa che:

$$x_i = (x_i^1, \dots, x_i^F), \forall i = 1, \dots, I \quad (4.6)$$

K-Means partiziona l'insieme  $X = \{x_i, i = 1, \dots, I\} \subset R^F$  nei cluster  $C_1, \dots, C_K$ , identificando rispettivamente i centri  $c_1, \dots, c_K$ . Ciascun centro rappresenta "l'oggetto medio" del cluster corrispondente, i.e. un oggetto fittizio le cui feature sono la media di tutti gli oggetti appartenenti al cluster. Un indice che esprime la dispersione all'interno degli oggetti di un cluster è la somma dei quadrati interna i.e. la somma dei quadrati delle distanze tra gli oggetti che appartengono al cluster e i suoi centri. L'obiettivo consiste nello scegliere la partizione che minimizza la somma interna al cluster in tutti i cluster. Sia  $P = (C_1, \dots, C_k)$  la partizione di  $X$ . Allora, la scelta dei cluster ha come funzione obiettivo:

$$P = \arg \min_P \sum_{k=1}^K \sum_{i \in C_k} \|x_i - c_k\|^2 \quad (4.7)$$

dove  $\|\cdot\|$  è la distanza euclidea.

$$\|x\| = \left( \sum_{f=1, \dots, F} (x^f)^2 \right)^{\frac{1}{2}} \quad (4.8)$$

K-Means è un algoritmo iterativo dato che consiste nel ripetuto aggiornamento dei centroidi. Prima di iniziare il processo è necessario scegliere i semi, i.e. i valori iniziali dei centri. Quindi, il K-Means itera gli step consistenti nell'aggiornare la scelta dei cluster assegnando ciascuno degli oggetti al centroide più vicino e successivamente ricalcolare i centroidi di ciascun cluster. Di seguito vengono dettagliati i passi.

- **Step di assegnamento:** assegna ogni oggetto  $x_i$  al cluster  $C_k$  se il centro  $c_k$  è più vicino.

$$k = \arg \min \|x_i - c_k\|^2 \quad (4.9)$$

- **Aggiornamento dei centri:** per ogni cluster, calcola il valore dei suoi centri come baricentro.

$$c_k = \frac{1}{C_k} \sum_{i \in C_k} x_i, \forall k = 1, \dots, K \quad (4.10)$$

Per determinare quando stoppare l'aggiornamento, la scelta migliore è ripetere questi step finché l'algoritmo non raggiunga la convergenza, i.e. finché i centroidi non subiscono modifiche. Se la convergenza non viene raggiunta in un tempo ragionevole, è possibile stoppare l'aggiornamento. Per fare ciò, l'opzione più comune è fissare il numero massimo di step prima della partenza del K-Means, sebbene sfortunatamente questa modalità non assicuri la convergenza.

### K-Means prestazioni

K-Means è il principale algoritmo di clustering basato sui centroidi, ma non è sempre la scelta migliore. Prima di tutto, non è un algoritmo stabile; infatti differenti scelte dei semi portano a risultati differenti. A causa di questo fatto è essenziale un metodo di inizializzazione appropriato. Al contrario, in assenza di un buon criterio, si può far girare l'algoritmo più volte usando differenti inizializzazioni. Una buona comprensione del problema può venire dalla comparazione di diversi risultati. Un altro limite dell'algoritmo è che l'assegnazione ai cluster si basa esclusivamente sulla distanza tra gli oggetti e i centri dei cluster. Dato che molti degli oggetti sono vicini al centro, la forma dei gruppi è di tipo sferica. In questo modo l'algoritmo non può identificare cluster di forma particolare; inoltre occorre che i dati siano dotati di una distanza metrica (l'algoritmo non può evidentemente funzionare per dati categorici). Inoltre, il parametro K va imposto a priori, ma determinare il numero di classi corrette per un algoritmo che opera su dati in cui nemmeno noi sappiamo che suddivisione aspettarci è sempre difficile. Rispetto ad altri algoritmi che operano con un numero predefinito di classi, K-means è più sensibile ad errori nella stima del parametro K. Infine, K-means non sopporta molto bene gli outlier.

## Kernel density Estimation

In statistica la Kernel density estimation (KDE) è una rappresentazione non parametrica attraverso la quale viene stimata la funzione di densità probabilità di una variabile casuale. Fondamentalmente è un problema di smoothing dei dati dove viene fatta dell'inferenza circa una popolazione sulla base di un campione finito. Sia  $x_1, \dots, x_n$  un campione indipendente e identicamente distribuito estratto da una qualche distribuzione di cui non si conosce la densità  $f$ . La stima Kernel è:

$$\hat{f}_h = \frac{1}{n} \sum_{i=1}^n K_h(x - x_i) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right) \quad (4.11)$$

dove  $K(\cdot)$  è il kernel - una funzione simmetrica ma non necessariamente positiva integrabile a uno - e  $h > 0$  è un parametro di smoothing chiamato bandwidth. Il kernel con il quoziente  $h$  è chiamato lo scaled kernel ed è definito come  $K_h(x) = \frac{1}{h} K\left(\frac{x}{h}\right)$ . In maniera intuitiva si cercherà di scegliere  $h$  piccolo quanto i dati permettano, comunque è presente un trade-off tra l'errore dello stimatore e la sua varianza. Possono essere impiegate diverse funzioni kernel: uniforme, triangolare, normale e altre. Per le sue proprietà matematiche solitamente viene utilizzata la normale. La scelta di  $h$  è stata fatta in modo da ottenere risultati migliori sotto l'ipotesi di distribuzione normale dei dati.

## Fasi della Metodologia

Il metodo consta di una serie di passi da realizzare in sequenza:

### 1. Pre-processamento dati

Il dataset del profilo viene scalato mediante il cosiddetto “Robust scaling” [57] nel quale i valori tradizionali di media e varianza vengono sostituiti, rispettivamente, con quelli di mediana e con il valore di mediana assoluta della deviazione dalla mediana, al fine di ottenerne una stima più robusta all'influenza degli outlier. Il risultato del procedimento consente di ottenere dei profili con delle feature aventi media prossima a zero e varianza unitaria. Sia  $X = \{x_1, \dots, x_n\}$  il dataset di riferimento dove  $x_i \in \mathfrak{R}^m$

$$\begin{cases} x_{medianj} = median(x_{1j}, \dots, x_{nj}), & \text{con } j = 1, \dots, m \\ s_{MADj} = 1.4826 median(|x_{1j} - x_{medianj}|, \dots, |x_{nj} - x_{medianj}|), & \text{con } j = 1, \dots, m \end{cases} \quad (4.12)$$

Pertanto i nuovi profili saranno:

$$x'_i = \frac{x_i - median}{s_{MAD}} \quad (4.13)$$

## 2. Definizione dei parametri della rete

Prima di effettuare l'addestramento della rete occorre definire dei parametri di progettazione della stessa, in particolare essa richiederà di fissare il numero di iterazioni dell'algoritmo e il numero di nodi del reticolo.

## 3. Addestramento della rete

Una volta definita l'architettura della rete stessa, si procederà con l'addestramento vero e proprio, mediante il quale si otterranno i nodi.

## 4. Clustering dei nodi mediante K-Means

Una volta che ogni profilo del dataset sarà stato assegnato ad uno dei nodi della rete, occorrerà effettuare un ulteriore step per avere delle informazioni qualitative circa il dataset. In particolare mediante l'algoritmo K-Means e impostando il valore delle classi pari a due verranno suddivisi i nodi della rete in due raggruppamenti. Il raggruppamento corrispondente alla condizione in controllo sarà determinato in base al valore minimo di varianza dei profili rispetto al nodo mediano. Dato che ad ogni nodo in controllo corrisponderanno i profili associati ad esso dalla rete, dai raggruppamenti trovati verrà determinata una prima suddivisione del dataset di training.

## 5. Stima della soglia mediante KDE

Identificato il raggruppamento corrispondente alla condizione in controllo si procede con il calcolo della statistica di interesse; sono state utilizzate quattro tipologie: due presenti in letteratura, quali la distanza minima e quella basata sulla distanza normalizzata dal nodo mediano, introdotta da Xuefeng. Gli altri due parametri di controllo introdotti sono il valore di mediana delle

distanze dai nodi e quella di correlazione minima. Una volta determinata la distribuzione della statistica univariata scelta per i campioni in controllo, si procede con la definizione della soglia mediante la stima Kernel al valore di probabilità di falsi allarmi desiderata ( $\alpha$ ).

## 6. Carta di controllo

Determinata la soglia si creerà la carta di controllo con il valore limite determinato nella fase precedente e tutti i valori che supereranno il valore limite saranno considerati dati fuori controllo.

## Statistiche Impiegate

### 1. Distanza minima

La statistica di controllo maggiormente impiegata in letteratura per il monitoraggio è sicuramente la distanza minima del dato di training dal nodo più vicino, detto anche errore di quantizzazione minimo. Questo indicatore funziona particolarmente bene quando la rete è stata addestrata con condizioni di controllo; infatti in tal caso il valore di distanza di un dato outlier sarà evidentemente molto maggiore delle condizioni già conosciute. La soglia verrà calcolata mediante il metodo presentato precedentemente del Kernel density estimation, sui valori di distanza dei dati del cluster in controllo determinati con l'algoritmo K-Means applicato sui nodi.

$$d_i = \min_j \|x_i - w_j\|^2 \text{ con } j = 1, \dots, RxC, \quad i = 1, \dots, N \quad (4.14)$$

### 2. Distanza di Xuefeng

La distanza di Xuefeng prende spunto dalla tecnica utilizzata in [56] per eliminare i nodi outlier. In particolare viene generalizzata per calcolarla direttamente sui dati di training. Pertanto verranno calcolate le distanze dei dati dal nodo mediano:

Il vettore mediano di tutti i pesi, sarà il seguente:

$$w_{j,median} = median(w_{j,1}, \dots, w_{j,RxC}), j = 1, 2, \dots, l \quad (4.15)$$

Si valuta la distanza di ogni vettore di training con il nodo determinato al passo precedente attraverso la formula:

$$r_i = \left[ \sum_{k=1}^l (x_{k,i} - w_{k,median})^2 \right]^{\frac{1}{2}}, \quad i = 1, 2, \dots, N \quad (4.16)$$

Si determina il valore mediano e la stima della deviazione standard delle distanze determinate:

$$\begin{cases} r_{median} = median(r_1, \dots, r_N) \\ r_{MAD} = 1.4826 median(|r_1 - r_{median}|, \dots, |r_N - r_{median}|) \end{cases} \quad (4.17)$$

I valori di distanza di interesse saranno i seguenti:

$$d_i = \left| \frac{d_i - d_{median}}{d_{MAD}} \right| \quad i = 1, 2, \dots, N \quad (4.18)$$

La soglia verrà stimata sui valori di distanza dei dati del cluster in controllo determinati con l'algoritmo K-Means applicato ai nodi. I valori di distanza da tutti i nodi di questi vengono normalizzati mediante la mediana e la mediana delle deviazioni assolute dalla mediana delle stesse distanze.

### 3. Distanza mediana

Un'altra misura di distanza d'interesse si ricava, partendo dai valori di distanza di ciascun dato dai nodi della mappatura bidimensionale. Partendo dal presupposto che la contaminazione del dataset riguardi una percentuale ridotta dello stesso, ci si attenderà che la rete abbia popolato di nodi in maniera più densa le zone a cui corrisponderà una maggiore presenza di dati, dato che la Som approssima tramite i nodi la distribuzione di probabilità dei dati di training come dimostrato sotto stringenti condizioni in [59]. Pertanto il valore di mediano di distanza di ciascun dato dovrà appartenere ad un nodo in controllo o almeno più vicino alla zona più densa di nodi. Quindi con tale distanza si riuscirà ad evidenziare gli scostamenti dei dati fuori controllo. La soglia verrà stimata sui valori di mediana delle distanze da tutti i nodi dei

dati del cluster in controllo determinati con l'algoritmo K-Means applicato sui nodi.

$$d_i = \text{median}(d_1, \dots, d_{RxC}) \text{ con } i = 1, \dots, N \quad (4.19)$$

#### 4. Correlazione minima

Questa tipologia di statistica è stata introdotta come modifica al valore di quantizzazione minimo, sostituendo la distanza euclidea semplice con la correlazione. Può essere utilizzata esclusivamente per profili e non con dati a bassa dimensionalità, poiché si sfrutta il legame di correlazione esistente tra i segnali in ingresso e i nodi addestrati tramite la Som. Il metodo deve la sua forza alla tipologia di addestramento della Som che consente di adattare i nodi e il suo vicinato ai profili più rappresentati nel dataset. Pertanto grazie a questa proprietà la maggiore presenza dei dati in controllo comporterà che i valori di correlazione minima di questi saranno più elevati rispetto ai dati fuori controllo che essendo meno rappresentati sono riusciti ad influenzare meno i nodi della rete. La soglia verrà stimata sui valori di correlazione dei dati del cluster in controllo rispetto a tutti i nodi. La formula usata sarà la seguente:

$$d_i = \min_j \text{cor}(x_i, w_j) \text{ con } i = 1, \dots, N \quad j = 1, \dots, RxC \quad (4.20)$$

#### Confronto Statistiche

È stato effettuato il confronto tra le diverse statistiche introdotte per valutarne l'efficacia, a tal fine è stata stimata una soglia empirica per ogni tipologia con un dataset in controllo da diecimila dati, in maniera da avere un confronto a parità di falsi positivi. In tabella 4.1, si può vedere come per tutte e quattro le tipologie l'alpha sia correttamente controllato al valore del 5 %. Dal confronto nei diversi scenari nelle figure 4.1, 4.2 e 4.3 emerge come la soglia basata sul minimo della distanza euclidea abbia le prestazioni peggiori in tutti gli scenari, a causa dell'influenza delle condizioni di fuori controllo nell'addestramento. La soglia che meglio riesce a discriminare i profili buoni da quelli non buoni è sicuramente quella basata sulla correlazione minima, che fornisce risultati migliori in tutti e tre gli scenari



in tutte le condizioni. Le altre due soluzioni hanno tra loro prestazioni paragonabili ma inferiori. Per i confronti successivi si prenderà dunque in considerazione soltanto la statistica basata su correlazione.

Tipologia	FP(IC)
Correlazione minima	0.0492(0.0009)
Xuefeng	0.0516(0.0019)
Distanza minima	0.0499(0.0016)
Mediana della distanza	0.0508(0.0013)

Tabella 4.1: Tabella falsi positivi con IC batch means e  $\alpha = 5\%$

## 4.2 Metodi di decontaminazione basati su clustering

### 4.2.1 Approccio di Chen-Birch

Il metodo di Chen-Birch [60] è uno dei pochi lavori presenti in letteratura riguardante l'analisi retrospettiva di profili in fase I che, pertanto, verrà preso in considerazione come riferimento per la validazioni delle prestazioni della metodologia sviluppata. L'idea alla base è quella di creare inizialmente un raggruppamento per ottenere un insieme di profili del cluster principale con una forma simile. Per ridurre la dimensionalità dei profili vengono stimati i parametri di ciascuno di essi mediante la regressione spline. Questa, a differenza delle reti neurali che sono delle black box che non richiedono conoscenza a priori del processo, comporterà una complessità computazionale e applicativa, che renderà tale metodo meno attraente nei confronti del mondo produttivo reale. Le condizioni in cui è stato effettuato il confronto sono quelle migliori per il competitor ovvero in presenza di shift sostenuto posto nella parte terminale dei dati. I passi della metodologia saranno i seguenti:

1. **Step 1** Data la difficoltà di lavorare con profili, il primo step prevede di fittare la curva con un metodo appropriato, in tal modo ogni profilo sarà

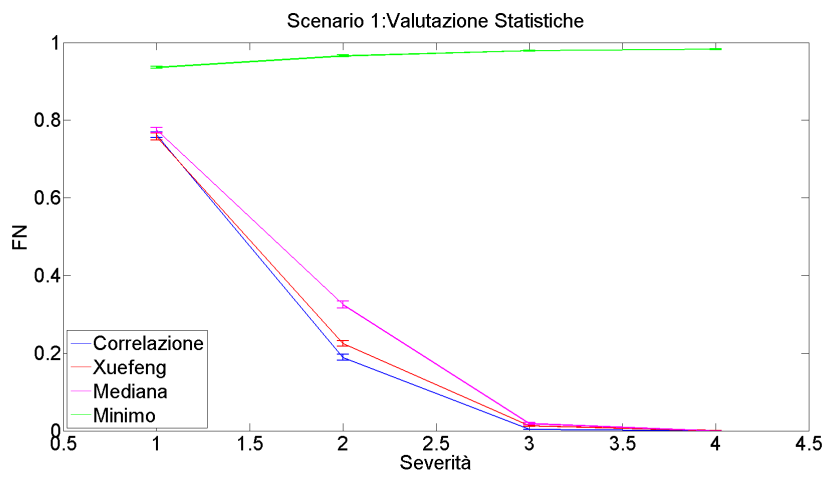


Figura 4.1: Confronto statistiche Scenario I

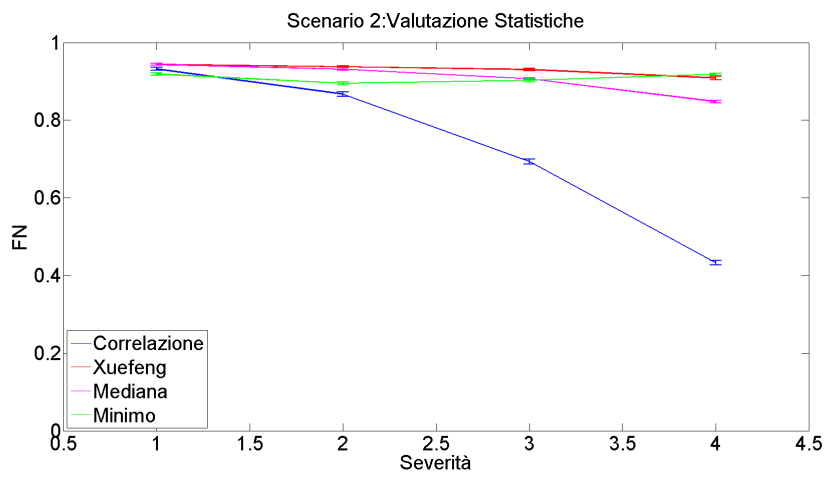


Figura 4.2: Confronto statistiche Scenario II

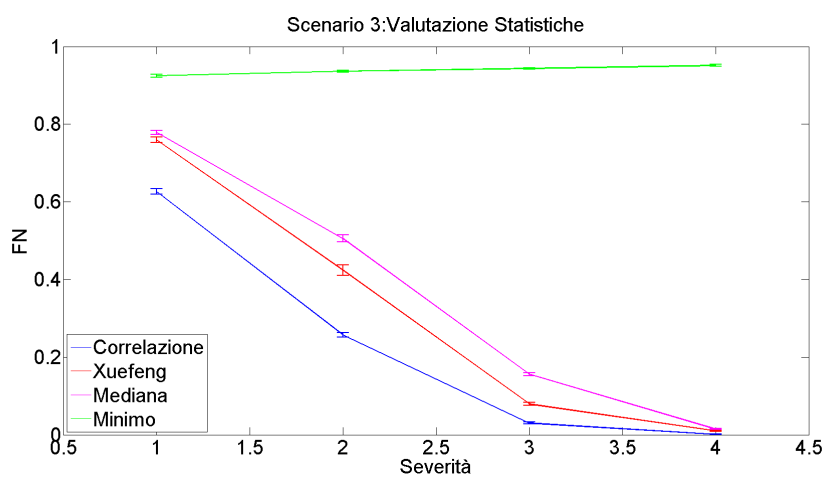


Figura 4.3: Confronto statistiche Scenario III

rappresentato da un vettore dei parametri del modello stimato. In totale si avrà una matrice di parametri  $m \times p$   $\hat{B}$ . Dove la riga  $i$ -esima di  $\hat{B}$  è un vettore  $p \times 1$  di parametri del profilo  $i$ -esimo. Inoltre verrà stimata la matrice di varianza-covarianza usando il metodo delle differenze successive sui parametri stimati.

$$\hat{V} = \frac{1}{2(m-1)} \sum_{i=1}^{m-1} (\hat{\beta}_{i+1} - \hat{\beta}_i)^T (\hat{\beta}_{i+1} - \hat{\beta}_i) \quad (4.21)$$

2. **Step 2** Il passo successivo prevede di calcolare la matrice di similarità  $S$ ,  $m \times m$  sui parametri di ogni profilo, usando la matrice  $\hat{V}$  ottenuta nello step I. Dove i valori di  $s_{ij}$  saranno i seguenti:

$$s_{ij} = (\hat{\beta}_i - \hat{\beta}_j)^T \hat{V}^{-1} (\hat{\beta}_i - \hat{\beta}_j) \quad (4.22)$$

dove  $\hat{\beta}_i$  e  $\hat{\beta}_j$  sono rispettivamente la  $i$ -esima e  $j$ -esima riga di  $\hat{B}$ .

3. **Step 3** Alla matrice dello step 2 verrà applicato il clustering gerarchico con il linkage completo che consiste nel prendere in considerazione la distanza massima tra tutte le possibili coppie di dati dei cluster, presi a due a due, e nell'unire quelli che presenteranno il minimo di questa distanza. Il clustering si arresterà una volta determinato un raggruppamento  $C$  con un numero di profili pari alla metà più uno della numerosità del dataset.
4. **Step 4** A partire da questo raggruppamento  $C$  verranno stimati i parametri della popolazione media mediante l'approccio a modelli misti e verrà calcolata la matrice di varianza covarianza. Per i profili esclusi verrà calcolata la statistica  $T^2$ .

$$T_i^2 = (\hat{\beta}_i - \hat{\beta}_{PA})^T \hat{V}^{-1} (\hat{\beta}_i - \hat{\beta}_{PA}) \quad (4.23)$$

Dove con  $i$  si denota il profilo  $i$ -esimo non contenuto nel raggruppamento determinato nello step precedente. Nel caso in cui il valore di  $T_i^2 < \chi_{[1-\frac{\alpha}{m}], df=p}^2$  allora il profilo  $i$ -esimo verrà aggiunto al raggruppamento  $C$  e verrà determinato un nuovo insieme  $C_{new}$ .

5. **Step 5** Se i profili in  $C_{new}$  sono differenti da quelli in  $C$ , si imporrà che  $C = C_{new}$  e si ripartirà dallo step 4. Altrimenti il raggruppamento  $C_{new}$  verrà ridefinito come  $C_{final}$
6. **Step 6** Usando ancora una volta l'approccio a modelli misti vengono stimati i parametri della popolazione media in  $C_{final}$

# Capitolo 5

## Analisi delle Prestazioni: simulation study

### 5.1 Definizione dataset di riferimento

Per valutare l'accuratezza del procedimento precedente, sono state fatte diverse simulazioni. Il modello utilizzato per i profili dei segnali è il seguente:

$$y_j = \sum_{i=1}^5 \beta_{i,j} \exp \{ \gamma_{i,j} (t + \omega_{i,j})^2 \} + e_j(t), \quad 0 \leq t \leq 1 \quad e \quad j = 1, 2, \dots \quad (5.1)$$

dove:

$$\beta_j = [\beta_{1,j}, \dots, \beta_{5,j}] \sim MN[\mu_\beta \Sigma_\beta]; \quad (5.2)$$

$$\gamma_j = [\gamma_{1,j}, \dots, \gamma_{5,j}] \sim MN[\mu_\gamma \Sigma_\gamma]; \quad (5.3)$$

$$\mu_\beta = [0.88, -0.5, 0.6, 0.6, -0.5], \quad \Sigma_\beta = \text{diag}[(8.8, 5, 6, 6, 5)e^{-2}] \quad (5.4)$$

$$\mu_\gamma = [-20, -50, -100, -150, -200], \quad \Sigma_\gamma = \text{diag}[(5.5, 8.5, 11, 9, 15)e^{-2}] \quad (5.5)$$

$$\omega = [-0.5, -0.45, -0.3, 0.7, -0.45] \quad (5.6)$$

$$\varepsilon_j(t) \sim N(0, s_\varepsilon) \quad \text{dove} \quad s_\varepsilon = 0.05 \quad (5.7)$$

Sono stati presi in considerazione tre tipologie di scenari fuori controllo, che fornissero una rappresentazione di condizioni realistiche e allo stesso tempo che fornissero uno spettro ampio di deviazioni. Il primo scenario prende in considerazione una

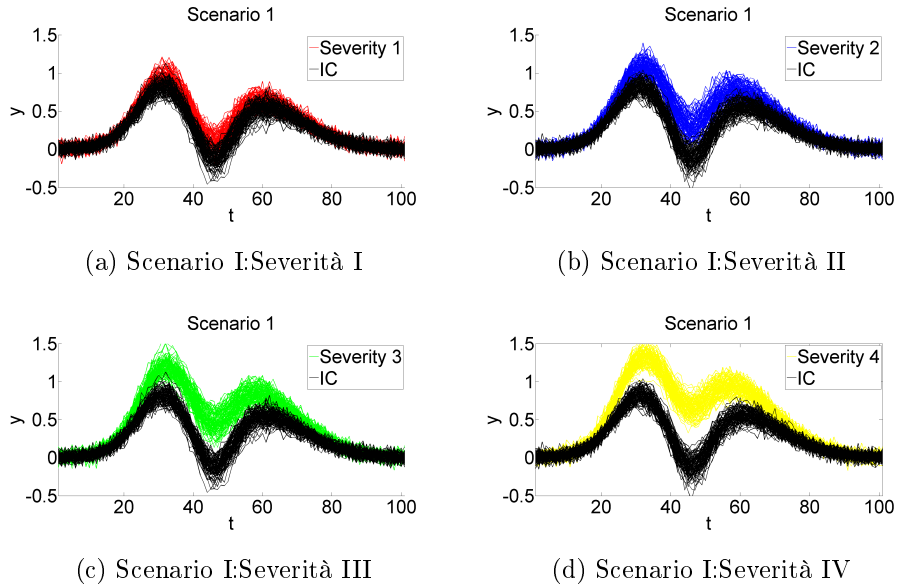


Figura 5.1: Scenario I

deviazione dei coefficienti beta del modello di una quantità proporzionale alla deviazione standard dei coefficienti stessi. In figura 5.1, viene illustrato il profilo nello scenario I nei diversi livelli di shift, rispetto alla situazione in controllo. Il modello considerato sarà il seguente:

$$\mu'_\beta = \mu_\beta + \delta_i * \Sigma_\beta \quad \text{con } i = 1, \dots, 4 \quad (5.8)$$

dove i valori del parametro delta sono tabellati in 5.1:

Severità	Shift( $\delta_i$ )
1 livello	$\delta_1 = 1$
2 livello	$\delta_2 = 2$
3 livello	$\delta_3 = 3$
4 livello	$\delta_4 = 4$

Tabella 5.1: Tabella parametri di scostamento Scenario 1

Il secondo scenario illustrato in fig:5.2, si prevede l'inserimento di un disturbo sinusoidale sul segnale originale; variandone l'ampiezza si ottengono i diversi livelli

Severità	Ampiezza( $A_i$ )
1 livello	$A_1 = 0.025$
2 livello	$A_2 = 0.04$
3 livello	$A_3 = 0.055$
4 livello	$A_4 = 0.07$

Tabella 5.2: Tabella parametri di scostamento Scenario 2

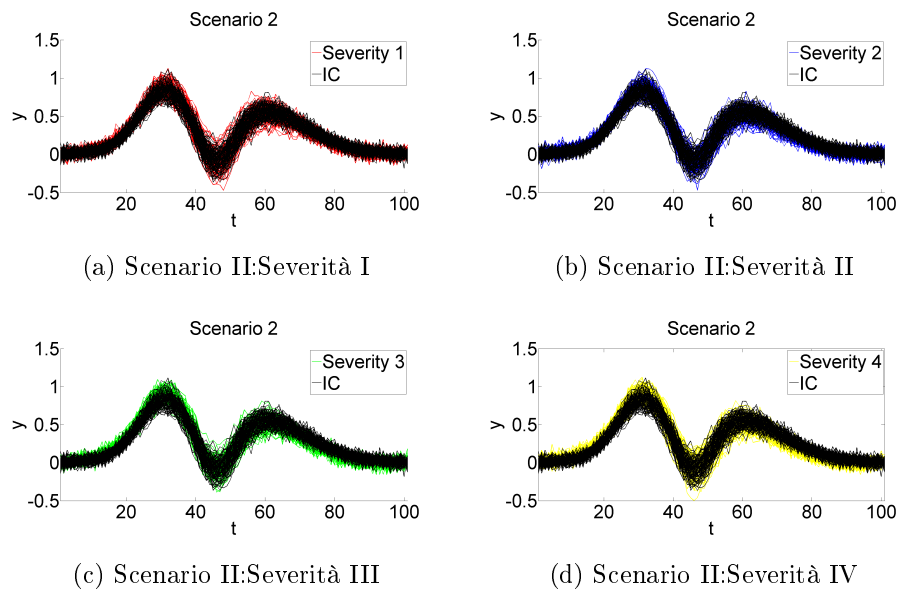


Figura 5.2: Scenario II

di severità della deviazione. La nuova media dei coefficienti beta del modello sarà la seguente:

$$y'_j = y_j + A_i * \sin(2\pi\omega t + \phi), i = 1, \dots, 4 \text{ e } j = 1, \dots, n \quad (5.9)$$

dove  $\omega = 1$  e  $\phi = 0$ . In tabella 5.2, vi sono i diversi livelli di ampiezza del segnale.

Il terzo scenario presente in fig:5.3, considera un aumento localizzato dell'ampiezza del profilo. Tabellati in 5.3 i diversi livelli di ampiezza. I profili ottenuti saranno i seguenti:

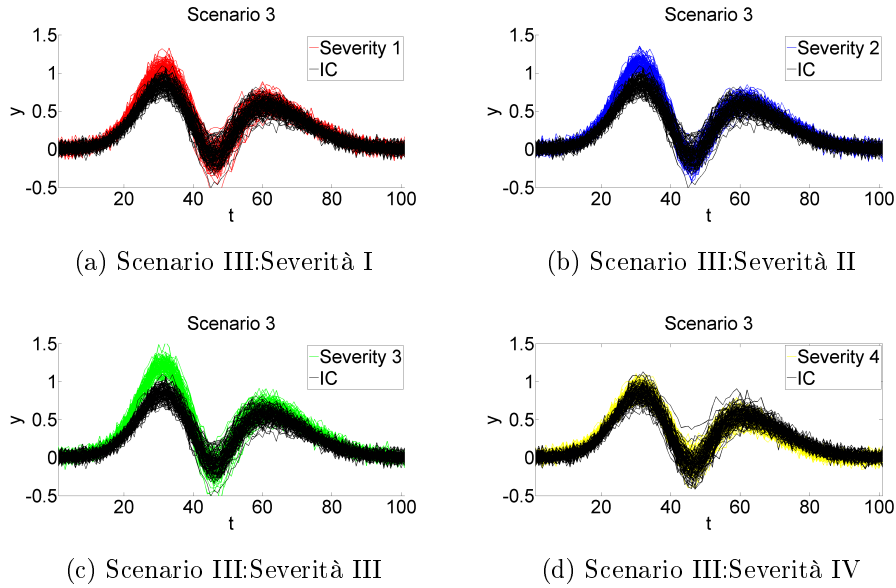


Figura 5.3: Scenario III

$$y'_j = y_j + A_i \omega_j, i = 1, \dots, 4, j = 1, \dots, n \quad (5.10)$$

con  $\omega = 1$  da  $j = 1, \dots, 50$  e  $\omega = \text{linspace}(0.2, 1)$  da  $j = 51, \dots, n$ .

Severità	Ampiezza( $A_i$ )
1 livello	$A_1 = 0.25$
2 livello	$A_2 = 0.35$
3 livello	$A_3 = 0.4$
4 livello	$A_4 = 0.55$

Tabella 5.3: Tabella parametri di scostamento Scenario 3

Per testare la capacità di classificazione del metodo è stata usata la categorizzazione tabellata in 5.4 per i profili classificati.

Si individua con A il numero di profili normali correttamente classificati come tali, con B quelli fuori controllo inseriti nella classe esatta, mentre con C i falsi negativi e con D i falsi positivi ovvero rispettivamente i profili segnalati come buoni ma in realtà fuori controllo e quelli segnalati come non buoni anche se in controllo. Da queste quantità determinate ricaviamo i seguenti coefficienti che



	Profilo In-Controllo	Profilo Outlier
Profilo In-Controllo	A	B
Profilo Outlier	C	D

Tabella 5.4: Tabella tipologia di classificazione

saranno utilizzati per valutare i risultati della sperimentazione.

$$FCC = \frac{A + D}{A + B + C + D} \quad (5.11)$$

$$Sensitivity = \frac{A}{A + B} \quad (5.12)$$

$$Specificity = \frac{D}{C + D} \quad (5.13)$$

$$FP = \frac{C}{A + C} \quad (5.14)$$

$$FN = \frac{B}{B + D} \quad (5.15)$$

## 5.2 Effetto dei parametri decisionali sulle prestazioni della Som

L'algoritmo della Som utilizzato per la sperimentazione è quello batch che, come dimostrato in [61], ci consente una maggiore semplicità e rapidità computazionale oltreché migliori valori di distorsione finale con il pregio di non richiedere la definizione del learning rate, assicurando nel contempo risultati deterministici e quindi riproducibili. Ovviamente a questi vantaggi si contrappongono una serie di svantaggi che sono, in particolare, una cattiva organizzazione finale, una peggiore visualizzazione, classi sbilanciate e una forte dipendenza dall'inizializzazione dei pesi che in alcuni casi può portare a raggiungere minimi locali con conseguenti risultati finali peggiori. Per le finalità di monitoraggio l'organizzazione finale non rappresenta un aspetto critico e in ogni caso i vantaggi che si ottengono in termini di tempo impiegato per l'addestramento della rete fanno protendere verso questa soluzione. Il metodo di inizializzazione scelto per i pesi della Som è quello lineare lungo i due autovalori più grandi; se invece la matrice è singolare gli autovettori non

possono essere calcolati e pertanto, si utilizzerà quello casuale. Vi sono, comunque, molti parametri da fissare per la progettazione e l'addestramento della rete, avendo noto che valori differenti porteranno a risultati finali diversi. I parametri vengono generalmente definiti attraverso il metodo "trial and error". Pertanto gli effetti dei parametri restanti verranno studiati in base alle prestazioni del metodo mediante simulazione. Questi saranno: il numero di nodi, il numero di iterazioni dell'algoritmo e la numerosità del dataset di training.

### 5.2.1 Effetto del numero dei nodi

La dimensione della Som determina il grado di generalizzazione che verrà prodotta dall'algoritmo della stessa; più nodi ci saranno più sarà dettagliata la rappresentazione, mentre se vi saranno meno nodi sarà più ampio il livello di generalizzazione. In ogni caso, i pattern più variabili saranno rilevati ad ogni livello. Un numero maggiore di nodi comporterà d'altro canto un tempo di training più lungo proporzionale al quadrato della numerosità stessa. Una regola euristica è stata introdotta in [52] dove viene fissato pari a cinque volte la radice del numero di dati di training, tuttavia è sempre rischioso affidarsi a formule euristiche al variare della tipologia di applicazione e delle condizioni senza un'evidenza empirica. Per questa ragione si è scelto di effettuare una sperimentazione per stabilire se i valori del numero di nodi avessero incidenza nei parametri di prestazione per la decontaminazione del dataset. In tabella 5.8 vi sono le condizioni imposte nella simulazione. Possiamo vedere dai grafici relativi ai falsi negativi del metodo più performante basato sulla soglia di correlazione nelle figure 5.4, 5.5, 5.6, che i valori di beta raggiunto un certo numero di nodi si attestano sulle medesime prestazioni, infatti le curve saranno quasi interamente sovrapposte. Solo con valori molto bassi del numero di nodi si riscontrano performance peggiori sia in termini di falsi negativi che positivi nelle figure 5.7, 5.8, 5.9. Anche per quanto concerne la percentuale correttamente classificata illustrata nelle figure 5.10, 5.11 e 5.12 i valori sono uniformi per tutti i livelli di nodi al di sopra del valore dieci. Per questa ragione si è deciso di utilizzare per le successive sperimentazioni un valore del numero di nodi pari a 150, in modo tale da garantire un buon livello di generalizzazione al metodo

Lunghezza training fase di ordinamento	50
Raggio iniziale e finale fase di ordinamento	10-1
Lunghezza training fase di convergenza	20
Raggio iniziale e finale fase di convergenza	1-1
Numerosità training set	500
Numero Nodi	10-50-100-150-200-250
Percentuale di contaminazione	15 %
Alpha	0.05
Numero di cluster K-Means	2
Repliche	500
Funzione di vicinato	Gaussiana
Lattice	Esagonale

Tabella 5.5: Tabella parametri di simulazione scelti analisi nodi

e nel frattempo non avere un onere computazionale eccessivo.

### 5.2.2 Effetto del numero di iterazioni

Il processo di apprendimento coinvolto nella creazione della feature map è di natura stocastica, ciò vorrà dire che l'accuratezza della mappa dipenderà dal numero di iterazione dell'algoritmo. Partendo da uno stato di completo disordine l'algoritmo della SOM raggiungerà gradualmente una rappresentazione organizzata dello spazio di input, a patto che i parametri vengano scelti correttamente. Il processo di adattamento può essere scomposto in due fasi: una fase di ordinamento seguita da una di convergenza. Nella prima fase avviene l'ordinamento topologico dei modelli, mentre nella seconda si ha una regolazione più precisa della mappa. Il valore del raggio di vicinato dovrà essere fissato ad un valore molto grande pari solitamente alla metà del lato più esteso della mappa rettangolare, in modo tale che i pesi dei neuroni inizialmente si riconducano tutti insieme a grandi passi verso l'area dello spazio di input da dove i vettori in ingresso alla rete provengono. Questi valori, poi, dovranno decrescere fino a comprendere solo i vicini più pros-

5.2. EFFETTO DEI PARAMETRI DECISIONALI SULLE PRESTAZIONI DELLA SOM69

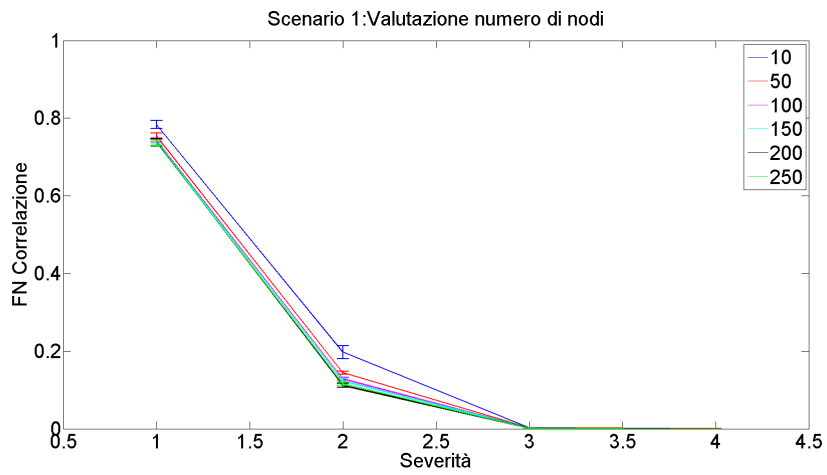


Figura 5.4: Falsi negativi Scenario I valutazione nodi

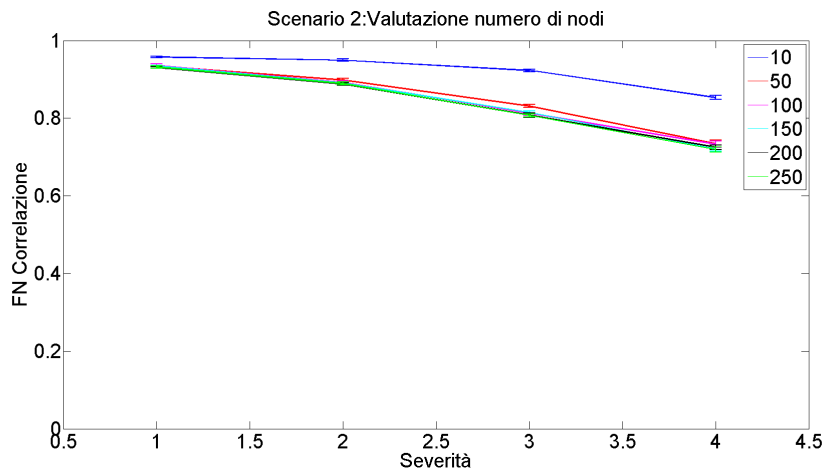


Figura 5.5: Falsi negativi Scenario II valutazione nodi

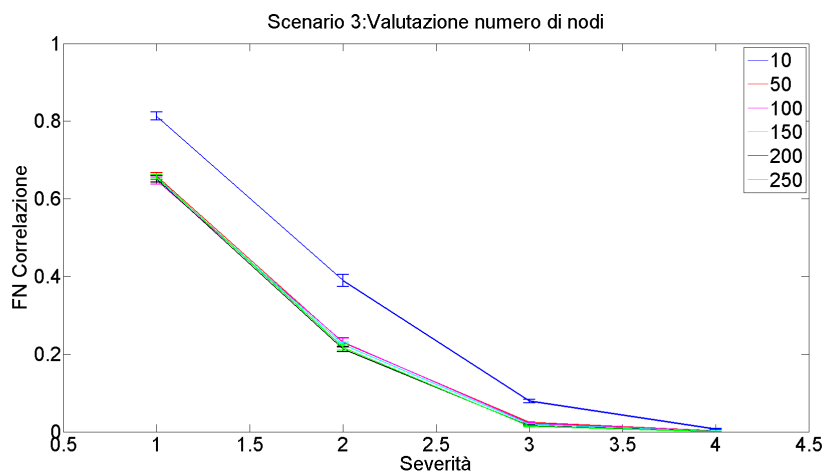


Figura 5.6: Falsi negativi Scenario III valutazione nodi

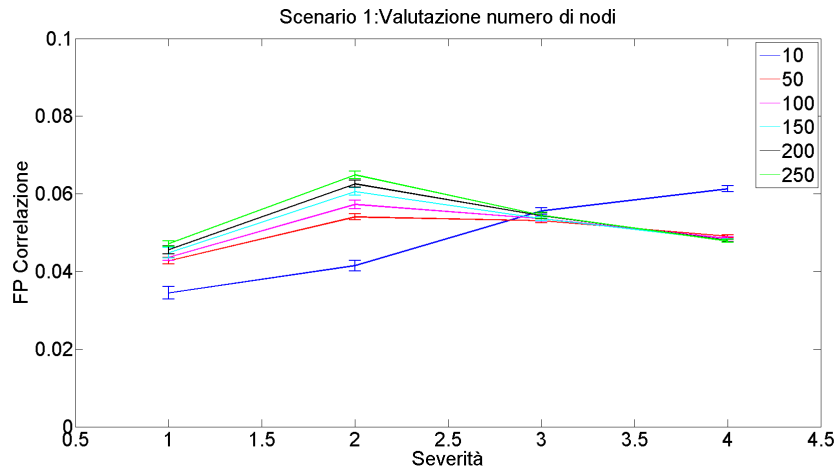


Figura 5.7: Falsi positivi Scenario I valutazione nodi

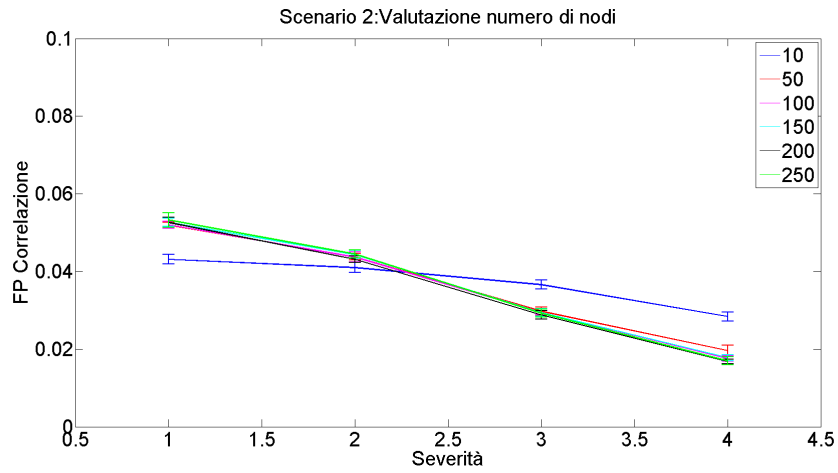


Figura 5.8: Falsi positivi Scenario II valutazione nodi

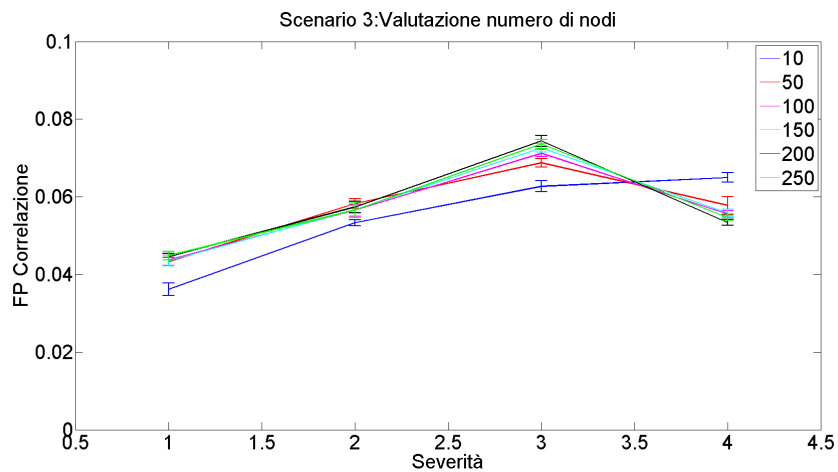


Figura 5.9: Falsi positivi Scenario III valutazione nodi

5.2. EFFETTO DEI PARAMETRI DECISIONALI SULLE PRESTAZIONI DELLA SOM71

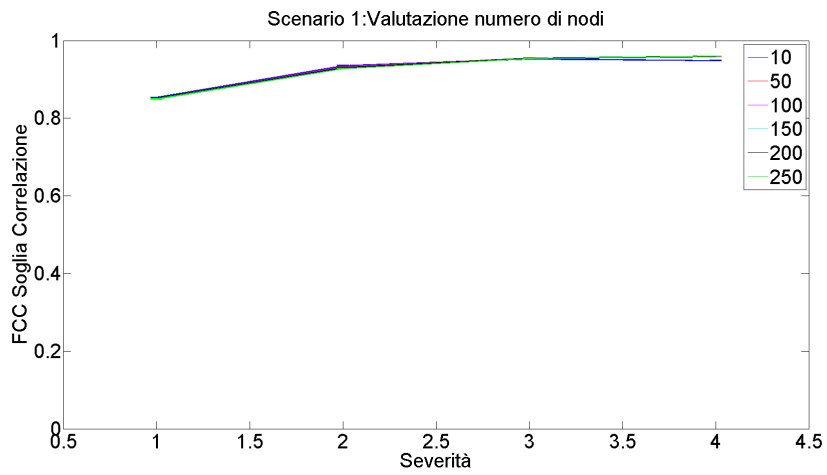


Figura 5.10: FCC Scenario I valutazione nodi

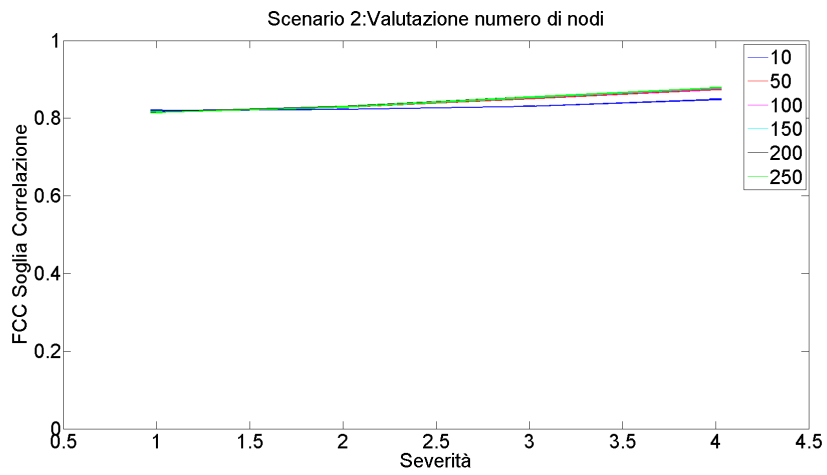


Figura 5.11: FCC Scenario II valutazione nodi

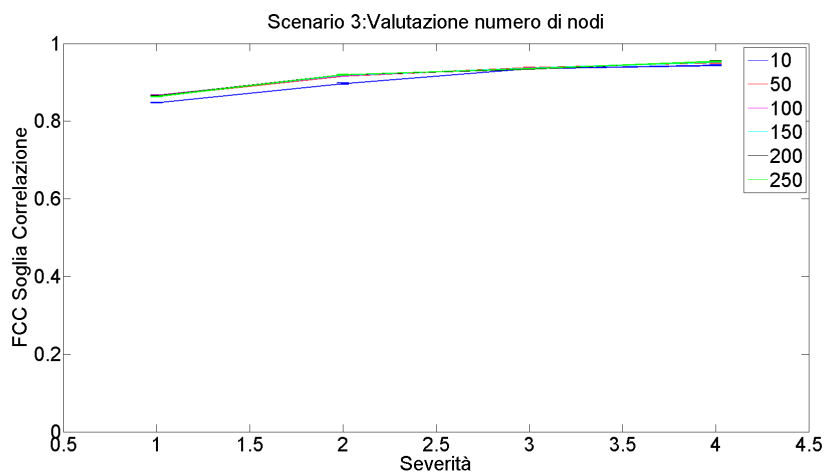


Figura 5.12: FCC Scenario III valutazione nodi

simi della best matching unit. Con tale condizioni di vicinato avviene anche la seconda fase di convergenza, che sarà necessaria per rifinire la rete e fornire una quantificazione dello spazio di input accurata statisticamente, questa durerà per il resto del training. Durante quest'ultima fase i vettori dei pesi convergono al loro valore corretto. In generale il numero di iterazioni dipende dalla dimensione del training set, quindi con pochi dati, occorreranno un numero elevato di iterazioni per arrivare a convergenza che, a sua volta, non è garantita. Per determinare il numero di iterazioni è stata ancora una volta effettuata una sperimentazione per verificare la convergenza dell'algoritmo. Per la valutazione del risultato della Som è stato considerato l'andamento dell'errore di quantizzazione in funzione del numero di iterazioni e l'aggiornamento massimo del peso, anche questo in funzione delle iterazioni. Questi due parametri sono stati plottati in entrambe le fasi del training. In tabella 5.6 vi sono le condizioni imposte nella simulazione.

Lunghezza training fase di ordinamento	1000
Raggio iniziale e finale fase di ordinamento	10-1
Lunghezza training fase di convergenza	1000
Raggio iniziale e finale fase di convergenza	1-1
Numerosità training set	500
Numero Nodi	150
Percentuale di contaminazione	0 %
Repliche	500
Funzione di vicinato	Gaussiana
Lattice	Esagonale

Tabella 5.6: Tabella parametri di simulazione scelti analisi iterazioni

Dal grafico relativo all'errore di quantizzazione durante la fase di ordinamento in figura 5.13 è possibile vedere come non si arrivi mai alla convergenza dello stesso ad un valore costante, ma anzi questo decresca indefinitamente al crescere delle iterazioni. Tale andamento sarà tipico di questa fase di addestramento e analogo qualsiasi sia il numero di iterazioni scelte. Ciò è dovuto alla diminuzione progressiva del raggio di vicinato durante la fase di aggiornamento dei pesi. D'altro canto

vediamo come l'aggiornamento dei pesi in figura 5.14 dopo una rapida decrescita iniziale, si assesti ad un livello prossimo allo zero, quindi dopo una fase iniziale di modifica più consistente dei pesi si arrivi ad una soluzione complessivamente stabile. Nella fase di convergenza vera e propria, l'errore di quantizzazione illustrato in figura 5.15, raggiunge un valore costante dopo un certo numero di iterazioni, la stessa cosa avviene per l'aggiornamento dei pesi in figura 5.16, che ovviamente raggiunto il valore di convergenza si arresterà. Da queste considerazioni si è scelto di utilizzare per le successive simulazioni un valore del numero di iterazioni della fase di ordinamento pari a cinquanta per garantire un buon livello di quantizzazione di partenza, mentre per la fase terminale di convergenza è stato verificato come siano sufficienti venti iterazioni per arrivare al risultato definitivo in termini di errore di quantizzazione.

### 5.2.3 Effetto della numerosità del dataset

La numerosità del dataset è un altro degli aspetti che andrà ad incidere sul risultato finale, infatti occorre che il numero di dati sia sufficiente al fine di raggiungere mediante soglia il valore di falsi positivi di progetto. Ciò è stato verificato mediante simulazioni in tabella 5.7 dove vi sono le condizioni imposte in tale simulazione. Possiamo vedere in figura 5.17 come per un numero limitato di dati il valore di falsi positivi progettuale non viene raggiunto e anzi i risultati presentano variabilità più alta. Al crescere del numero di dati si converge verso il valore desiderato; si è scelto di non andare oltre il livello dei cinquecento dati per mantenersi in una situazione facilmente adottabile nella realtà industriale.

## 5.3 Effetto della percentuale di contaminazione

Altro aspetto preso in considerazione è il diverso livello di percentuale di contaminazione del dataset. In tabella 5.8 sono riportate le condizioni imposte nella simulazione. Nelle figure 5.24, 5.25, 5.26 si può vedere come il livello di contaminazione abbia una forte influenza sul risultato finale della metodologia, infatti in presenza di più fuori controllo nei dati si compromette la capacità della metodolo-



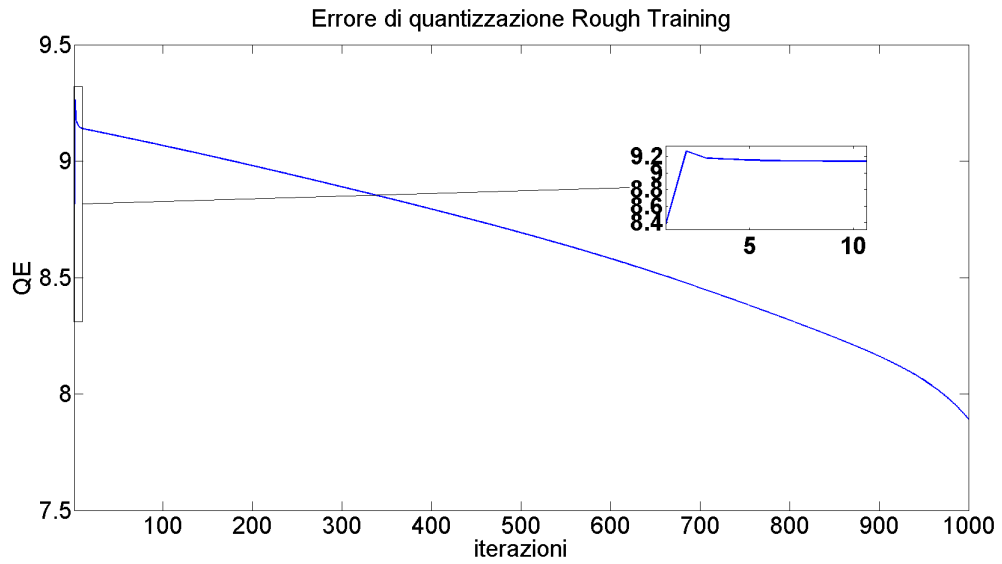


Figura 5.13: Andamento QE fase di ordinamento

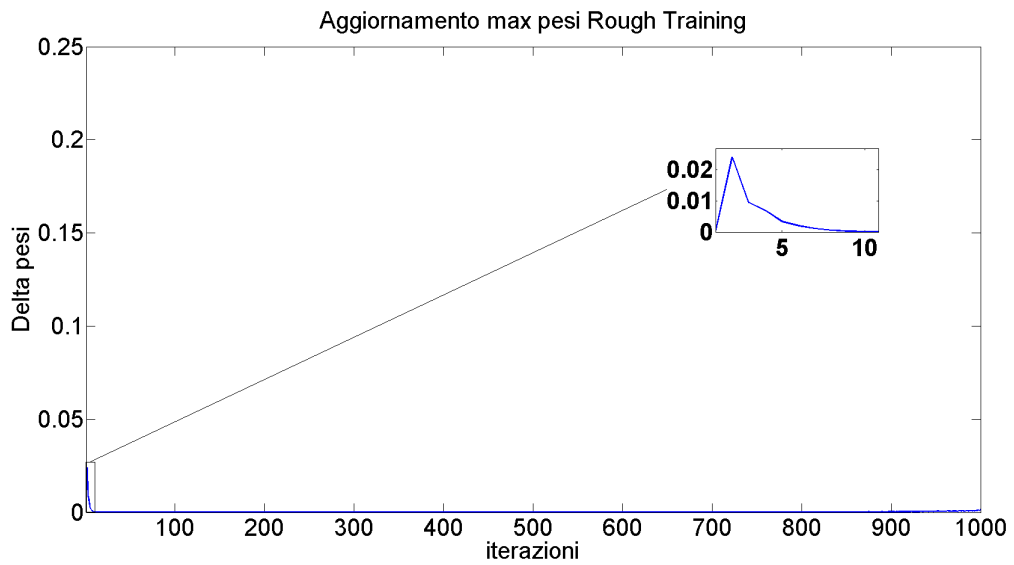


Figura 5.14: Andamento aggiornamento pesi fase di ordinamento

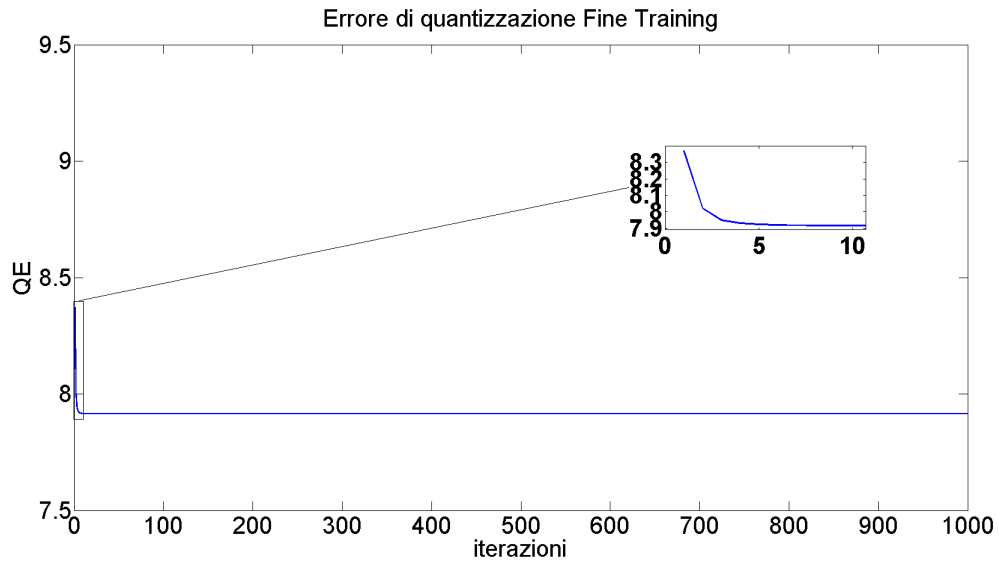


Figura 5.15: Andamento QE fase di convergenza

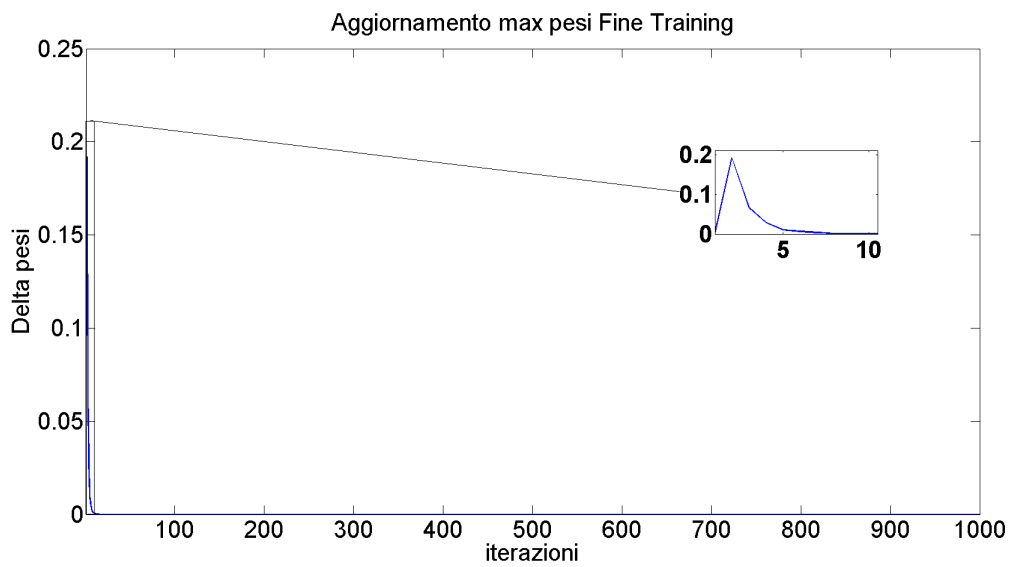


Figura 5.16: Andamento aggiornamento pesi fase di convergenza

Lunghezza training fase di ordinamento	50
Raggio iniziale e finale fase di ordinamento	10-1
Lunghezza training fase di convergenza	20
Raggio iniziale e finale fase di convergenza	1-1
Numerosità training set	100-200-300-400-500
Numero Nodi	150
Percentuale di contaminazione	15 %
Alpha	0.05
Numero di cluster K-Means	2
Repliche	500
Funzione di vicinato	Gaussiana
Lattice	Esagonale

Tabella 5.7: Tabella parametri di simulazione scelti analisi numerosità dataset

gia di discriminare tra le diverse condizioni, in particolare modo per bassi livelli di severità. Per quanto riguarda i valori di falsi positivi nelle figure 5.18, 5.19, 5.20 si avranno valori che sostanzialmente si assestano correttamente al valore di progetto del 5%. In conseguenza di questi due fattori il valore percentuale di correttezza nella classificazione nelle figure 5.21, 5.22, 5.23 vede prevalere le condizioni con bassa contaminazione.

## 5.4 Effetto del numero di cluster

Una delle criticità della metodologia è quella di dover fissare a priori il numero di raggruppamenti presenti all'interno dei dati. Per ovviare a questa limitazione è stato testato un metodo di scelta automatica del numero di cluster, facendo uso di due degli indici proposti in letteratura. Il procedimento prevede di determinare mediante l'indice Gap, dato che questo rappresenta il solo metodo analitico per tale valutazione, la presenza o meno di raggruppamenti distinti. Nel caso in cui la statistica Gap fornisca indicazione di presenza cluster distinti per una scelta più accurata del loro numero esatto si farà uso dell'indice Silhouette, entrambi di

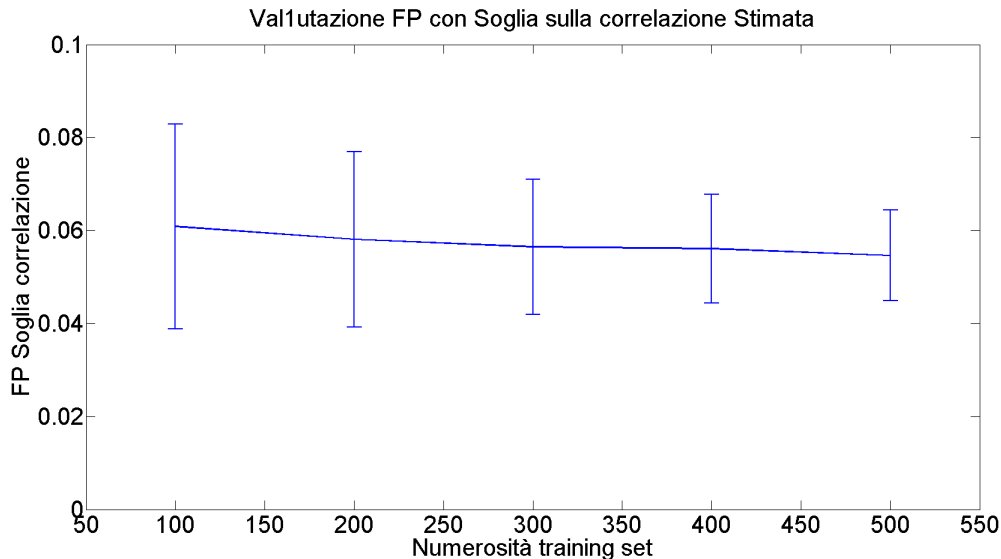


Figura 5.17: Dipendenza numerosità training set

seguito illustrati. I risultati di questo approccio evidenziano come questi indici per funzionare correttamente abbiano bisogno di una separazione consistente dei raggruppamenti. A causa di questo fattore, per il primo livello di severità l'indice Gap fornisce nella quasi totalità dei casi un'indicazione non corretta di singolo gruppo, compromettendo i risultati conseguenti della metodologia. In tabella 5.9 vi sono le condizioni imposte nella simulazione. È possibile vedere in figura 5.29 come le prestazioni in termini di falsi negativi con scelta del  $K$  automatico siano peggiori del metodo originale a causa degli errori compiuti dagli indici di valutazione nel determinare il numero corretto di classi, questo ovviamente comporterà l'indebolimento della metodologia nel determinare gli outlier presenti nei dati di addestramento. In termini di falsi positivi in figura 5.27, vediamo come il metodo di scelta automatico fornisca valori più bassi del valore di progetto pari al 5% generando un numero minore di falsi allarmi; a causa di questo i valori di FCC in figura 5.28, saranno sostanzialmente uguali. In appendice è presente la tabella con i  $K$  selezionati dall'algoritmo.

### Indice Gap

La statistica Gap proposta da Tibshirani in [55] è uno degli approcci esistenti per stimare il numero di cluster nei dati. Siano  $x_{ij}$   $i = 1, \dots, n$ ,  $j = 1, \dots, p$  i dati

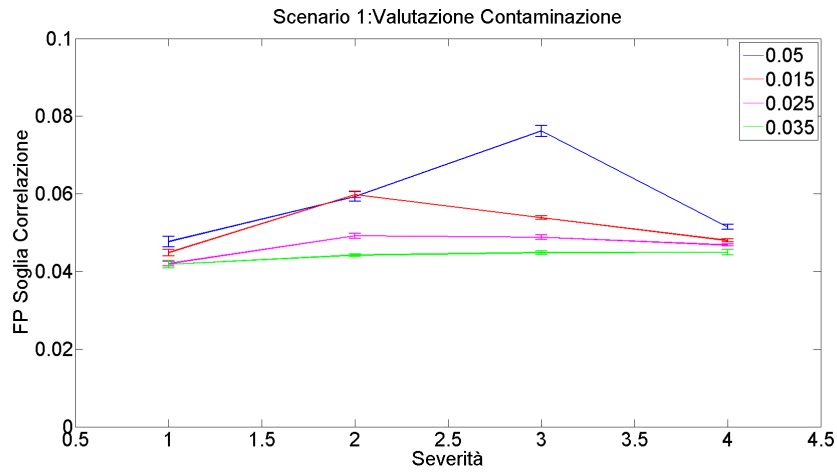


Figura 5.18: Falsi positivi Scenario I valutazione contaminazione

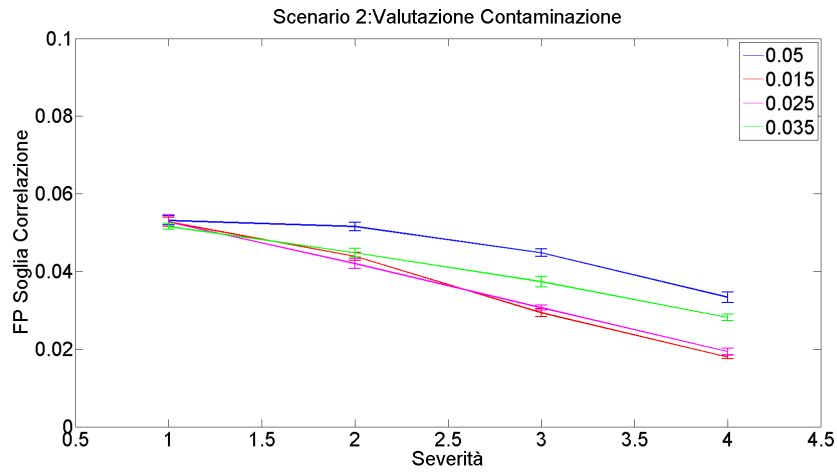


Figura 5.19: Falsi positivi Scenario II valutazione contaminazione

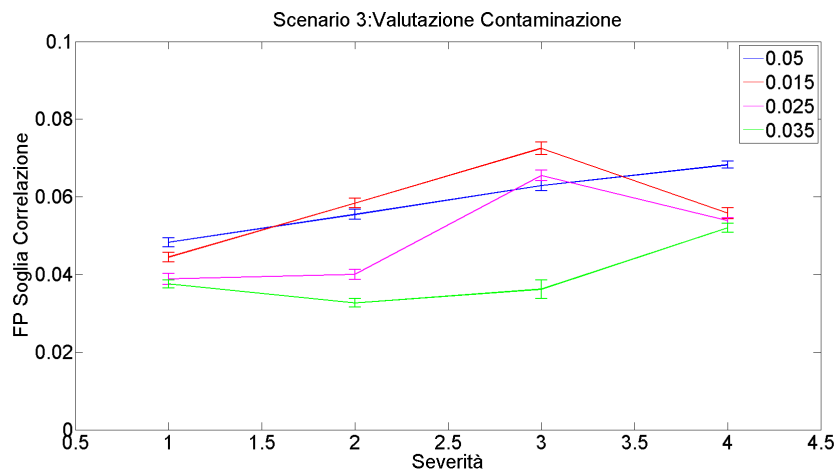


Figura 5.20: Falsi positivi Scenario III valutazione contaminazione

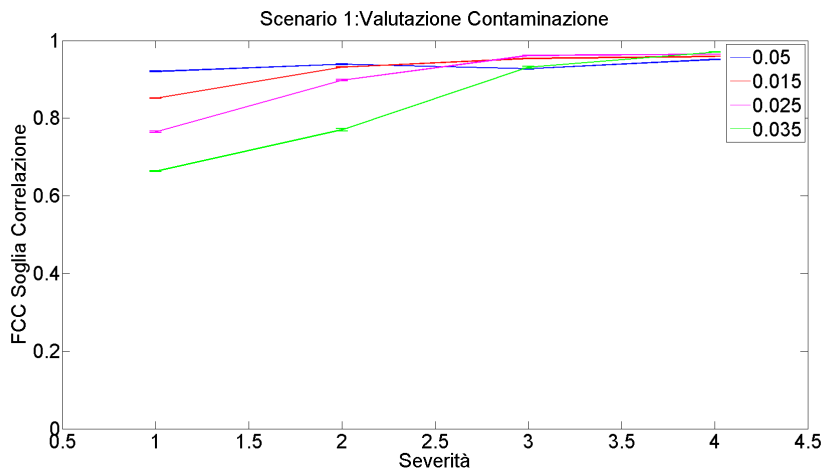


Figura 5.21: FCC Scenario I valutazione contaminazione

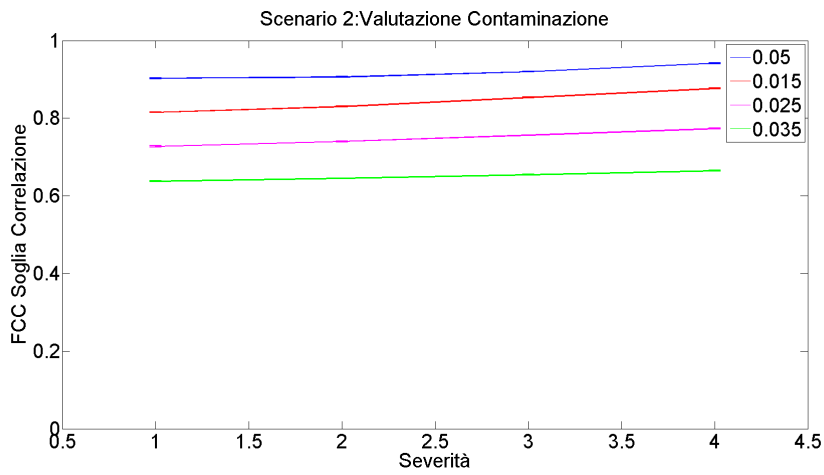


Figura 5.22: FCC Scenario II valutazione contaminazione

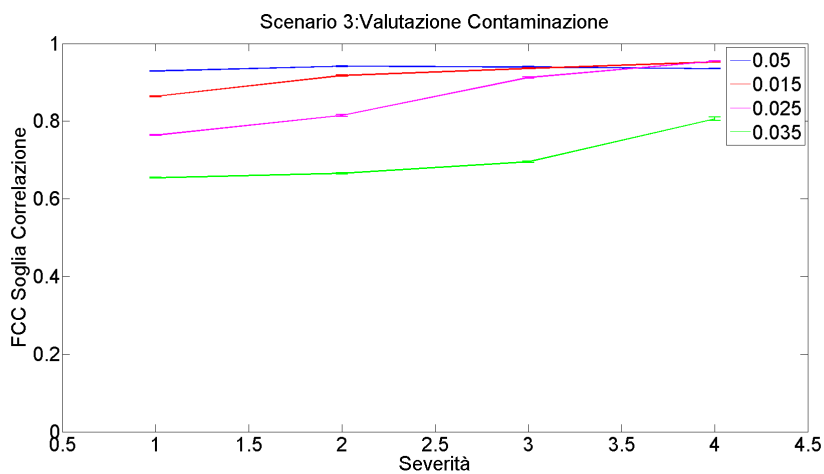


Figura 5.23: FCC classificata Scenario III valutazione contaminazione

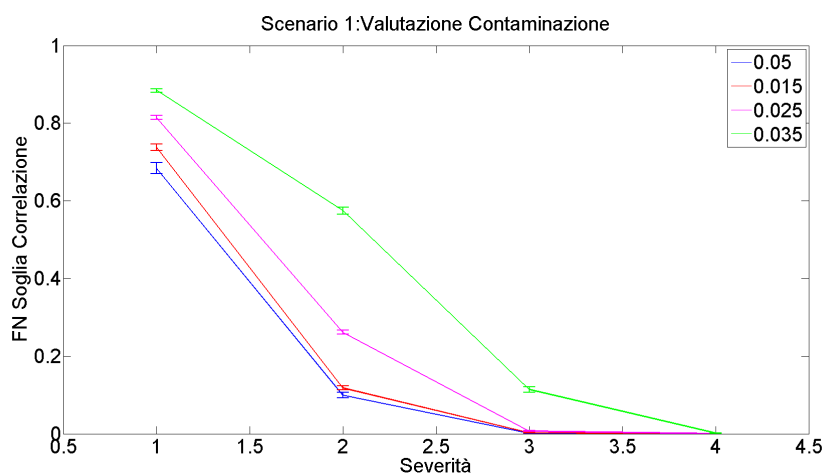


Figura 5.24: Falsi negativi Scenario I valutazione contaminazione

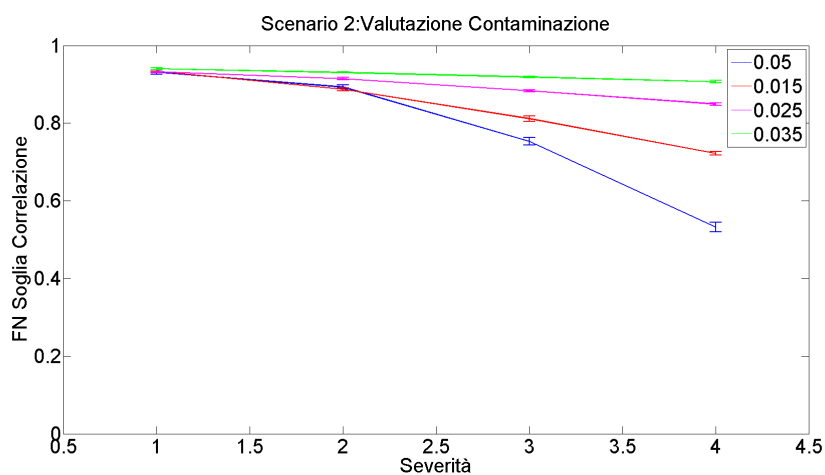


Figura 5.25: Falsi negativi Scenario II valutazione contaminazione

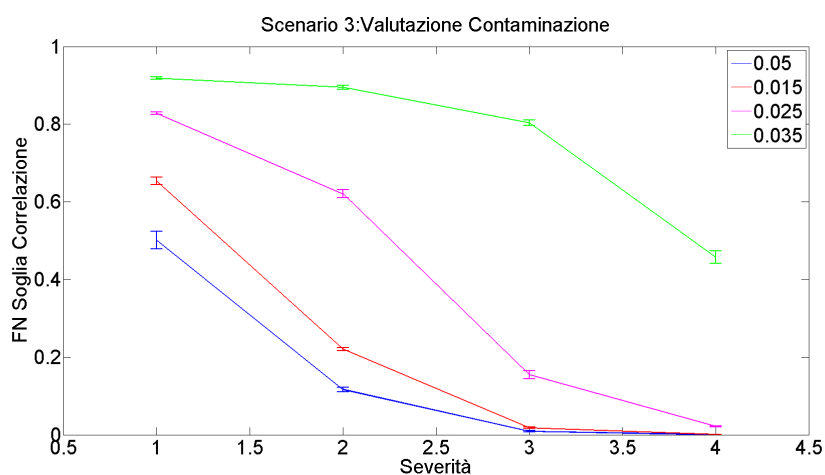


Figura 5.26: Falsi negativi Scenario III valutazione contaminazione

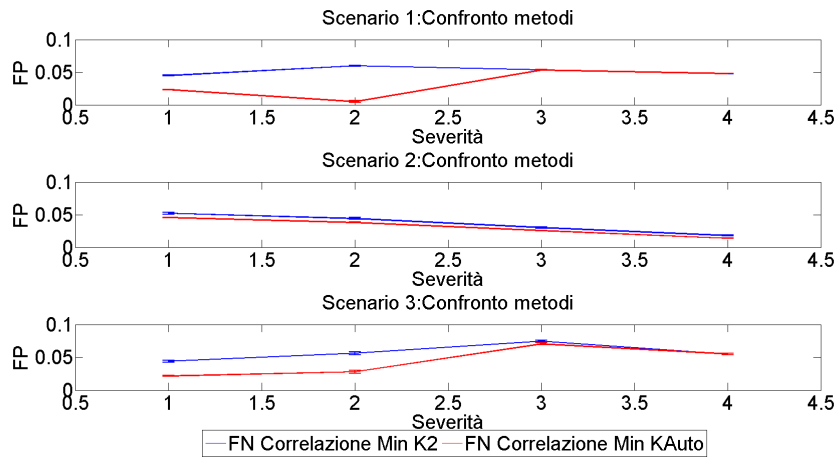


Figura 5.27: Confronto Falsi positivi tra k automatico e k=2

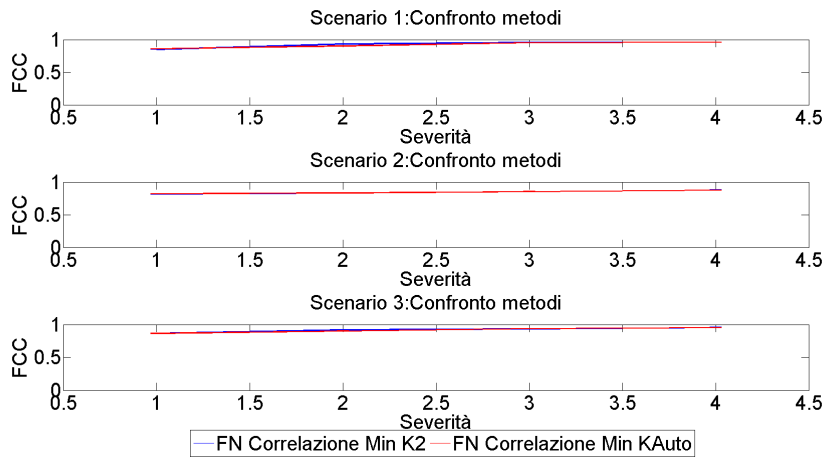


Figura 5.28: Confronto FCC tra k automatico e k=2

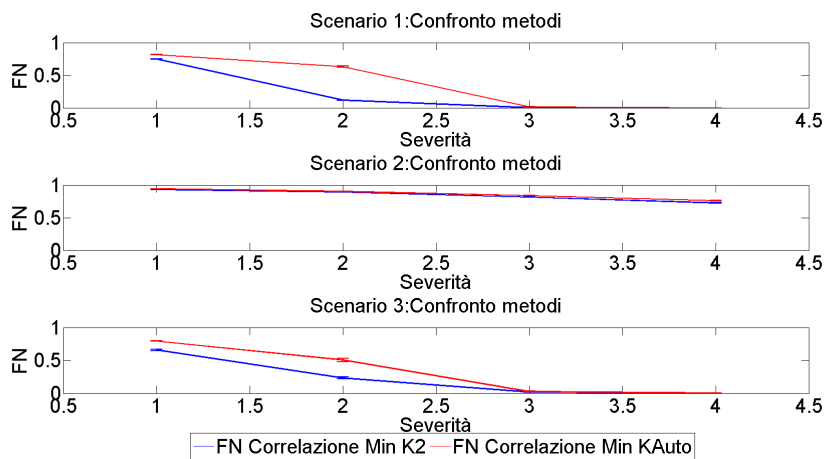


Figura 5.29: Confronto Falsi negativi tra k automatico e k=2



Lunghezza training fase di ordinamento	50
Raggio iniziale e finale fase di ordinamento	10-1
Lunghezza training fase di convergenza	20
Raggio iniziale e finale fase di convergenza	1-1
Numerosità training set	500
Numero Nodi	150
Percentuale di contaminazione	5-15-25-35 %
Alpha	0.05
Numero di cluster K-Means	2
Repliche	500
Funzione di vicinato	Gaussiana
Lattice	Esagonale

Tabella 5.8: Tabella parametri di simulazione scelti analisi contaminazione

consistenti in  $p$  feature misurate in  $n$  osservazioni indipendenti. Si supponga che i dati siano stati clusterizzati in  $k$  raggruppamenti in modo che  $C_k$  sia il generico cluster. Si definiscono le seguenti quantità, rispettivamente la somma delle distanza interne al cluster e il valore di dispersione interna:

$$D_r = \sum_{i,i' \in C_r} d_{ii'} \quad (5.16)$$

$$W_k = \sum_{r=1}^K \frac{1}{2n_r} D_r \quad (5.17)$$

L'idea è consiste nel confrontare la somma dei quadrati interna al cluster con quella attesa sotto una distribuzione di riferimento tramite la seguente funzione:

$$Gap(k) = E^* \{ \log(W_k) - \log(W_k) \} \quad (5.18)$$

dove  $E^*$  denota il valore atteso rispetto alla distribuzione di riferimento. Il valore di  $K$  migliore sarà quello per il quale il  $\log(W_k)$  decresce molto più del valore atteso. Pertanto si dovrà massimizzare il Gap per trovare il  $K$ . Due saranno le possibili scelte della distribuzione nulla:

Lunghezza training fase di ordinamento	50
Raggio iniziale e finale fase di ordinamento	10-1
Lunghezza training fase di convergenza	20
Raggio iniziale e finale fase di convergenza	1-1
Numerosità training set	500
Numero Nodi	150
Percentuale di contaminazione	15 %
Alpha	0.05
Repliche	500
Funzione di vicinato	Gaussiana
Lattice	Esagonale

Tabella 5.9: Tabella parametri di simulazione scelti analisi numero di cluster

- Generare  $n$  valori per ognuna delle  $p$  variabili da una distribuzione uniforme nel range di variazione osservato in ciascuna variabile.
- Generare gli  $n$  valori per ognuna delle  $p$  variabili da una distribuzione uniforme con supporto allineato secondo le componenti principali.

Il procedimento sarà il seguente:

1. Per  $k = 1, \dots, K$ , dove  $K$  è il massimo numero di cluster considerati. Si applica l'algoritmo di clustering al variare di  $k$  e si determina  $W_k$ .
2. Si generano  $B$  dataset di riferimento usando una delle due distribuzioni precedenti. Si clusterizzano tali distribuzioni calcolando  $W_b^*(k)$ ,  $b = 1, 2, \dots, B$  e  $k = 1, 2, \dots$ . Si determina la statistica gap:

$$Gap(k) = \frac{1}{B} \sum_b \log(W_b^*(k)) - \log(W(k)) \quad (5.19)$$

3. Viene calcolata la deviazione standard:

$$sd_k = \left[ \frac{1}{B} \sum_b (\log(W_b^*(k)) - \bar{l})^2 \right]^{\frac{1}{2}} \quad (5.20)$$

Dove  $\bar{l} = \frac{1}{B} \sum_b \log(W_b^*(k))$

4. Definito  $s_k = sd_k \sqrt{1 + \frac{1}{B}}$ . Il valore ottimo del numero di cluster è dato dal più piccolo  $k$  tale che  $Gap(k) \geq Gap(k+1) - s_{k+1}$ .

### Indice Silhouette

Kaufman e Rousseeuw [62] hanno proposto l'indice silhouette per determinare il numero ottimo di cluster per i dati. Sia  $a_i$  la dissimilarità media tra l'osservazione  $i$  e tutte le osservazioni nel cluster a cui appartiene. Denotiamo con  $d(i, C)$  la dissimilarità media tra  $i$  e tutti gli oggetti del cluster  $C$  diverso da quello di appartenenza. Sia  $b_i$  la minima di tali dissimilarità. Definiamo silhouette la quantità seguente:

$$s(i) = \frac{b_i - a_i}{\max a_i, b_i} \quad (5.21)$$

Quando il cluster contiene una sola osservazione viene assegnato ad  $s$  il valore arbitrario zero. La silhouette media è la media di tutte le  $s(i)$ :

$$sil_k = \frac{1}{n} \sum_i s(i) \quad (5.22)$$

Per  $k \geq 2$ , il numero di cluster stimato è il valore massimo di silhouette.

## 5.5 Confronto tra approcci di decontaminazione

In questo paragrafo viene effettuato il confronto con il metodo di Chen Birch presentato in [60] che rappresenta il competitor principale nel campo dell'analisi retrospettiva dei dati. Al fine di effettuare il confronto a parità di falsi allarmi sono state stimate per entrambi i metodi delle soglie empiriche mediante un dataset con diecimila dati in controllo. I parametri della rete per queste simulazioni sono presentate nella tabella 5.10.

Una volta verificato tramite i dati tabulati in 5.11, che in presenza di dati in controllo l'alpha sia correttamente controllato al valore fissato del 5%, si può procedere facendo delle considerazioni sulle prestazioni dei metodi adottati negli scenari di fuori controllo. Dalla figura 5.30 si evince che nel primo scenario simulato le prestazioni in termini di falsi negativi della metodologia sviluppata con SOM siano migliori del competitor; tuttavia in figura 5.31, il metodo di Chen Birch arriva

Lunghezza training fase di ordinamento	50
Raggio iniziale e finale fase di ordinamento	10-1
Lunghezza training fase di convergenza	20
Raggio iniziale e finale fase di convergenza	1-1
Numerosità training set	500
Numero Nodi	150
Percentuale di contaminazione	15 %
Alpha	0.05
Numero di cluster K-Means	2
Repliche	500
Funzione di vicinato	Gaussiana
Lattice	Esagonale

Tabella 5.10: Tabella parametri di simulazione scelti confronto metodi

molto più rapidamente alla convergenza della curva beta, mostrando maggiore sensibilità nei confronti del disturbo di natura sinusoidale. Infine per quanto riguarda il terzo scenario in figura 5.32, le prestazioni dei due metodi sono paragonabili con una migliore rapidità di discesa di Chen Birch. Tuttavia è rilevante sottolineare come il metodo mediante Som non abbia bisogno che i dati di fuori controllo siano disposti consecutivamente all'interno dataset, e come lo stesso sia in grado di riconoscere i fuori controllo in qualsiasi posizione essi siano e senza la necessità che si presentino uno di seguito l'altro. Questo è dovuto alla mancanza di influenza dell'ordine dei dati nell'applicazione dell'algoritmo di tipologia batch, dato che esso fornirà in ingresso alla rete tutti i dati contemporaneamente ad ogni iterazioni. In assenza di tale vincolo sull'ordinamento, cioè con fuori controllo presenti in ordine sparso tra i dati da classificare, il metodo di Chen Birch perde totalmente di efficacia e si renderà preferibile in tutte le condizioni il metodo sviluppato.

Tipologia	FP(IC)
Chen-Birch	0.0497(0.0008)
Correlazione minima	0.0491(0.0009)

Tabella 5.11: Tabella falsi positivi con IC batch means e  $\alpha = 5\%$

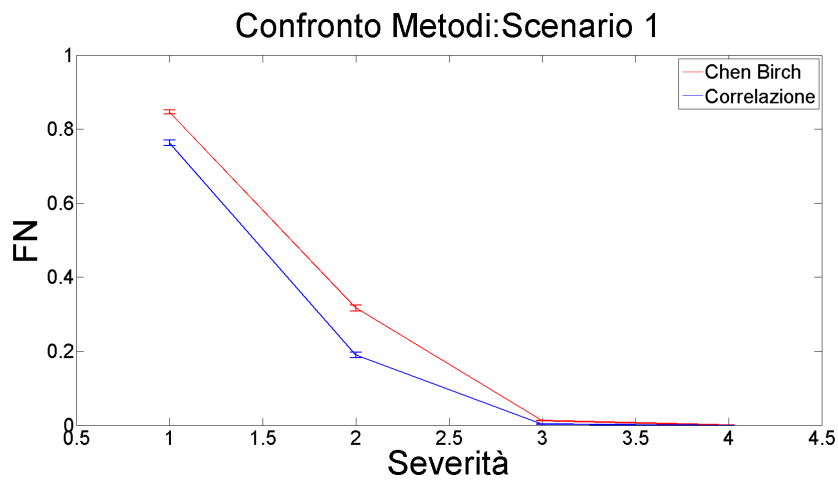


Figura 5.30: Confronto falsi negativi Chen Birch e Correlazione Scenario I

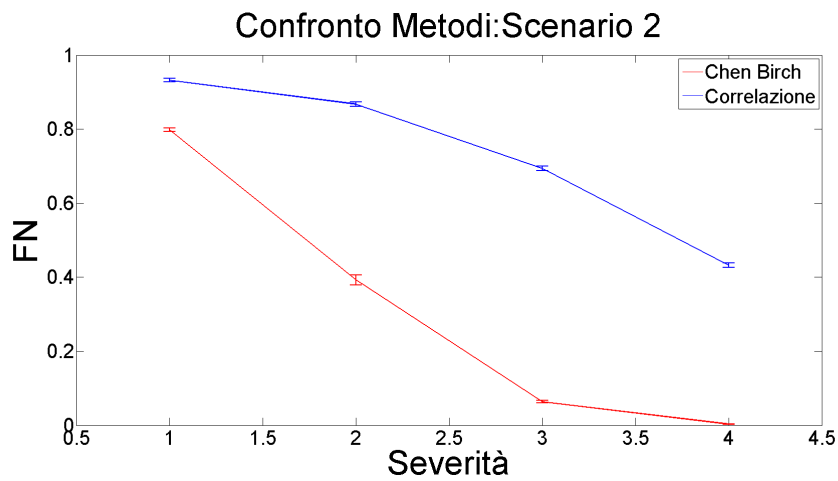


Figura 5.31: Confronto falsi negativi Chen Birch e Correlazione Scenario II

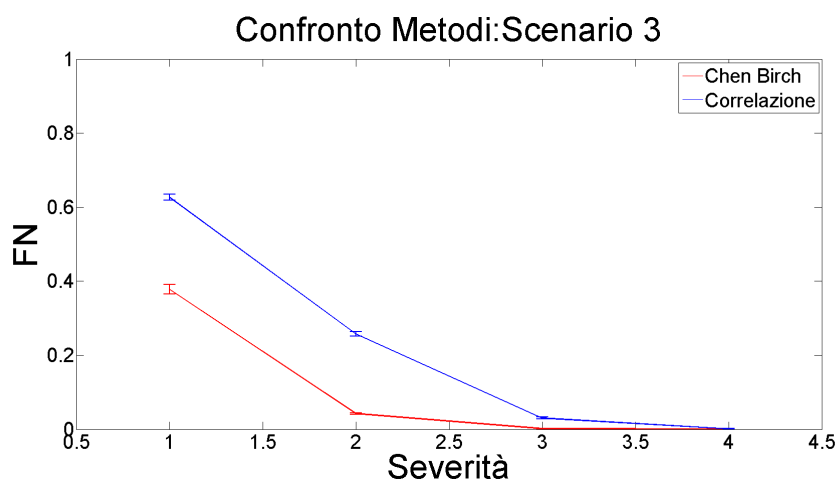


Figura 5.32: Confronto falsi negativi Chen Birch e Correlazione Scenario III

# Capitolo 6

## Conclusioni e Sviluppi Futuri

### 6.1 Conclusioni

Questa tesi riassume i risultati del lavoro di ricerca in ambito di "Profile Monitoring" con l'utilizzo di reti neurali. Le motivazioni alla base di questo lavoro sono molteplici e coprono sia l'ambito di ricerca che quello industriale. A livello di ricerca, il *Profile Monitoring* rappresenta un filone molto attivo in ambito di Statistical Process Control ed è motivato dai numerosi articoli pubblicati negli ultimi anni. Inoltre l'analisi retrospettiva della fase di addestramento di un sistema di monitoraggio in presenza di dati anomali, è una tematica molto importante ma ancora poco dibattuta in letteratura. A livello industriale, invece, l'interesse è da ricercare nel salto di qualità proposto nel nuovo framework di monitoraggio: il controllo del prodotto non avviene più a valle della produzione ma, grazie alla disponibilità di sensori e alle nuove tecniche di analisi più sofisticate, si è in grado di monitorare direttamente il processo produttivo stesso. A questo scopo ben si adattano le reti neurali poiché si sono dimostrate uno strumento "black box" che, grazie alla loro semplicità di implementazione e apprendimento, stanno raccogliendo il favore dell'industria.

Lo scopo del progetto è consistito quindi nel valutare la capacità della rete neurale *Self Organizing Map*, proposta da Kohonen in [6], di effettuare automaticamente la decontaminazione del dataset con cui essa è addestrata. Trattandosi di un argomento ancora non dibattuto in letteratura, è stato proposto un approccio

che consenta alla rete neurale di effettuare una analisi retrospettiva del training set ed in seguito ne sono state valutate le performance.

La scelta di questo specifico strumento è motivata dalla natura *non supervisionata* della rete e dal fatto che essa ben si adatta a clusterizzare i profili forniti in ingresso nei nodi della rete. Ad ogni nodo è associato un vettore di pesi sinaptici della stessa dimensione dello spazio degli ingressi. Questi nodi, una volta conclusa la fase di addestramento della rete, formeranno un'approssimazione della distribuzione dei campioni d'ingresso. Tuttavia la Som, così com'è stata proposta, non si adatta ad una analisi retrospettiva dei dati di training, motivo per cui i lavori di ricerca fino a questo momento presenti in letteratura riguardano esclusivamente la fase II del monitoraggio e sono quindi fortemente dipendenti dalla bontà dei dati di addestramento.

Questa problematica accomuna tutte le reti neurali e rappresenta il motivo principale per cui, nonostante le potenzialità di questi strumenti, non sia stata raggiunta una larga adozione nella realtà industriale. Nel presente lavoro, prima di introdurre la metodologia sviluppata, è stato presentato un breve resoconto sui campi di applicazione delle reti neurali con riferimento alla letteratura esistente in tale contesto. In seguito è stata illustrata la Self-Organizing Map e si è rilevato come la stessa non possa funzionare quale metodo di decontaminazione senza degli opportuni accorgimenti.

La metodologia innovativa proposta incorpora congiuntamente all'utilizzo della Self-Organizing Map una fase di clustering mediante l'algoritmo *K-Means*. Sfruttando la capacità di sintesi della rete neurale e l'abilità di separazione del clustering, si è riusciti ad effettuare in maniera automatica una discriminazione tra profili in controllo e anomali. Si è quindi in grado di identificare la reale firma del processo basandosi unicamente sui profili giudicati come rappresentativi di condizioni stabili. Questa tecnica è stata testata su un dataset di profili già discusso in letteratura su cui sono state, inoltre, effettuate diverse simulazioni per la configurazione ottimale della rete al fine di determinare i parametri progettuali che consentissero buone prestazioni.

Sono stati illustrati i risultati del metodo testato su dati simulati in diversi



scenari di fuori controllo. Tali risultati sono stati poi confrontati con il miglior competitor presente in letteratura.

In sintesi l'applicazione del metodo basato su Som, nonostante fornisca prestazione peggiori in due condizioni delle tre testate, ha dimostrato di ottenere risultati paragonabili a quelle del metodo competitor, con il vantaggio evidente di non richiedere la parametrizzazione dei profili attraverso un modello di fitting, aspetto che potrà rilevarsi estremamente critico in presenza di profili complessi.

## 6.2 Sviluppi futuri

Con il presente lavoro si è aperto un ramo di ricerca relativo alle reti neurali che sicuramente ancora necessita di molti sviluppi che potranno essere portati avanti in lavori futuri. Le possibili direzioni di ricerca per il miglioramento del modello possono riguardare diversi aspetti, sia dal punto di vista dell'algoritmo che dell'applicazione pratica. In primis l'applicazione del metodo proposto dovrà essere testata su un dataset di dati reali, abbandonando quindi l'ambito simulativo e scontrandosi con profili acquisiti da sensori installati presso macchine utensili. In parallelo si potranno approfondire gli studi sull'applicazione di un algoritmo di clustering differente dal K-Means, come ad esempio uno basato sulla densità. Inoltre occorreranno ulteriori studi sul metodo automatico di scelta del numero di cluster  $K$  al fine di renderlo più sensibile a shift limitati, oppure più robusto al rumore. Per quanto riguarda possibili varianti del metodo proposto invece, dato che è stato fatto uso unicamente della distanza euclidea nell'adozione dell'algoritmo di clustering e nell'addestramento della rete, si potrebbero adottare metriche di distanza differenti per tali algoritmi, come ad esempio la correlazione o altre tipologie, data l'evidenza empirica delle performance migliori della soglia basata su di essa. Infine, altre linee di studio potranno riguardare l'introduzione di tecniche di riduzione dimensionale dei profili per lavorare mediante feature a dimensionalità più bassa e quindi alleggerire il carico di dati somministrati alla rete.

# Acronimi

**ANN** Artificial Neural Network

**ARL** Average Run Length

**ART** Adaptive Resonance Theory

**FCC** Percentuale correttamente classificata

**FN** Percentuale profili OOC classificati com IC

**FP** Percentuale profili IC classificati com OOC

**KDE** Kernel Density Estimation

**MCM** Machine Condition Monitoring

**SOM** Self Organizing Map

**Svdd** Support vector data description

**Svm** Support vector machine

**TCM** Tool Condition Monitoring

# Bibliografia

- [1] Bianca Maria COLOSIMO, Giovanni MORONI, and Marco GRASSO. Real-time tool condition monitoring in milling by means of control charts for auto-correlated data. In *XXI CIRP Sponsorship Conference on Supervising and Diagnostics of Machining Systems, Karpacz, Poland, 2010*.
- [2] G Scrinzi, E Gregori, F Giannetti, D Galvagni, G Zorn, G Colle, and M Andrenelli. An evaluation model of protective function in forest management planning: slope stability in regard to shallow landslide events. *Forest@*, 2006.
- [3] Anil K Jain, Jianchang Mao, and KM Mohiuddin. Artificial neural networks: A tutorial. *Computer*, 29(3):31–44, 1996.
- [4] D. Floreano and C. Mattiussi. *Manuale sulle reti neurali*. Strumenti (Società editrice il Mulino).: Psicologia. Il Mulino, 2002.
- [5] T. Kohonen, M. R. Schroeder, and T. S. Huang, editors. *Self-Organizing Maps*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 3rd edition, 2001.
- [6] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1):59–69, 1982.
- [7] William H Woodall. Current research on profile monitoring. *Produção*, 17(3):420–425, 2007.
- [8] M. Grasso, P. Albertelli, and B.M. Colosimo. An adaptive {SPC} approach for multi-sensor fusion and monitoring of time-varying processes. *Procedia {CIRP}*, 12(0):61 – 66, 2013. Eighth {CIRP} Conference on Intelligent Computation in Manufacturing Engineering.

- [9] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology Press, 2002.
- [10] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 1988.
- [11] L.I Burke. Competitive learning based approaches to tool-wear identification. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(3):559–563, May 1992.
- [12] Y.S. Tarng, Y.W. Hsieh, and S.T. Hwang. Sensing tool breakage in face milling with a neural network. *International Journal of Machine Tools and Manufacture*, 34(3):341 – 350, 1994.
- [13] TaeJo Ko and DongWoo Cho. Adaptive modelling of the milling process and application of a neural network for tool wear monitoring. *The International Journal of Advanced Manufacturing Technology*, 12(1):5–13, 1996.
- [14] R.J. Kuo. Multi-sensor integration for on-line tool wear estimation through artificial neural networks and fuzzy neural network. *Engineering Applications of Artificial Intelligence*, 13(3):249 – 261, 2000.
- [15] Dornfeld D Rangwala, S. Sensor integration using neural networks for intelligent tool condition monitoring. *Journal of Manufacturing Science and Engineering*, 112(3):219–228, Nov 1990.
- [16] Laura I Burke. An unsupervised neural network approach to tool wear identification. *IIE transactions*, 25(1):16–25, 1993.
- [17] T.J. Ko, D.W. Cho, and M.Y. Jung. On-line monitoring of tool breakage in face milling using a self-organized neural network. *Journal of Manufacturing Systems*, 14(2):80 – 90, 1995.
- [18] Ibrahim Nur Tansel, Christine Mekdeci, Oscar Rodriguez, and Balemir Uragun. Monitoring drill conditions with wavelet based encoding and neural networks. *International Journal of Machine Tools and Manufacture*, 33(4):559 – 575, 1993.

- [19] Wang Haili, Shao Hua, Chen Ming, and Hu Dejin. On-line tool breakage monitoring in turning. *Journal of Materials Processing Technology*, 139(1):237–242, 2003.
- [20] YS Tarn, MC Chen, and HS Liu. Detection of tool failure in end milling. *Journal of materials processing technology*, 57(1):55–61, 1996.
- [21] Choon Seong Leem and David A Dornfeld. Design and implementation of sensor-based tool-wear monitoring systems. *Mechanical Systems and Signal Processing*, 10(4):439–458, 1996.
- [22] D.E. Dimla Jr, P.M. Lister, and N.J. Leighton. Neural network solutions to the tool condition monitoring problem in metal cutting—a critical review of methods. *International Journal of Machine Tools and Manufacture*, 37(9):1219 – 1241, 1997.
- [23] Abhinav Saxena and Ashraf Saad. Evolving an artificial neural network classifier for condition monitoring of rotating mechanical systems. *Applied Soft Computing*, 7(1):441 – 454, 2007.
- [24] J. Rafiee, F. Arvani, A. Harifi, and M.H. Sadeghi. Intelligent condition monitoring of a gearbox using artificial neural network. *Mechanical Systems and Signal Processing*, 21(4):1746 – 1754, 2007.
- [25] Gerald M. Knapp and Hsu-Pin (Ben) Wang. Machine fault classification: a neural network approach. *International Journal of Production Research*, 30(4):811–823, 1992.
- [26] T.I. Liu and N.R. Iyer. Diagnosis of roller bearing defects using neural networks. *The International Journal of Advanced Manufacturing Technology*, 8(4):210–215, 1993.
- [27] Roya Javadpour and Gerald M. Knapp. A fuzzy neural network approach to machine condition monitoring. *Computers & Industrial Engineering*, 45(2):323 – 330, 2003. 25th International Conference on Computers and Industrial Engineering.

- [28] V. Emamian, Mostafa Kaveh, and AH. Tewfik. Robust clustering of acoustic emission signals using the kohonen network. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on*, volume 6, pages 3891–3894 vol.6, 2000.
- [29] Odin Taylor and John MacIntyre. Adaptive local fusion systems for novelty detection and diagnostics in condition monitoring. In *Aerospace/Defense Sensing and Controls*, pages 210–218. International Society for Optics and Photonics, 1998.
- [30] Odin Taylor, John MacIntyre, and John Tait. Improved classification for a data fusing kohonen self organizing map using a dynamic thresholding technique. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI '99*, pages 828–832, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.
- [31] M.L.D. Wong, L.B. Jack, and A.K. Nandi. Modified self-organising map for automated novelty detection applied to vibration signal monitoring. *Mechanical Systems and Signal Processing*, 20(3):593 – 610, 2006.
- [32] S Zhang, R Ganesan, and GD Xistris. Self-organising neural networks for automated machinery monitoring systems. *Mechanical systems and signal processing*, 10(5):517–532, 1996.
- [33] T. Harris. A kohonen som based, machine health monitoring system which enables diagnosis of faults not seen in the training set. In *Neural Networks, 1993. IJCNN '93-Nagoya. Proceedings of 1993 International Joint Conference on*, volume 1, pages 947–950 vol.1, Oct 1993.
- [34] M. L. Dennis. A novelty detection technique for machine condition monitoring using s. o. m. *Jurnal Kejuruteraan (Journal of Engineering)*, 20(0), 2011.
- [35] Sila Kittiwachana, Diana L. S. Ferreira, Louise A. Fido, Duncan R. Thompson, Richard E. A. Escott, and Richard G. Brereton. Self-organizing map quality control index. *Analytical Chemistry*, 82(14):5972–5982, 2010. PMID: 20557073.

- [36] Hilario López García and Iván Machón González. Self-organizing map and clustering for wastewater treatment monitoring. *Engineering Applications of Artificial Intelligence*, 17(3):215 – 225, 2004.
- [37] Jing Tian, Michael H Azarian, and Michael Pecht. Anomaly detection using self-organizing maps-based k-nearest neighbor algorithm.
- [38] Yew Seng Ng and Rajagopalan Srinivasan. Multivariate temporal data analysis using self-organizing maps. 1. training methodology for effective visualization of multistate operations. *Industrial & Engineering Chemistry Research*, 47(20):7744–7757, 2008.
- [39] J. B. Yu and L. F. Xi. Using an mqe chart based on a self-organizing map nn to monitor out-of-control signals in manufacturing processes. *International Journal of Production Research*, 46(21):5907–5933, 2008.
- [40] Jian-Bo Yu and Shijin Wang. Using minimum quantization error chart for the monitoring of process states in multivariate manufacturing processes. *Computers & Industrial Engineering*, 57(4):1300 – 1312, 2009.
- [41] DavidM.J. Tax, Alexander Ypma, and RobertP.W. Duin. Pump failure detection using support vector data descriptions. In DavidJ. Hand, JoostN. Kok, and MichaelR. Berthold, editors, *Advances in Intelligent Data Analysis*, volume 1642 of *Lecture Notes in Computer Science*, pages 415–425. Springer Berlin Heidelberg, 1999.
- [42] Shijin Wang, Jianbo Yu, Edzel Lapira, and Jay Lee. A modified support vector data description based novelty detection approach for machinery components. *Applied Soft Computing*, 13(2):1193 – 1205, 2013.
- [43] David MJ Tax and Robert PW Duin. Support vector domain description. *Pattern recognition letters*, 20(11):1191–1199, 1999.
- [44] Hsin-Hao(Tom) Huang and Hsu-Pin(Ben) Wang. Machine fault classification using an art 2 neural network. *The International Journal of Advanced Manufacturing Technology*, 8(4):194–199, 1993.

- [45] M Pacella\*, Q Semeraro, and A Anglani. Adaptive resonance theory-based neural algorithms for manufacturing process quality control. *International Journal of Production Research*, 42(21):4581–4607, 2004.
- [46] T. Kohonen. Things you haven't heard about the self-organizing map. In *Neural Networks, 1993., IEEE International Conference on*, pages 1147–1156 vol.3, 1993.
- [47] Georg Pözlbauer. Survey and comparison of quality measures for self-organizing maps.
- [48] Massimo Pacella and Quirico Semeraro. Understanding art-based neural algorithms as statistical tools for manufacturing process quality control. *Eng. Appl. Artif. Intell.*, 18(6):645–662, September 2005.
- [49] Marco Grasso, Bianca Maria Colosimo, Quirico Semeraro, and Massimo Pacella. A comparison study of distribution-free multivariate spc methods for multimode data. *Quality and Reliability Engineering International*, 2014.
- [50] Melody Y. Kiang. Extending the kohonen self-organizing map networks for clustering analysis. *Comput. Stat. Data Anal.*, 38(2):161–180, January 2002.
- [51] A. Ultsch and H. P. Siemon. Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis. In *Proceedings of International Neural Networks Conference (INNC)*, pages 305–308, Paris, 1990. Kluwer Academic Press.
- [52] Juha Vesanto and Esa Alhoniemi. Clustering of the self-organizing map. *Neural Networks, IEEE Transactions on*, 11(3):586–600, 2000.
- [53] F. Murtagh. Interpreting the kohonen self-organizing feature map using contiguity-constrained clustering. *Pattern Recognition Letters*, 16(4):399 – 408, 1995.
- [54] David L Davies and Donald W Bouldin. A cluster separation measure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (2):224–227, 1979.



- [55] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.
- [56] Xuefeng Yan. Multivariate outlier detection based on self-organizing map and adaptive nonlinear map and its application. *Chemometrics and Intelligent Laboratory Systems*, 107(2):251 – 257, 2011.
- [57] Leo H. Chiang, Randy J. Pell, and Mary Beth Seasholtz. Exploring process data with the use of robust outlier detection algorithms. *Journal of Process Control*, 13(5):437 – 449, 2003.
- [58] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.
- [59] Hujun Yin and Nigel M. Allinson. On the distribution and convergence of feature space in self-organizing maps. *Neural Computation*, pages 1178–1187, 1995.
- [60] Yajuan Chen and Jeffrey B Birch. Cluster-based profile monitoring in phase i analysis.
- [61] Jean-Claude Fort, Patrick Letrémy, and Marie Cottrell. Advantages and drawbacks of the batch kohonen algorithm. In Michel Verleysen, editor, *ESANN*, pages 223–230, 2002.
- [62] Leonard Kaufman and Peter J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley-Interscience, 9th edition, March 1990.

# Appendice A

## Risultati Simulazioni

Scenario	Severità	$K = 1$	$K = 2$	$K = 3$	$K = 4$
1	1	499	1	0	0
1	2	466	34	0	0
1	3	8	492	0	0
1	4	0	500	0	0
2	1	496	4	0	0
2	2	496	4	0	0
2	3	493	7	0	0
2	4	492	4	4	0
3	1	480	20	0	0
3	2	302	184	14	0
3	3	4	443	53	0
3	4	0	500	0	0

Tabella A.1: Tabella K scelti automaticamente