UNIVERSITÀ DEGLI STUDI DI UDINE

COMUNICAZIONE MULTIMEDIALE E TECNOLOGIE DELL'INFORMAZIONE

Tecniche di Deep Learning per il rilevamento di anomalie nelle immagini

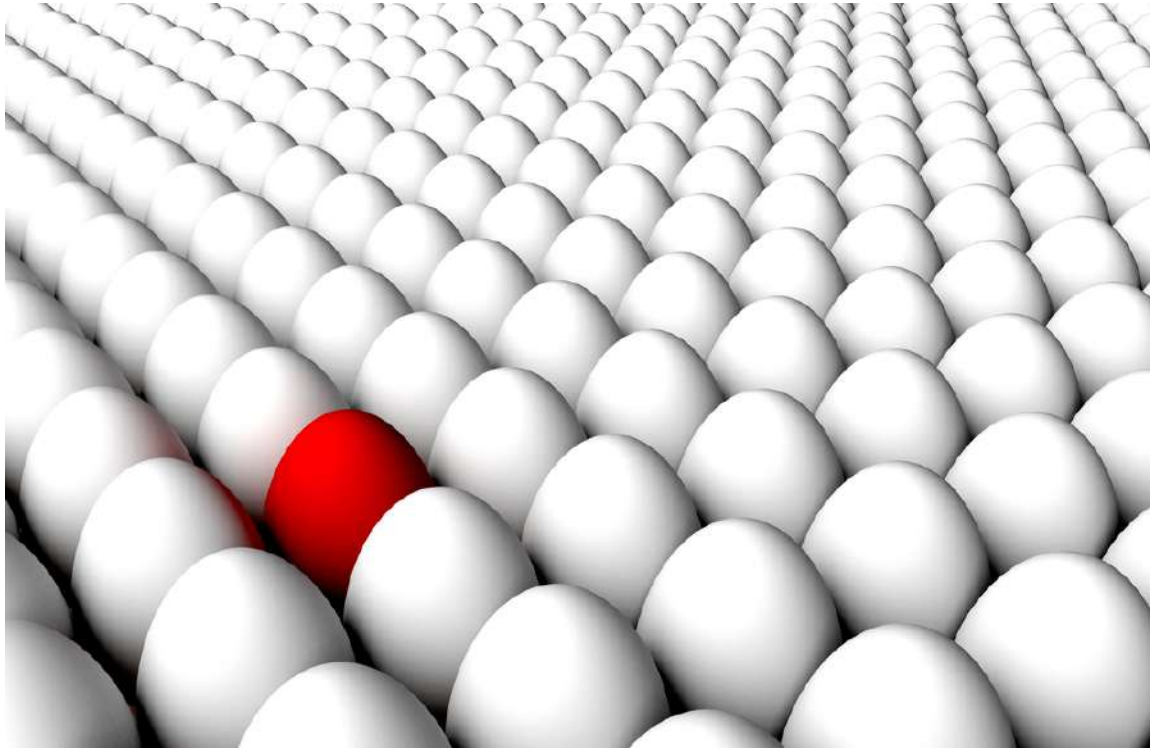
Relatore: Prof. Gian Luca Foresti

Laureando: Riccardo Verk

Anno accademico 2018/19

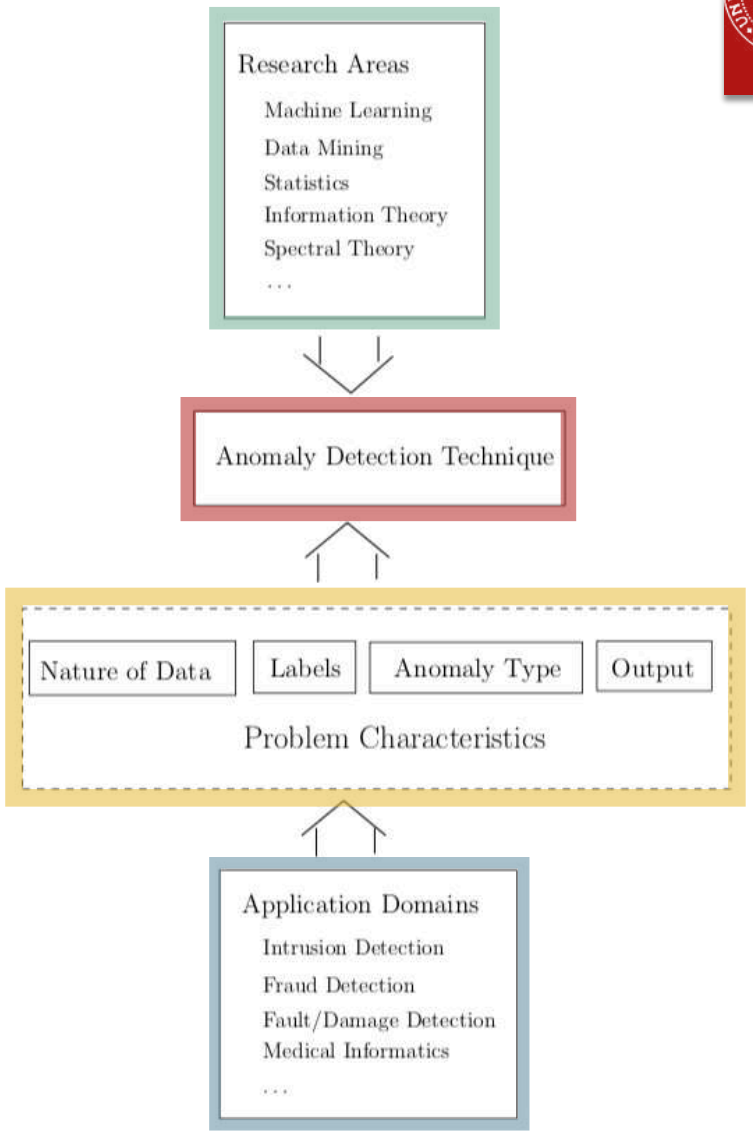
Definizione del problema

- Rilevamento di possibili anomalie in immagini attraverso tecniche di Deep Learning



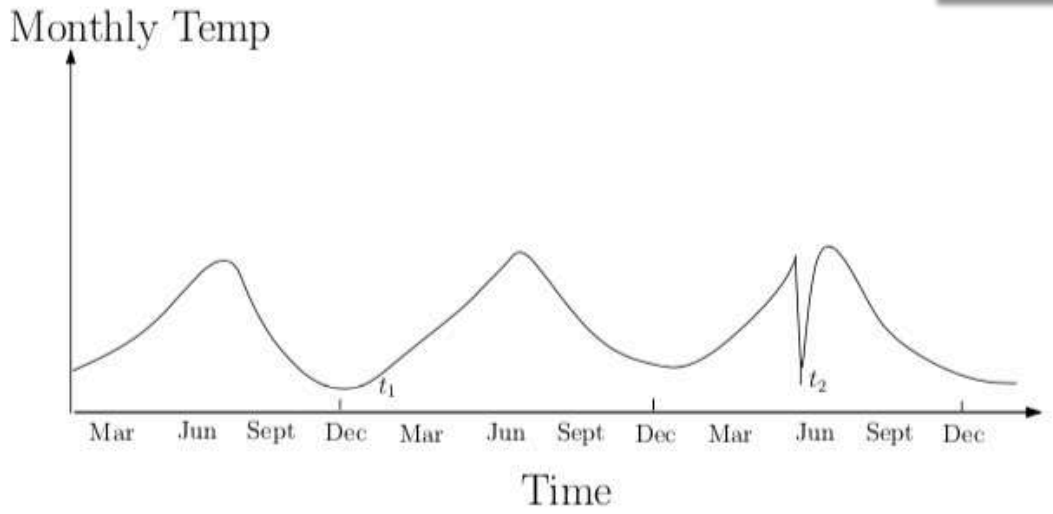
Anomaly Detection

- È una tecnica utilizzata per l'identificazione di elementi, eventi o osservazioni anomale che si differenziano in modo significativo dalla maggior parte dei dati

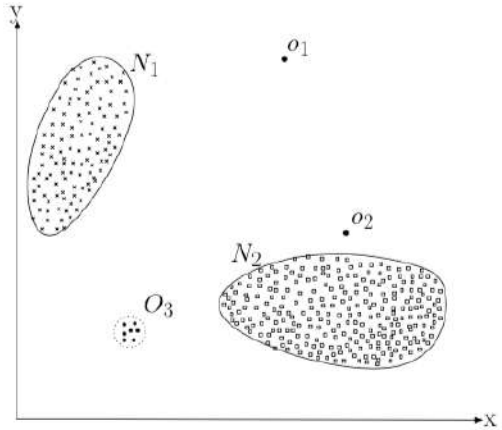


Tipi di anomalia

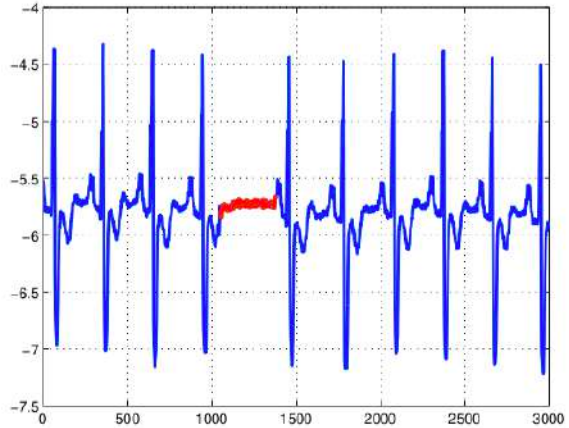
- **Anomalie puntiformi:** una singola istanza di dati è considerata anomala rispetto al resto dei dati
- **Anomalie contestuali:** un'istanza di dati è anomala in un contesto specifico (ma non altrove)
- **Anomalie collettive:** una raccolta di istanze di dati correlati è anomala rispetto all'intero set di dati



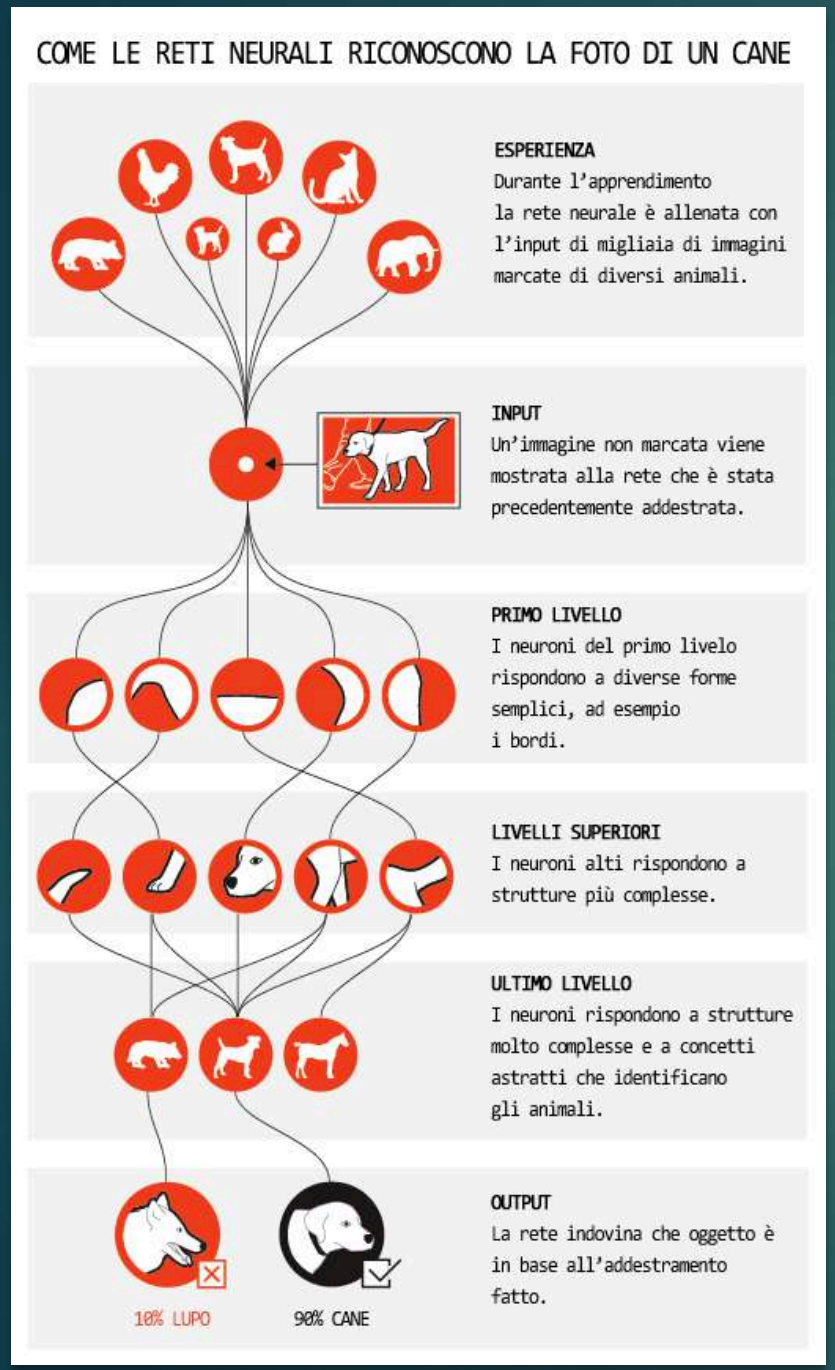
Anomalie contestuali



Anomalie puntiformi



Anomalie collettive

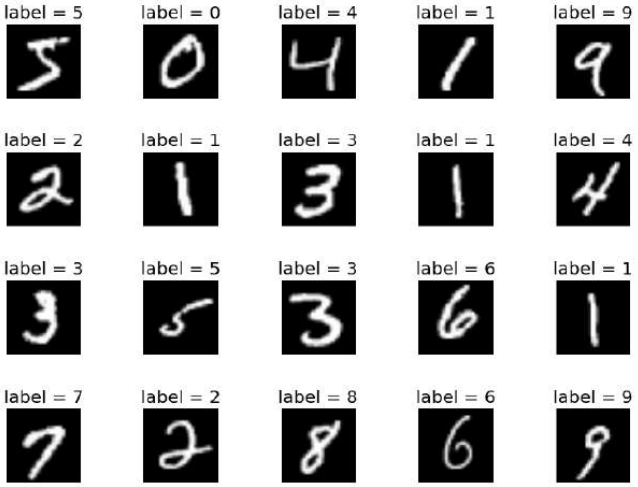


Il Deep Learning

- È definito come una classe di algoritmi di apprendimento automatico
- Usa vari livelli di unità non lineari a cascata per svolgere compiti di estrazione di caratteristiche e di trasformazione
- Le caratteristiche di livello più alto vengono derivate da quelle di livello più basso per creare una rappresentazione gerarchica
- Apprendono più livelli di rappresentazione che corrispondono a differenti livelli di astrazione
- L'output di una rete è di solito una label che indovina l'oggetto di input

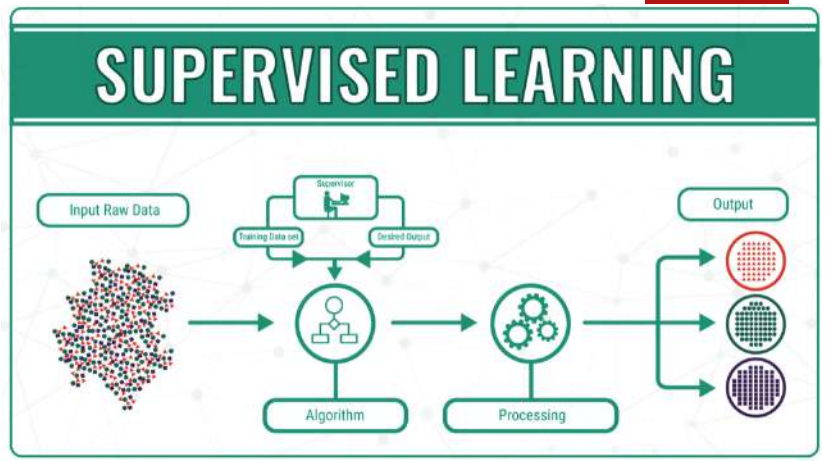
Label dei dati nel caso dell'Anomaly Detection

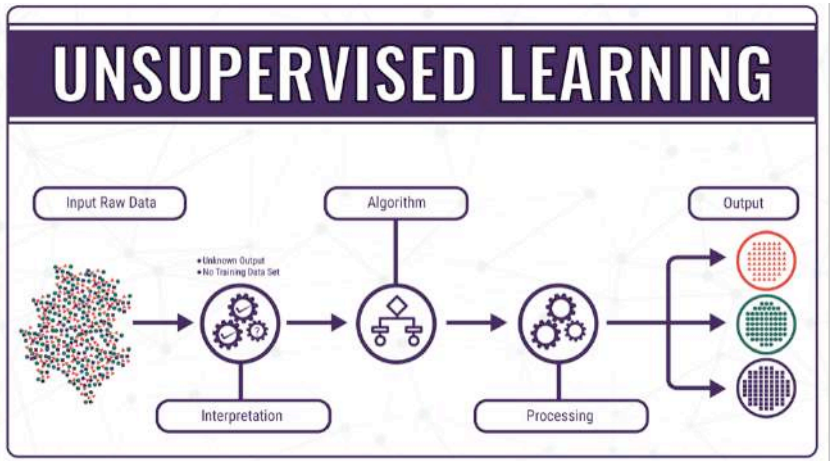
- Le label associate ad un'istanza di dati indicano se tale istanza è normale o anomala
- **Problemi:** ottenere le label dei dati che siano accurate e rappresentative di tutti i tipi di comportamento è spesso proibitivo
- Il labeling è spesso eseguito **manualmente** da esperti umani e richiede quindi un notevole sforzo per ottenere i dati di training etichettati

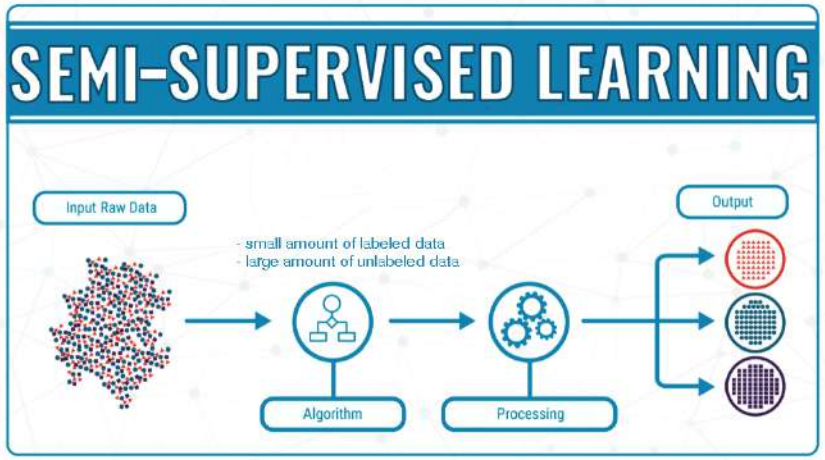


Apprendimento di una rete di Deep Learning

- In base alla disponibilità delle label, le tecniche di rilevamento delle anomalie possono operare in una delle tre modalità seguenti:
  1. Supervised anomaly detection
  2. Unsupervised anomaly detection
  3. Semi-Supervised anomaly detection



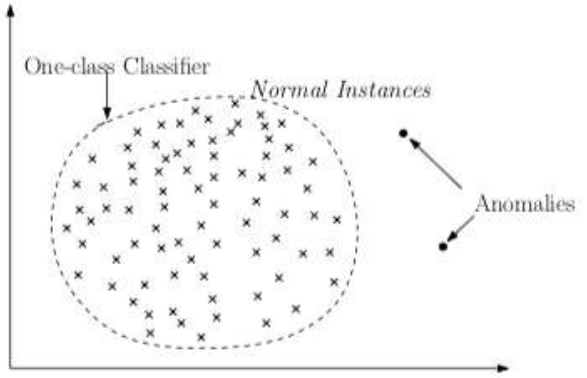




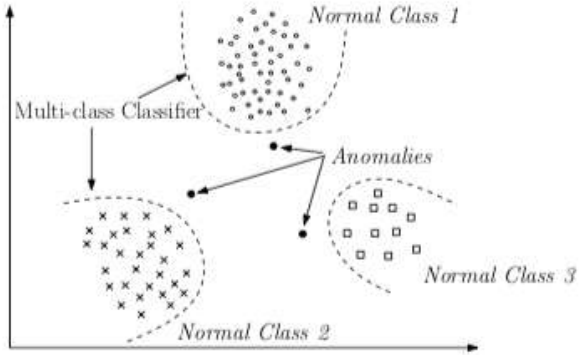
Tecniche statistiche per l'Anomaly Detection

Tecniche di rilevazione delle anomalie basate sulla classificazione (Approccio supervisionato)

- La classificazione è usata per addestrare un modello (classificatore) da un insieme di istanze di dati etichettati (training) e poi, classificare un'istanza di test in una delle classi usando il modello appreso (testing)
- Possono essere raggruppate in due grandi categorie:
  - One-class anomaly detection
  - Multi-class anomaly detection

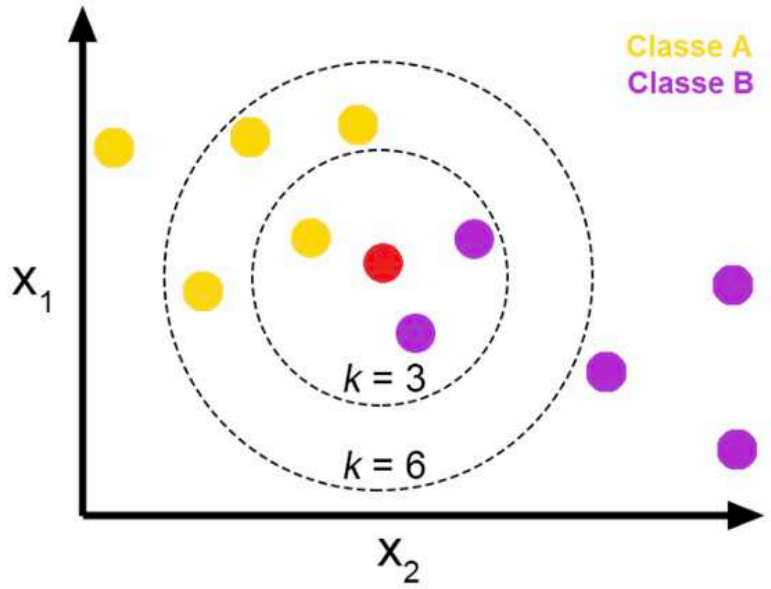


(b) One-class Anomaly Detection



(a) Multi-class Anomaly Detection

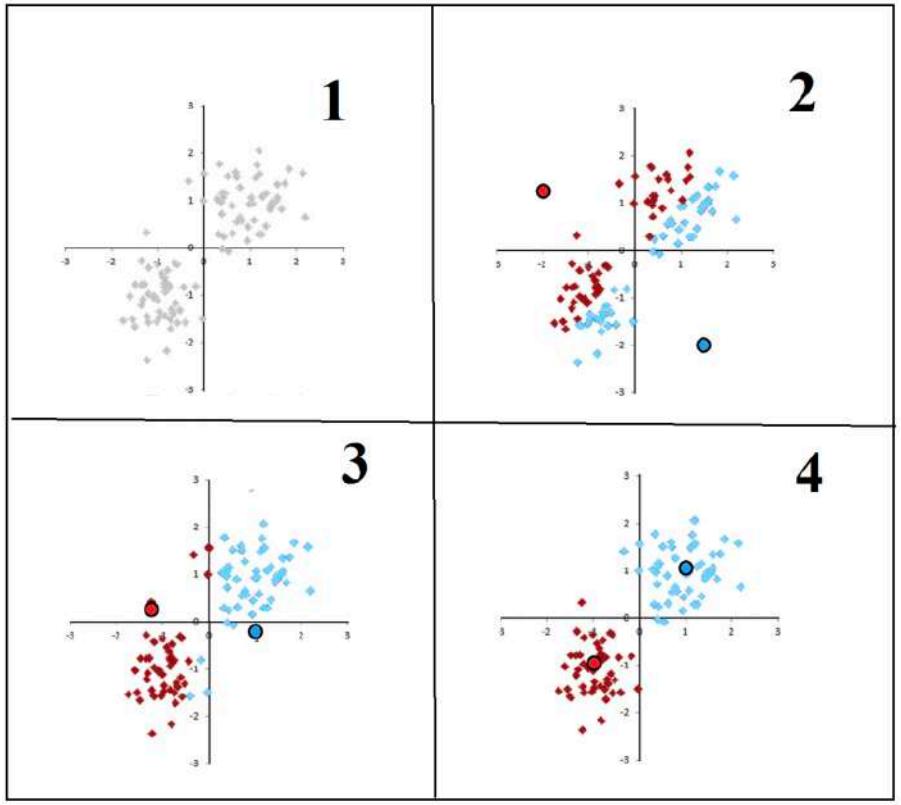
Tecniche di rilevamento delle anomalie basate sulla densità (Approccio non supervisionato)



- Il rilevamento delle anomalie basato sulla densità si basa sull'algoritmo KNN (K-Nearest Neighbors)
- I dati normali si presentano in un insieme denso, mentre le anomalie sono lontane
- L'insieme più vicino di dati viene valutato utilizzando un punteggio, che si basa su una distanza (es. euclinese)

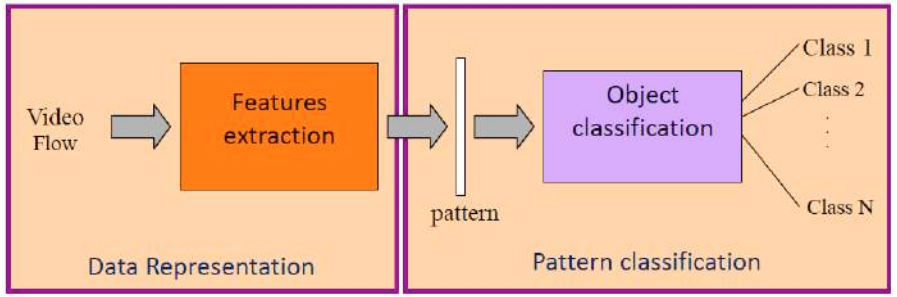
Tecniche di rilevamento delle anomalie basate sul clustering (Approccio non supervisionato)

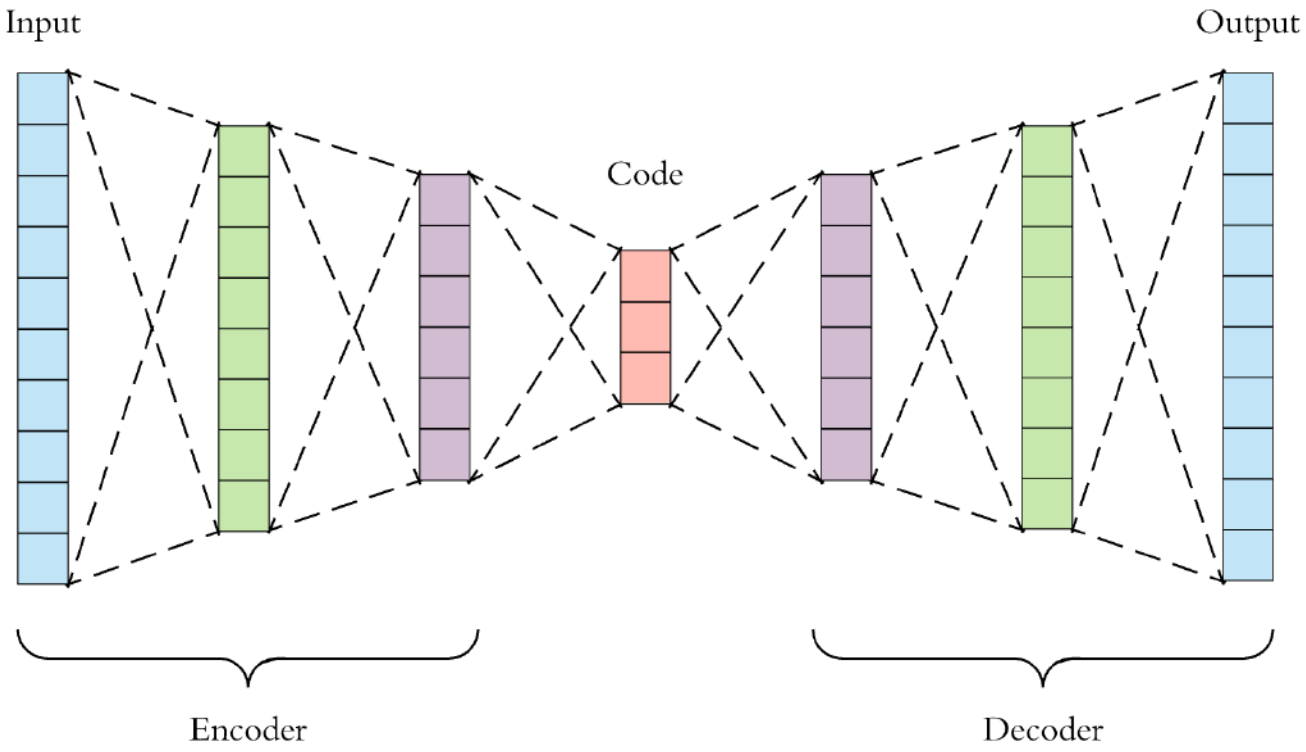
- Il **clustering** è uno dei concetti più popolari nel campo dell'apprendimento non supervisionato
- I dati simili in uno spazio di n dimensioni tendono ad appartenere a gruppi simili, in base alla media aritmetica delle loro posizioni (centroide)
- Il clustering è usato per raggruppare istanze di dati simili in gruppi
- Un algoritmo ampiamente usato è il **K-mean**:
  - Crea 'k' gruppi simili di dati in uno spazio
  - Le istanze di dati che non rientrano in questi gruppi potrebbero essere potenzialmente contrassegnate come anomalie.



Problemi aperti

- Le tecniche viste in precedenza si basano su metodi statistici, e quindi hanno bisogno di parametri pre-impostati per poter addestrare il modello ad estrarre determinate feature
- La soluzione proposta invece, sfrutta un tipo di rete neurale e quindi l'estrazione delle feature avviene in modo automatico
- La tecnica proposta mostra quindi i suoi meriti quando i dati sono complessi e di natura non lineare

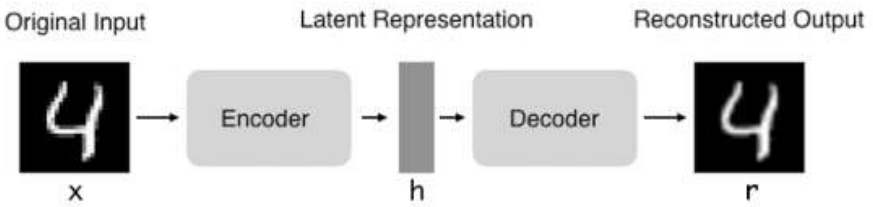




Soluzione proposta: Gli Autoencoder

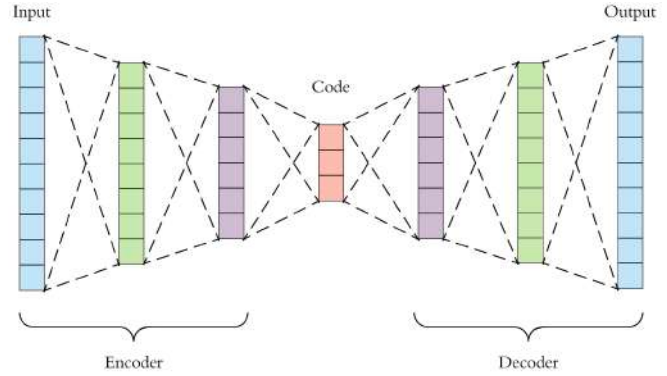
Autoencoder

- Sono reti neurali con lo scopo di generare nuovi dati dapprima comprimendoli e, successivamente, ricostruendo l'output sulla base delle informazioni acquisite
- L'obiettivo dell'Autoencoder è quello di ottenere un apprendimento delle caratteristiche utili per la ricostruzione dell'input

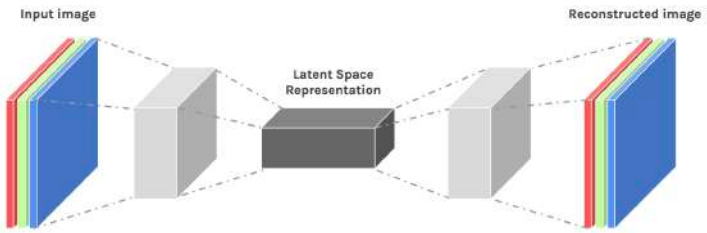


Differenti tipologie di Autoencoder

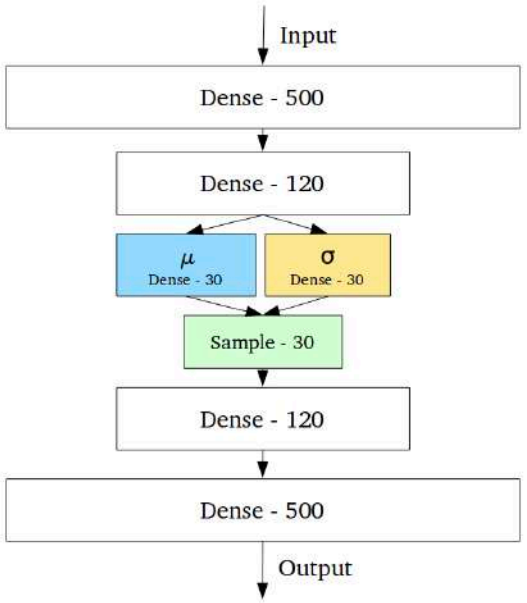
- **Basic Autoencoder:** (o multilayer) la forma più semplice
- **Convolutional Autoencoder:** al posto di vettori unidimensionali, vengono utilizzati vettori tridimensionali
- **Variational Autoencoder:** sono il risultato della combinazione di Deep Learning e inferenza bayesiana, nel senso che sono costituiti da una rete neurale allenata con l'algoritmo di backpropagation modificato con una tecnica chiamata riparametrizzazione



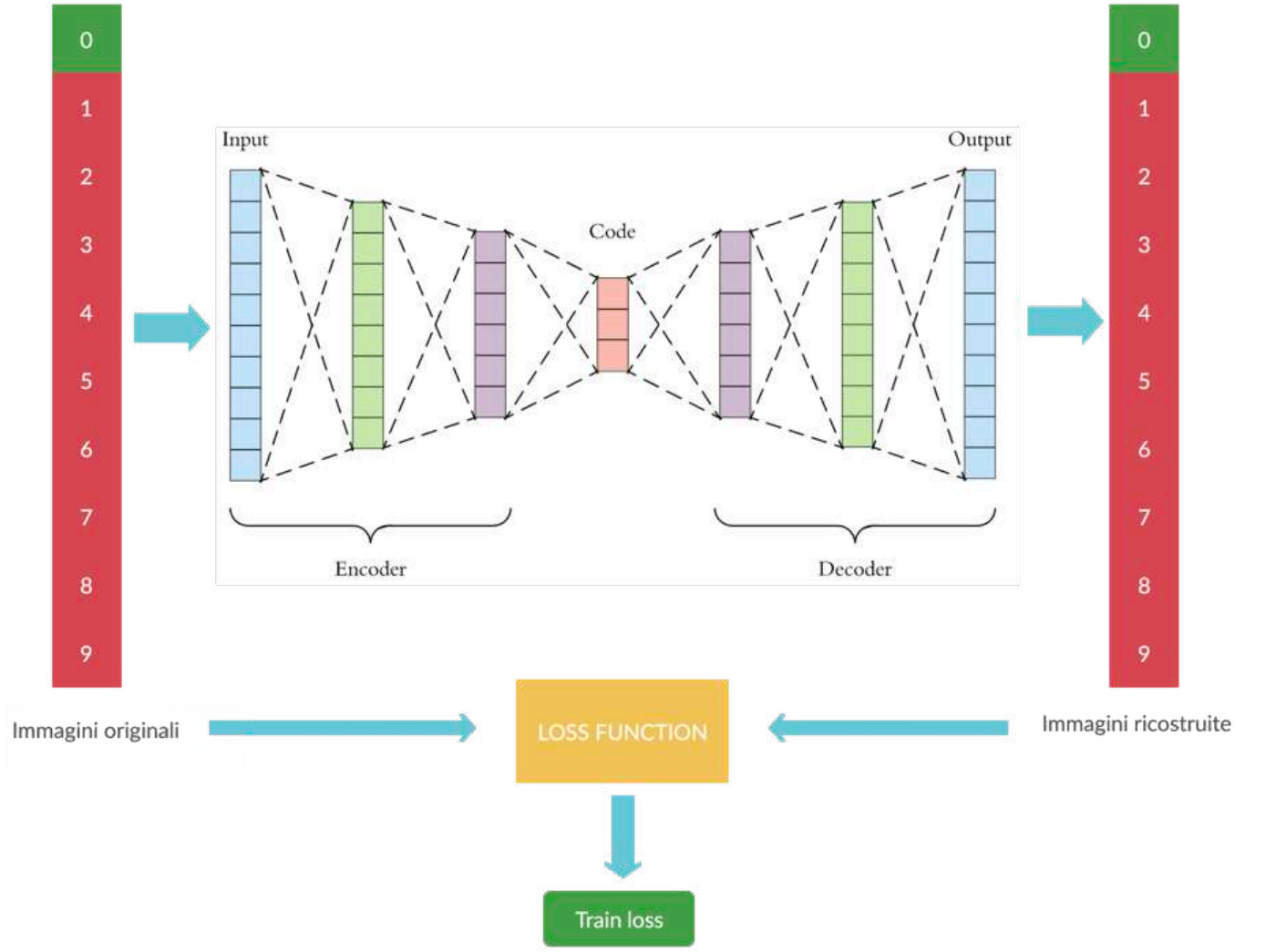
Basic Autoencoder



Convolutional Autoencoder



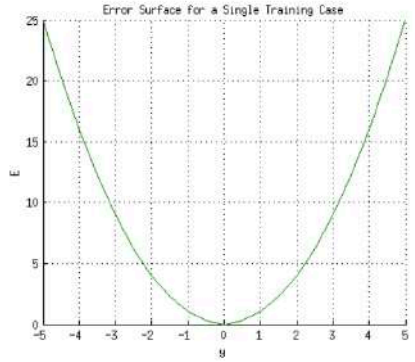
Variational Autoencoder

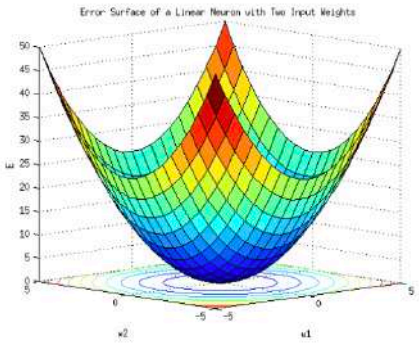

0
1
2
3
4
5
6
7
8
9
Input
Output
Code
Encoder
Decoder
0
1
2
3
4
5
6
7
8
9
Immagini originali
LOSS FUNCTION
Immagini ricostruite
Train loss

Training

Training di un Autoencoder

- L'apprendimento avviene tramite la minimizzazione della loss function
- È un metodo per valutare il modo in cui uno specifico algoritmo modella i dati forniti
- Se le previsioni si discostano troppo dai risultati effettivi, la funzione di perdita potrebbe produrre un numero elevato
- Con l'aiuto di alcune funzioni di ottimizzazione, la funzione di perdita impara un po' alla volta a ridurre l'errore nella previsione





Loss function prese in considerazione

- **MSE:** viene misurato come la media della differenza quadrata tra previsioni e osservazioni effettive

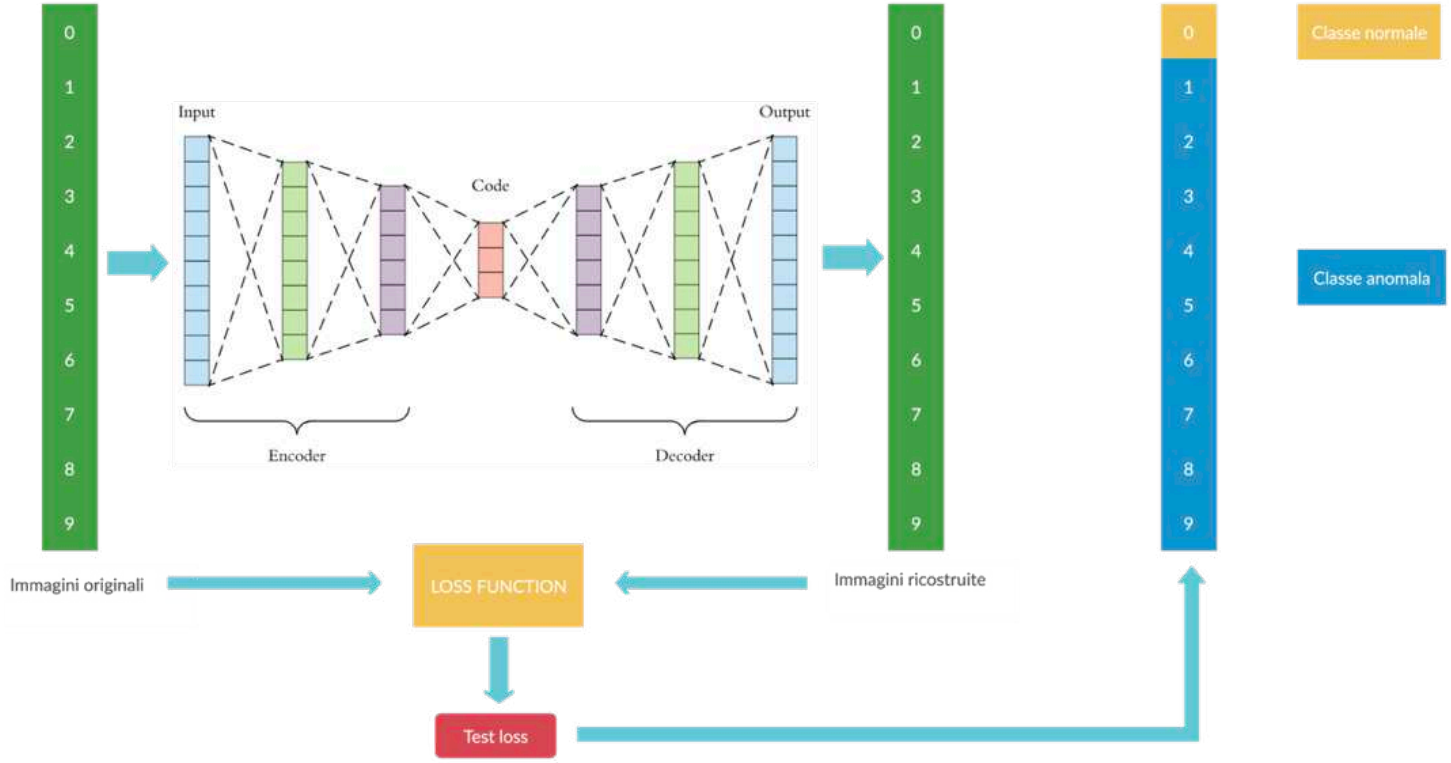
$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

- **BCE:** la perdita aumenta quando la probabilità prevista differisce dall'etichetta effettiva

$$BCE = -\frac{1}{N}\sum_{i=0}^{N} y_i \cdot log(\hat{y}_i) + (1 - y_i) \cdot log(1 - \hat{y}_i)$$

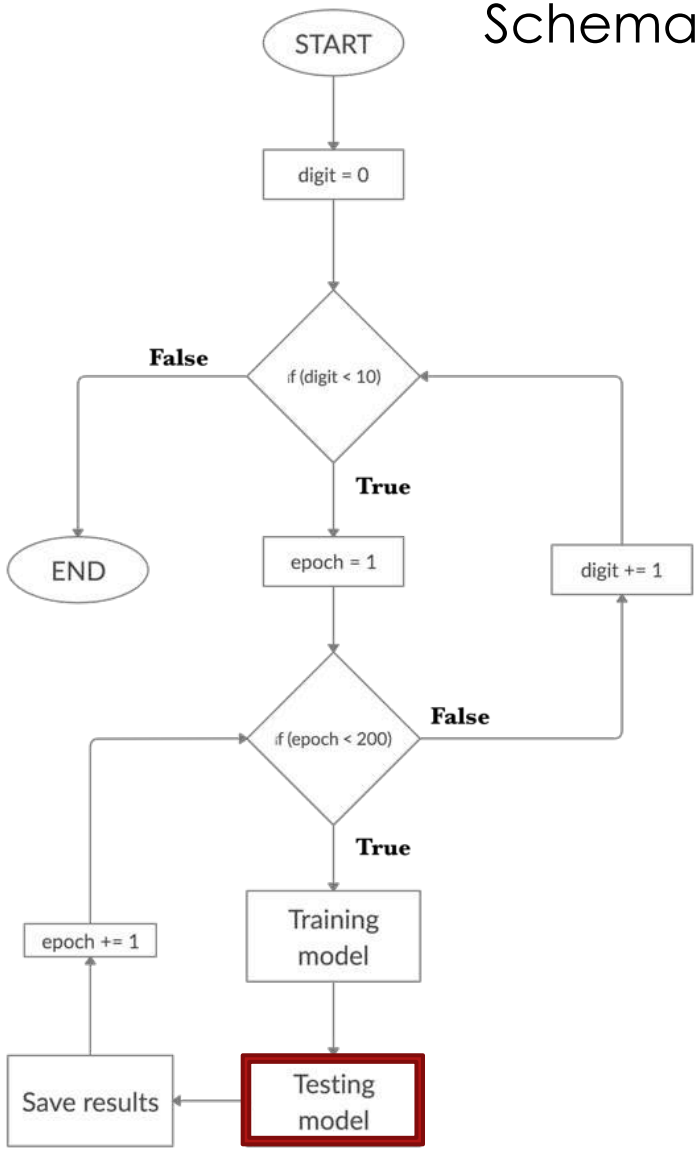
- **SSIM:** l'indice di similarità strutturale è un metodo per misurare la somiglianza tra due immagini

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

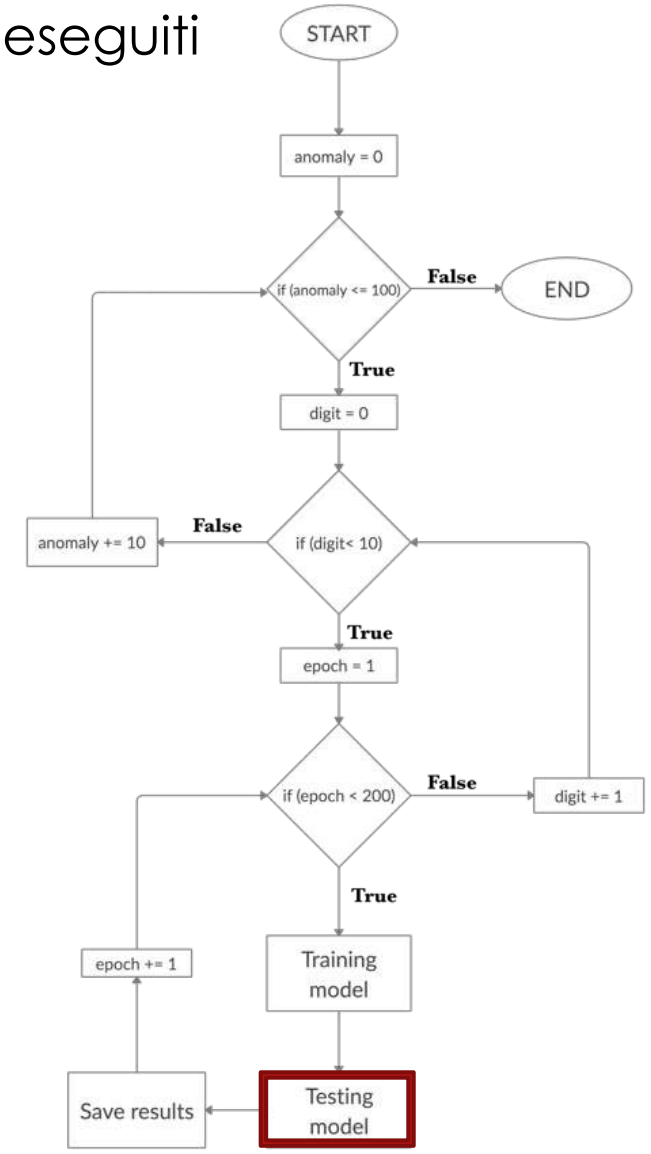

0
1
2
3
4
5
6
7
8
9
Input
Output
Code
Encoder
Decoder
0
1
2
3
4
5
6
7
8
9
0
1
2
3
4
5
6
7
8
9
Classe normale
Classe anomala
Immagini originali
LOSS FUNCTION
Immagini ricostruite
Test loss

Testing

Schema completo dei test eseguiti



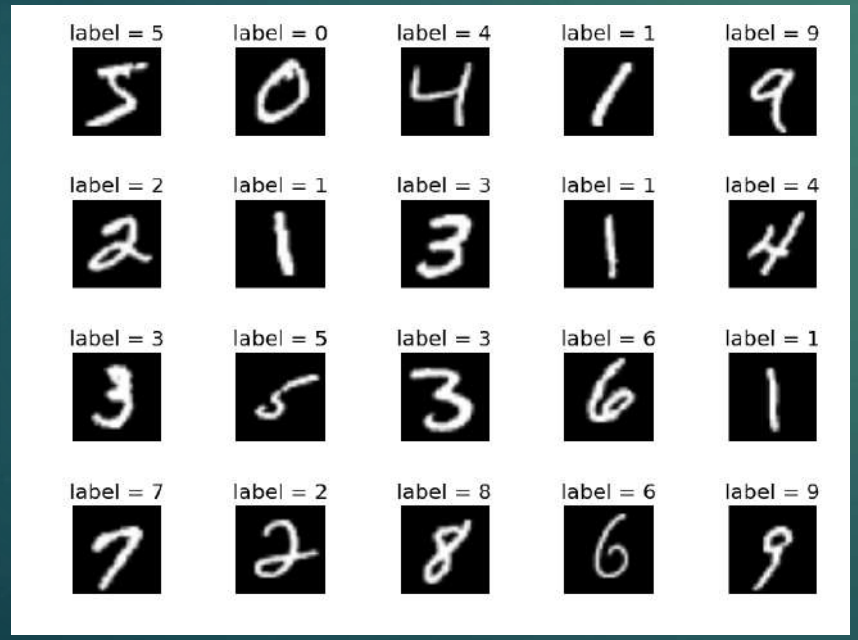
Approccio supervisionato



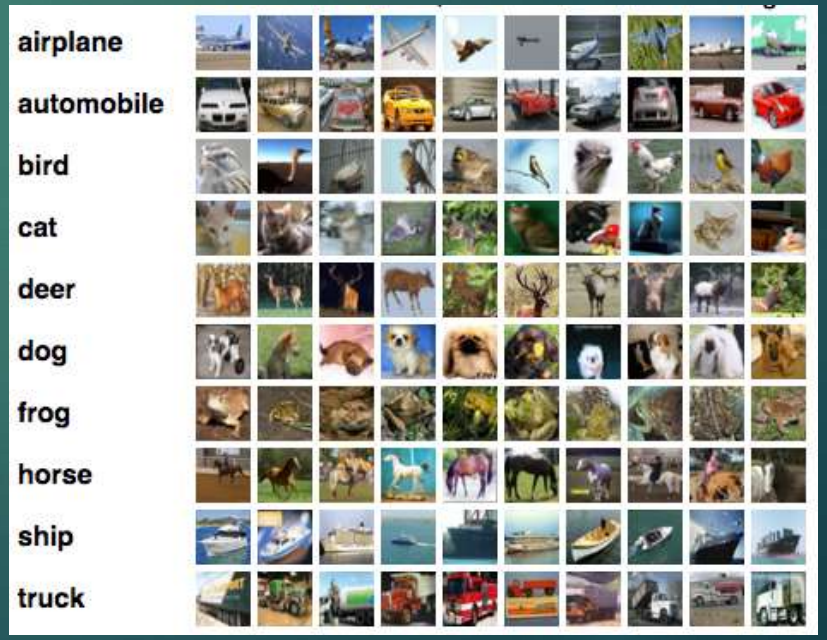
Approccio non supervisionato

MNIST VS CIFAR10

- Training set: 60,000 immagini
- Testing set: 10,000 immagini
- 28x28 in scala di grigi
- Classi: 10

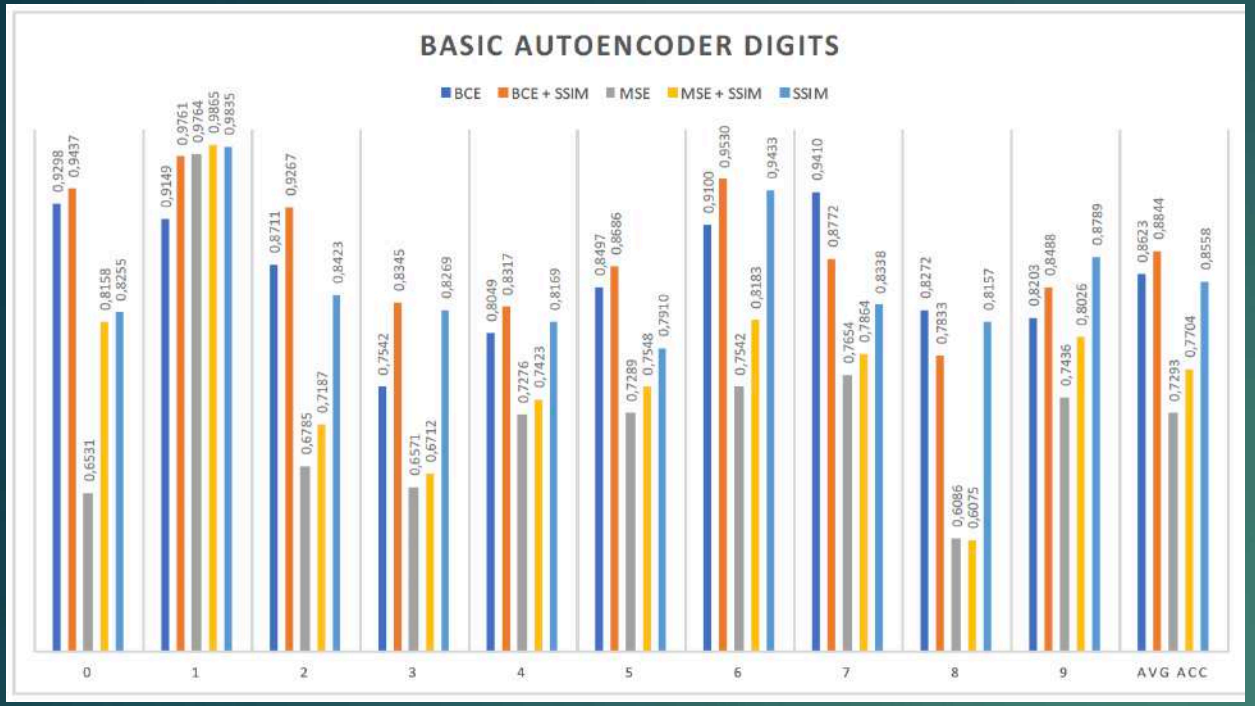


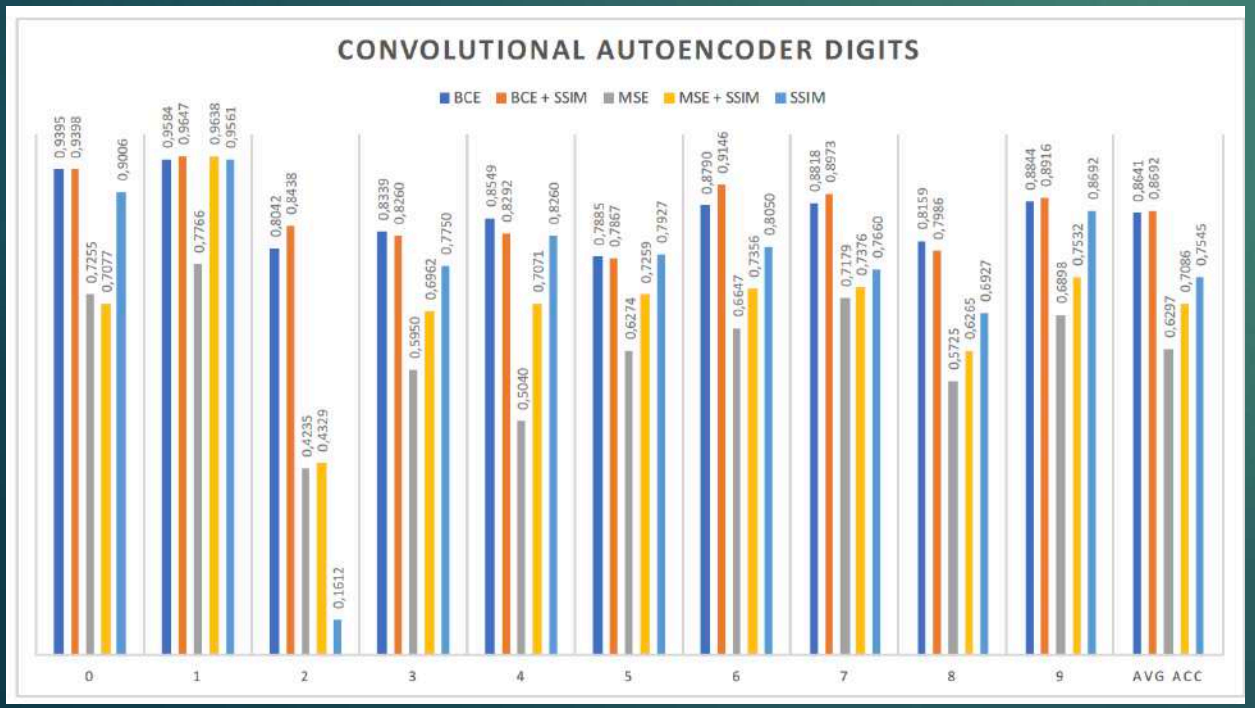
- Training set: 50,000 immagini
- Testing set: 10,000 immagini
- 32x32 a colori (3 canali RGB)
- Classi: 10

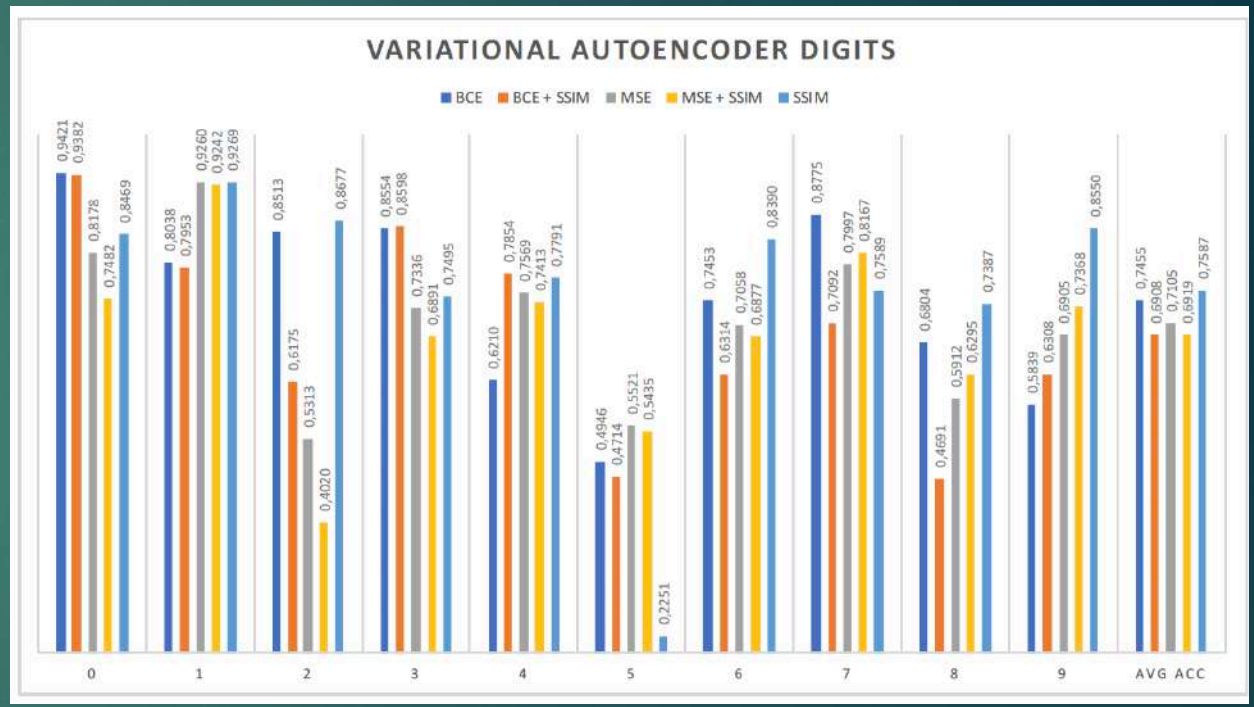


Risultati: approccio supervisionato

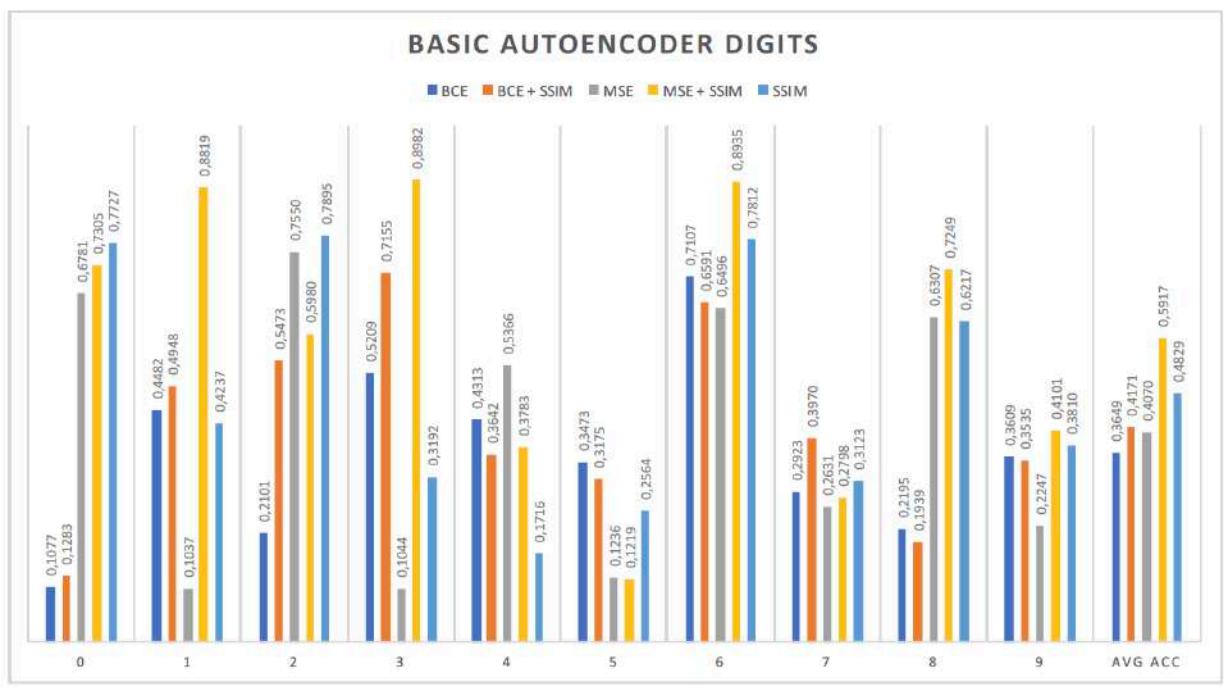
DATASET MNIST

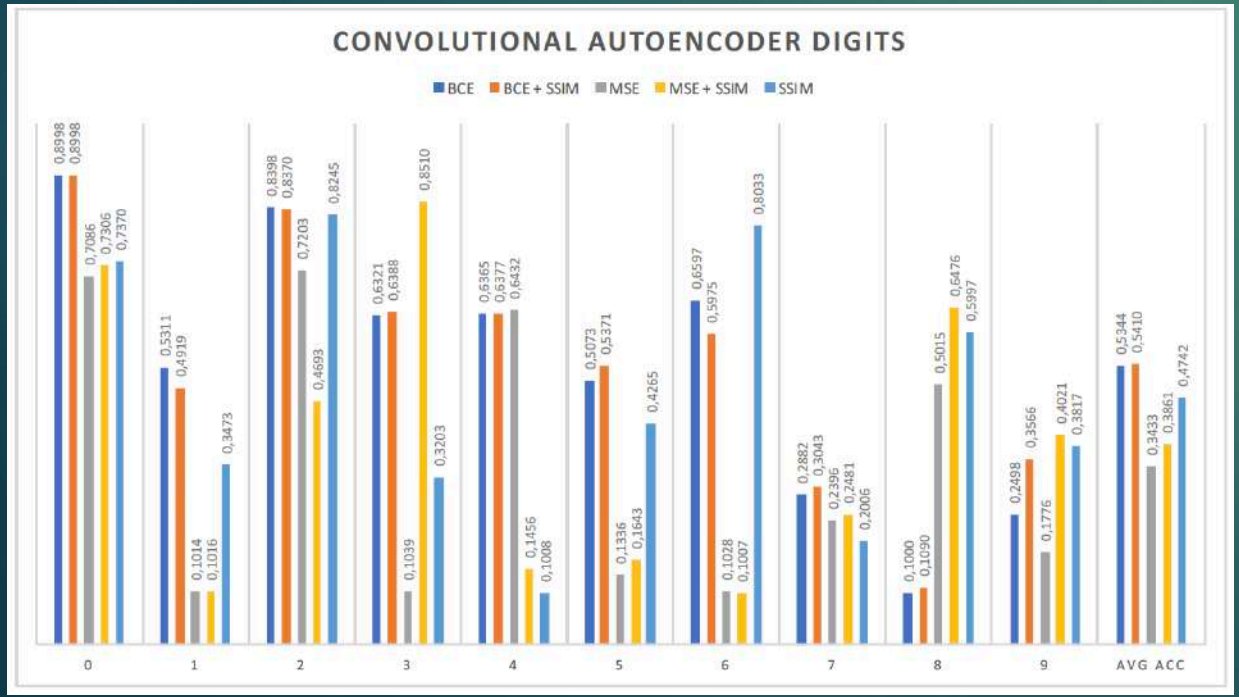


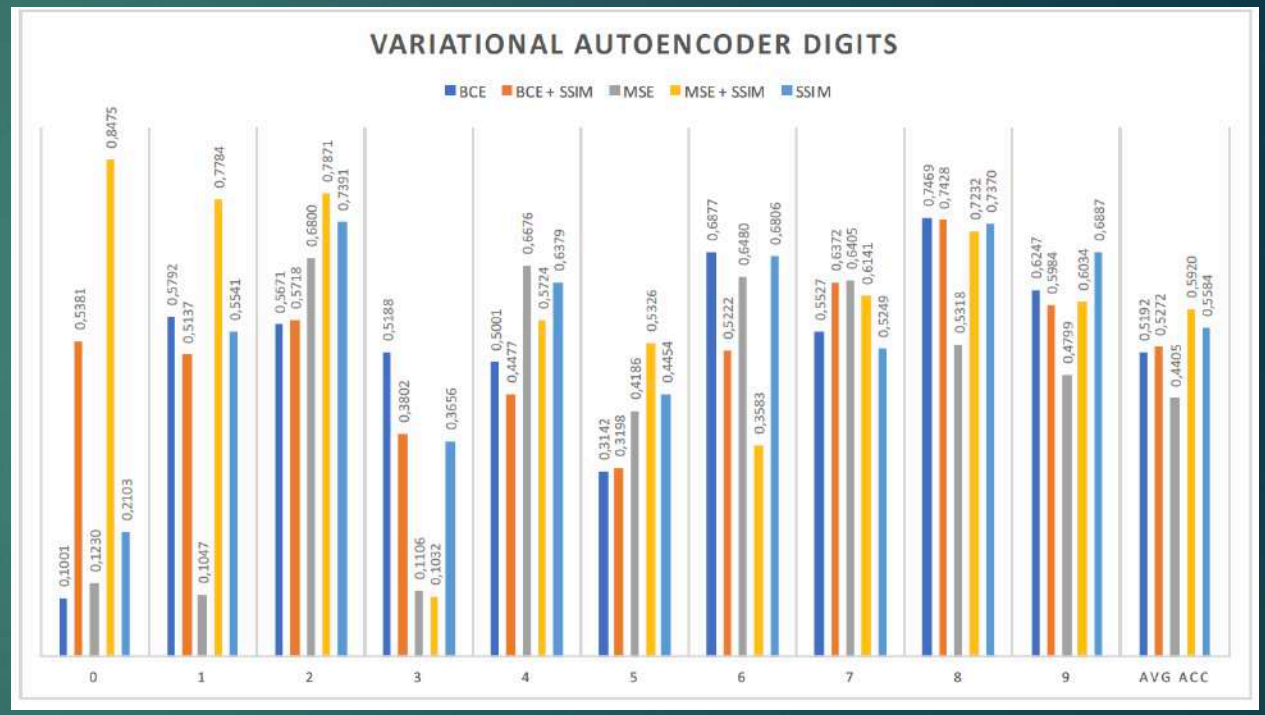




DATASET CIFAR10

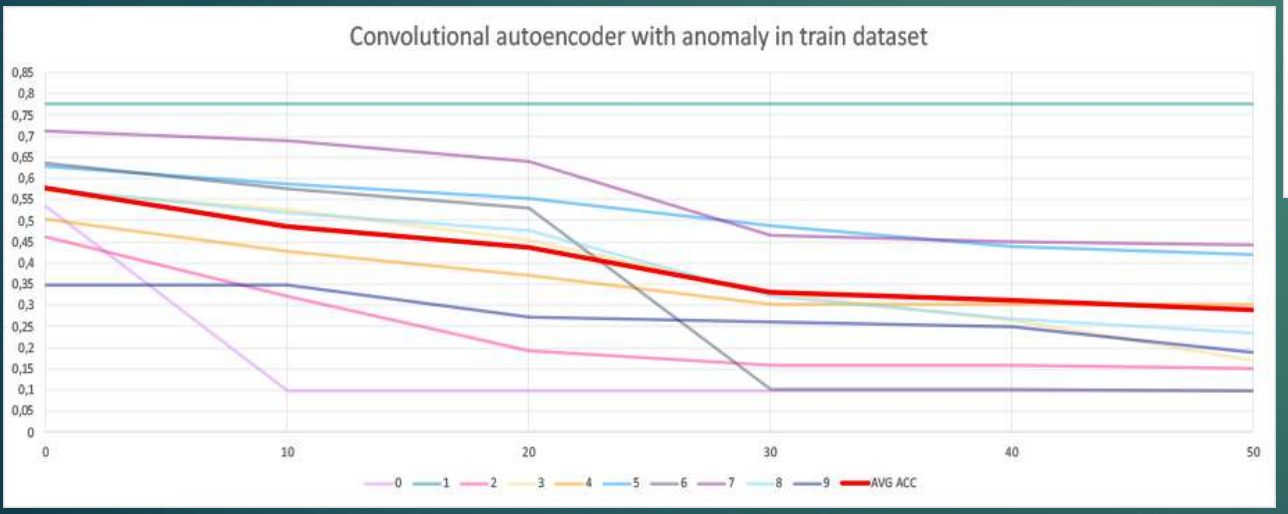


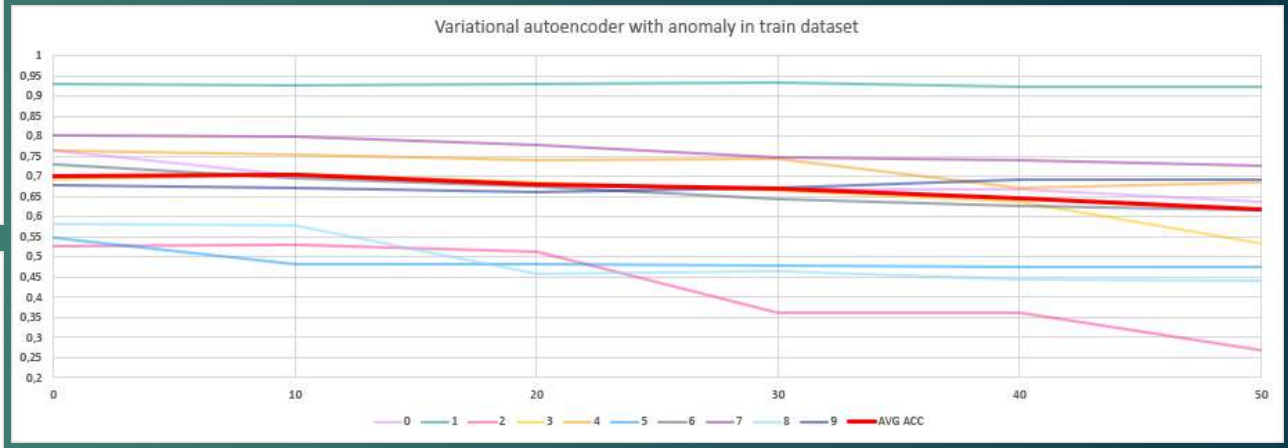




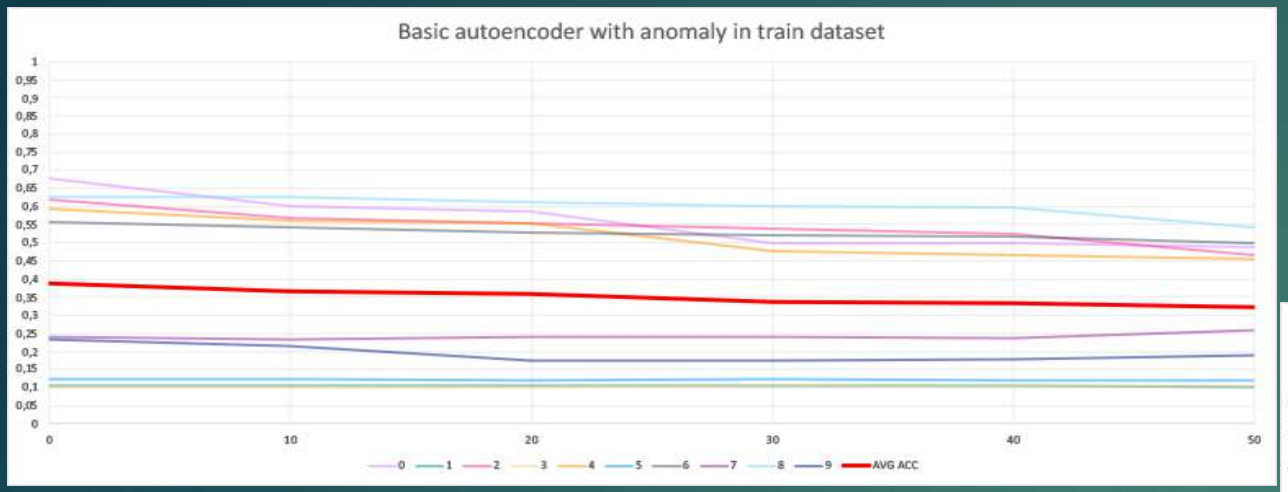
Risultati: approccio non supervisionato

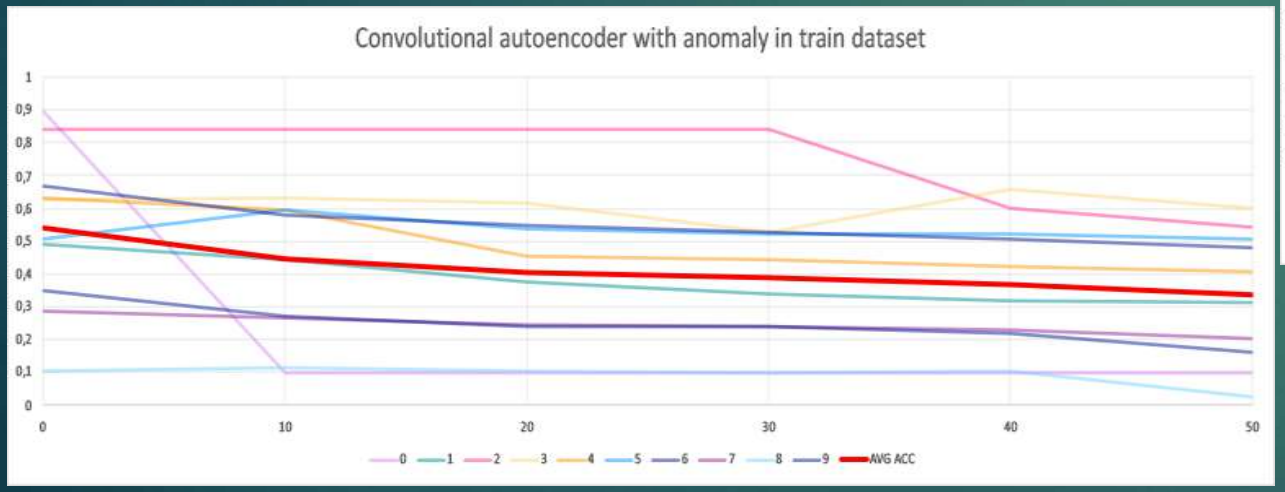
DATASET MNIST

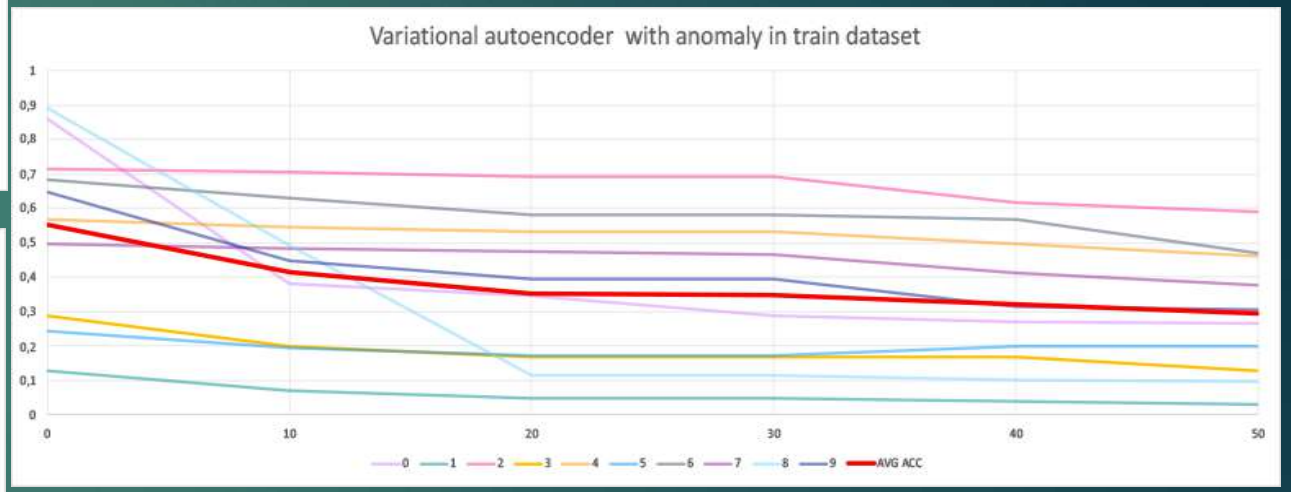




DATASET CIFAR10





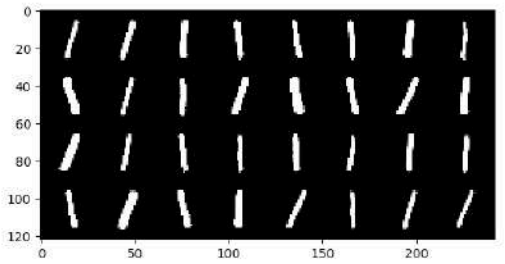


Confronto dei migliori risultati ottenuti

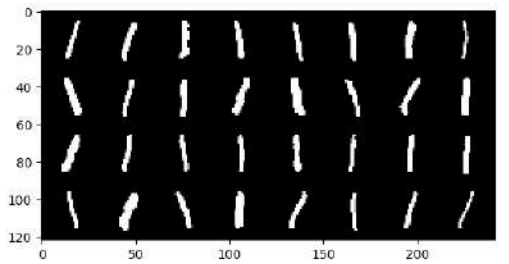
DATASET MNIST

DIGIT	AUTOENCODER	LOSS FUNCTION	ACCURACY	AUC
0	BASIC	BCE + SSIM	0,9437	0,9775
1	BASIC	MSE + SSIM	**0,9865**	**0.9975**
2	BASIC	BCE + SSIM	0,9267	0,9749
3	VAE	BCE + SSIM	0,8598	0,9103
4	CONVOLUTIONAL	BCE	0,8549	0,9348
5	BASIC	BCE + SSIM	0,8686	0,9176
6	BASIC	BCE + SSIM	0,9530	0,9839
7	BASIC	BCE	0,9410	0,9745
8	BASIC	BCE	0,8272	0,8802
9	CONVOLUTIONAL	BCE + SSIM	0,8916	0,9535
AVG	BASIC	BCE + SSIM	0,8844	0,9346

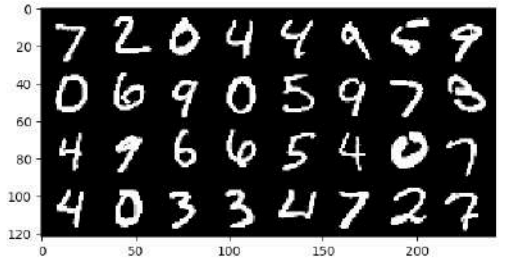
	MNIST									
Model	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
GANomaly [4]	0.881	0.675	0.953	0.801	0.827	0.864	0.849	0.682	0.856	0.558
AnoGAN [3]	0.623	0.31	0.521	0.458	0.442	0.431	0.492	0.401	0.392	0.368
EGBAD [23]	0.783	0.294	0.523	0.506	0.453	0.436	0.593	0.398	0.523	0.358
DenseNet-169	**0.998265**	0.994258	**0.984126**	0.980750	0.983918	0.992295	**0.984011**	0.997476	0.991551	**0.999386**
ResNet-152	0.998050	0.994176	0.982025	**0.981253**	0.984338	0.989994	0.980970	**0.998940**	0.989815	0.998982
Inception-V4	0.997676	**0.994609**	0.983431	0.980548	**0.984617**	**0.992676**	0.983624	0.997108	**0.994305**	0.999080



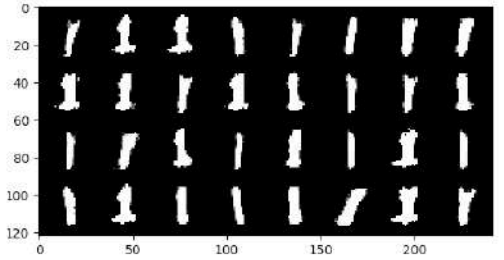
classe normale originale



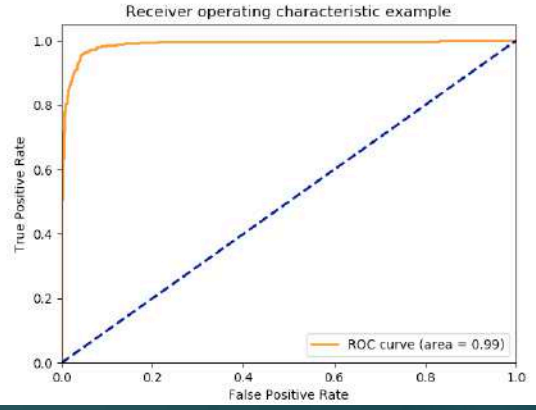
classe normale ricostruita

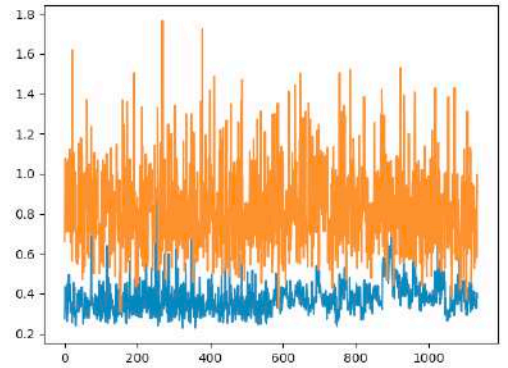


classe anomala originale



classe anomala ricostruita

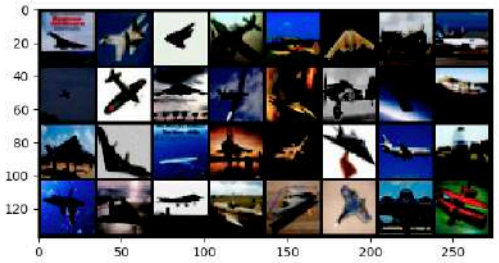




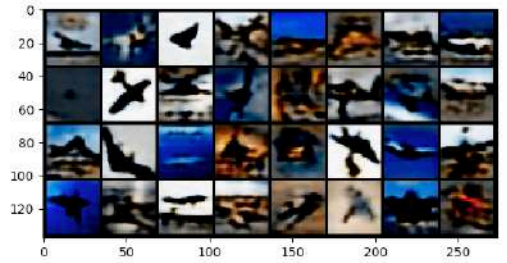
DATASET CIFAR10

CLASS	AUTOENCODER	LOSS FUNCTION	ACCURACY	AUC
0	CONVOLUTIONAL	MSE + SSIM	**0,7306**	**0,7714**
1	CONVOLUTIONAL	BCE + SSIM	0,4919	0,6093
2	BASIC	MSE	0,7550	0,6116
3	VAE	BCE	0,5188	0,5808
4	VAE	MSE	0,6676	0,7311
5	VAE	MSE + SSIM	0,5326	0,5893
6	VAE	BCE	0,6877	0,7069
7	VAE	BCE	0,5527	0,6104
8	VAE	SSIM	**0,7370**	**0,7757**
9	VAE	BCE + SSIM	0,5984	0,6566
AVG	VAE	MSE + SSIM	0,5920	0,5767

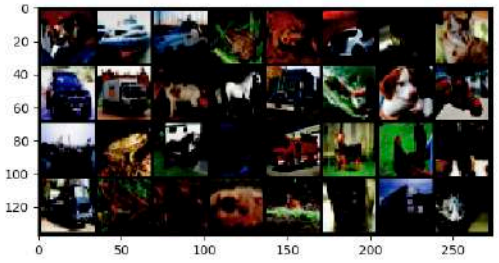
	CIFAR10									
Model	plane	car	bird	cat	deer	frog	horse	ship	truck	dog
GANomaly [4]	0.633	0.631	0.51	0.587	0.593	0.683	0.605	0.616	0.617	0.628
AnoGAN [3]	0.516	0.492	0.411	0.399	0.335	0.321	0.399	0.567	0.511	0.393
EGBAD [23]	0.577	0.514	0.383	0.448	0.374	0.353	0.526	0.413	0.555	0.481
DenseNet-169	**0.998449**	**0.998933**	0.994980	**0.992014**	0.998145	**0.991758**	0.999031	0.998386	**0.998948**	**0.998291**
ResNet-152	0.998071	0.998203	**0.995249**	0.991605	**0.998480**	0.991375	**0.999607**	**0.999289**	0.998934	0.997900
Inception-V4	0.930263	0.971474	0.842340	0.853591	0.895042	0.893674	0.949273	0.921899	0.954804	0.931945



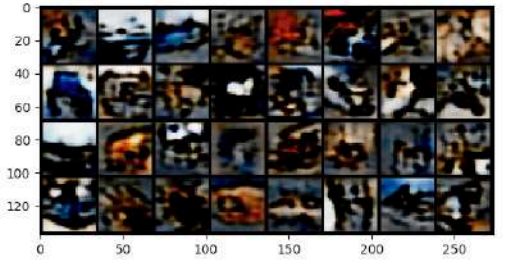
classe normale originale



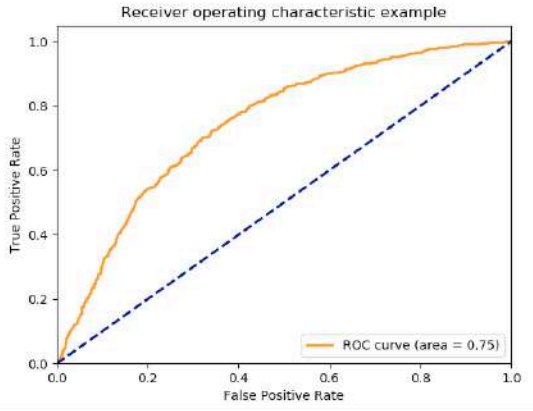
classe normale ricostruita

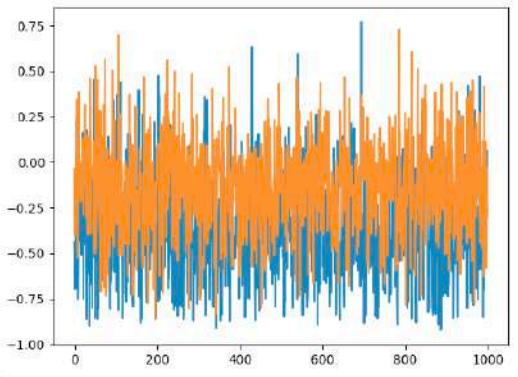


classe anomala originale



classe anomala ricostruita





Autoencoder vs DenseNet

- Gli Autoencoder dei test eseguiti hanno un'architettura con 4-5 livelli, contro i 169 della DenseNet
- Gli Autoencoder riducono la dimensionalità dei dati, guadagnando così spazio in memoria
- Inoltre, gli Autoencoder a differenza delle classiche reti neurali, hanno il vantaggio di poter separare l'Encoder dal Decoder

Conclusione e futuri sviluppi

- In conclusione, possiamo dire che gli Autoencoder possono essere utilizzati nella risoluzione di problemi dell'Anomaly Detection con ottimi risultati
- In futuro, per ottenere dei risultati ancora più significativi si potrebbe puntare su:
  - Aumento del numero di livelli della rete
  - Perfezionamento dei parametri:
    - Numero di epoche (ampliare lo studio già fatto in questa tesi)
    - Learning rate
    - Loss function (ampliare lo studio già fatto in questa tesi)