

# Quantum Topological Graph Neural Networks per il Rilevamento di Anomalie Finanziarie Analisi, Implementazione e Valutazione su Dispositivi NISQ

Martino Papa

Università degli Studi di Firenze

26 gennaio 2026

# Introduzione

# Rilevamento di frodi finanziarie

---

Le frodi finanziarie moderne sono difficili da identificare, caratterizzate da:

- **Transazioni complesse:** Schemi di riciclaggio che coinvolgono decine di conti.
- **Pattern nascosti:** Anomalie topologiche invisibili alle analisi statistiche standard.

Lo standard attuale è rappresentato dalle **Graph Neural Network (GNN)**, modelli di deep learning geometrico capaci di elaborare dati strutturati a grafo.

# Modellazione dei Dati

---

## Transaction Graph

Definiamo il grafo delle transazioni finanziarie come  $G = (V, E)$  con  $|V| = N$ , dove:

- **Nodi** ( $v \in V$ ): Rappresentano i conti bancari.
- **Archi pesati** ( $(u, v) \in E$ ): Rappresentano il flusso di denaro aggregato:

$$w_{uv} \doteq \sum_{t \in T_{uv}} \text{amount}(t)$$

- Ogni nodo possiede uno stato interno ( $\alpha_i$ ) normalizzato sul grado totale del grafo.

# Limiti dello Stato dell'Arte (GNN)

---

Nonostante l'efficacia, le GNN classiche presentano criticità strutturali:

- ① **Visione Locale:** Tendenza ad uniformare le feature dei nodi distanti, perdendo informazioni su cicli o pattern a lungo raggio.
- ② **Black-box:** La natura delle reti neurali contrasta con le normative finanziarie (GDPR/AI Act) che richiedono spiegabilità.
- ③ **Onerosità:** L'addestramento richiede enormi moli di dati etichettati e risorse computazionali elevate.

## La Proposta: Architettura QTGNN

---

Si propone l'utilizzo di una **Quantum Topological Graph Neural Network (QTGNN)**, un'architettura ibrida che integra:

- **GNN**: Per la modellazione delle relazioni locali.
- **TDA (Topological Data Analysis)**: Per l'estrazione di invarianti globali robusti (Numeri di Betti) che garantiscono stabilità al rumore.
- **QML (Quantum Machine Learning)**: Per un embedding efficiente e per catturare correlazioni non lineari tramite entanglement.

# Modellazione Quantistica del Grafo

# Generalized Amplitude Encoding (GAES)

---

Per codificare la struttura del grafo  $G = (V, E)$  in uno stato quantistico, si adotta una variante dell'*Amplitude Encoding* arricchita da *Entanglement Seeding*. Lo stato target  $|\psi_G\rangle$  è mappa la matrice di adiacenza, ed è definito:

$$|\psi_G\rangle = \frac{1}{\sqrt{Z}} \sum_{i,j} A_{ij} \mathcal{E}_{ij}(\theta_e) |i\rangle |j\rangle$$

L'obiettivo è mappare le relazioni pesate del grafo direttamente nell'entanglement tra qubit.

## Entanglement Seeding ( $\mathcal{E}_{ij}$ )

---

L'informazione sugli archi è codificata tramite l'operatore di entanglement  $\mathcal{E}_{ij}$ , che agisce tra i qubit del registro sorgente e quelli di ambiente.

$$\mathcal{E}_{ij}(\theta_e) \doteq \exp(-i\theta_e \sigma_i^x \sigma_j^x)$$

Dove:

- $\theta_e$ : Parametro addestrabile che modula l'intensità dell'interazione.
- $\sigma^x$ : X-Gate che creano correlazioni non classiche.

## Matrice di Interazione $A_{ij}$

---

Il tensore  $A_{ij}$  unifica le informazioni topologiche e le feature dei nodi.

$$A_{ij} \doteq \begin{cases} w_{ij} & \text{se } i \neq j \quad (\text{Peso cumulativo transazioni}) \\ \alpha_i & \text{se } i = j \quad (\text{Centralità di grado normalizzata}) \end{cases}$$

Dove  $\alpha_i = \frac{\deg(i)}{\sum_k \deg(k)}$ . Questo permette di processare simultaneamente flussi e importanza dei nodi.

## Matrice Densità Ridotta

---

Dallo stato puro  $|\psi_G\rangle$ , ricaviamo la matrice di densità ridotta  $\rho_G$ :

$$\rho_{total} = |\psi_G\rangle \langle \psi_G|$$

$$\rho_G = Tr_{env}(\rho_{total}) \doteq \sum_{m \in env} (\mathbb{I} \otimes \langle m |) \rho_{total} (\mathbb{I} \otimes |m \rangle)$$

$\rho_G$  incapsula le proprietà topologiche del grafo in uno stato misto di dimensione ridotta ( $n$  qubit).

# Variational Quantum Graph Convolution

# Evoluzione variazionale del grafo

---

Si utilizzano circuiti quantistici parametrizzati per evolvere lo stato  $\rho_G$  attraverso  $L$  layer di convoluzione quantistica variazionale (VQGC).

## Definizione ricorsiva

$$\rho^{(0)} = \rho_G$$

$$\rho^{(l+1)} = \mathcal{N}_\theta^{(l)} \left( U_\theta^{(l)} \rho^{(l)} U_\theta^{(l)\dagger} \right)$$

# Definizione dell'operatore unitario

---

## Operatore di Evoluzione Unitaria

$$U_{\theta}^{(l)} = \exp \left( -i \sum_{(i,j) \in E} \theta_{ij}^{(l)} H_{ij} - i \sum_{i \in V} \phi_i^{(l)} \sigma_i^z - i \sum_{(i,j,k) \in \Delta} \psi_{ijk}^{(l)} \sigma_i^z \sigma_j^z \sigma_k^z \right)$$

- $\theta_{ij}^{(l)}$ : Dinamica sulle connessioni dirette (Archi).
- $\phi_i^{(l)}$ : Dinamica specifica del conto (Nodi).
- $\psi_{ijk}^{(l)}$ : Interazioni multi-partito su strutture triangolari

# Phase Damping

---

Per introdurre non-linearità, si applica inoltre un canale di dephasing:

## Dephasing mediante operatori di Kraus

$$\mathcal{N}_\theta^{(l)}(\rho) = \sum_k p_k(\theta) \Pi_k \rho \Pi_k$$

Con operatori di Kraus:  $\Pi_0 = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \Pi_1 = \sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$

E probabilità associate:  $p_0(\theta) = \frac{e^{\theta_0}}{e^{\theta_0} + e^{\theta_1}}, \quad p_1(\theta) = \frac{e^{\theta_1}}{e^{\theta_0} + e^{\theta_1}}$

Il parametro  $\theta$  controlla l'intensità del rumore modulando la preservazione delle coerenze quantistiche.

# Quantum Feature Embedding ( $Z_Q$ )

---

Al termine dei layer variazionali, si ottiene uno stato che codifica le feature del grafo.

## Embedding Finale

Lo stato finale  $Z_Q$  dopo  $L$  layer di convoluzione quantistica è definito come:

$$Z_Q = \text{Tr}_{anc} \left( \prod_{l=1}^L \mathcal{N}_\theta^{(l)} (U_\theta^{(l)} \rho^{(l)} U_\theta^{(l)\dagger}) \right)$$

# Entropia di Correlazione ( $C_Q$ )

---

Si calcola l'Entropia di Correlazione Quantistica per catturare le interazioni multi-partito.

## Entropia di Correlazione

$$C_Q = \sum_{i \neq j} Tr \left( \rho_{ij}^{(L)} \log \rho_{ij}^{(L)} - \rho_i^{(L)} \log \rho_i^{(L)} - \rho_j^{(L)} \log \rho_j^{(L)} \right)$$

Valori elevati di  $C_Q$  indicano strutture di entanglement complesse, spesso correlate ad anomalie organizzate.

# Analisi Topologica Persistente (TDA)

---

Sulla matrice densità finale  $\rho^{(L)}$  viene costruita una metrica di distanza quantistica  $d_\rho(i, j)$ . Tramite filtrazione di Vietoris-Rips, si estraggono gli invarianti topologici:

- **Numeri di Betti**  $\beta_k$ : Conteggio di componenti connesse ( $\beta_0$ ), cicli ( $\beta_1$ ) e cavità ( $\beta_2$ ).
- **Caratteristica di Eulero Persistente**  $\chi^Q$ :

$$\chi^Q(\epsilon) = \sum_k (-1)^k \beta_k^Q(\epsilon)$$

Tali invarianti offrono una "firma digitale" della forma globale del grafo.

# Classificatore Ibrido

---

Le feature quantistiche estratte vengono concatenate in un vettore:

$$\phi(G) = [Z_Q, C_Q, \{\beta_i^Q\}, \chi^Q]$$

Il modello viene addestrato minimizzando una funzione di costo ibrida:

$$\mathcal{L} = \mathcal{L}_{\text{sup}}(\hat{y}, y) + \lambda_1 \min_{m \in \mathcal{M}} \|\phi(G) - m\|^2 + \lambda_2 \|\Theta\|^2$$

## Definizione della Loss supervisionata

---

Il vettore  $\phi(G)$  è dato in input ad un classificatore classico che stima la probabilità di frode  $\hat{y}$ :

$$\hat{y} = \sigma(\mathbf{W}^\top \phi(G) + b)$$

La loss supervisionata è definita come la Binary Cross-Entropy tra la previsione e l'etichetta reale  $y$ :

$$\mathcal{L}_{\text{sup}}(\hat{y}, y) = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]$$

## Classificazione Finale

---

Una volta allenato il modello la classificazione finale di un grafo  $G$  avviene calcolando un punteggio di anomalia:

$$s(G) = \min_{\phi(G^*) \in S_{normal}} \|\phi(G) - \phi(G^*)\|_2^2$$

Dove  $S_{normal}$  è l'insieme delle feature di grafi etichettati come normali presi da un validation set.

La decisione finale è basata su una soglia  $\tau$ :

$$Fraud(G) = \begin{cases} 1(\text{Frode}) & \text{se } s(G) > \tau \\ 0(\text{Normale}) & \text{altrimenti} \end{cases}$$

# Dettagli Implementativi e Training

# Iperparametri

Parametro	Valore
Nodi per Sottografo ( $N$ )	16
Qubit Totali ( $2 \cdot \log_2 N$ )	8
Dimensione Training Set	700
Dimensione Test Set	100
Layer VQGC ( $L$ )	3
Numeri di Betti Calcolati	Fino a $\beta_1$ (Cicli)
Epochs	20
Learning Rate	0.005
Lambda Loss ( $\lambda_1, \lambda_2$ )	0.1, 0.01
Dimensione Memory Bank ( $K$ )	10
Soglia Anomalia ( $\tau$ )	0.4
Dimensione $S_{\text{normal}}$	10

## L'ambigua definizione di $S_{normal}$

---

### Problema

$S_{normal}$  va utilizzato sia durante il training (loss) che durante l'inferenza (classificazione). Nel paper originale  $S_{normal}$  è richiesto in input e definito come:

$(\phi(G_1), \phi(G_2), \dots, \phi(G_K))$ , dove  $G_i$  sono grafi leciti

Tuttavia, questa scelta presenta un'enorme criticità:

Non è possibile conoscere  $\phi(G)$  prima della fase di training.

## Soluzione Proposta: Memory Bank Dinamica

---

In fase di training si utilizza una **Memory Bank** dinamica:

$$S_{normal} \equiv \mathcal{M}^{(t)} = \{m_1, m_2, \dots, m_K\}$$

- ① **Inizializzazione:**  $\mathcal{M}^{(0)}$  è popolato con  $\{\phi(G_i)\}_{i=1}^K$ , t.c.  $G_i$  leciti.
- ② **Aggiornamento:** Utilizzando una politica FIFO ad ogni iterazione di Stochastic Gradient Descent si aggiorna una parte della Memory Bank.

Un primo tentativo basato su “*Deep One-Class Classification*” si è rilevato inadeguato poichè l’assunzione che i dati leciti fossero contenuti in un’ipersfera si è rivelata falsa.

# Analisi Sperimentale e Conclusioni

## Dataset PaySim

---

L'algoritmo è stato testato sul dataset sintetico **PaySim**. Un'analisi preliminare (script `calculate_density.py`) ha evidenziato gravi limitazioni strutturali:

- **Sparsità:** Il 94.83% dei nodi possiede grado 1. Solo il 5.17% ha grado  $\geq 2$ .
- **Assenza di Cicli:** Anche isolando le componenti debolmente connesse più grandi, i cicli e le strutture complesse sono rarissimi.

**Impatto:** Questa topologia “piatta” invalida parzialmente l'efficacia del modello, che si basa sull'identificazione di strutture topologiche complesse.

# Risultati Sperimentali e Analisi di $\Phi(G)$

---

**Metriche Finali:** Accuracy: 70.0% (TP=3, FP=0, TN=84, FN=13)

## 1. Strutture Topologiche Simili (Falsi Negativi)

**Graph 117 | NORMAL | Score: 0.000000 | Pred: 0**

$\Phi(117) = [0.1628, 2.1829, 1.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 1.0, 0.0625, 1.0625, 0.0, -1.0]$

**Graph 002 | FRAUD | Score: 0.000009 | Pred: 0**

$\Phi(002) = [0.1641, 2.1785, 1.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 1.0, 0.0625, 1.0625, 0.0, -1.0]$

## 2 Variazioni Topologiche (Rilevamento Efficace)

**Graph 068 | FRAUD | Score: 0.406291 | Pred: 1**

$\Phi(068) = [0.1571, 2.1855, 1.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 1.0, \mathbf{0.1875}, \mathbf{0.6875}, 0.0, -0.50]$

**Graph 016 | FRAUD | Score: 2.115981 | Pred: 1**

$\Phi(016) = [0.1239, 2.3273, 1.0, 1.0, 0.0, 1.0, 1.0, 0.0, 0.0, 1.0, \mathbf{0.3125}, \mathbf{0.1875}, 0.0, \mathbf{0.1250}]$

# Considerazioni Finali

---

## Conclusioni

Sebbene la natura sintetica e semplice di PaySim non abbia permesso di sfruttare appieno il potenziale del modello, l'esperimento dimostra che:

- ① L'architettura ibrida stima correttamente i Numeri di Betti e la Caratteristica di Eulero.
- ② La Memory Bank stabilizza l'apprendimento evitando il collasso su un centro inesatto.
- ③ L'algoritmo possiede un alto potenziale per scenari reali (non sintetici), dove le frodi si manifestano attraverso strutture topologiche complesse (es. anelli di riciclaggio).

Grazie per l'attenzione

# Riferimenti Bibliografici I

---

- [1] S. Alarfaj F. K. Shahzadi. “Enhancing fraud detection in banking with deep learning: Graph neural networks and autoencoders for real-time credit card fraud prevention”. In: *Journal of Financial Data Science* (2024).
- [2] Mohammad Doost e Mohammad Manthouri. “Quantum topological graph neural networks for detecting complex fraud patterns”. In: *arXiv preprint arXiv:2512.03696* (2025). URL: <https://arxiv.org/abs/2512.03696v1>.
- [3] Edward Farhi, Jeffrey Goldstone e Sam Gutmann. “A Quantum Approximate Optimization Algorithm”. In: (2014). arXiv: 1411.4028 [quant-ph]. URL: <https://arxiv.org/abs/1411.4028>.

## Riferimenti Bibliografici II

---

- [4] Hdaib M. e L. Rajasegarar S. Pan. “Quantum deep learning-based anomaly detection for enhanced network security”. In: *Quantum Machine Intelligence* 6.1 (2024).
- [5] M. J. Bastian N. D. Monkam G. F. De Lucia. “A topological data analysis approach for detecting data poisoning attacks against machine learning based network intrusion detection systems”. In: *Computers Security* 144 (2024).
- [6] B. Motie S. Raahemi. “Financial fraud detection using graph neural networks: A systematic review”. In: *Expert Systems with Applications* 240 (2024).
- [7] María Laura Olivera-Atencio, Lucas Lamata e Jesús Casado-Pascual. “Impact of Amplitude and Phase Damping Noise on Quantum Reinforcement Learning: Challenges and Opportunities”. In: (2025). arXiv: 2503.24069 [quant-ph]. URL: <https://arxiv.org/abs/2503.24069>.

## Riferimenti Bibliografici III

---

- [8] T. Pourhabibi et al. “Fraud detection: A systematic literature review of graph-based anomaly detection approaches”. In: *Decision Support Systems* 133 (2020), p. 113303.
- [9] Lukas Ruff et al. “Deep One-Class Classification”. In: *Proceedings of the 35th International Conference on Machine Learning*. A cura di Jennifer Dy e Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, ott. 2018, pp. 4393–4402. URL:  
<https://proceedings.mlr.press/v80/ruff18a.html>.
- [10] Y. Umeda, J. Kaneko e H. Kikuchi. “Topological data analysis and its application to time-series data analysis”. In: *Fujitsu Scientific and Technical Journal* 55 (2019), pp. 65–71.

## Riferimenti Bibliografici IV

---

- [11] Xingbao Zhang, Wei Li e Yue Zhao. “One-class Anomaly Detection with Redundancy Reduction and Momentum Mechanism”. In: (2022), pp. 1–6.  
DOI: [10.1109/DOCS55193.2022.9967719](https://doi.org/10.1109/DOCS55193.2022.9967719).