LA TECHNOLOGY TRANSFER PRESENTS

RUSSEL JURNEY

GRAPH MACHINE LEARNING

THEORY, PRACTICE, TOOLS AND TECHNIQUES

ONLINE LIVE STREAMING

JUNE 5-9, 2023

DUE TO TIME ZONES, THIS CLASS WILL TAKE PLACE IN THE AFTERNOON
FROM 2 PM TO 6 PM ITALIAN TIME

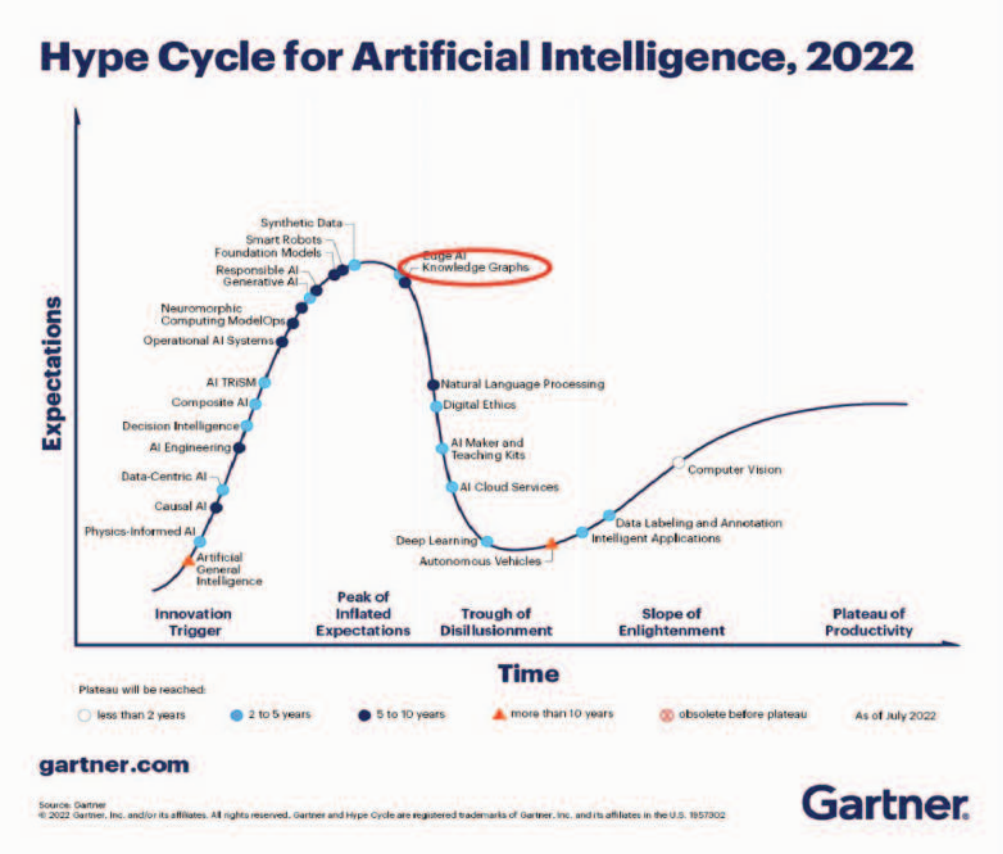
info@technologytransfer.it
www.technologytransfer.it

ABOUT THIS SEMINAR

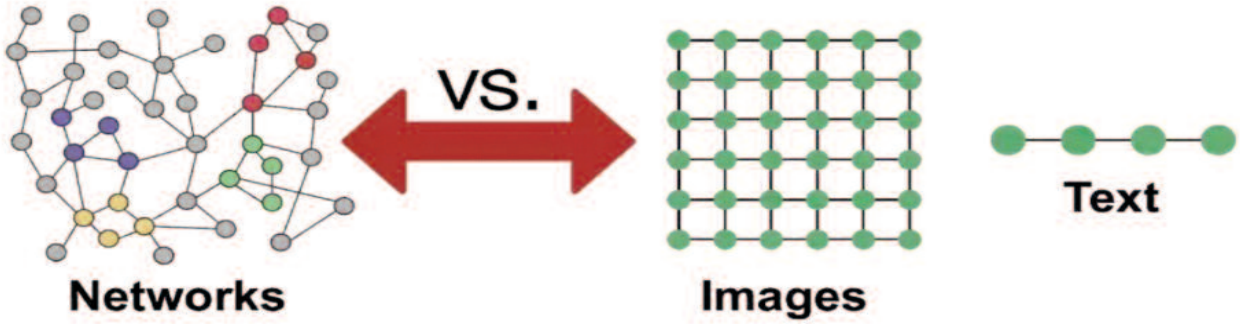
Starting from tabular DataFrames and traditional Machine Learning - we build a theoretical understanding of modern Data Science and Machine Learning methods for graph structured datasets as well as the practical skills that enable participants to implement them. Participants graduate from the course able to add Graph Analytics and Graph Machine Learning to their day to day workflows for datasets small and large using the most popular tools.

Why is there so much talk about Knowledge Graphs and GNNs?
Knowledge Graphs are at the peak of the Gartner hype cycle and graph neural networks (GNNs) are the are soon to be high on the ramp because they tap and unlock the potential of enterprise Knowledge Graphs. Data Lakes put data in one place, Knowledge Graphs link datasets together and graph neural networks automate business processes using data from across an enterprise. Most graph databases are fast becoming Cloud-based GNN platforms:

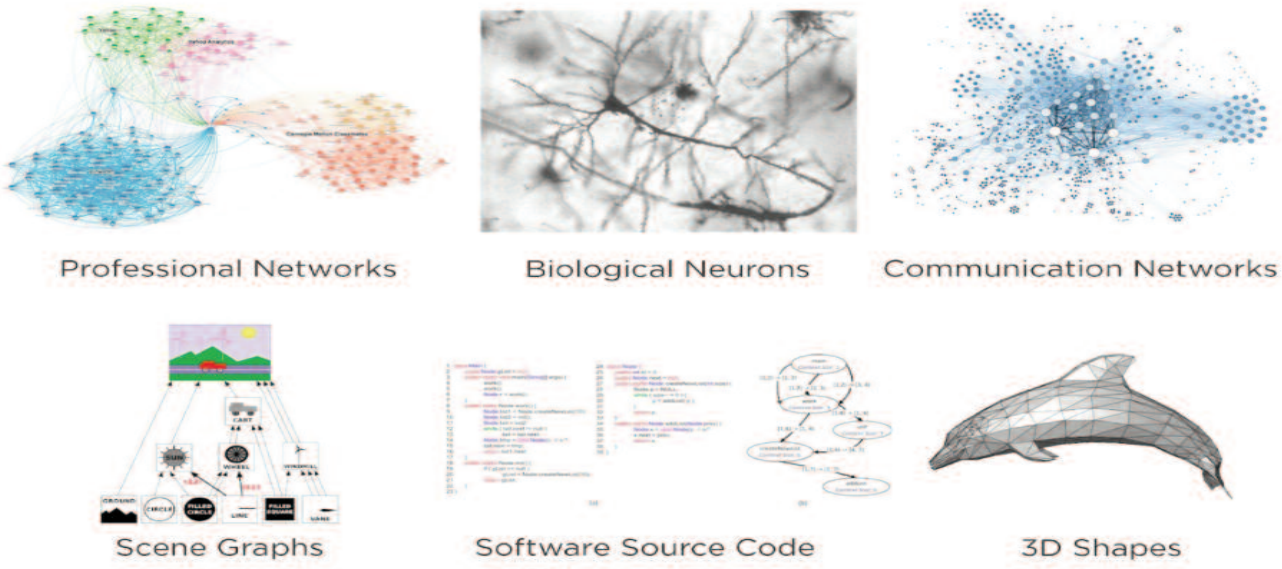
- Neo4j → Neo4j Graph Data Science
- TigerGraph → Machine Learning Workbench
- ArangoDB → ArrangoGraphML
- Kumo → SQL query the future



What's the real story?
I'll let you in on a secret that is driving the popularity of enterprise Knowledge Graphs, property graphs, graph databases and Graph Neural Networks (GNNs): **MOST DATA IS GRAPH DATA**. To compose a single table to get the corresponding vectors, matrices and tensors we load into GPUs to drive Machine Learning algorithms, several tables have usually been combined [squashed] into one table. There's a problem with this... it is a **lossy process**. We threw away the relationships. Graph Neural Networks are able to learn better to build more powerful models because they have a greater potential by matching the structure of the data's entities and their In this course, we will take the skills you've developed in working with data tables and DataFrames and extend



them to cover graphs, networks, knowledge graphs, property graphs and graph databases. We will work with different types of graphs from across problem domains. This includes natural networks like social networks, collaboration networks or communications networks as well as structural networks like the plan of a Python program or the 3D mesh of a model of a 3-dimensional scene.



We will introduce common Python and R tools for Graph Analytics and Graph Machine Learning as well as the most popular graph databases. We will focus on property graphs but will also compare them with RDF/triple stores using SPARQL. We will cover the core methods from social network analysis and network science that will guide your informed-intuition in doing Graph Machine Learning. We will build a Knowledge Graph using Natural Language Processing (NLP), combine its duplicate nodes using deep networks for entity resolution and mine the resulting graph for patterns. Finally, we will build a full-stack graph ML application that shows network visualizations of explainable GNNs for chemical engineering.
You will graduate from the course able to work with graphs as you now work with tables and DataFrames.

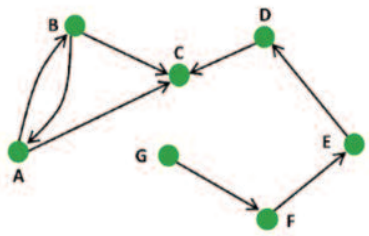
How log is this coure?

The course is spread over five half-days. The order of topics is:

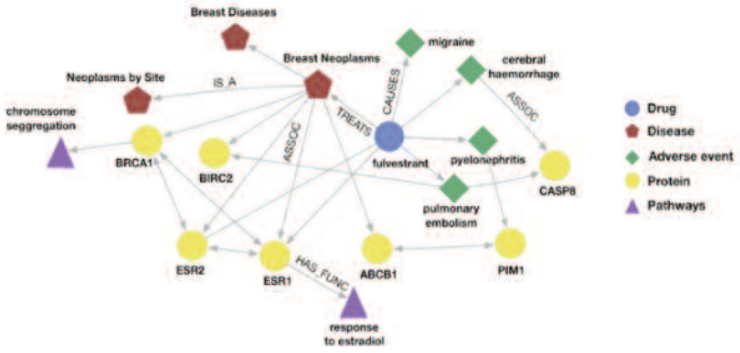
- Day 0 - 2-3 days before we begin - Platform Setup of the Docker, notebook Github software for the course. Participants begin before the course and we debug by email. At the start of Day 1 we will verify everything works for all participants.
- Day 1 - Theory and use cases
- Day 2 - Quantitative networks: social network analysis and network science
- Day 3 - Building, refining and serving a Knowledge Graph from end-to-end
- Day 4 - Graph Machine Learning from traditional graph ML to Graph Neural Networks (GNNs)
- Day 5 - Full-stack Graph ML applications

Who should attend this course?

This course is for Data Scientists and Machine Learning engineers who want to extend their work with data tables and DataFrames to datasets with a relational or graph structure - entities (nodes/vertices) and their connections (edges/links).



Simple, Directed Graph



Heterogeneous Graph

Beginning with a grounding in theory and real-world use cases, we will build Knowledge Graphs using Natural Language Processing (NLP), analyze networks using Graph Analytics and extract the full potential of relational structures in data by automating business processes using Graph Machine Learning including graph kernels, embeddings and graph neural networks (GNNs). When the course is completed, you will be able to incorporate Graph Machine Learning in your daily work to build more powerful models and systems.

What skills are required to be successful in this course?

The course focuses on the **Python** data stack, paricipants should be able to write Python programs. They should be grounded in Data Science or statistics using data tables like pandas DataFrames. They should be familiar with Jupyter, Databricks, Snowflake or other notebooks. Some exposure to traditional Machine Learning concepts and tools like scikit-learn is necessary, but expertise is NOT required. Mathematics are helpful for attaining intuition but a basic grasp of graphs and vectors will be sufficient for a working knowledge as we employ visualizations to assist with intuitive learning of concepts.

I use a certain tool or platform. Can this course help me?

The course is useful to Data Scientists on any platform, but here are a few communities of users that might benefit from the course to create business value using graphs and networks:

- Python tools like Pandas and NetworkX, graph-tool, NetworKit or EasyGraph
- R tools like iGraph, tidygraph and ggraph
- Big Data tools like PySpark, Databricks, Dask, Snowflake or GraphFrames
- GPU-accelerated compute tools like RAPIDS cuGraph
- Property graph databases like Neo4j, TigerGraph, Oracle Graph Studio or Oracle Graph Studio
- Enterprise Knowledge Graphs that use RDF Triple Stores/SPARQL like StarDog or Ontotext
- Large knowledge bases like WikiData Query Service
- Jupyter, Databricks and Snowflake Notebooks
- Natural Language Processing (NLP) users of tools such as spaCy, FlairNLP, BLINK, Gensim or NLTK
- Network visualization tools like Gephi, Graphistry or Cambridge Intelligence Keylines/ReGraph

What will participants learn?

Participants will go from a working knowledge of Data Science and Machine Learning with data tables to a working knowledge of Data Science and Machine Learning for graphs to build real world applications. It is not enough to teach participants Graph Neural Networks (GNNs) - they need to work their way up from graph theory to GNNs using common Python tools in design patterns based on real world use cases.
Participants will learn to:

- Build and extend Knowledge Graphs using Natural Language Processing (NLP) named entity recognition (NER), information extraction and entity linking
- Describe social networks using social network analysis (SNA)
- Describe and analyze any network using network science
- Perform graph analytics on both property graph and RDF triple store/SPARQL databases
- Find significant patterns in real world networks
- Build predictive systems using traditional ML
- Replace manual feature engineering with graph embeddings
- Solve a range of problems using Graph Neural Networks
- Build a full stack app for visualizing explainable GNN models

Outline

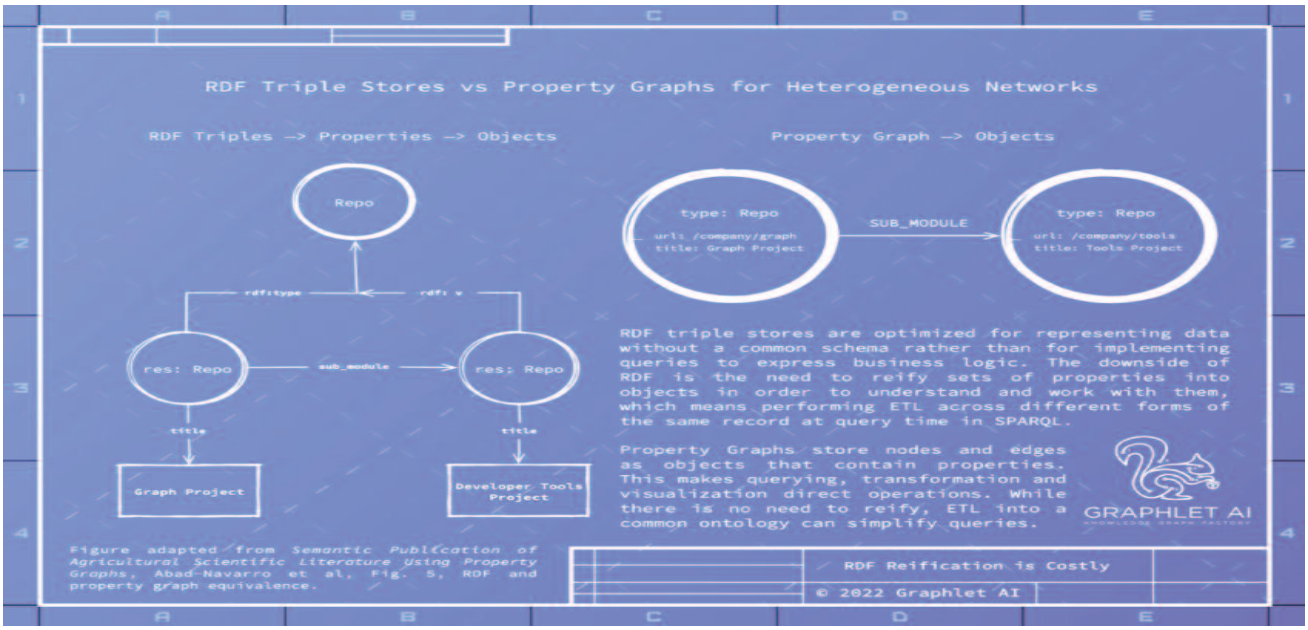
The course is divided into two sections: theory and practice.

The theory portion introduces theoretical models for the course's content along with real-world examples from science and industry of their use. If you don't know the basis and relevance of a technique - why learn it?
The practice portion works through the everyday work of implementing the topics covered in the first day using the most popular, effective tools for graph processing, graph databases and Graph Machine Learning. We focus on Python but do a little R.

Theory: Graphs, SNA, Network Science, Graph ML, GNNs

We start with a review of graph theory and present simple, intuitive, visual and mathematical explanations of topics such as graph data models, Social Network Analysis (SNA), network science, graph analytics, pattern machine and traditional and GNN Machine Learning methods. We also introduce knowledge graph construction using Natural Language Processing.

- Graph theory - what is a graph? Examples of networks? Heterogeneous networks can model anything!
- Data models - property graphs (yaaah!) vs RDF triple stores + SPARQL

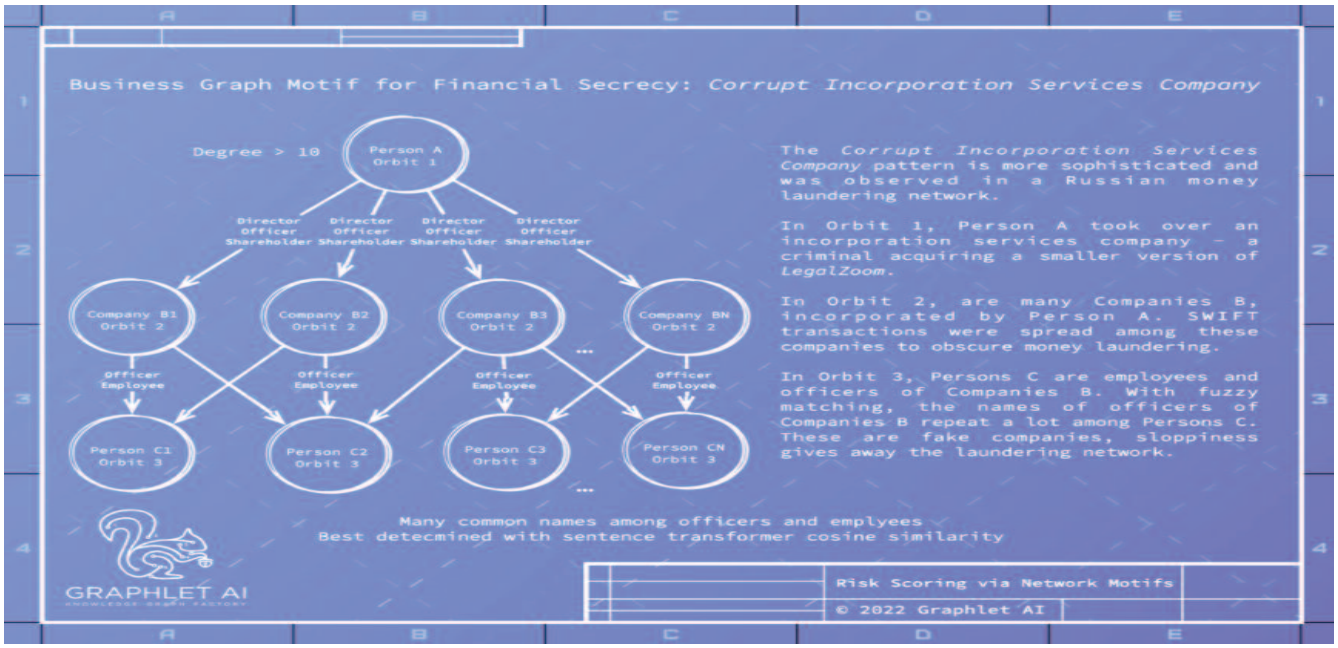


RDF requires you to reify properties into the objects they describe

- Social Network Analysis (SNA) - social science (yaaah!)
- Network science - techniques that span fields and applications
- Natural Language Processing (NLP) for Knowledge Graph construction
- Entity resolution - merging duplicate nodes and splitting erroneous combinations

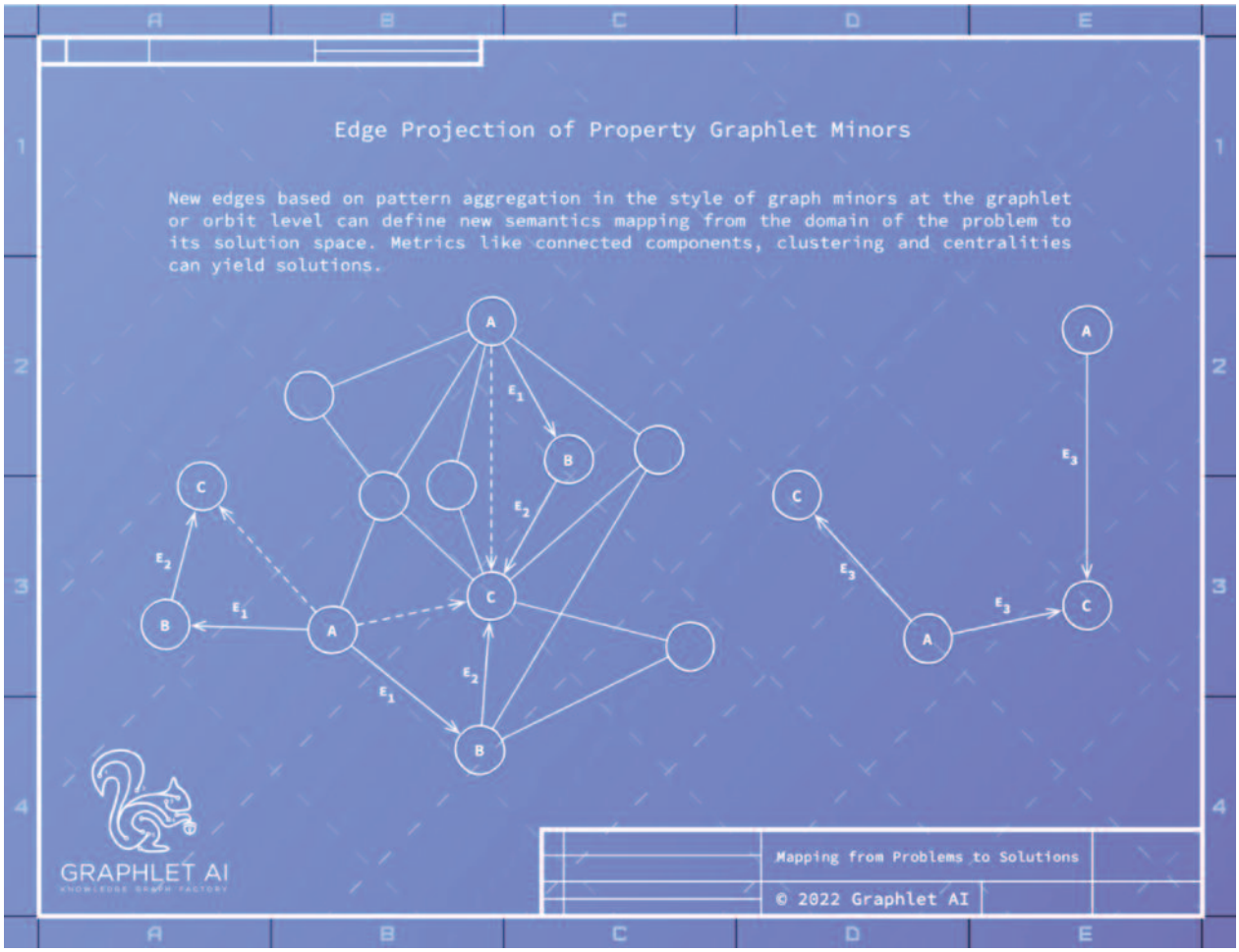
Without entity resolution, pattern matching like motif search doesn't work!

- Network null models - how do I know what is significant?
- Network motifs - what patterns [queries] matter in my network?



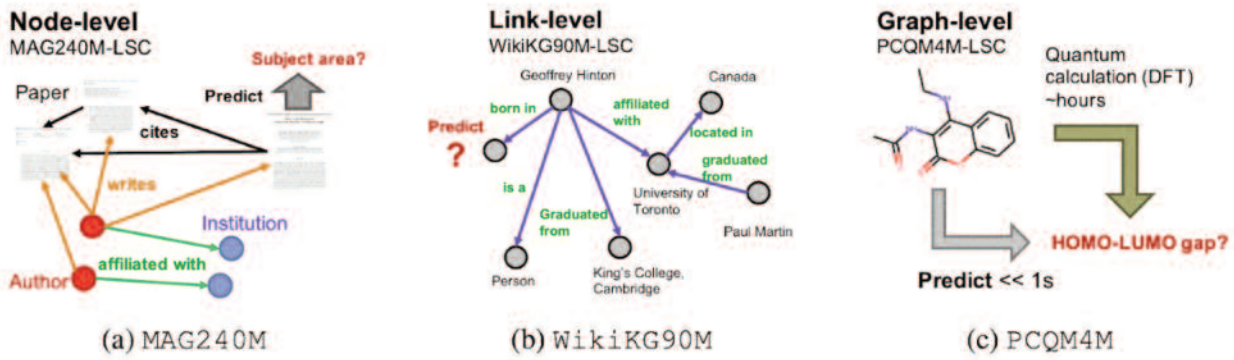
An example of a property graph motif in anti-money laundering (AML)

- Graph transformations + projections - problem space → solution space



In this network transformation, edges are projected between different orbits of network motifs

• Graph machine learning tasks - node, link, sub-graph, graph



Tasks for neural networks can also include sub-graphs when sampling

• Graph features and kernels - feature engineering for networks

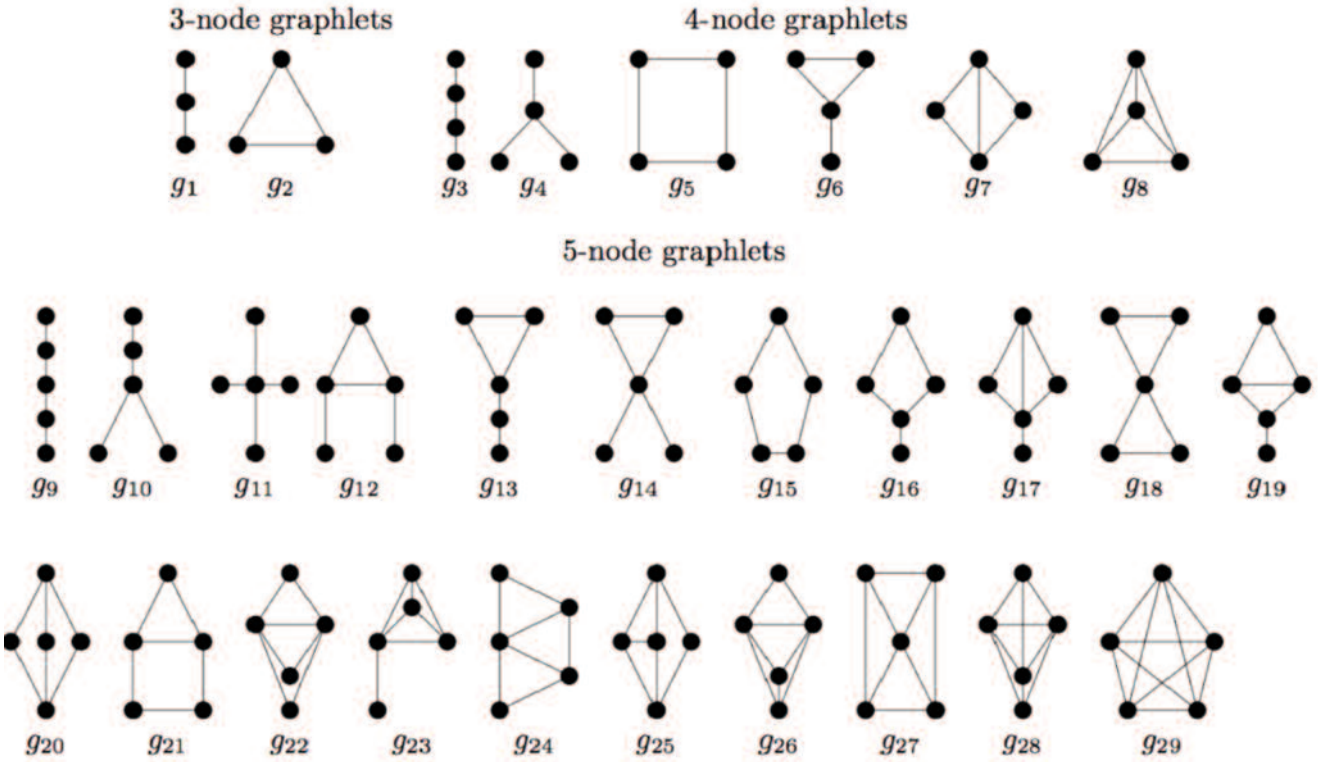
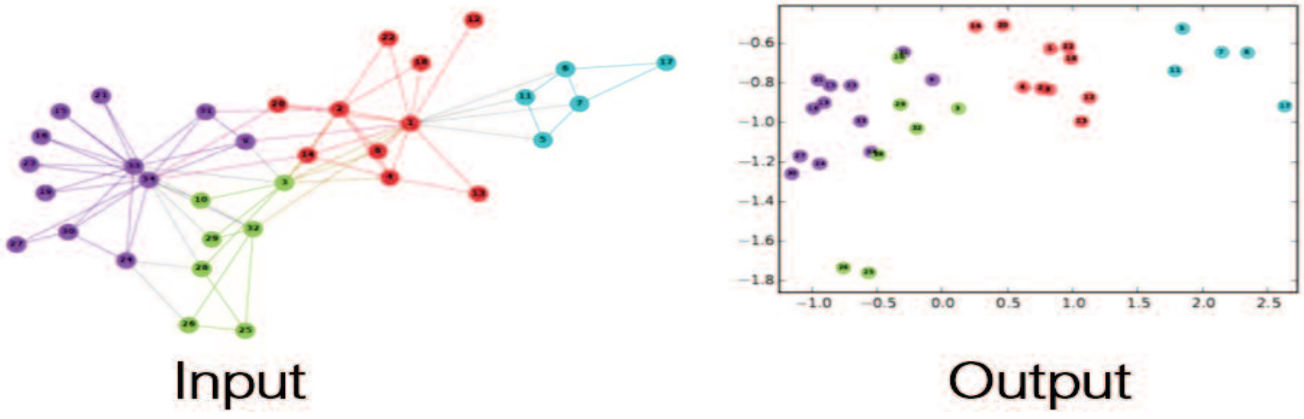


Fig. 1. All graphlets of size 3,4,5

Graphlet distributions can be used as graph kernels for graph level tasks

- Graph embeddings - from Word2Vec to Node2Vec and beyond!



Node2vec embedding a network's topology in two dimensions

- Graph Neural Networks (GNNs) - neural networks shaped like graphs that learn directly from the properties and structure of the data

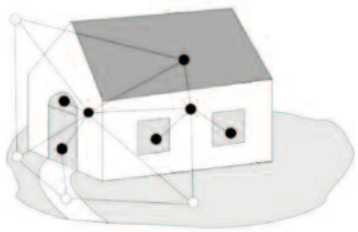


Fig. 1. An image and its graphical representation by a RAG

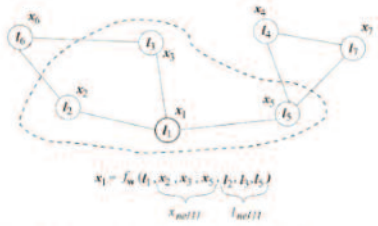


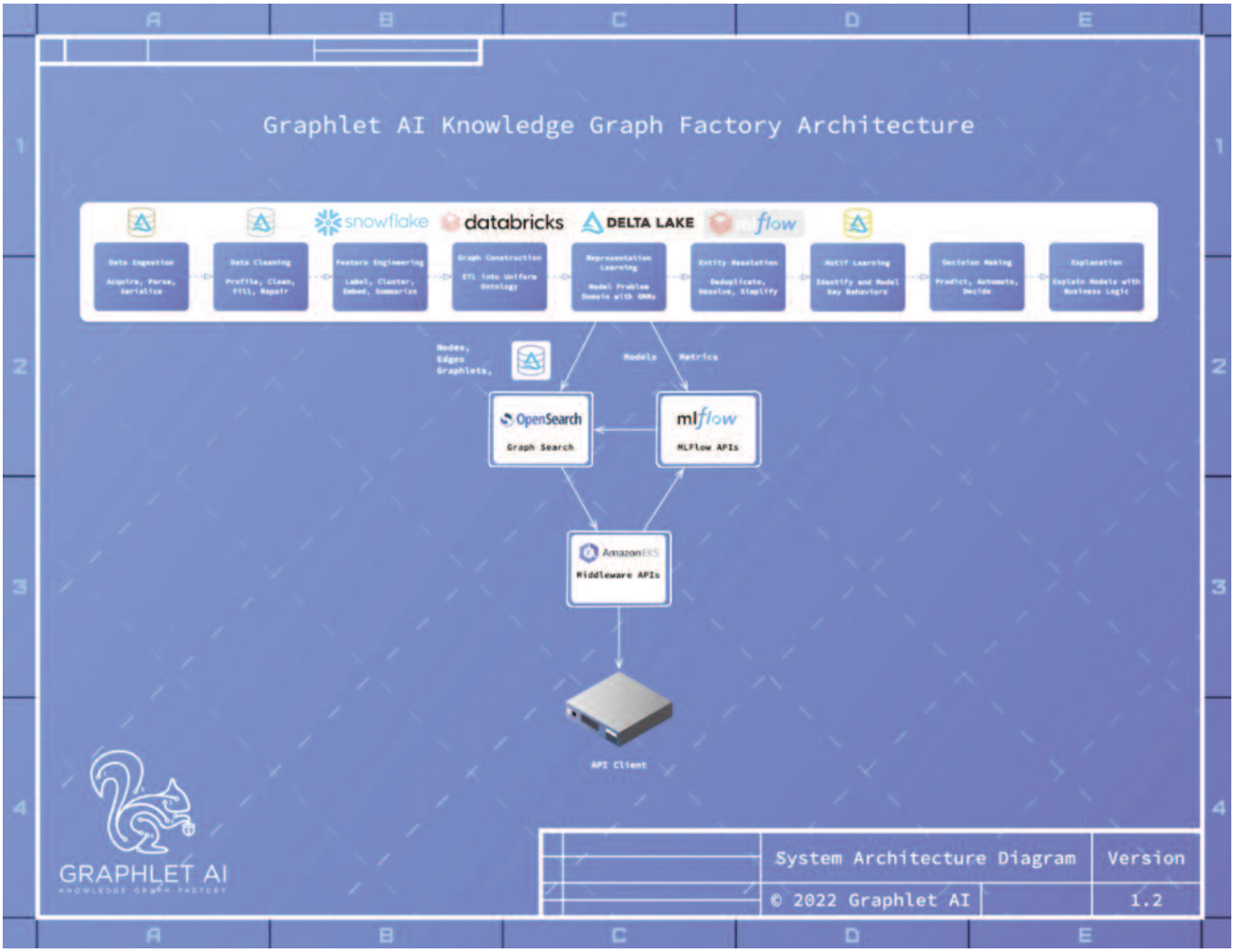
Fig. 2. State x_1 depends on the neighborhood information

Fig. 3. A graph and its corresponding encoding network.

The original graph neural network paper

• Network visualization - data viz for small and large scale networks

• Machine Learning Operations (MLOps) for large networks

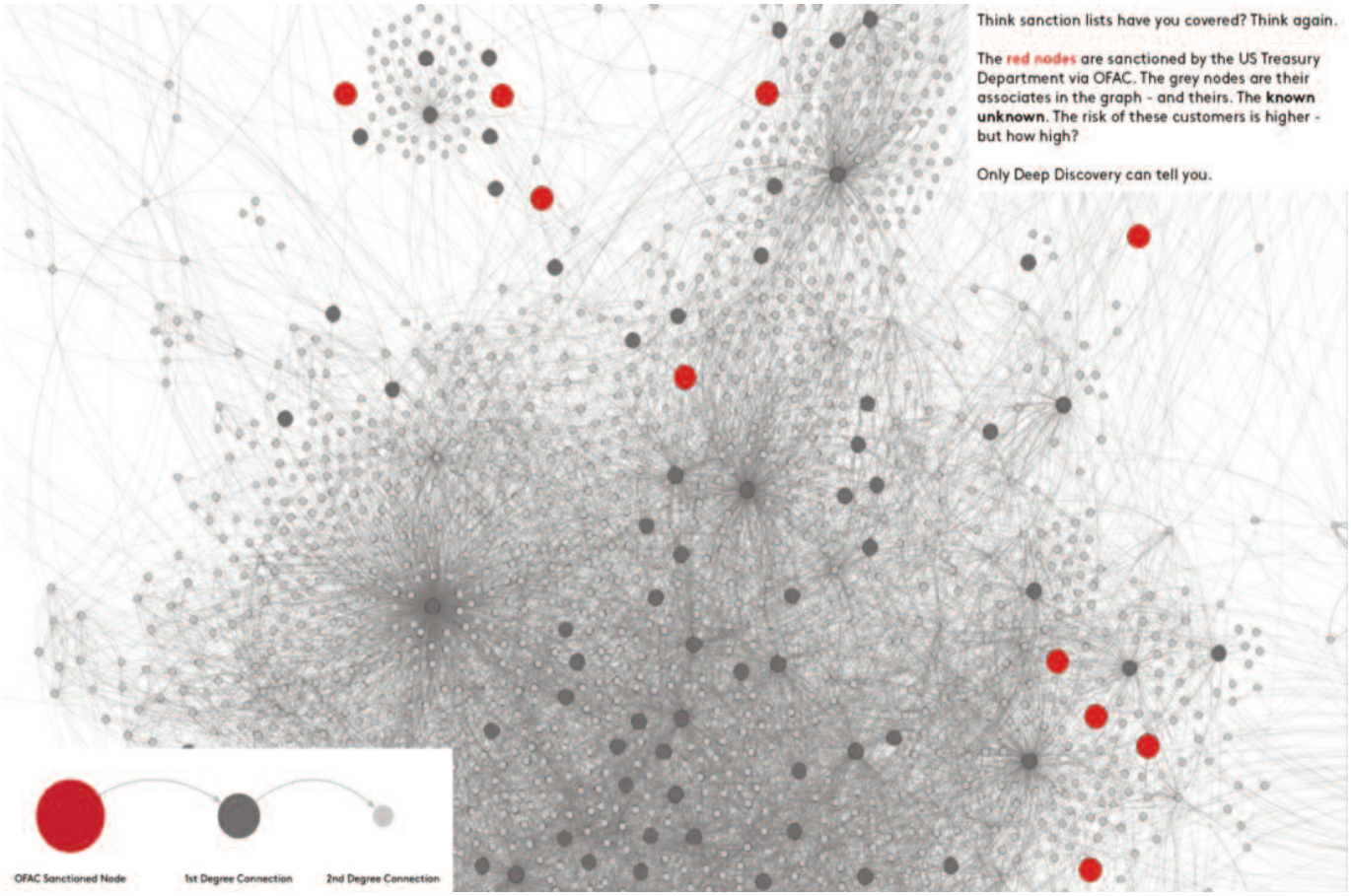


Big data build-refine-publish architecture for Graph Machine Learning

Practice: KG Construction/NLP, Quantitative Analysis, Graph ML

We move from theory to practice. Participants will build a practical, working knowledge of common tools for implementing the methods and techniques outlined above using Python 3 (and a little R).
From top to bottom, we cover:

- Knowledge graph (KG) construction using natural language processing (NLP)
  - o Named entity recognition (NER) using spaCy
  - o NER and Information extraction (IE) using FlairNLP
  - o Entity linking using Facebook BLINK
  - o Outsourcing it with Diffbot!
- Social network analysis (SNA) with networkx and Gephi



Gephi visualization of criminal networks for money laundering

- Neo4j and the Cypher Query Language (CQL)
- Fundamentals of network science with networkx and graphistry
    - Centrality metrics - local, neighborhood and global
- Transforming networks with PySpark and GraphFrames
- Simple and heterogeneous graph null models using Pandas and Neo4j
- Simple network motif search with R and iGraph
- Property graph motifs for huge networks with PySpark and GraphFrames

Property graph motif search on a 3 billion node heterogeneous network

- GPU accelerated graph analytics with RAPIDS cuGraph
- Traditional graph ML using networkx, scikit-learn and XGBoost
- Graph embeddings with KarateClub. Cobra Kai forever!
- Graph Neural Networks in PyG [formerly PyTorch Geometric]

```python
dataset = Planetoid(root='.', name='Cora')

class GCN(torch.nn.Module):
    def __init__(self, in_channels, hidden_channels, out_channels):
        super().__init__()
        self.conv1 = GCNConv(in_channels, hidden_channels)
        self.conv2 = GCNConv(hidden_channels, out_channels)

    def forward(self, x: Tensor, edge_index: Tensor) -> Tensor:
        x = self.conv1(x, edge_index).relu()
        x = self.conv2(x, edge_index)
        return x

model = GCN(dataset.num_features, 16, dataset.num_classes)
```

A simple Graph Convolutional Network (GCN) in PyG

• Finale: Full Stack, interactive visualization of GNN explainability with PyG and Cambridge Intelligence ReGraph [as in React + Keylines]

Explainable graph ML for anti-money laundering using a business graph

INFORMATION

PARTICIPATION FEE

€ 1400

The fee includes all seminar documentation.

SEMINAR TIMETABLE

2.00 pm - 6.00 pm (Italian time)

HOW TO REGISTER

You must send the registration form with the receipt of the payment to:
info@technologytransfer.it

TECHNOLOGY TRANSFER S.r.l.
Piazza Cavour, 3 - 00193 Rome (Italy)

PAYMENT

Wire transfer to:
Technology Transfer S.r.l.
Banca: Cariparma
Agenzia 1 di Roma
IBAN Code:
IT 03 W 06230 03202 000057031348
BIC/SWIFT: CRPPIT2P546

GENERAL CONDITIONS

DISCOUNT

The participants who will register 30 days before the seminar are entitled to a 5% discount.

If a company registers 5 participants to the same seminar, it will pay only for 4.

Those who benefit of this discount are not entitled to other discounts for the same seminar.

CANCELLATION POLICY

A full refund is given for any cancellation received more than 15 days before the seminar starts. Cancellations less than 15 days prior the event are liable for 50% of the fee. Cancellations less than one week prior to the event date will be liable for the full fee.

CANCELLATION LIABILITY

In the case of cancellation of an event for any reason, Technology Transfer's liability is limited to the return of the registration fee only.

RUSSEL JURNEY
GRAPH MACHINE LEARNING

June 5-9 2023

Registration fee:
€ 1400

If registered participants are unable to attend, or in case of cancellation of the seminar, the general conditions mentioned before are applicable.

first name .....

surname .....

job title .....

organisation .....

address .....

postcode .....

city .....

country .....

telephone .....

fax .....

e-mail .....

Stamp and Signature

Send your registration form with the receipt of the payment to:
Technology Transfer S.r.l.
Piazza Cavour, 3 - 00193 Rome (Italy)
Tel. +39-06-6832227 - Fax +39-06-6871102
info@technologytransfer.it
www.technologytransfer.it

Speaker

He works at the intersection of Big Data, large networks - Property Graphs or knowledge graphs, representation learning with Graph Neural Networks (GNNs), Natural Language Processing (NLP) and Understanding (NLU), model explainability using network visualization and vector search for information retrieval.

He is a startup product and engineering executive focused on building products driven by billion node+ networks. He has worked at cool places like Ning, LinkedIn and Hortonworks. He co-founded Deep Discovery to use networks, GNNs and visualizations to build an explainable risk score for KYC/AML.

He has a four-time O'Reilly author with 120 citations on Google Scholar for being the first to write about Agile Data Science - agile development as applied to Data Science and Machine Learning. He has an applied researcher and product manager with 17 years of experience building and shipping data-driven products.