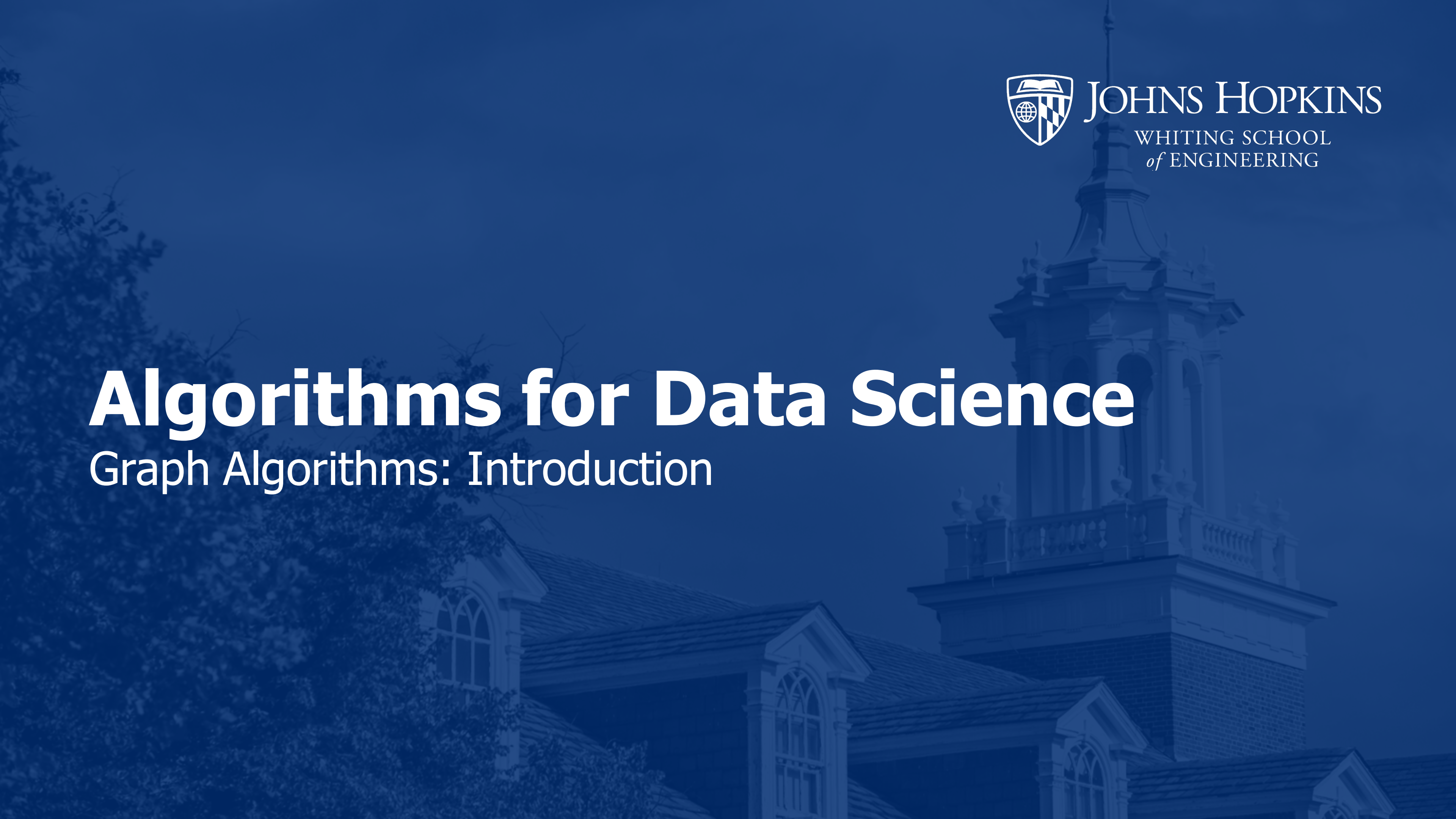


JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Algorithms for Data Science

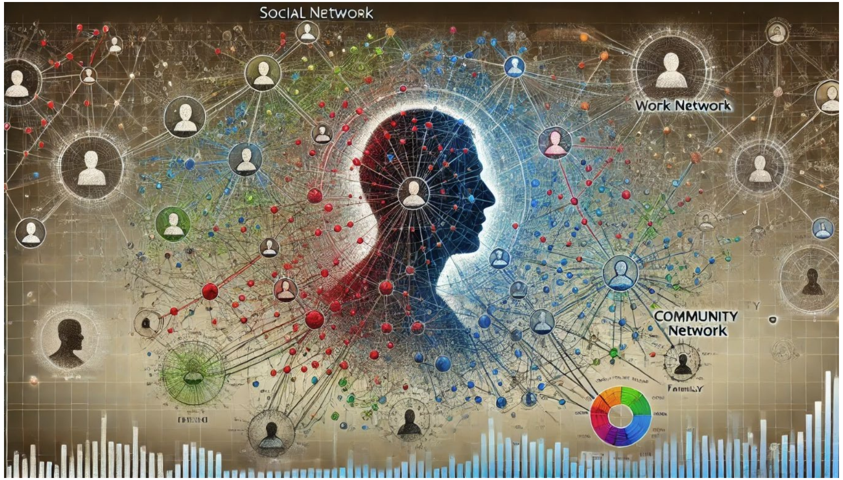
Graph Algorithms: Introduction

Module Learning Objectives

- 7.1 Describe the mathematical foundations of key graph algorithms, including Breadth-First Search (BFS), Depth-First Search (DFS), Minimum Spanning Trees (MST), and Bayesian Networks.
- 7.2 Analyze the computational complexity of graph algorithms and evaluate their efficiency in terms of time and space complexity.
- 7.3 Implement fundamental graph traversal algorithms such as BFS and DFS in Python and compare their practical applications.
- 7.4 Apply decision trees and Bayesian networks to model probabilistic dependencies and make informed inferences in data science problems.
- 7.5 Evaluate the effectiveness of graph-based clustering methods such as the Louvain Algorithm and Spectral Clustering in identifying network communities.
- 7.6 Construct and optimize graph-based feature engineering techniques for machine learning applications, leveraging graph embeddings and community detection.
- 7.7 Critique recent advances in graph algorithms, discussing their impact on scalability, deep learning, and real-world applications.

Why Graph Algorithms?

Their role in understanding connections, relationships, and structures in data is useful for a variety of Data Science and AI applications



They underpin models in Natural Language Processing, Computer Vision, and Bayesian inference.

Types of Graph Algorithm Problems

- Traversal Algorithms
- Minimums Spanning Trees
- Decision Making
- Community Detection
- Semi-Supervised Learning
- Clustering and Dimensionality Reduction

Graph Algorithm Problem Components

Graph Structures

- **Types of Graphs**: Directed vs Undirected Weighted vs Unweighted
- **Graph Representations**: Adjacency List, Adjacency Matrix
- Does the algorithm require a directed graph, or does it work with any type?

Graph Operations

- **Common Operations**: Traversing (BFS, DFS), Shortest Paths, Connectivity Checks.
- **Graph Modifications**: Edge Removal, Node Addition.
- Does the algorithm focus on finding paths, grouping nodes, or optimizing structures?

Constraints & Complexity

- **Time Complexity Considerations**: $O(V+E)$ for BFS/DFS vs. $O(V^2)$ for adjacency matrices.
- **Memory Usage**: Trade-off between speed and storage (Adjacency List vs. Adjacency Matrix).
- Is the graph too large for exact algorithms, requiring approximations instead?

Application of Graph Algorithms

Logistics

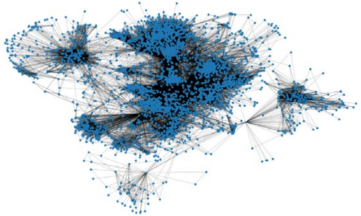
Planning the most efficient delivery routes

Cybersecurity

Security threats are best understood as anomalies in a network

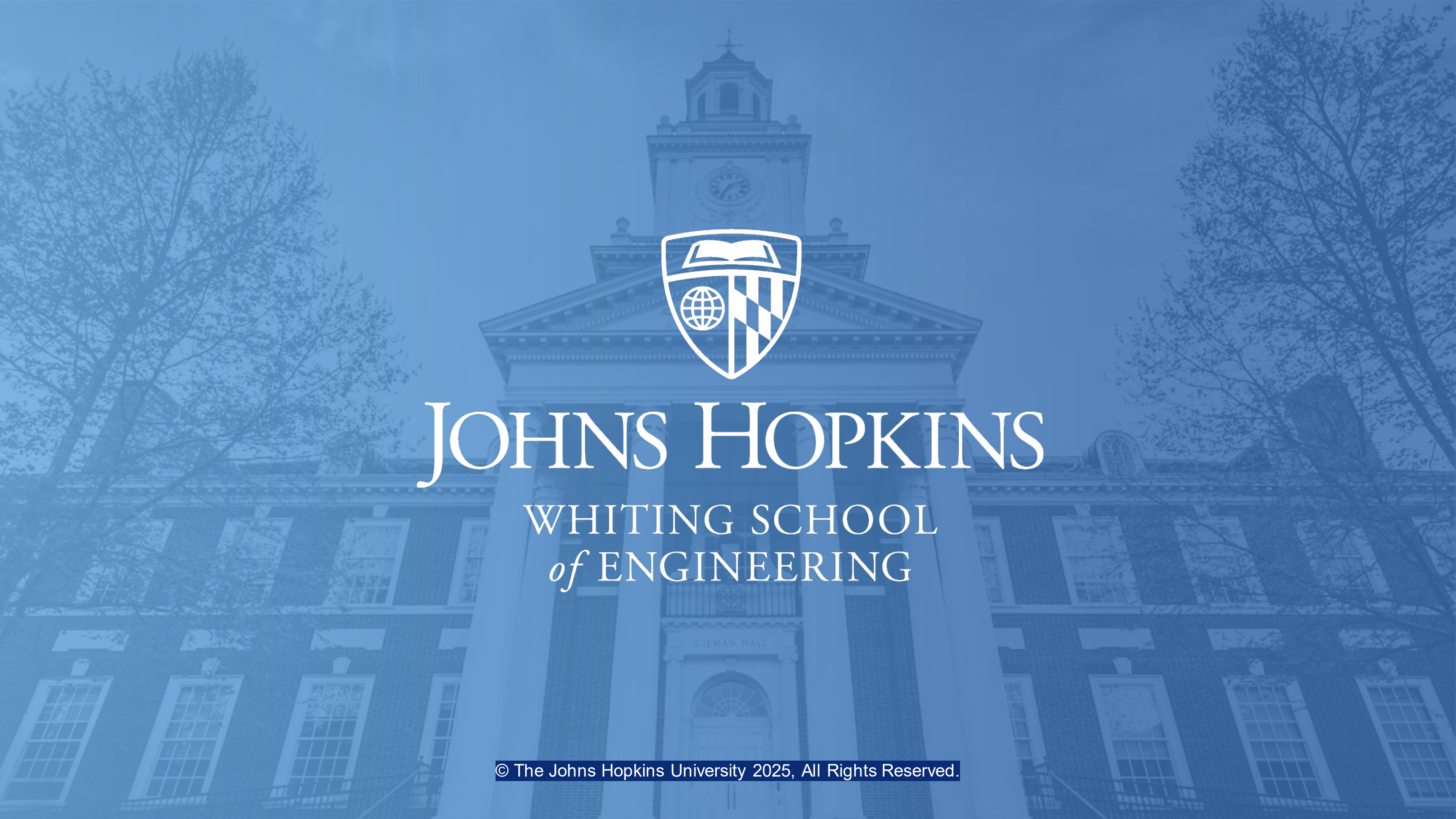
Healthcare

Analyze molecular interactions in drug discovery



Social Networks

Analyze connections between users, behaviors, and preferences



JOHNS HOPKINS

WHITING SCHOOL *of* ENGINEERING