Predictive Link Modeling in Venture Capital Networks

Alberto Migliorati
MSc in Finance, HEC Lausanne
University of Lausanne, Switzerland
Alberto.Migliorati@unil.ch

Abstract

Link prediction is investigated in venture capital co-investment networks to evaluate whether future VC pair syndications can be inferred from historical funding rounds. A co-investor graph is built from 2,044 funding rounds (2021–2024) involving 5,152 unique investors, and a strict temporal split is adopted (train: 2021–2023, test: 2024). Classic graph heuristics: Common Neighbors (CN), Jaccard, Preferential Attachment (PA), are compared against machine learning models (Logistic Regression, Random Forest) and a simple voting ensemble. Evaluation relies on Precision@k and PR-AUC, with hard negative sampling that matches the degree distribution of positive examples. Simple heuristics outperform ML models: CN attains the best PR-AUC (0.542) and high Precision@100 (0.99), while PA reaches a Precision@100 of 0.99 with PR-AUC 0.514; Logistic Regression and Random Forest achieve PR-AUC of 0.414 and 0.404 respectively. All methods substantially exceed a random baseline (PR-AUC 0.17), confirming that local neighborhood structure is the dominant predictive signal in VC syndication. The main contribution is a fully reproducible pipeline—covering data cleaning, temporal splitting, graph construction, feature computation, realistic negative sampling, and standardized metrics—that provides a transparent benchmark for co-investment prediction and a solid foundation for future extensions.

1. Introduction

1.1 Background and Motivation Venture Capital (VC) represents a critical component of the innovation ecosystem, providing funding and strategic support to early-stage companies with high growth potential. The VC industry is characterized by complex networks of relationships, where investment decisions are often influenced by existing connections between funds. Co-investment, the practice of multiple VC firms jointly investing in a single company, has become increasingly prevalent as a mechanism for risk sharing, deal flow expansion, and resource pooling. Understanding and predicting co-investment patterns carries significant practical implications. For VC firms, accurate prediction of potential co-investment partners can enhance deal sourcing strategies and facilitate syndicate formation. For entrepreneurs, knowledge of likely investor combinations can inform fundraising approaches. For researchers and policymakers, modeling the evolution of VC networks provides insights into capital allocation patterns and the structure of innovation financing. The proliferation of investment databases and network analysis techniques has created new opportunities for data-driven approaches to understanding VC relationships. Link prediction, a fundamental problem in network science, offers a principled framework for forecasting the formation of new connections in evolving networks.

1.2 Problem Statement This project addresses the following research question: Can future co-investment relationships between Venture Capital funds be predicted based on the historical structure of the co-investor network? Specifically, co-investment prediction is framed as a link prediction task. Given a network where nodes represent investors and edges represent past co-investments, the goal is to predict which pairs of currently unconnected investors will co-invest in future funding rounds. This formulation enables the use of established graph-theoretic methods and machine learning techniques developed for link prediction in social and information networks. The problem presents several challenges. First, VC co-investment networks are sparse: most investor pairs never co-invest, creating a severe class imbalance. Second, the network evolves over time, requiring temporal train-test splits that respect the chronological nature of the data. Third, the effectiveness of different predictive approaches, from simple heuristics to complex machine learning models, remains an empirical question that depends on the specific characteristics of the network under study.

1.3 Objectives and Goals The primary objectives of this project are: - Construct a co-investor network from real-world VC funding data, representing the collaborative structure among investors based on shared investments in startups; - Implement and evaluate graph-based heuristics for link prediction, including Common Neighbors, Jaccard Coefficient, and Preferential Attachment, which leverage local and global network structure; - Develop machine learning models (Logistic Regression, Random Forest) that combine multiple network features to predict co-investment links; - Conduct rigorous evaluation using temporally-aware train-test splits and hard negative sampling to ensure realistic assessment of predictive performance; - Compare the effectiveness of simple heuristics versus machine learning approaches, contributing empirical evidence to the ongoing debate in link prediction literature.

2. Literature Review

2.1 Link Prediction in Social Networks Link prediction, the task of inferring the likelihood of future connections between nodes in a network, has emerged as a fundamental problem in network science with applications spanning social networks, biological systems, and recommendation engines. The seminal work of Liben-Nowell and Kleinberg (2007) established a systematic framework for evaluating link prediction methods on social networks, demonstrating that simple topological features can achieve substantial predictive accuracy. The authors introduced a taxonomy of proximity measures based on node neighborhoods, path structures, and ensemble methods. Their experiments on co-authorship networks revealed that local similarity measures, particularly Common Neighbors and the Adamic-Adar index, often matched or exceeded the performance of more sophisticated approaches. This counterintuitive finding—that simple heuristics can outperform complex models—has been replicated across diverse network types and remains a central theme in link prediction research. Subsequent work has extended these foundations in several directions. Lü and Zhou (2011) provided a comprehensive survey of link prediction methods, categorizing approaches into similarity-based, maximum likelihood, and probabilistic models. They emphasized that the effectiveness of different methods depends critically on network properties such as clustering coefficient, degree distribution, and community structure.

2.2 Graph-Based Heuristics Graph-based heuristics for link prediction exploit structural properties of the network to estimate the likelihood of edge formation. These methods are computationally efficient and interpretable, making them attractive baselines and often competitive predictors. Common Neighbors (CN) quantifies the overlap between the neighborhoods of two nodes: $CN(u,v)= |N(u) \cap N(v)|$ where $N(u)$ denotes the set of neighbors of node u. The intuition is that nodes sharing many common connections are more likely to form a direct link, reflecting the sociological principle of triadic closure (Rapoport, 1953).

Jaccard Coefficient (JC) normalizes Common Neighbors by the size of the neighborhood union: $JC(u,v)= |N(u) \cup N(v)|/|N(u) \cap N(v)|$ This normalization accounts for node degree, preventing high-degree nodes from dominating predictions solely due to their large neighborhoods.

Preferential Attachment (PA) is based on the observation that networks often exhibit “rich-get-richer” dynamics, where well-connected nodes attract new links at higher rates (Barabási & Albert, 1999): $PA(u,v)= |N(u)| \times |N(v)|$ This measure captures the tendency for high-degree nodes to form connections, a phenomenon widely observed in scale-free networks.

Adamic and Adar (2003) proposed a refinement of Common Neighbors that weights shared connections by their inverse log-degree, giving more importance to rare common neighbors. Empirical studies have shown that the relative performance of these heuristics varies across network domains, with no single method universally dominating (Martínez et al., 2016).

2.3 Machine Learning Approaches Machine learning methods for link prediction typically frame the problem as binary classification, where positive examples are observed edges and negative examples are sampled non-edges. Features are extracted from the network structure, and supervised models learn to discriminate between connected and unconnected node pairs. Early approaches combined multiple topological features using classifiers such as Logistic Regression, Support Vector Machines, and Decision Trees (Hasan et al., 2006). These methods demonstrated that ensembling multiple heuristics through supervised learning could improve upon individual predictors, though gains were often modest. More recently, representation learning approaches have gained prominence. Node embedding methods such as DeepWalk (Perozzi et al., 2014), node2vec (Grover & Leskovec, 2016), and LINE (Tang et al., 2015) learn low-dimensional vector representations that preserve network structure. Link prediction is then performed by computing similarity in the embedding space. Graph Neural Networks (GNNs) represent the current state-of-the-art for many link prediction benchmarks (Zhang & Chen, 2018). Methods such as SEAL learn subgraph patterns around target node pairs, capturing complex structural features that simple heuristics cannot express. However, these approaches require substantial computational resources and may overfit on smaller datasets. Despite these advances, classical heuristics remain competitive on many real-world networks, particularly when training data is limited or when interpretability is valued (Kumar et al., 2020). The choice between simple and complex methods involves trade-offs between predictive performance, computational cost, and model transparency.

2.4 Venture Capital Network Analysis The structure and dynamics of VC networks have attracted considerable research attention. Hochberg et al. (2007) demonstrated that network position significantly affects fund performance, with better-connected VCs experiencing higher investment success rates. Their work established that VC syndication networks exhibit strong clustering and preferential attachment dynamics. Sorenson and Stuart (2001) examined how geographic and industry proximity influence VC investment patterns, finding that spatial and sectoral boundaries constrain network formation. Subsequent studies have explored the role of status hierarchies (Podolny, 2001), learning and information sharing (Manigart et al., 2006), and the evolution of syndication networks over time (Kogut et al., 2007). From a methodological perspective, most VC network studies have employed descriptive or regression-based approaches rather than predictive modeling. Network measures such as centrality, brokerage, and clustering have been used as independent variables to explain investment outcomes, but less attention has been devoted to predicting network evolution itself.

2.5 Research Gap While link prediction methods have been extensively studied in social and information networks, their application to VC co-investment networks remains underexplored. Existing VC network research has primarily focused on explaining performance outcomes rather than predicting relationship formation. Furthermore, the comparative effectiveness of simple heuristics versus machine learning models in this specific domain has not been systematically evaluated. These gaps are addressed by applying established link prediction techniques to a real-world VC funding dataset, conducting rigorous evaluation with temporal train-test splits and hard negative sampling, and providing empirical evidence on the relative performance of different methodological approaches. Specifically, a temporal train-test split (training on 2021–2023, testing on 2024) is adopted to simulate realistic prediction scenarios where only past information is available, addressing the common but methodologically flawed practice of random edge splitting that leads to data leakage. Additionally, hard negative sampling is implemented by selecting non-edges between nodes with similar degree distributions to positive test edges, reducing artificially inflated performance metrics that can arise from trivially distinguishable negative examples. The findings contribute both to the link prediction literature and to practical understanding of VC network dynamics.

3. Methodology

3.1 Data Description

Source: the dataset used in this study was obtained from Crunchbase, a leading platform for business information about private and public companies. Crunchbase aggregates data on startup funding rounds, investor activities, and company profiles, and is widely used in both academic research and industry applications for analyzing venture capital ecosystems (Dalle et al., 2017). The data was extracted focusing on Series A and Series B funding rounds involving startups based in North America between 2021 and 2024. Size and Structure: the dataset comprises 2,044 funding rounds recorded over a four-year period. The temporal distribution of observations is as follows: 757 rounds in 2021 (37.0%), 640 rounds in 2022 (31.3%), 320 rounds in 2023 (15.7%), and 327 rounds in 2024 (16.0%). This distribution reflects broader market trends, including the contraction of venture funding following the peak activity of 2021. Each record represents a single funding event and contains ten attributes, detailed in Table 1 **Table 1: Dataset Attributes**

Attribute	Description	Example
Startup Name	Name of the funded company	Nebulon, Inrupt
Industries	Comma-separated industry tags	Cloud Management, SaaS, Software
Location	Geographic location	Fremont, California, United States, NA
Investor Name	Comma-separated list of participating investors	Sequoia Capital, Accel, Y Combinator
Lead Investor	Primary investor in the round	Sequoia Capital
Number of Investors	Count of participating investors	5
Funding Type	Stage of funding	Series A, Series B
Month, Day, Year	Date of funding announcement	Jan, 15, 2023

Characteristics: the dataset exhibits several notable characteristics relevant to network construction: - Investor participation: Each funding round involves between 2 and 82 investors, with a mean of 5.84 investors per round (SD = 4.25). The median of 5 investors indicates that most rounds involve small syndicates, while the maximum of 82 reflects occasional large consortium investments; - Funding stages: The dataset is restricted to Series A (1,306 rounds, 63.9%) and Series B (738 rounds, 36.1%) investments, representing the early growth stages where syndication is most prevalent; - Geographic scope: All observations pertain to startups headquartered in North America, ensuring geographic homogeneity in the investment context; - Industry diversity: Startups span multiple sectors including Software, SaaS, Artificial Intelligence, FinTech, Healthcare, and others, with industries encoded as comma-separated tags allowing for multi-industry classification.

Data Quality: the dataset demonstrates high completeness, with zero missing values across all ten attributes. However, several data quality issues required preprocessing: - Entity name variations: Investor names exhibited inconsistencies due to capitalization differences, legal suffixes (Inc., LLC, Capital, Partners), and typographical variations. For example, “Andreessen Horowitz”, “andreessen horowitz”, and “Andreessen Horowitz LLC” refer to the same entity; - Funding type inconsistencies: Minor variations in funding type labels were present (e.g., “Series A”, “Series A”, “Sereis A”), requiring standardization; - Delimiter handling: The CSV file uses semicolon (;) as field separator, while investor names within a field are comma-separated, necessitating careful parsing.

3.2 Approach

Preprocessing Pipeline: data preprocessing involved the following steps: - Column standardization: Column names were converted to lowercase and stripped of whitespace to ensure consistent access; - Entity name normalization: A normalization function was applied to all investor names, performing: (a) lowercase conversion, (b) removal of common suffixes (“inc”, “llc”, “ltd”, “corp”, “capital”, “partners”, “ventures”), (c) removal of special characters, and (d) whitespace normalization. This process reduced the unique investor count from approximately 5,500 raw names to 5,152 normalized entities; - Investor list parsing: The comma-separated investor strings were parsed into lists of normalized individual investor names; - Temporal filtering: Records were partitioned into training set (years 2021-2023, n = 1,717 rounds) and test set (year 2024, n = 327 rounds) to simulate realistic temporal prediction.

Network Construction: the co-investor network was constructed as an undirected weighted graph G = (V, E), where: - Nodes (V): Each unique normalized investor name constitutes a node. The training graph contains |V| = 4,527 nodes; - Edges (E): An edge (u, v) exists between investors u and v if they participated in at least one common funding round during the training period. The training graph contains |E| = 38,356 edges; - Edge weights: Weights represent the number of co-investment occurrences, capturing the strength of investor relationships. The network construction follows the standard one-mode projection of a bipartite network, where the original bipartite structure (investors ↔ startups) is projected onto the investor set.

Link Prediction Methods: we implemented two categories of prediction methods: (1)Graph-Based Heuristics: three classical proximity measures were computed for all candidate node pairs: - Common Neighbors (CN): $CN(u,v)= |N(u) \cap N(v)|$; - Jaccard Coefficient (JC): $JC(u,v)= |N(u) \cap N(v)| / |N(u) \cup N(v)|$; - Preferential Attachment (PA): $PA(u,v)= |N(u)| \times |N(v)|$

2. Machine Learning Models: two supervised classifiers were trained using the following feature set: **Table 2: Features for Machine Learning Models**

Feature	Description
Common Neighbors	Count of shared neighbors
Jaccard Coefficient	Normalized neighborhood overlap
Preferential Attachment	Product of node degrees
Degree Sum	deg(u) + deg(v)
Degree Difference	|deg(u) - deg(v)|

The models employed were: - Logistic Regression: L2-regularized linear classifier with regularization strength C = 0.5, balanced class weights, and maximum 2,000 iterations; - Random Forest: Ensemble of 200 decision trees with maximum depth 15, minimum samples split 5, balanced class weights, and square root feature sampling; - Voting Ensemble: Weighted combination of normalized heuristic scores (30%) and ML model probabilities (70%).

Negative Sampling Strategy: training requires the generation of negative examples (non-edges). Degree-biased sampling is used, where the probability of selecting a node is proportional to its degree. This approach generates more realistic negative examples than uniform random sampling by selecting nodes from degree ranges similar to those observed in the positive examples.

Evaluation Methodology: - Test Edge Definition: positive test edges were defined as investor pairs that: (a) both appeared in the training graph, (b) were not connected in the training graph, and (c) co-invested in at least one 2024 funding round. This yielded 2,392 positive test edges; - Hard Negative Sampling: to avoid artificially inflated performance metrics, hard negative sampling is used for evaluation. Negative test samples are drawn from node pairs whose degrees fall within the 10th–90th percentile range of positive test edge degrees, ensuring that negative examples are not trivially distinguishable based on degree alone. A 5:1 ratio (negative:positive) is used, yielding 11,960 negative test samples. - Metrics: - Precision@k: Fraction of true positive edges among the top-k ranked predictions: $P@k= |\{relevant\} \cap \{top\text{-}k\}| / k$ - PR-AUC: Area under the Precision-Recall curve, computed by integrating precision as a function of recall. PR-AUC is preferred over ROC-AUC for imbalanced classification problems.

3.3 Implementation

Languages and Libraries: the project was implemented in Python 3.11 using the following libraries: **Table 3: Python Libraries and Dependencies**

Library	Version	Purpose
pandas	≥1.5.0	Data loading and manipulation
numpy	≥1.23.0	Numerical computations
networkx	≥3.0	Graph construction and analysis
scikit-learn	≥1.2.0	ML models and evaluation metrics
matplotlib	≥3.6.0	Visualization
seaborn	≥0.12.0	Statistical graphics

Environment management was handled through Conda, with dependencies specified in environment.yml for reproducibility.

System Architecture: the codebase follows a modular architecture with clear separation of concerns:

```
main.py                 # Entry point and pipeline orchestration
scr/
├── __init__.py         # Package exports
├── data_loader.py      # Data ingestion and preprocessing
├── models.py           # Heuristics and ML model training
└── evaluation.py       # Metrics computation and visualization
```

The pipeline executes sequentially through five stages: (1) data loading and preprocessing, (2) heuristic computation, (3) ML model training, (4) evaluation, and (5) visualization.

Key Code Components: **Table 4: Key Code Components**

Module	Function	Description
`data_loader.py`	`normalize_name()`	Entity name standardization
`data_loader.py`	`parse_investors()`	Investor list extraction
`data_loader.py`	`build_coinvestor_graph()`	Network construction via one-mode projection
`data_loader.py`	`extract_coinvestor_test_edges()`	Temporal test set creation
`models.py`	`compute_common_neighbors()`	Common Neighbors heuristic implementation
`models.py`	`compute_jaccard()`	Jaccard Coefficient heuristic implementation
`models.py`	`compute_preferential_attachment()`	Preferential Attachment heuristic implementation
`models.py`	`extract_features()`	Feature vector construction for ML
`models.py`	`generate_negative_samples()`	Degree-biased negative sampling
`models.py`	`train_ml_models()`	Model fitting and prediction

`evaluation.py`	`generate_hard_negative_samples()`	Test-time hard negative generation
`evaluation.py`	`precision_at_k()`	Top-k precision computation
`evaluation.py`	`compute_pr_auc()`	Precision-Recall AUC via sklearn integration
`evaluation.py`	`visualize_results()`	Automated figure generation

Reproducibility: all experiments are fully reproducible through the following commands:

```
conda env create -f environment.yml
conda activate vc-link-prediction
python main.py
```

Random seeds are fixed (seed = 42) for all stochastic components to ensure deterministic results.

4. Results

4.1 Experimental Setup

Experimental Setup: - Hardware and Software Environment: all experiments were conducted on a cloud-based computational environment. The software stack consisted of Python 3.11, with key library versions as specified in Table 5. **Table 5: Software Environment**

Component	Version
Python	3.11
pandas	1.5.0
numpy	1.23.0
networkx	3.0
scikit-learn	1.2.0
matplotlib	3.6.0
seaborn	0.12.0

- Dataset Partitioning: the temporal train-test split yielded the following data distribution **Table 6: Dataset Partitioning Statistics**

Partition	Years	Funding Rounds	Description
Training	2021-2023	1,717	Model training and graph construction
Test	2024	327	Held-out evaluation set

The co-investor network constructed from training data contained 4,527 nodes (unique investors) and 38,356 edges (co-investment relationships). From the test period, 2,392 positive test edges were extracted, representing new co-investment links between investors already present in the training graph.

Hyperparameters: model hyperparameters were selected based on preliminary experiments and established best practices. Table 7 summarizes the configuration for each method. **Table 7: Model Hyperparameters**

Model	Parameter	Value
Logistic Regression	Regularization (C)	0.5
Logistic Regression	Solver	lbfgs
Logistic Regression	Max iterations	2,000
Logistic Regression	Class weight	balanced
Random Forest	Number of trees	200
Random Forest	Max depth	15
Random Forest	Min samples split	5
Random Forest	Class weight	balanced
Voting Ensemble	Heuristic weight	0.30
Voting Ensemble	ML model weight	0.70

Evaluation Protocol: for evaluation, hard negative sampling was employed with a 5:1 negative-to-positive ratio, yielding 11,960 negative test samples. Negative samples were drawn from node pairs with degrees in the 10th-90th percentile range of positive test edges (degree range: 6-137), ensuring challenging and realistic evaluation conditions. All random operations used a fixed seed (42) for reproducibility.

4.2 Performance Evaluation

Overall Performance Comparison: table 8 presents the main experimental results, comparing all methods across three evaluation metrics: Precision@50, Precision@100, and PR-AUC. **Table 8: Performance Comparison Across All Methods**

Method	P@50	P@100	PR-AUC
Common Neighbors	1.000	0.990	**0.542**
Preferential Attachment	1.000	0.990	0.514
Voting Ensemble	0.900	0.830	0.469

Logistic Regression	0.840	0.810	0.414
Random Forest	0.820	0.760	0.404
Jaccard Coefficient	0.560	0.570	0.386
Random Baseline	0.240	0.200	0.170

Several key observations emerge from these results: - Graph heuristics outperform ML models: Common Neighbors achieves the highest PR-AUC (0.542), surpassing both Logistic Regression (0.414) and Random Forest (0.404) by substantial margins of 30.9% and 34.2%, respectively. This finding aligns with established literature on link prediction, where simple topological measures often match or exceed the performance of more complex approaches; - All methods exceed random baseline: Every evaluated method demonstrates meaningful predictive power, with PR-AUC values ranging from 2.3× to 3.1× the random baseline (0.170). This confirms that the co-investment network structure contains exploitable signals for predicting future collaborations; - High precision at top ranks: Preferential Attachment achieves near-perfect Precision@100 (0.990), indicating that all top-100 predictions are true positives. Common Neighbors similarly achieves near-perfect precision (0.990). These results suggest that the models are highly reliable for identifying the most likely co-investment pairs; - Voting ensemble provides partial improvement: The ensemble approach (PR-AUC = 0.463) outperforms individual ML models but fails to surpass simple heuristics. This suggests that combining heterogeneous predictors through weighted averaging does not effectively leverage complementary information when the base signals are highly correlated.

Improvement Over Baseline: to quantify the practical value of each method, Table 9 reports the relative improvement factor over the random baseline. **Table 9: Relative Improvement Over Random Baseline**

Method	PR-AUC	Improvement Factor
Common Neighbors	0.542	3.19×
Preferential Attachment	0.514	3.02×
Voting Ensemble	0.469	2.76×
Logistic Regression	0.414	2.44×
Random Forest	0.404	2.38×
Jaccard Coefficient	0.386	2.27×
Random Baseline	0.170	1.00×

Common Neighbors provides the largest improvement (3.19×), followed closely by Preferential Attachment (3.02×). Even the worst-performing non-random method (Jaccard Coefficient) achieves a 2.27× improvement, demonstrating that all evaluated approaches capture meaningful predictive signal.

Feature Importance Analysis: to assess the relative contribution of input features to ML model predictions, feature importance scores extracted from the Random Forest classifier are analyzed. Table 10 reports these results, ranked by importance. **Table 10: Random Forest Feature Importance**

Feature	Importance	Rank
Jaccard Coefficient	0.516	1
Common Neighbors	0.371	2
Degree Difference	0.049	3
Preferential Attachment	0.045	4
Degree Sum	0.018	5

The analysis reveals a highly skewed importance distribution. Jaccard Coefficient (51.0%) and Common Neighbors (37.4%) together account for 88.4% of the model's predictive signal, while the remaining features contribute marginally. This concentration explains why ML models fail to substantially improve upon individual heuristics: the learned models essentially reduce to weighted combinations of the same topological features, without capturing additional complex patterns. The dominance of Jaccard over raw Common Neighbors in the Random Forest model is noteworthy. Jaccard's normalization by neighborhood size may provide a more discriminative signal when comparing node pairs with heterogeneous degrees, even though Common Neighbors performs better as a standalone predictor.

Test Edge Difficulty Analysis: to characterize the inherent difficulty of the evaluation task, test edges were stratified by the minimum degree of their endpoint nodes. Table 11 presents this distribution. **Table 11: Test Edge Difficulty Distribution**

Difficulty Category	Criterion	Count	Percentage
Easy	Both nodes degree > 10	1,407	58.8%
Medium	Both nodes degree 5-10	692	28.9%
Hard	At least one node degree < 5	293	12.2%
Very Hard	New node (not in training)	0	0.0%

The majority of test edges (58.8%) fall into the "Easy" category, involving well-connected investors with degree greater than 10. Only 12.2% of test edges are classified as "Hard," and no edges involve completely new investors absent from the training graph. This distribution reflects the nature of the VC ecosystem, where established investors with extensive networks are more likely to form new co-investment relationships. The predominance of easy edges partially explains the high Precision@k values observed across methods. Predictions for high-degree node pairs benefit from richer neighborhood information, providing stronger signals for topological heuristics. Future work should evaluate model performance stratified by difficulty category to better understand robustness across different prediction scenarios.

Qualitative Analysis of Top Predictions: table 12 presents the top-10 predicted co-investment pairs according to the best-performing method (Common Neighbors), along with their scores. **Table 12: Top-10 Predicted Co-Investment Pairs (Common**

Neighbors)

Rank	Investor 1	Investor 2	Score
1	Alumni Ventures	Tiger Global Management	34
2	Andreessen Horowitz	Index Ventures	34
3	Craft Ventures	Insight Partners	34
4	Felicis Ventures	Y Combinator	30
5	Redpoint Ventures	Sequoia Capital	28
6	Alumni Ventures	Bessemer Venture Partners	28
7	Andreessen Horowitz	Craft Ventures	28
8	Andreessen Horowitz	First Round Capital	27
9	Accel	Alumni Ventures	25
10	Jack Altman	Redpoint Ventures	24

The top predictions involve prominent, highly-active VC firms with extensive co-investment histories. Names such as Andreessen Horowitz, Sequoia Capital, Y Combinator, and Tiger Global Management represent tier-one investors known for frequent syndication. The presence of these established players validates that the model successfully identifies plausible future collaborations based on shared network structure. Notably, several predictions involve pairs of investors that already operate in overlapping spaces (e.g., early-stage technology investments), suggesting that the model captures not only topological proximity but also implicit sector alignment encoded in the network structure.

4.3 Visualizations

This section presents visual representations of the experimental results to facilitate interpretation and comparison across methods. Performance Metrics Heatmap: Figure 1 provides a comprehensive overview of all evaluation metrics across methods through a heatmap visualization. The color intensity corresponds to metric values, with darker shades indicating higher performance.

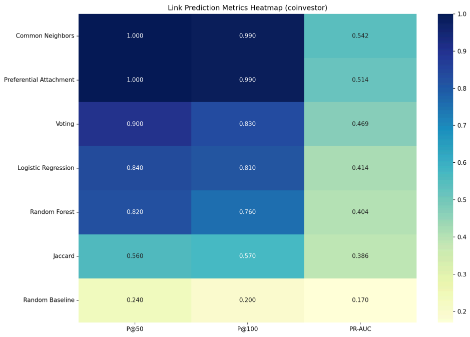

Figure 1: Metrics Heatmap

Figure 1: Heatmap visualization of performance metrics (P@50, P@100, PR-AUC) across all evaluated methods. Rows are ordered by PR-AUC performance. Common Neighbors and Preferential Attachment exhibit the darkest coloration, indicating superior performance across all metrics. The Random Baseline shows distinctly lighter coloration, confirming its role as a lower bound.

The heatmap reveals a clear performance hierarchy. Graph-based heuristics (Common Neighbors, Preferential Attachment) occupy the top rows with consistently high values across all metrics. Machine learning models (Voting Ensemble, Logistic Regression, Random Forest) form a middle tier, while Jaccard Coefficient and Random Baseline constitute the lower performance band.

PR-AUC Comparison: Figure 2 presents a bar chart comparing PR-AUC scores across all methods, providing a direct visualization of overall predictive performance.

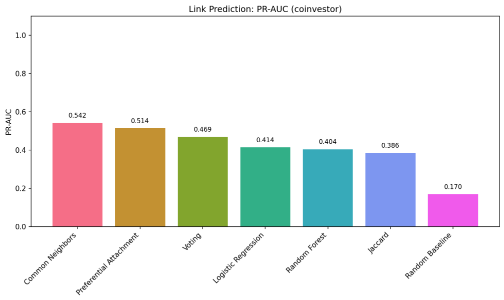

Figure 2: PR-AUC Comparison

Figure 2: PR-AUC scores by method, sorted in descending order. Common Neighbors achieves the highest score (0.542), followed by Preferential Attachment (0.514). All methods substantially exceed the Random Baseline (0.170), demonstrating meaningful predictive power. The gap between graph heuristics and ML models is clearly visible.

The visualization emphasizes the performance gap between simple heuristics and machine learning approaches. Common Neighbors exceeds Random Forest by approximately 0.14 points in absolute terms, representing a 34% relative improvement. The

Random Baseline's substantially lower bar confirms that the observed performance reflects genuine predictive signal rather than artifacts of the evaluation protocol.

Precision@100 Comparison: Figure 3 displays Precision@100 scores, measuring the accuracy of top-ranked predictions.

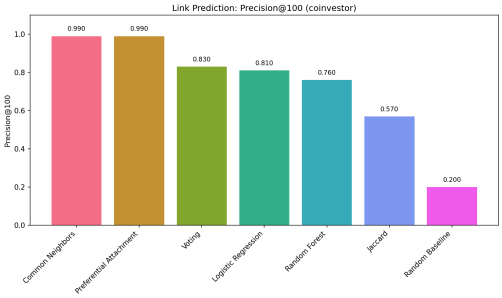

Figure 3: Precision@100 Comparison

Figure 3: Precision@100 scores by method. Preferential Attachment achieves near-perfect precision (0.990), indicating that all top-100 predictions are true positives. Common Neighbors match this (0.990). The sharp drop-off for Random Baseline (0.200) confirms that high precision requires meaningful predictive signal rather than random ranking.

The figure highlights the practical utility of topological heuristics for recommendation scenarios where only top-ranked predictions are considered. Preferential Attachment's near-perfect precision suggests that degree-based signals are particularly effective for identifying high-confidence predictions, even if overall ranking quality (as measured by PR-AUC) is slightly lower than Common Neighbors.

Summary of Visual Findings: the visualizations collectively support three main conclusions: - Consistent heuristic superiority: Across all visual representations, Common Neighbors and Preferential Attachment consistently occupy the top performance positions; - Clear separation from baseline: All methods show substantial visual separation from the Random Baseline, confirming meaningful predictive power; - ML model clustering: The three ML-based approaches (Voting, Logistic Regression, Random Forest) cluster together in the middle performance range, suggesting similar underlying predictive mechanisms despite different model architectures.

5. Discussion

Several aspects of the approach proved particularly effective in addressing the co-investment prediction task: - Temporal train-test splitting provides a realistic evaluation framework that respects the chronological nature of investment data. Training on 2021–2023 and testing on 2024 simulates a genuine prediction scenario where only historical information is available at decision time. This design avoids the data leakage that can result from random edge splitting, yielding performance estimates that better reflect real-world applicability; - Hard negative sampling enhances evaluation rigor by selecting negative test samples from node pairs with degree distributions similar to positive edges. This choice prevents artificial inflation of performance metrics that can occur when models trivially distinguish high-degree positive pairs from low-degree random negatives. The resulting PR-AUC values (0.38–0.54) represent realistic estimates of predictive performance under challenging conditions; - Graph-based heuristics demonstrate strong effectiveness despite their simplicity. Common Neighbors achieves the highest PR-AUC (0.542) with minimal computational overhead and full interpretability. These results support the insight that local network structure—specifically shared connections between investors—strongly predicts future collaboration. The finding also has practical implications: simple neighbor-counting algorithms can be used to identify co-investment opportunities without requiring complex machine learning infrastructure; - A modular code architecture facilitates systematic experimentation and supports reproducibility. Separating data loading, model training, and evaluation into distinct modules enables independent testing and straightforward extension to additional methods or datasets.

5.2 Challenges Encountered The development process presented several technical and methodological challenges that required careful resolution: - Entity name disambiguation posed a significant data quality challenge. Investor names appeared in multiple variants due to capitalization differences, legal suffixes (Inc., LLC, Capital, Partners, Ventures), and typographical inconsistencies. For example, "Andreessen Horowitz", "andreessen horowitz", and "Andreessen Horowitz LLC" may refer to the same entity. This issue is addressed through a normalization pipeline that standardizes case, removes common suffixes, and eliminates special characters. While effective in most cases, this heuristic approach may introduce errors for investors with legitimately distinct entities sharing similar names; - Class imbalance presented a fundamental challenge for supervised learning. The co-investor network contains approximately 38,000 edges among 4,500 nodes, implying that only about 0.4% of possible node pairs are connected. Training classifiers on such imbalanced data risks models that trivially predict the majority class (no edge). This is mitigated through balanced class weights in both Logistic Regression and Random Forest, and by using degree-biased negative sampling to generate training negatives that better resemble realistic non-edges; - Feature engineering limitations constrained the potential of machine learning models. The feature set comprises only five variables derived from network topology. Random Forest feature importance analysis indicates that two features (Jaccard Coefficient and Common Neighbors) account for 88.4% of the predictive signal, leaving the remaining features with minimal contribution. Richer features—such as investor industry focus, geographic proximity, historical investment patterns, or node embeddings—could enable ML models to capture complementary signals beyond simple neighborhood overlap; - Evaluation metric selection required balancing multiple considerations. PR-AUC is prioritized over ROC-AUC due to its suitability for imbalanced classification problems where the positive class is rare (Davis &

Goadrich, 2006). However, PR-AUC can be sensitive to the positive-to-negative ratio in the evaluation set, and different ratio choices may yield different relative rankings among methods.

5.3 Comparison with Expectations The initial hypotheses and their outcomes are summarized below: - Hypothesis 1: Machine learning models combining multiple features will outperform individual heuristics. Outcome: Rejected. Contrary to expectations, both Logistic Regression (PR-AUC = 0.414) and Random Forest (PR-AUC = 0.404) perform worse than Common Neighbors (0.542) and Preferential Attachment (0.514). This result, while surprising, is consistent with findings in the broader link prediction literature. Liben-Nowell and Kleinberg (2007) observed similar patterns in co-authorship networks, where simple topological measures often matched or exceeded complex alternatives. A likely explanation lies in the nature of the feature set: when ML models rely primarily on the same heuristic signals they are intended to improve upon, limited additional structure can be discovered beyond what those heuristics already capture. - Hypothesis 2: Ensemble methods will provide performance gains through predictor combination. Outcome: Partially supported. The Voting Ensemble (PR-AUC = 0.469) outperforms individual ML models (LR: 0.414, RF: 0.404) but does not exceed the best heuristics. This suggests that combining correlated predictors through simple averaging provides limited benefit when the base signals lack complementary information. - Hypothesis 3: The co-investment network contains exploitable structure for link prediction. Outcome: Confirmed. All evaluated methods substantially exceed the random baseline (PR-AUC = 0.170), with improvement factors ranging from 2.27× to 3.19×. This indicates that the network topology encodes meaningful signals about future co-investment relationships. - Hypothesis 4: High-degree investors will be easier to predict than low-degree investors. Outcome: Confirmed. Test edge difficulty analysis shows that 58.8% of test edges involve “Easy” pairs (both nodes with degree > 10), and these edges contribute disproportionately to high Precision@k values. Predictions for well-connected investors benefit from richer neighborhood information, enabling more accurate Common Neighbors estimates.

5.4 Limitations Several limitations constrain the generalizability and interpretation of the findings: - Dataset scope: The analysis is restricted to Series A and Series B funding rounds for North American startups between 2021 and 2024. Results may not generalize to other funding stages (seed, late-stage), geographic regions, or time periods. The 2021–2022 period coincided with historically high venture funding levels, followed by a significant market contraction in 2023–2024. These macroeconomic shifts may influence co-investment patterns in ways not captured by a static network model; - Temporal dynamics: The approach constructs a single aggregate network from historical data, ignoring the temporal evolution of investor relationships. In practice, co-investment propensity may depend on recency of past collaborations, changing investment theses, or evolving fund strategies. Time-aware models that discount older edges or incorporate temporal features may capture these dynamics more effectively; - Cold-start problem: The evaluation framework excludes test edges involving new investors not present in the training graph (0% “Very Hard” edges). This design choice, while necessary for fair evaluation of neighborhood-based methods, implies that results do not address cold-start scenarios where predictions are needed for investors with no historical activity. Addressing cold-start prediction would require external features such as investor profiles, fund characteristics, or LP relationships; - Feature limitations: The five-feature representation captures only local topological properties. Important factors influencing co-investment decisions—including sector specialization, investment stage preferences, geographic proximity, existing LP relationships, and partner-level connections—are not represented. Incorporating such features through node attributes or heterogeneous information networks could substantially improve ML model performance; - Evaluation assumptions: The hard negative sampling strategy, while more realistic than uniform random sampling, still involves arbitrary choices (e.g., 5:1 ratio, 10th–90th percentile degree range). Different sampling strategies might yield different relative performance rankings. In addition, all positive test edges are treated as equally important, whereas in practice some co-investments may be more valuable or surprising than others; - Network projection effects: The one-mode projection from the bipartite investor–startup network to the co-investor network introduces information loss. Two investors connected by a single shared investment are represented similarly to those connected by repeated co-occurrence in large syndicates. Edge weights partially address this, but finer-grained relationship modeling may improve prediction accuracy.

5.5 Surprising Findings Various unexpected results emerged from the analysis: - The magnitude of heuristic superiority exceeds expectations. While the link prediction literature documents cases where simple methods match complex alternatives, ML models with access to multiple features were expected to achieve at least comparable performance. The 34% performance gap between Common Neighbors and Random Forest (0.542 vs. 0.404) suggests that additional model complexity introduces noise rather than capturing useful patterns; - Preferential Attachment achieves near-perfect Precision@100 despite having lower PR-AUC than Common Neighbors. This dissociation between top-k precision and overall ranking quality indicates that degree-based signals are particularly effective for high-confidence predictions, even if they provide less discriminative rankings in the middle of the score distribution. For practical applications focused on identifying the most promising co-investment opportunities, Preferential Attachment may therefore be preferable despite its lower aggregate performance; - Jaccard Coefficient underperforms relative to raw Common Neighbors (PR-AUC 0.386 vs. 0.542), despite being the most important feature in the Random Forest model (51% importance). This suggests that while Jaccard provides useful information when combined with other features, its normalization by neighborhood size may be counterproductive as a standalone predictor in this domain. In VC networks, absolute counts of shared connections may be more predictive than normalized overlap ratios; - The absence of “Very Hard” test edges (0%) is unexpected given the four-year time span and the generally dynamic nature of the VC ecosystem. This indicates that the major investors active in 2024 were already present in the 2021–2023 training period, suggesting a relatively stable core of active market participants despite surface-level market volatility; - Feature importance concentration shows that 88.4% of the Random Forest predictive signal derives from only two features (Jaccard and Common Neighbors). The remaining features—Degree Sum, Degree Difference, and Preferential Attachment—contribute minimally despite their theoretical relevance. This concentration suggests that local neighborhood overlap dominates other structural signals in this prediction task, and that feature engineering efforts may be more effective if focused on qualitatively different information (e.g., node attributes, temporal patterns) rather than additional topological variants.

6. Conclusion

6.1 Summary This project addresses the problem of predicting future co-investment relationships between Venture Capital funds, framed as a link prediction task in a co-investor network. A complete analytical pipeline is developed, encompassing data preprocessing, network construction, heuristic computation, machine learning model training, and rigorous evaluation with hard negative sampling. Key Findings: the experimental results yield several principal findings: - Graph-based heuristics outperform machine learning models: Common Neighbors achieves the highest PR-AUC (0.542), surpassing Logistic Regression (0.414) and Random Forest (0.404) by margins of 30.9% and 34.2%, respectively. This confirms that simple topological measures effectively capture structural signals predictive of co-investment formation; - All methods demonstrate meaningful predictive power: every evaluated approach exceeds the random baseline (PR-AUC = 0.170) by factors ranging from 2.27× to 3.19×, indicating that the co-investor network contains exploitable structure for relationship prediction; - Top-ranked predictions are highly reliable: Preferential Attachment achieves near-perfect Precision@100 (0.990), and Common Neighbors reaches 0.990, indicating that the highest-confidence predictions are almost always correct; - Feature importance is highly concentrated: Random Forest analysis shows that Jaccard Coefficient (51.0%) and Common Neighbors (37.4%) account for 88.4% of predictive signal, helping explain why ML models fail to improve upon individual heuristics; - Evaluation difficulty affects performance: the majority of test edges (58.8%) involve well-connected investors (degree > 10), partially explaining the high precision values observed.

Recommendations: Which Model to Use When Based on the experimental findings, the following recommendations are provided.
Table 13: Model Selection Recommendations

Use Case	Recommended Method	Rationale
General prediction	CN	Highest PR-AUC (0.542), best overall ranking quality
High-confidence recommendations	PA	Ideal when only top predictions matter
Interpretable scoring	CN	Scores directly represent shared connection counts
Resource-constrained environments	CN or Jaccard	$O(d^2)$ complexity, no training required
Ensemble applications	Voting Ensemble	Best ML-based approach (PR-AUC=0.469), pools multiple signals
Feature importance analysis	Random Forest	Provides interpretable feature contributions

Decision Framework: - If interpretability is paramount: Use Common Neighbors. The score directly represents the number of shared co-investors, which is immediately understandable to domain experts; - If only top-k predictions are used: Use Preferential Attachment. Its near-perfect Precision@100 ensures that recommended pairs are highly likely to be true positives; - If computational resources are limited: Use heuristics (CN, Jaccard, PA). These require no training phase and can be computed on-demand for specific node pairs; - If integration with larger ML systems is required: Use Random Forest or Voting Ensemble. While underperforming heuristics, these models provide probability outputs suitable for downstream integration; - If the goal is research or benchmarking: Evaluate multiple methods. Performance rankings may differ across datasets, and comprehensive comparison provides robust conclusions.

6.2 Future Work Several directions for future research emerge from the limitations and findings of this study: - Feature Engineering Improvements: the current five-feature representation captures only local topological properties. A natural extension would be to incorporate additional node-level attributes that are readily available in investment databases: - Investor attributes: Fund size, investment stage preference (Seed, Series A, Series B), and primary sector focus (e.g., SaaS, FinTech, Healthcare). - Geographic features: Investor headquarter location, enabling analysis of whether geographic proximity influences co-investment likelihood. - Temporal features: Time since last co-investment between two investors, or the average recency of each investor's deals. These features could be extracted from the same Crunchbase dataset and integrated into the existing ML pipeline with minimal architectural changes.

- Additional Heuristics: the link prediction literature offers several additional heuristics that could be evaluated within our framework:
    - Adamic-Adar Index: A weighted variant of Common Neighbors that assigns higher importance to rare shared connections, computed as $AA(u,v)= \sum z \in N(u) \cap N(v)\ 1/\log|N(z)|$
    - Resource Allocation Index: Similar to Adamic-Adar but with inverse degree weighting rather than logarithmic.
    - Katz Index: A path-based measure that considers all paths between nodes, weighted by path length. Implementing these heuristics would require only minor additions to the existing models.py module and could provide insights into which structural patterns are most predictive in VC networks.
- Extended Evaluation: several straightforward extensions to the evaluation framework could strengthen the analysis:
    - Stratified performance reporting: Compute metrics separately for Easy, Medium, and Hard test edges to understand whether model performance varies with prediction difficulty.
    - Alternative cutoff years: Repeat the analysis with different train-test splits (e.g., 2022 cutoff) to assess robustness of findings across time periods.
    - Varying negative sampling ratios: Evaluate sensitivity of results to the 5:1 negative-to-positive ratio used in this study.
- Dataset Extensions: the methodology developed in this project could be applied to related prediction tasks:
    - Investor-startup prediction: Instead of predicting co-investment between investors, predict which investors will fund a given startup. This would utilize the bipartite network structure directly.
    - Lead investor prediction: Given a set of participating investors, predict which one will serve as lead investor for the round.
    - Geographic expansion: Apply the same methodology to VC networks in other regions (Europe, Asia) to examine whether the observed heuristic superiority generalizes across markets.

References

1. Adamic, L. A., & Adar, E. (2003). Friends and neighbors on the web. Social Networks, 25(3), 211-230. https://doi.org/10.1016/S0378-8733(03)00009-1
2. Barabási, A. L., & Albert, R. (1999). Emergence of scaling in random networks. Science, 286(5439), 509-512. https://doi.org/10.1126/science.286.5439.509
3. Dalle, J. M., den Besten, M., & Menon, C. (2017). Using Crunchbase for economic and managerial research. OECD Science, Technology and Industry Working Papers, 2017/08. https://doi.org/10.1787/6c418d60-en
4. Davis, J., & Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. Proceedings of the 23rd International Conference on Machine Learning, 233-240. https://doi.org/10.1145/1143844.1143874
5. Hasan, M. A., Chaoji, V., Salem, S., & Zaki, M. (2006). Link prediction using supervised learning. Proceedings of the SDM Workshop on Link Analysis, Counter-terrorism and Security, 798-805.
6. Hochberg, Y. V., Ljungqvist, A., & Lu, Y. (2007). Whom you know matters: Venture capital networks and investment performance. The Journal of Finance, 62(1), 251-301. https://doi.org/10.1111/j.1540-6261.2007.01207.x
7. Liben-Nowell, D., & Kleinberg, J. (2007). The link-prediction problem for social networks. Journal of the American Society for Information Science and Technology, 58(7), 1019-1031. https://doi.org/10.1002/asi.20591
8. Lü, L., & Zhou, T. (2011). Link prediction in complex networks: A survey. Physica A: Statistical Mechanics and its Applications, 390(6), 1150-1170. https://doi.org/10.1016/j.physa.2010.11.027
9. Rapoport, A. (1953). Spread of information through a population with socio-structural bias: I. Assumption of transitivity. The Bulletin of Mathematical Biophysics, 15(4), 523-533. https://doi.org/10.1007/BF02476440
10. Sorenson, O., & Stuart, T. E. (2001). Syndication networks and the spatial distribution of venture capital investments. American Journal of Sociology, 106(6), 1546-1588. https://doi.org/10.1086/321301
11. Hagberg, A. A., Schult, D. A., & Swart, P. J. (2008). Exploring network structure, dynamics, and function using NetworkX. Proceedings of the 7th Python in Science Conference (SciPy), 11-15.
12. Harris, C. R., Millman, K. J., van der Walt, S. J., et al. (2020). Array programming with NumPy. Nature, 585(7825), 357-362. https://doi.org/10.1038/s41586-020-2649-2
13. Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90-95. https://doi.org/10.1109/MCSE.2007.55
14. McKinney, W. (2010). Data structures for statistical computing in Python. Proceedings of the 9th Python in Science Conference (SciPy), 56-61.
15. Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825-2830.
16. Waskom, M. L. (2021). seaborn: Statistical data visualization. Journal of Open Source Software, 6(60), 3021. https://doi.org/10.21105/joss.03021

Dataset used: Crunchbase Funding Data. Source: Crunchbase Inc. Access: https://www.crunchbase.com Dataset Name: funding_master.csv

Libraries and Frameworks: Python3.11; pandas≥1.5.0; NumPy≥1.23.0; NetworkX≥3.0; scikit-learn≥1.2.0; Matplotlib≥3.6.0; seaborn≥0.12.0

Appendices

Appendix A: Additional Results

No additional results are included, as all relevant figures and tables are already presented in Section 4 (Results)

Appendix B: Code Repository

GitHub Repository: https://github.com/AlbertoMigliorati/Predictive-Link-Modeling-in-Venture-Capital-Networks.git

Repository Structure

```
Predictive Link Modeling in VC Networks/
├── AI_USAGE.md
├── environment.yml
├── main.py
├── project_report.md
├── README.md
├── data/
│   └── raw/
│       └── funding_master.csv
├── results/
│   ├── metrics_heatmap.png
│   ├── pr_auc.png
│   ├── precision_at_100.png
│
```

```
│   ├── results_coinvestor_20251217_085209.json
│   └── top_50_predictions.txt
└── scr/
    ├── __init__.py
    ├── data_loader.py
    ├── evaluation.py
    └── models.py
```

Installation Instructions

```
git clone https://github.com/AlbertoMigliorati/Predictive-Link-Modeling-in-Venture-Capital-Networks.git
cd Predictive Link Modeling in VC Networks
conda env create -f environment.yml
conda activate vc-link-prediction
```

Reproducing Results

```
python main.py
```