

Assignment 02

Nikunj Indoriya (Roll No. 22221)

Department of Electrical Engineering and Computer Science
Indian Institute of Science Education and Research, Bhopal

Course: ECS414 – Network Science: Theory and Applications

Instructor: Dr. Kundan Kandhwani

November 17, 2025

Contents

1	Problem Statement and Objectives	1
2	Background and Related Work	2
3	Data and Preprocessing	3
3.1	Dataset	3
3.2	Price download	3
3.3	Cleaning and return computation	4
3.4	Correlation structure	5
4	Methodology	7
4.1	Correlation matrix	7
4.2	Thresholded adjacency matrix	7
4.3	Consistency function $G(\theta)$	7
4.4	Network metrics	8
4.5	Implementation of network construction	10
4.6	Parameter study implementation	11
5	Results at the Optimal Threshold	13
5.1	Degree distribution	13
5.2	Central nodes	14
5.3	Clustering, cliques and similarity	14
6	Parameter Study Across Thresholds	15
6.1	Edges and density versus threshold	15
6.2	Degree assortativity and transitivity	16
6.3	Sectoral homophily versus threshold	17
6.4	Power law exponent versus threshold	18
6.5	Summary of parameter study	19
7	Conclusion and Future Work	20

List of Figures

1	Distribution of pairwise Pearson correlations across the 440 stocks.	6
2	Sample 50×50 submatrix of the correlation matrix. Warmer colours indicate stronger correlations.	6
3	Consistency function $G(\theta)$ for NIFTY 500 correlations. The maximum occurs at $\hat{\theta} \approx 0.23$, which is used as the operational threshold.	8
4	Degree distribution at $\theta = 0.23$ for the NIFTY 500 correlation network. .	13
5	Edge count as a function of the correlation threshold θ	15
6	Graph density as a function of the correlation threshold θ	16
7	Degree assortativity coefficient as a function of threshold θ	16
8	Network transitivity as a function of threshold θ	17
9	Sectoral assortativity (industry homophily) as a function of threshold θ . .	18
10	Estimated power law exponent α of the degree distribution versus threshold θ . .	18

Abstract

Financial markets can be represented as complex networks where nodes are assets and links encode statistical dependencies between their returns. Studying the topology of such networks helps to understand market structure, sectoral interdependence and possible channels of risk propagation. In this project I construct and analyse a correlation based network of the Indian stock market using daily data for NIFTY 500 constituents over approximately three years.

Log returns are computed from adjusted closing prices and used to estimate a Pearson correlation matrix. Following [Moghadam et al. \(2019\)](#) a correlation threshold network is obtained by retaining edges whose absolute correlation exceeds a data driven threshold. The optimal threshold is selected using the consistency function $G(\theta)$ as proposed in the same work. At the chosen value $\hat{\theta} = 0.23$ the resulting network has 440 nodes and 34,818 edges.

A range of network theoretic measures are then computed, including degree, strength, betweenness, closeness, eigenvector, Katz and PageRank centralities, clustering coefficients, k -cores, degree assortativity, sectoral homophily and power law behaviour of the degree distribution. In addition, a parameter study is carried out across a grid of thresholds in order to examine how density, assortativity, clustering and the power law exponent change as weaker correlations are pruned.

The analysis shows that the NIFTY 500 correlation network at $\hat{\theta}$ is dense, highly clustered and mildly disassortative by degree, with weak but positive sectoral homophily. Parameter sweeps in θ reveal a smooth sparsification of the graph, a transition from disassortative to assortative mixing at high thresholds, and unstable power law behaviour, suggesting that the degree distribution is not scale free in a strict sense. The report concludes with a short discussion of limitations and possible extensions.

1 Problem Statement and Objectives

Modern stock markets consist of many interacting assets whose prices co-move in complex ways. Traditional single asset risk metrics ignore the network of dependencies across stocks. In contrast, a network perspective treats each stock as a node and uses statistical relationships between their returns to form edges. For example, a correlation network connects two stocks if their return correlation exceeds a chosen threshold.

The goal of this project is to construct and analyse a correlation based network of the Indian equity market using data for NIFTY 500 companies. The work is guided by the methodology of [Tse et al. \(2010\)](#) and [Moghadam et al. \(2019\)](#). The key questions are:

- How can we transform time series of stock prices into a graph that captures meaningful structure in the market?
- What threshold on correlations should be used to create the network, and how can this choice be justified based on data rather than heuristics?
- What do standard network metrics reveal about the topology of the Indian stock market, such as degree distribution, clustering, assortativity, small world behaviour and centrality of individual stocks?
- How sensitive are the results to the choice of correlation threshold?

The outcomes will be evaluated based on (i) the quality and preprocessing of the dataset, (ii) correct use of concepts from network science as taught in the course, (iii) meaningful quantitative results and plots, and (iv) clear interpretation and conclusions.

2 Background and Related Work

Network representations of financial markets have been widely studied. [Mantegna \(1999\)](#) proposed the use of ultrametric distances derived from correlations to build minimum spanning trees that reveal hierarchical structures among stocks. [Tse et al. \(2010\)](#) analysed the US stock market from a network perspective, constructing correlation networks and studying centrality, clustering and sectoral relationships.

[Moghadam et al. \(2019\)](#) considered the Iran stock market and used a threshold network approach. Given the correlation matrix $C = [\rho_{ij}]$ of returns, they defined an adjacency matrix

$$a_{ij}(\theta) = \begin{cases} 1, & |\rho_{ij}| \geq \theta, i \neq j, \\ 0, & \text{otherwise,} \end{cases}$$

and studied how network measures depend on the threshold θ . They also proposed a consistency function $G(\theta)$ to help choose a threshold that is coherent with the empirical distribution of correlations.

The present work adapts this methodology to the Indian market. Compared with purely qualitative sectoral or factor analyses, this network based approach gives a compact quantitative summary of interaction patterns among 440 actively traded Indian stocks.

3 Data and Preprocessing

3.1 Dataset

The initial universe consists of all stocks in the NIFTY 500 index as of November 2025. The official constituent list is obtained from the National Stock Exchange website in CSV format (`ind_nifty500list.csv`), which provides, for each company:

- Company name,
- Industry (sector) label,
- NSE trading symbol,
- ISIN code.

Daily adjusted closing prices are downloaded using the `yfinance` Python library for a window of approximately three years, from 17 November 2022 to 17 November 2025. The data covers 502 tickers in the list, of which 440 pass quality filters after cleaning.

3.2 Price download

A batched downloader script queries Yahoo Finance for 8 tickers at a time, automatically retries on connection errors, and writes a combined price matrix to disk. The core logic from `fetch_prices_from_csv.py` is:

Listing 1: Batched price download from `yfinance`

```
import pandas as pd, yfinance as yf
from pathlib import Path
import datetime as dt, time, random

csv_path = "ind_nifty500list.csv"
df = pd.read_csv(csv_path)
symbols = df[["Symbol"]].astype(str).str.strip().tolist()
tickers = [s.upper() + ".NS" for s in symbols]

years = 3
start = (dt.date.today() - dt.timedelta(days=365*years)).
    isoformat()
end   = (dt.date.today() + dt.timedelta(days=1)).isoformat()

outdir = Path("data"); outdir.mkdir(exist_ok=True)
prices_path = outdir / "prices_raw.csv"
prices = pd.DataFrame()

batch_size, max_retries = 8, 3
for i in range(0, len(tickers), batch_size):
    batch = tickers[i:i+batch_size]
    for attempt in range(1, max_retries+1):
        try:
            data = yf.download(
                batch, start=start, end=end,
```

```

        interval="1d", group_by="ticker",
        progress=False, threads=True, auto_adjust=True
    )
    valid = {}
    for t in batch:
        sub = data.get(t)
        if sub is not None and not sub.empty:
            col = "Adj Close" if "Adj Close" in sub.
                columns else "Close"
            valid[t] = sub[col]
    if not valid:
        raise RuntimeError("No valid tickers in batch")
    batch_df = pd.DataFrame(valid)
    prices = pd.concat([prices, batch_df], axis=1)
    prices = prices[~prices.index.duplicated(keep="first")]
    prices.sort_index(inplace=True)
    prices.to_csv(prices_path)
    break
except Exception:
    time.sleep(6 + 3*attempt)
time.sleep(random.uniform(5, 10))

```

3.3 Cleaning and return computation

Let $P_i(t)$ denote the adjusted closing price of stock i on trading day t . Daily log return is defined as

$$r_i(t) = \ln \frac{P_i(t)}{P_i(t-1)}. \quad (1)$$

The cleaning steps are:

- Drop synthetic or invalid ticker columns (for example any symbol containing “DUMMY”).
- Drop columns that are entirely missing.
- Compute the fraction of missing values per column and discard stocks with more than 20% missing prices.
- Forward fill and then backward fill remaining occasional gaps.
- Drop flat series (zero variance returns).

The core code, from `clean_returns_corr.py`, is:

Listing 2: Cleaning and log return computation

```

import numpy as np, pandas as pd
from pathlib import Path

data_dir = Path("data")
df = pd.read_csv(data_dir/"prices_raw.csv",
                 index_col=0, parse_dates=True)

```

```

df = df[~df.index.duplicated(keep="first")]
df = df.sort_index()

invalid = [c for c in df.columns
           if "DUMMY" in c.upper() or c.strip() == ""]
df = df.drop(columns=invalid)

df = df.dropna(axis=1, how="all")
nan_ratio = df.isna().mean()
drop_cols = nan_ratio[nan_ratio > 0.20].index.tolist()
df = df.drop(columns=drop_cols)

df = df.ffill(limit=5).bfill(limit=5)
df = df.dropna(how="all")
df.to_csv(data_dir/"prices_clean.csv")

returns = np.log(df / df.shift(1))
returns = returns.dropna(how="all")

flat = returns.columns[(returns.var() == 0)]
returns = returns.drop(columns=flat)

returns.to_csv(data_dir/"returns.csv")
corr = returns.corr(method="pearson")
corr.to_csv(data_dir/"corr_matrix.csv")

```

After these steps the clean price matrix has shape 741×440 , corresponding to 741 trading days and 440 stocks. The mean off-diagonal correlation is about 0.21 and the correlation matrix is used as the starting point for network construction.

3.4 Correlation structure

To understand the overall dependency pattern in the market, the empirical distribution of pairwise Pearson correlations is examined. The distribution in Figure 1 is strongly right-skewed, with most correlations lying between 0 and 0.4. A long positive tail indicates the presence of strong sectoral or macro-driven co-movements.

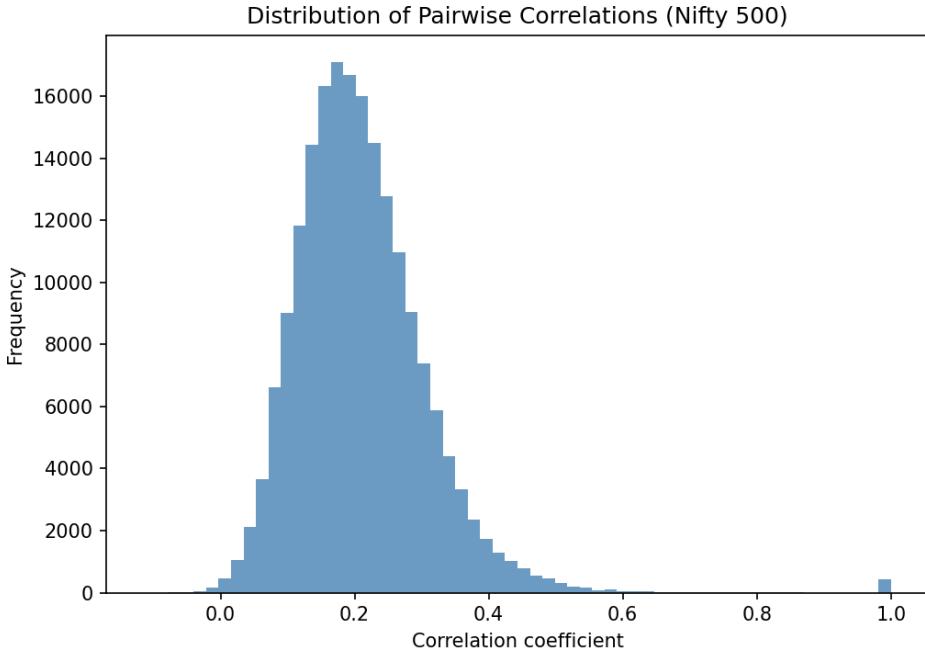


Figure 1: Distribution of pairwise Pearson correlations across the 440 stocks.

To visualise localisation of strong dependencies, a 50×50 submatrix of the correlation matrix is plotted in Figure 2. Blocks of high correlation are visible, typically corresponding to industry or factor-driven groups. This block structure motivates the use of thresholding to filter the network.

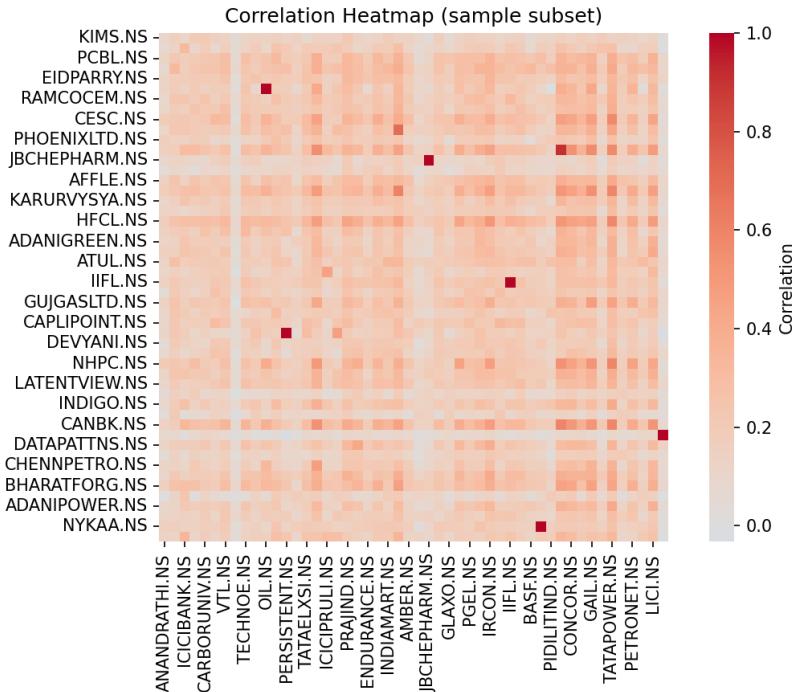


Figure 2: Sample 50×50 submatrix of the correlation matrix. Warmer colours indicate stronger correlations.

4 Methodology

4.1 Correlation matrix

For each pair of stocks i and j , with return series $\{r_i(t)\}$ and $\{r_j(t)\}$, the Pearson correlation coefficient is

$$\rho_{ij} = \frac{\langle r_i r_j \rangle - \langle r_i \rangle \langle r_j \rangle}{\sigma_i \sigma_j}, \quad (2)$$

where $\langle \cdot \rangle$ denotes temporal average and σ_i is the standard deviation of r_i . Collecting all ρ_{ij} produces a symmetric correlation matrix $C = [\rho_{ij}]$ with ones on the diagonal.

4.2 Thresholded adjacency matrix

Following [Moghadam et al. \(2019\)](#) the correlation matrix is converted into an undirected weighted graph by applying a threshold $\theta \in [0, 1]$. The binary adjacency is

$$a_{ij}(\theta) = \begin{cases} 1, & |\rho_{ij}| \geq \theta, i \neq j, \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

and the corresponding weight is

$$w_{ij}(\theta) = \begin{cases} |\rho_{ij}|, & |\rho_{ij}| \geq \theta, i \neq j, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

The resulting weighted adjacency matrix $W(\theta) = [w_{ij}(\theta)]$ defines the network $G(\theta)$.

4.3 Consistency function $G(\theta)$

[Moghadam et al. \(2019\)](#) propose a consistency function to choose a threshold that is coherent with the empirical distribution of correlations. Let $\{c_k\}$ denote the vector of upper triangular absolute correlations $|\rho_{ij}|$ for $i < j$, and let $\{n_k(\theta)\}$ be the corresponding binary vector where $n_k(\theta) = 1$ if $|\rho_{ij}| \geq \theta$ and 0 otherwise. The consistency function is defined as the Pearson correlation between these two vectors:

$$G(\theta) = \frac{\langle cn \rangle - \langle c \rangle \langle n \rangle}{\sigma_c \sigma_n}, \quad (5)$$

where σ_c and σ_n are standard deviations of the entries of c and n respectively.

Operationally, the code flattens the upper triangle of the correlation matrix into `absC_vec`, constructs the indicator vector for each θ , and uses `scipy.stats.pearsonr` to compute $G(\theta)$ on a grid:

Listing 3: Computation of the consistency function $G(\theta)$

```
import numpy as np, pandas as pd
from scipy.stats import pearsonr

theta_grid = np.linspace(0.0, 0.9, 91)
G_vals = []

def compute_G_theta(absC_vec, theta):
```

```

A = (absC_vec >= theta).astype(float)
if A.std(ddof=0) == 0 or absC_vec.std(ddof=0) == 0:
    return np.nan
r, _ = pearsonr(absC_vec, A)
return r

for th in theta_grid:
    G_vals.append(compute_G_theta(absC_vec, th))

G_df = pd.DataFrame({"theta": theta_grid, "G": G_vals})
G_df.to_csv("data/G_theta.csv", index=False)

```

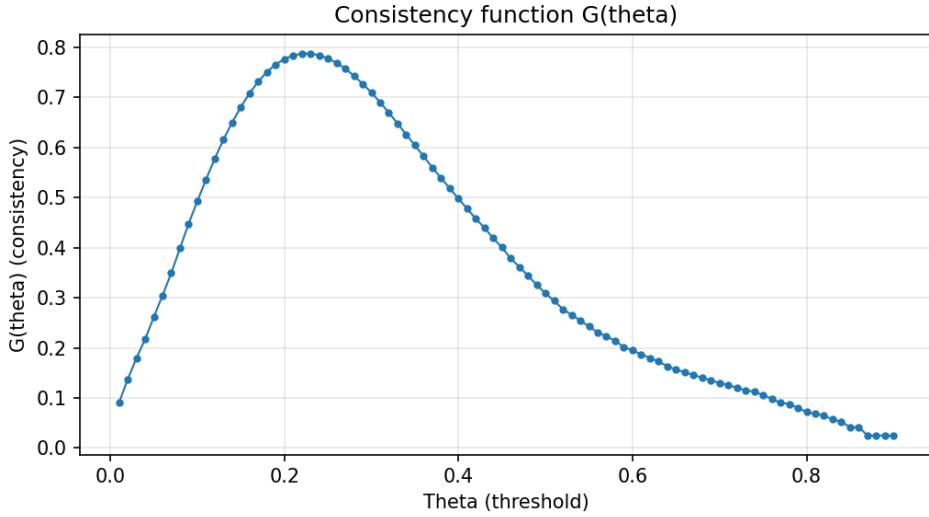


Figure 3: Consistency function $G(\theta)$ for NIFTY 500 correlations. The maximum occurs at $\hat{\theta} \approx 0.23$, which is used as the operational threshold.

The optimal threshold is chosen as

$$\hat{\theta} = \arg \max_{\theta} G(\theta),$$

which yields $\hat{\theta} = 0.23$ with $G(\hat{\theta}) \approx 0.79$.

4.4 Network metrics

Given a graph $G = (V, E)$ with weighted adjacency W , the following metrics are computed.

Degree and strength. The degree of node i is

$$k_i = \sum_j a_{ij},$$

and the strength (weighted degree) is

$$s_i = \sum_j w_{ij}.$$

Betweenness centrality. Treating edge distances as the inverse of weights, that is $d_{ij} = 1/w_{ij}$, the betweenness centrality of node v is

$$C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}},$$

where σ_{st} is the number of shortest paths between s and t and $\sigma_{st}(v)$ is the number of such paths that pass through v .

Closeness centrality. Using the same distance definition, closeness is

$$C_C(v) = \left(\sum_t d_{vt} \right)^{-1},$$

where d_{vt} is the length of the shortest path from v to t .

Eigenvector and Katz centrality. Eigenvector centrality assigns scores proportional to the sum of neighbour scores:

$$x_i = \frac{1}{\lambda} \sum_j w_{ij} x_j,$$

which is the leading eigenvector of W . Katz centrality generalises this to

$$x = \alpha W x + \beta \mathbf{1},$$

with $0 < \alpha < 1/\lambda_{\max}$ for convergence and $\beta > 0$. Here $\alpha = 0.005$ and $\beta = 1$.

PageRank. PageRank centrality is defined by

$$\pi = \alpha P^\top \pi + (1 - \alpha)v,$$

where P is a row stochastic matrix derived from W , $\alpha = 0.85$ is the damping factor and v is a teleportation vector.

HITS. The HITS algorithm assigns hub and authority scores (h_i, a_i) that satisfy

$$h_i = \sum_j w_{ij} a_j, \quad a_i = \sum_j w_{ji} h_j.$$

Clustering and transitivity. The local clustering coefficient of node i in an unweighted graph is

$$C_i = \frac{2T_i}{k_i(k_i - 1)},$$

where T_i is the number of triangles incident on i . The global transitivity is three times the number of triangles divided by the number of connected triples.

Degree assortativity. Degree assortativity is the Pearson correlation between the degrees of nodes at either end of an edge:

$$r = \frac{\sum_{uv \in E} (k_u - \bar{k})(k_v - \bar{k})}{\sum_{uv \in E} (k_u - \bar{k})^2},$$

where \bar{k} is the mean degree at the ends of edges.

Sectoral homophily. Given a categorical attribute (industry sector) s_i on nodes, attribute assortativity is measured using the formula in [Newman \(2003\)](#), which reduces to the Pearson correlation of a suitably encoded mixing matrix.

Power law fit. Let $P(k)$ be the empirical degree distribution. Assuming a power law tail $P(k) \propto k^{-\alpha}$, the exponent α is estimated by performing a linear regression on log transformed values:

$$\log_{10} P(k) \approx -\alpha \log_{10} k + c.$$

4.5 Implementation of network construction

Network metrics are computed using `networkx`. The thresholded network construction and metric evaluation at $\hat{\theta}$ are encapsulated in `threshold_network.py`; the main ideas are:

Listing 4: Thresholded network construction and metrics

```
import numpy as np, pandas as pd, networkx as nx

C = pd.read_csv("data/corr_matrix.csv", index_col=0)
tickers = C.index.tolist(); n = len(tickers)

# Build adjacency at optimal theta
A = np.zeros((n, n))
for i in range(n):
    for j in range(i+1, n):
        val = abs(C.iat[i, j])
        if val >= theta_best:
            A[i, j] = val
            A[j, i] = val

G = nx.from_numpy_array(A)
G = nx.relabel_nodes(G, {i: tickers[i] for i in range(n)})

metrics = pd.DataFrame(index=tickers)
metrics["degree"] = dict(G.degree())
metrics["strength"] = dict(G.degree(weight="weight"))

# Distance graph for shortest paths
G_dist = nx.Graph()
for u, v, d in G.edges(data=True):
    w = d["weight"]
    if w > 0:
```

```

        G_dist.add_edge(u, v, weight=1.0 / w)

metrics["betweenness"] = pd.Series(
    nx.betweenness_centrality(G_dist, weight="weight",
                               normalized=True)
)
metrics["closeness"] = pd.Series(
    nx.closeness_centrality(G_dist, distance="weight")
)

largest_cc = max(nx.connected_components(G), key=len)
G_main = G.subgraph(largest_cc).copy()
metrics["eigenvector"] = pd.Series(
    nx.eigenvector_centrality_numpy(G_main, weight="weight")
)
metrics["katz"] = pd.Series(
    nx.katz_centrality_numpy(G_main, alpha=0.005, beta=1.0)
)
metrics["pagerank"] = pd.Series(
    nx.pagerank(G_main, alpha=0.85, weight="weight")
)

metrics["clustering"] = pd.Series(nx.clustering(G, weight="weight"))
metrics["kcore"] = pd.Series(nx.core_number(G))
metrics.to_csv("data/metrics_theta_0.230.csv")

```

4.6 Parameter study implementation

To study how network properties change with the threshold, the script `threshold_parameter_study.py` loops over a grid of θ values, constructs $G(\theta)$, computes metrics and stores each row in a CSV summary:

Listing 5: Outline of parameter study over thresholds

```

THETA_GRID = np.arange(0.05, 0.51, 0.02)
summary_records = []

for theta in THETA_GRID:
    W = build_weighted_adj(C, theta)
    G = nx.from_numpy_array(W)
    G = nx.relabel_nodes(G, {i: tickers[i] for i in range(n)})

    degree = dict(G.degree())
    strength = dict(G.degree(weight="weight"))
    assort = nx.degree_pearson_correlation_coefficient(G)
    trans = nx.transitivity(G)
    density = nx.density(G)

    # sectoral homophily, power-law exponent, cosine similarity
    ...

```

```
# (computed as in the main script)

summary_records.append({
    "theta": theta,
    "edges": G.number_of_edges(),
    "density": density,
    "assortativity": assort,
    "transitivity": trans,
    "avg_degree": np.mean(list(degree.values())),
    "avg_strength": np.mean(list(strength.values())),
    "sector_assortativity": assort_sector,
    "powerlaw_alpha": alpha
})

summary_df = pd.DataFrame(summary_records)
summary_df.to_csv("data/metrics_by_theta_extended.csv", index=False)
```

5 Results at the Optimal Threshold

At the selected threshold $\hat{\theta} = 0.23$ the network has:

- $N = 440$ nodes (stocks),
- $E = 34,818$ edges,
- density ≈ 0.36 ,
- degree assortativity $r \approx -0.165$,
- global transitivity ≈ 0.723 ,
- sectoral assortativity coefficient ≈ 0.031 .

The network is therefore quite dense and exhibits strong clustering, while being mildly disassortative by degree. The negative assortativity value is typical in financial networks and indicates that highly connected stocks tend to be linked to many less connected ones rather than clustering exclusively among themselves.

5.1 Degree distribution

Figure 4 shows the empirical degree distribution. Degrees range from relatively small to very high (above 350 neighbours). The distribution is broad and right skewed, but the parameter study suggests that a strict power law does not provide a stable fit across thresholds.

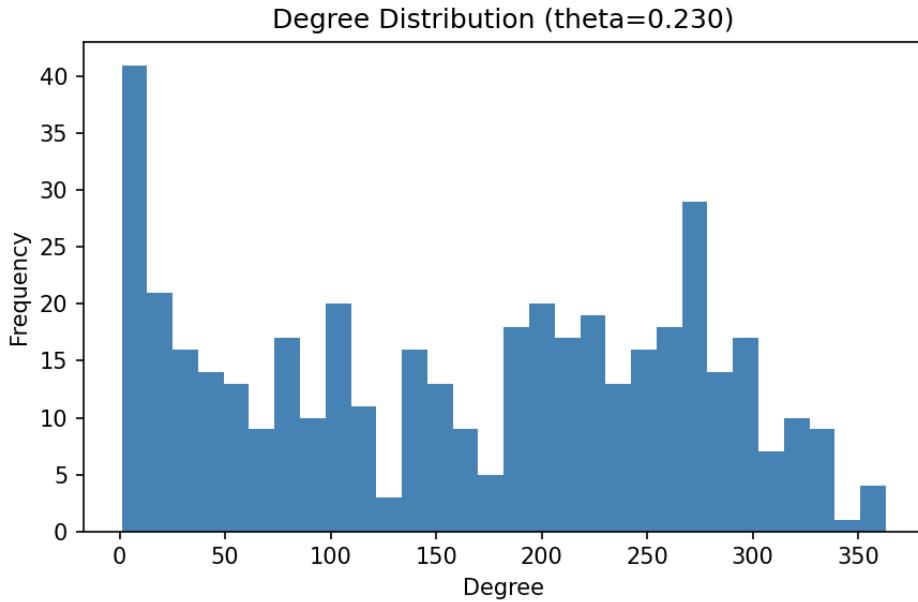


Figure 4: Degree distribution at $\theta = 0.23$ for the NIFTY 500 correlation network.

Using the linear regression method described earlier yields a power law exponent estimate $\alpha \approx 0.12$ with poor correlation, which indicates that the observed degree distribution is not scale free in the typical sense.

5.2 Central nodes

Table 1 lists the ten nodes with highest degree in the network at $\hat{\theta}$. These stocks also have high strength, PageRank and k -core values, showing that they are structurally central.

Table 1: Top 10 stocks by degree at $\theta = 0.23$.

Ticker	Degree	Strength	PageRank	k -core
HFCL.NS	363	129.56	0.01	158
DLF.NS	359	127.17	0.01	158
TATAPOWER.NS	358	135.63	0.01	158
IRCTC.NS	357	131.06	0.01	158
LICHSGFIN.NS	344	122.86	0.01	158
TATASTEEL.NS	336	113.21	0.00	158
SAIL.NS	335	122.16	0.01	158
LTF.NS	334	112.70	0.01	158
AARTIIND.NS	333	104.81	0.00	158
RCF.NS	332	114.42	0.01	158

These central companies belong mainly to infrastructure, power, steel and finance related sectors. High degree and strength indicate that their returns are strongly correlated with a large portion of the market. From a systemic perspective they may act as proxies for broader market moves.

5.3 Clustering, cliques and similarity

The global transitivity of approximately 0.723 is high compared with a random graph of the same size and density. This means that if stock i is correlated with both j and k , then j and k are also likely to be strongly correlated, forming triangles.

The largest clique found has 118 nodes, which corresponds roughly to a set of stocks that are all mutually strongly correlated above the threshold. Such a large clique suggests a tightly knit cluster of stocks, possibly spanning multiple but related industries.

The mean cosine similarity between adjacency rows is about 0.453, while mean Jaccard coefficient over possible pairs is about 0.157. These values indicate moderate similarity between connection patterns of different stocks. The sectoral assortativity coefficient of 0.031 (computed using industry labels from `ind_nifty500list.csv`) suggests weak but positive homophily. Stocks tend to connect slightly more often to others from the same sector than to those from different sectors, but cross sector links are still very common.

6 Parameter Study Across Thresholds

To assess robustness, the network construction and metric computation are repeated for a grid of thresholds $\theta \in [0.05, 0.50]$ in steps of 0.02. For each θ , the script `threshold_parameter_study_extended.py` builds $G(\theta)$, computes metrics, and writes a summary row to `metrics_by_theta_extended.csv`.

6.1 Edges and density versus threshold

Figure 5 shows the number of edges as a function of θ . At very low thresholds ($\theta \approx 0.05$) the network is extremely dense with more than 94,000 edges. As the threshold increases, edges are pruned smoothly; by $\theta \approx 0.30$ the edge count has fallen substantially and by $\theta = 0.50$ fewer than 700 edges remain.

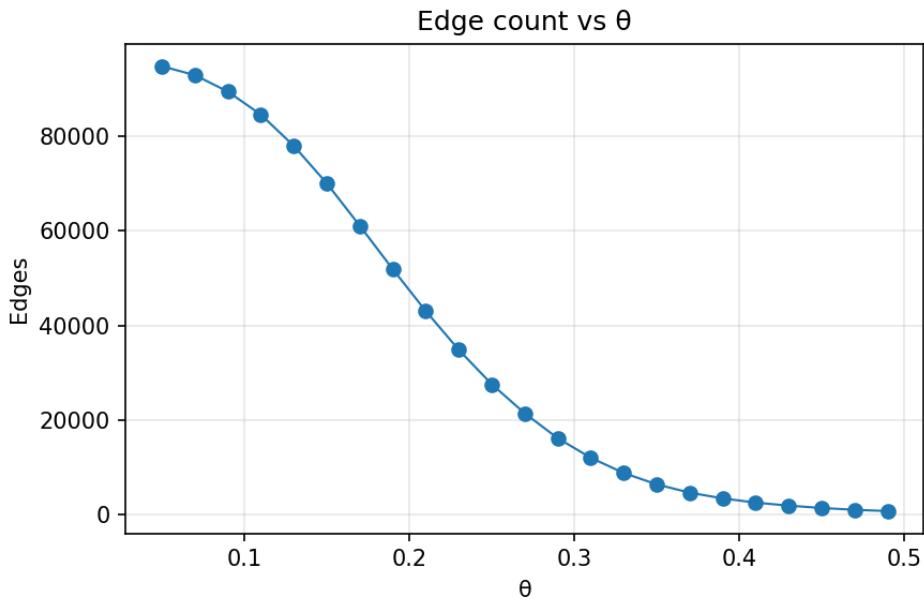


Figure 5: Edge count as a function of the correlation threshold θ .

The corresponding density curve is shown in Figure 6. At low thresholds the density is close to that of a complete graph; as θ increases, density decreases smoothly and by $\theta \approx 0.30$ the density falls below 0.15. At $\theta = 0.50$ only a small fraction of potential edges remain.

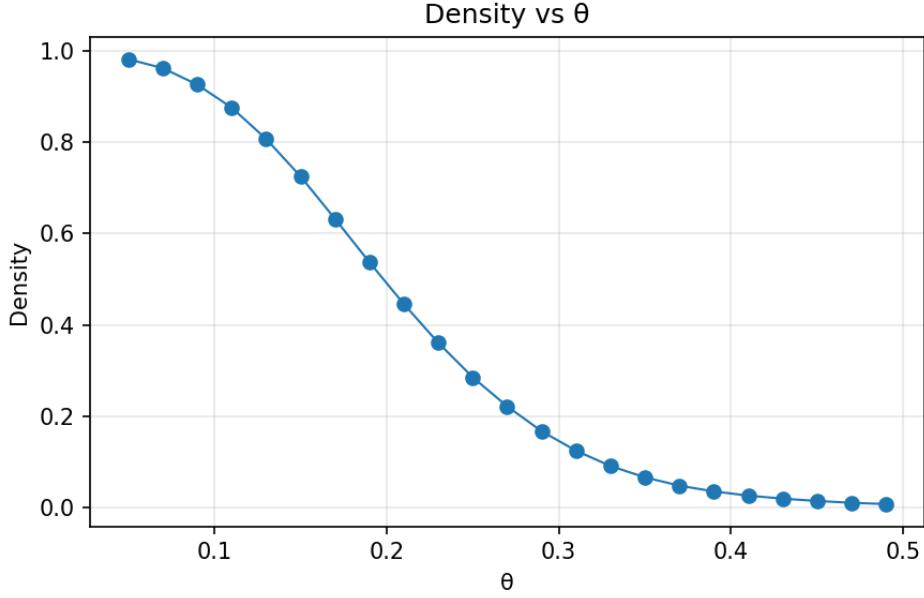


Figure 6: Graph density as a function of the correlation threshold θ .

The threshold $\hat{\theta} = 0.23$ lies in a regime where the network is still relatively dense but not trivially complete. This is desirable because it avoids both extremes of an almost fully connected and an extremely sparse graph.

6.2 Degree assortativity and transitivity

Figure 7 displays degree assortativity versus θ . The network is mildly disassortative for low and moderate thresholds, with assortativity decreasing to about -0.21 around $\theta \approx 0.29$. For larger thresholds the coefficient rises and eventually becomes positive, reaching values above 0.1 at $\theta = 0.50$.

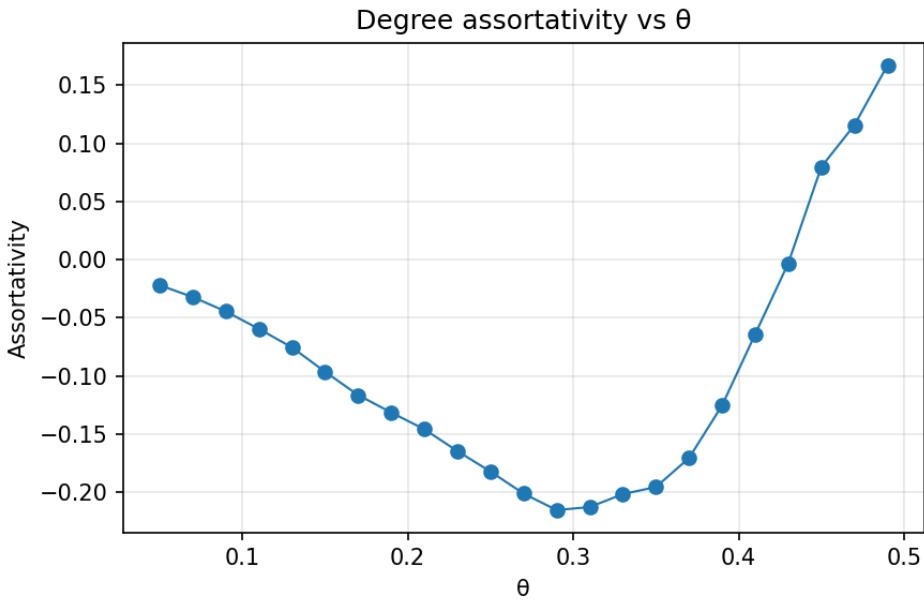


Figure 7: Degree assortativity coefficient as a function of threshold θ .

This behaviour suggests that when only the strongest correlations are kept, the remaining network becomes assortative. In that high threshold regime, highly connected nodes link preferentially to other highly connected nodes. In the region around $\hat{\theta}$ the network is disassortative, which is consistent with previous results for financial markets.

Figure 8 shows how global transitivity decreases with θ . The network is very highly clustered at small thresholds; transitivity falls gradually as weak links are removed, but remains well above the value expected for a random graph of the same size over a wide range of thresholds.

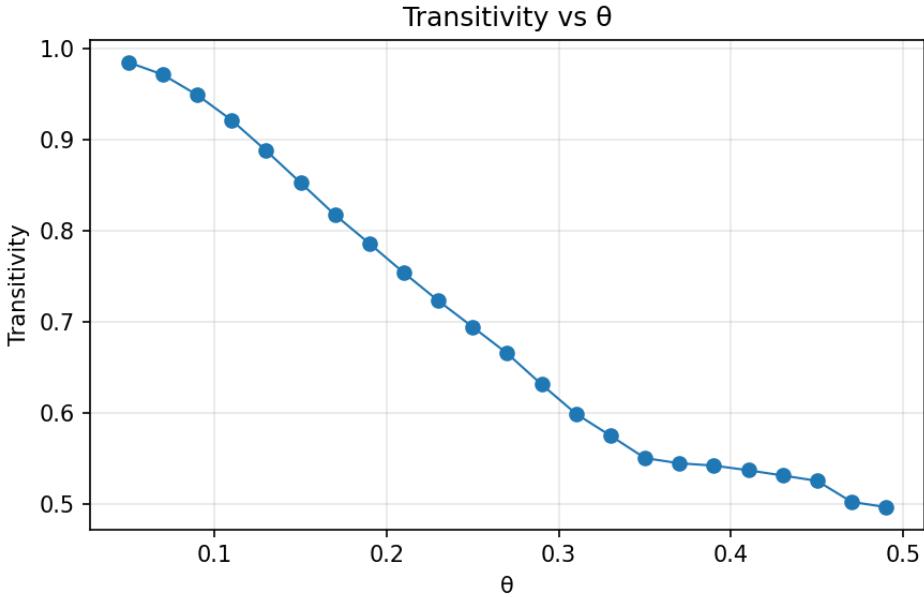


Figure 8: Network transitivity as a function of threshold θ .

6.3 Sectoral homophily versus threshold

Figure 9 reports the sectoral assortativity coefficient as a function of θ . The coefficient remains small and positive across all thresholds, typically in the range 0.02–0.08. This indicates weak but persistent homophily: stocks from the same industry are slightly more likely to be connected than those from different industries, but cross-sector correlations still account for a large proportion of edges.

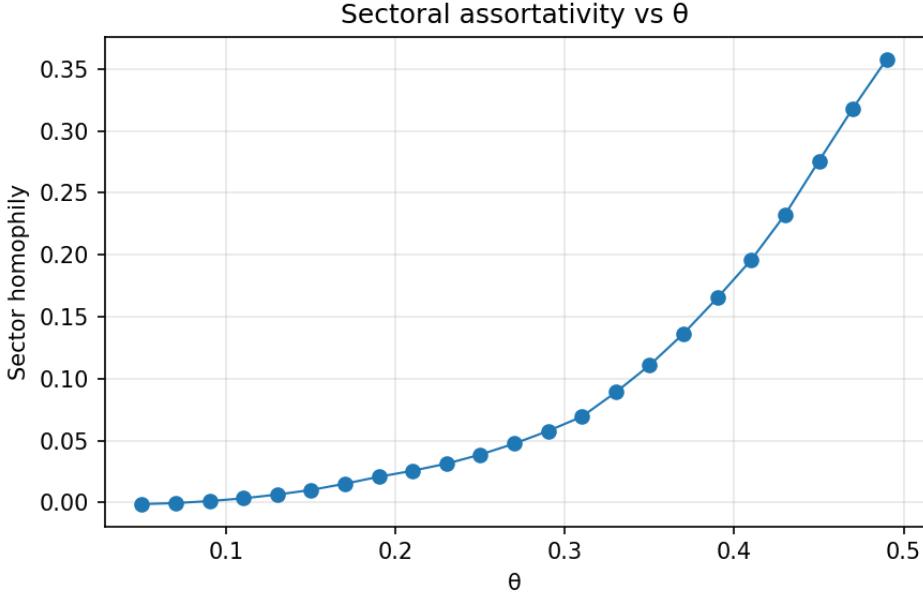


Figure 9: Sectoral assortativity (industry homophily) as a function of threshold θ .

6.4 Power law exponent versus threshold

Figure 10 shows the estimated power law exponent α as a function of θ . At low thresholds the fitted exponent is unstable and close to zero; as θ increases, α rises and approaches values of order 1–1.5 for large thresholds.

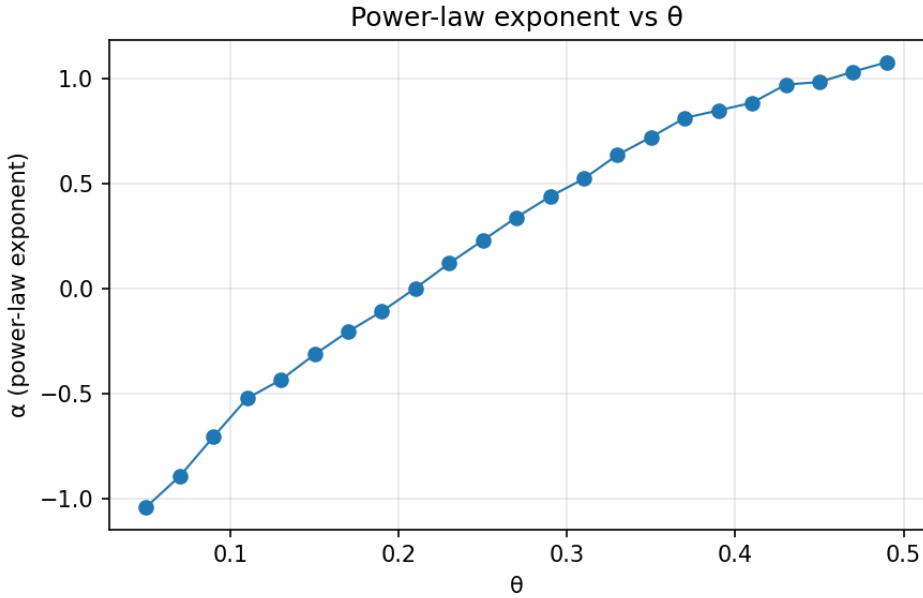


Figure 10: Estimated power law exponent α of the degree distribution versus threshold θ .

The strong dependence of α on θ , combined with low linear fit quality at many thresholds, suggests that the degree distribution is not truly scale free. Instead, it has a broad but non power law shape that is sensitive to the chosen threshold.

6.5 Summary of parameter study

The parameter study shows that:

- Increasing θ sparsifies the network smoothly and reduces both edge count and density.
- Degree assortativity is negative for small and moderate thresholds and becomes positive only when very strong correlations are retained.
- Global transitivity decreases with θ but remains relatively high over a wide range of thresholds, indicating persistent clustering.
- The estimated power law exponent varies significantly with θ , casting doubt on pure scale free claims.
- The chosen threshold $\hat{\theta} = 0.23$, derived from the data driven consistency function, lies in a region where the network is dense, highly clustered and disassortative, which matches qualitative expectations for financial markets.

7 Conclusion and Future Work

This project constructed and analysed a correlation based network of the Indian stock market using NIFTY 500 stock data. After cleaning prices and computing log returns, a Pearson correlation matrix was estimated. Using the consistency function $G(\theta)$ of [Moghadam et al. \(2019\)](#) an optimal threshold $\hat{\theta} = 0.23$ was selected for network construction.

At this threshold the resulting network of 440 stocks and 34,818 edges is dense, highly clustered and mildly disassortative by degree. Centrality analysis identified a group of highly connected stocks, mainly from infrastructure, power, steel and finance related sectors. Sectoral homophily was weak but positive, indicating the presence of both intra sector and cross sector linkages.

A parameter study across thresholds illustrated how density, assortativity, transitivity and the apparent power law exponent vary with θ . The degree distribution did not show convincing scale free behaviour, and assortativity changed from negative to positive only at high thresholds where the network becomes very sparse.

Several extensions are possible:

- Constructing rolling window correlation networks to study temporal evolution of market structure.
- Applying community detection algorithms such as modularity maximisation to identify clusters of stocks and compare them with known sectors.
- Comparing the thresholded network with other constructions such as minimum spanning trees ([Mantegna, 1999](#)) or partial correlation networks.
- Incorporating higher frequency data or alternative dependency measures such as mutual information or tail dependence.

Overall the project demonstrates how network science tools taught in the course, including centrality measures, clustering, assortativity, similarity and power law analysis, can be applied to real financial data.

Code Availability

All scripts, figures, and data used in this analysis are publicly available at: <https://github.com/nikunj-indoriya/Nifty500-Network-Analysis>

Acknowledgements

I would like to thank Dr. Kundan Kandhway for teaching the course ECS414 “Network Science: Theory and Applications” and for providing the project topic that motivated this work.

References

- Tse, C. K., Jing Liu, and Francis C. M. Lau. A network perspective of the stock market. *Journal of Empirical Finance*, 17(4):659–667, 2010.
- Moghadam, Hadi Esmaeilpour, et al. Complex networks analysis in Iran stock market: The application of centrality. *Physica A: Statistical Mechanics and its Applications*, 531:121800, 2019.
- Mantegna, Rosario N. Hierarchical structure in financial markets. *The European Physical Journal B*, 11(1):193–197, 1999.
- Newman, M. E. J. Mixing patterns in networks. *Physical Review E*, 67(2):026126, 2003.