Methodology

1. Dataset

The dataset used in this research consists of the closing prices of 3 stock indices of India namely SENSEX, NIFTY50 and NIFTY Consumption every day from 01-01-2014 to 31-12-2023 (10 years). SENSEX is a free-float market capitalization consisting of 30 most traded and relatively liquid stocks which contribute towards the balance of the country’s equity market. NIFTY50 on the other hand is a benchmark index of 50 companies. NIFTY Consumption reflects the performance of companies in the domestic consumption sector. The data for SENSEX and NIFTY50 is taken from MarketWatch and NIFTY Consumption is taken from Yahoo Finance.

1. Creation of graph

The graph is constructed using the study of Cao, Lin et. al. who uses N-day volatility V, and N-day return R in order to divide the movement of stock index in 4 separate variations.

Here, *t* refers to the current day in consideration and N refers to the number of continuous trading days (generally a week if there is no national holiday) and Close(t) refers to the closing price of the stock index on *tth* day. In order to find out V, we need to find one-day return which is r which is given by

After calculating r, we can calculate V for N days is given by

Where refers to the standard deviation of

We can now calculate the average Volatility of entire stock index in question by simply averaging over the entire time series.

Now, we can classify the changes in any stock index on the basis of these parameters in the following way –

The classification is done for all the 3 indices and the combination of the patterns formed represents a node of a graph. Since the total number of combinations can be 43 = 64, we used a 4-base number system as nodes u and v for the graph.

The graph is constructed for 60 days although experimenting with other window sizes is still a future prospect of this research.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Index  Window 2  Window 1 | SENSEX | NIFTY50 | NIFTY Consumption | Combined Pattern |
| 1 | P3 | P2 | P1 | P3P2P1 |
| 2 | P4 | P1 | P4 | P4P1P4 |
| 3 | P2 | P1 | P3 | P2P1P3 |
| 4 | P3 | P2 | P1 | P3P2P1 |
| 5 | P1 | P2 | P3 | P1P2P3 |
| 6 | P2 | P1 | P3 | P2P1P3 |
| 7 | P3 | P2 | P1 | P3P2P1 |
| … |  |  |  |  |
| 60 | P1 | P2 | P4 | P1P2P4 |
| 61 | P2 | P3 | P3 | P2P3P3 |

Table 1: Sample patterns for 7 days of data of SENSEX, NIFTY50 and NIFTY Consumption

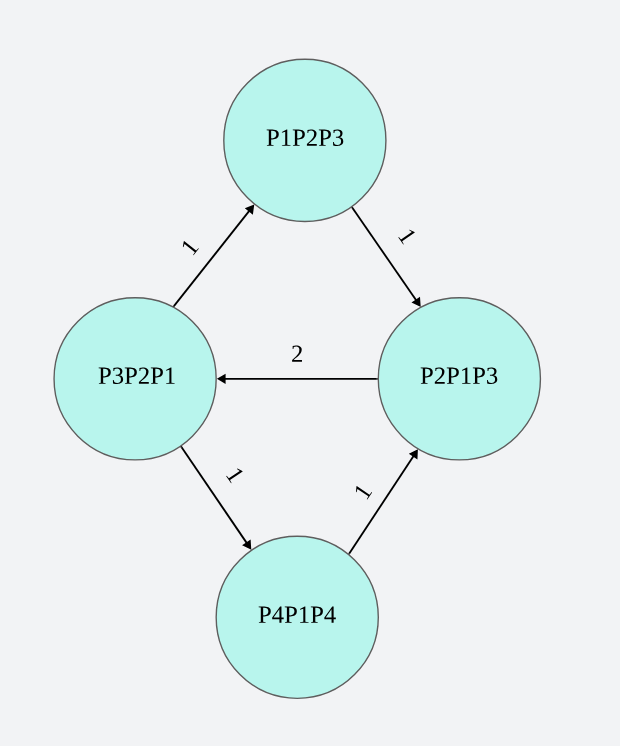


Figure 1: Example Weighted pattern graph for the data mentioned in Table 1 for the first 7 days

1. Centrality measures as input variables

The significance of this graph is that the denser this graph is, more is the dispersion and hence more care needs to be taken by investors while investing. Hence, we consider certain centrality measures in order to feed them as characteristics of our graph for classification on unseen data.

The measure we are considering for this research are degree centrality, strength (as described by Cao, Lin et. al.), closeness centrality and betweenness centrality. So far, we have applied the KNN algorithm for degree centrality and strength of the network.

Future Work

The following are the major contributions of this research which are yet to come in the future –

1. Applying various classification methods including SVM, random forest, deep learning strategies including neural networks in order to compare the strategies and find out the best one. We will also consider ensemble techniques by combining multiple classification strategies together.
2. Currently, we worked with only two centrality measures. In the future, we will work with more centrality measures such as betweenness centrality and closeness centrality in order to increase the input variables for classification algorithms.
3. We will also practically apply these machine learning techniques by using them through various investment strategies in trading simulators to better evaluate our results.
4. Currently, a window size of 60 days is only considered. However, graphs with other window sizes will also be considered on which these algorithms will be applied again. This will generate more results.

References