Stock Portfolio Selection using Ensemble Classifier, Gaussian Mixture, and Hidden Markov Models

Rony Mitra[a,b] (mitrarony92@gmail.com), Prajwal Yadav[c] (prajwal2610@gmail.com), Ratnesh Bhosale[d] (ratn.bhosale@gmail.com), Priyam Saha[e] (saha.priyam2@gmail.com), Debojyoti Ghosh[a] (debojyoti07.dg@gmail.com), Adrijit Goswami[a] (goswami@maths.iitkgp.ac.in), Manoj Kumar Tiwari[f,g] (mkt09@hotmail.com, director@iimmumbai.ac.in)

[a] Department of Mathematics, Indian Institute of Technology Kharagpur, India

[b] Data Science Lab, Indian Institute of Management Mumbai, India

[c] BlackRock Inc., India

[d] DTDC Express Limited, India

[e] Department of Economics, Indian Institute of Technology Kharagpur, India

[f] Director, Indian Institute of Management Mumbai, India

[g] Department of Industrial and Systems Engineering, Indian Institute of Technology Kharagpur, India

Corresponding Author:
Rony Mitra
Data Science Lab, Indian Institute of Management Mumbai
Vihar Lake Road, Mumbai-400087, Maharashtra, India
Tel: (+91) 7718433830
Email: mitrarony92@gmail.com, rony.mitra.gi@iimmumbai.ac.in

Stock Portfolio Selection using Ensemble Classifier, Gaussian Mixture, and Hidden Markov Models

Rony Mitra[a,b], Prajwal Yadav[c], Ratnesh Bhosale[d], Priyam Saha[b], Debojyoti Ghosh[b], Adrijit Goswami[b], Manoj Kumar Tiwari[a,b]

[a]*Indian Institute of Management Mumbai, India*
[b]*Indian Institute of Technology Kharagpur, India*
[c]*BlackRock Inc., India*
[d]*DTDC Express Limited, India*

Abstract

Stock price direction prediction is an essential factor in selecting a portfolio of stocks. In this paper, we introduced a novel approach for stock selection using price increment probability. We developed a novel methodology based on the Hidden Markov Model, Gaussian Mixture Model and the random forest algorithm to predict the price increment probability over a period, which is nascent in the literature. Two data segregation methods have been utilized to increase the accuracy of this model. We used Hidden Markov Models for market regime prediction and segregated the training data according to regimes. In the next step, the Gaussian Mixture Model clustering algorithm is employed to make stocks with similar price movement behaviour clusters. These clusters segregate the training data further and train Tree-based ensemble random forest classifiers for probability prediction. The performance of the model is analyzed by utilizing the NIFTY50 dataset. The results highlight the superior performance of the suggested methodology, showcasing excellence in profitability and prediction accuracy.

Keywords: Stock Selection, Hidden Markov Model, Gaussian Mixture Model, Random Forest

1. Introduction

Forecasting the stock market and stock price index can be challenging due to underlying uncertainties. Technical analysis involves evaluating a stock based on market activity statistics, such as past prices and volumes, to identify patterns and trends that may indicate future stock behaviour. Investors and traders in the financial industry attempt to maximize their returns while minimizing their risks. Leveraging stock-selection algorithms (Alamdari et al., 2023; Yang et al., 2019) is one strategy for achieving this goal. These algorithms aim to identify the most promising securities to invest in based on various factors, including financial ratios, news sentiment, and technical indicators. Fama (1970) proposed the efficient market hypothesis, which suggests that stock prices are informationally efficient. This means it is possible to predict stock prices based on trading data, as stock prices reflect numerous uncertain factors such as political conditions and a company's public image. Consequently, a stock or stock price index trend can be predicted by efficiently preprocessing information obtained from stock prices and leveraging appropriate algorithms (Chauhan et al., 2023).

In this paper, we propose a novel stock selection strategy that chooses stocks out of the top 50 stocks in the NIFTY50 index with the highest price increment probability. We employ state-of-the-art machine learning models to predict the price increment probability of each stock. Random forest is a popular algorithm for machine learning that generates n classification trees from samples with replacement. Based on the preponderance of the trees' predictions, the algorithm predicts the class. The collection of trees represents a singular hypothesis that is not necessarily contained in the hypothesis space of the models from which it is constructed. As a result, ensembles have more options for the functions they can represent.

Stock market regime predictions are created using Hidden Markov Models (HMM). A statistical model called an HMM can be used to forecast, given a sequence of seen states, the likelihood of a sequence of hidden states. Given a series of observed market data, HMMs can be used to forecast the likelihood of various market regimes in the context of the stock market. When discussing stock market regimes, a regime refers to a period when the market behaves in a specific way (Fikri et al., 2022). A regime could be a bull market, a bear market, or a time of extreme volatility, for instance. Investors might modify their investment plans by determining the current regime. In order for HMMs to function, the observed data must be inferred from a hidden state that is assumed to be the source of the observed data. Regarding the stock market, the observed data could be the daily returns of a stock or an index, while the hidden state could be the market regime (Zhao, 2022). To estimate the probability of different market regimes, an HMM is used. A historical dataset with market data and market regimes is used to train the model (Joshi, 2022). The statistical relationships between the observable data and the hidden states are initially taught to the HMM. Then, given a series of actual market data, it makes use of this information to predict the likelihood of other market regimes. The HMM can estimate the probability of various market regimes in real-time after its training is complete. Investors can then use this information to adjust their investment strategies accordingly. For example, suppose the HMM predicts a high probability of a bear market regime. In that case, an investor may choose to sell their stocks and move their investments into cash or other assets that are expected to perform well during a bear market (Virigineni et al., 2022).

However, most extant stock-selection algorithms need to account for the impact of various market regimes on stock performance. Market regimes refer to periods of time during which the market displays certain characteristics, such as high or low volatility, bullish or bearish trends, etc. The market regime can have a significant impact on the performance of stocks, making it essential to consider this variable when selecting equities. Traditional modelling approaches face challenges when dealing with multi-regime time series. These time series comprise multiple distinct states or regimes, each with its unique behaviour. For instance, economic time series may demonstrate different behaviours during bull and bear markets. Likewise, depending on loads, temperatures, and operational conditions, complicated equipment's sensor readings might differ greatly. It is possible that the complex internal structure of the data will be inaccurately represented if a single overarching model is used to describe the complete time series. It is possible that the complex internal structure of the data will be inaccurately represented if a single overarching model is used to describe the complete time series.

Specialised modelling techniques are needed to adequately represent the unique characteristics of each regime in order to overcome this issue. These strategies include using numerous models, each representing a distinct regime and alternating between them as the data changes states. Alternatively, models can be designed to adapt to the changing dynamics of the data, allowing for the identification of different regimes without the need for explicit modelling. In this paper, we have explored three strategies: the first one is stock selection using clustering, the second one is regime prediction using HMM, and the third one is the combination of regime prediction and stock clustering. We compare the results of these strategies with the NIFTY50 benchmark.

The remainder of the paper is structured as follows. After conducting a pertinent literature review and background study in the next section, section 3 demonstrates the proposed methodology. The results regarding Stock Clustering and Regime Prediction are analyzed in section 4. Finally, section 5 concludes the paper and suggests some future research directions.

2. Related Literature

Several techniques have been devised over the years to forecast stock market trends. Since stock data is classified as a non-stationary time series, non-linear machine-learning techniques have also been implemented. Artificial Neural Networks (ANN) and Support Vector Machine (SVM) are two of the most popular machine learning algorithms for predicting the movement of stocks and stock price indices. Wang & Leu (1996) developed an effective system for predicting intermediate-term price trends in the Taiwan stock market. Their approach relied on a recurrent neural network trained using features derived from ARIMA analyses. Tsai et al. (2011) conducted a study examining the effectiveness of classifier ensembles in predicting stock returns. The research compared the performance of two types of classifier ensembles with single baseline classifiers, such as logistic regression, decision trees, and neural networks. The findings showed that, in comparison to using single classifiers, employing ensembles of several classifiers produced better outcomes in terms of prediction accuracy and returns on investment. Our research highlights the value of ensemble approaches in stock market trend prediction by demonstrating that combining several models can produce predictions that are more accurate than using just one.

Artificial neural networks (ANN) replicate the ability of the human brain to recognise patterns. Hassan et al. (2007) suggested and implemented a fusion model to forecast financial market behaviour by combining the Hidden Markov Model (HMM), Artificial Neural Networks (ANN), and Genetic Algorithms (GA). The daily stock prices were transformed using ANN into sets of values that HMM could use as input (Chen et al., 2019). The HMM is a widely used machine learning technique for recognizing patterns in stochastic processes, especially time series data (Nystrup et al., 2018; Zhang et al., 2019; Liu et al., 2022). HMM has been utilized in various disciplines (Mor et al., 2021), including speech recognition (Woodland & Povey, 2002), handwriting recognition, musical score following, partial discharges, bioinformatics, economics, and finance (Wang & Hsieh, 2022). HMM has recently gained popularity in economics and finance, with applications in modelling foreign exchange data (Caporale & Spagnolo, 2004; Seerattan & Spagnolo, 2009), predicting stock prices of interrelated markets, predicting the regimes of market turbulence (Mamplata et al., 2022), inflation, and industrial production index, modelling interest rates, selecting stocks and mortgage-backed securities, analyzing stock market trends, and developing a trading strategy for stocks (Giudici & Abu Hashish, 2020; Thakkar & Chaudhari, 2020). HMM is useful for modelling market regimes, which are frequently characterized by cycles, and for predicting the regimes of certain economic indicators. In a regime-switching model, the observation variables are generated by an auto-regression model, while the parameters are optimized using a discrete Markov chain. HMM has proven to be a valuable tool in such models, where market regimes are simulated. HMM is a potent tool for analyzing time series data and predicting future financial market trends due to its ability to handle stochastic processes. Numerous fields continue to investigate its wide-ranging applications. This paper uses HMM to predict economic regimes and optimize global stock portfolios. While previous applications of HMM in local and national stock trading have been explored, we focus on the more complex factors driving global stocks. We expand our investigation from the 4.0 economic indicators used in previous work to six macroeconomic indicators with the most influence on the global stock market. As suggested in prior research, these indicators include inflation, production, sentiment, market demand, debt, and inflation expectations. Our aim is to use HMM to effectively analyze these indicators and predict economic regimes for optimal portfolio selection.

Also, researchers have applied ARIMA, LSTM, ANN, CNN (Banerjee & Nayak, 2024), deep learning (Alzaman, 2024), and RNN (Radojičić et al., 2024) for stock selection and price prediction. Chaudhari & Thakkar (2023) applied the median range, k-means algorithm, and top-M coefficient of variation for feature selection. They utilized a neural network on selected features for stock price prediction. Kim et al. (2020) developed a novel framework for predicting stock prices based on regime prediction and random forest. Two stages comprise the proposed framework: regime prediction and stock price prediction. The market regime is predicted in the initial phase utilizing a clustering algorithm. In the second stage, stock prices are predicted using a random forest model based on the predicted regime. Wang & Hsieh (2022) introduced a data-driven encoding-and-decoding method to identify hidden states in stock dynamics based on S&P500 index. On the S&P500 index, our proposed framework demonstrated promising results.

3. Methodology

3.1. Dataset Description

The study uses time series data consisting of Open, High, Low, Close, and Volume of the Top 50 companies by market capitalization from 1st July 2007 to 1st January 2023 from a well-known brokering company (Sisodia et al., 2022). The dataset considers stock splits, bonuses, dividends, and other corporate actions that impact the stock price. The stock pool selected is relatively diversified, with stocks belonging to various sectorial indices such as NIFTY Auto, NIFTY Bank, NIFTY Energy, NIFTY Metal, NIFTY FMCG, and NIFTY Pharma. This study also uses a volatility-based stop-loss mechanism based on India VIX; therefore, a daily time series of the India Volatility Index (India VIX) is used. Using India VIX as a stop-loss mechanism is essential to avoid Systematic Risk. The features of the dataset are described in Table 1. The summary of the data is presented in Table 2. MIN, MAX, AVG and STD denote minimum value, maximum value, average value and standard deviation on the feature, respectively. This statistic is presented for open, high, low and close features.

Feature Name	**Description**
Open	Price at which stock opens in the market when trading begins
High	The highest trading price of stock
Low	The lowest trading price of stock
Close	Price at which stock closes in the market when trading ends
Volume	Number of shares traded in a day
Symbol	Ticker of a stock
Datetime	Date of the date of trading

Table 1: Description of the features of NIFTY50 dataset

3.2. Dataset Pre-processing

The technical indicators for stock prices rely on essential features, drawing from the Open, High, Low, Close, and Volume data of the Top 50 com-

Features	**MIN**	**MAX**	**AVG**	**STD**
Open	2553.6	18871.95	8665.225	3968.011
High	2585.3	18887.6	8716.099	3978.139
Low	2252.75	18778.2	8600.178	3949.513
Close	2524.2	18812.5	8659.427	3964.88

Table 2: Summary of NIFTY50 from 1 January 2007 to 1 January 2023

panies. These features encompass (Pandey & Sharma, 2020) simple moving average (SMA), exponential moving average (EMA), average true range (ATR), bollinger bands, and chaikin money flow (CMF).

$$SMA = \frac{1}{n}\sum_{i=1}^{n} P_i \tag{1}$$

The widely utilized SMA is a technical indicator that reflects the extent of stock price variations within a specified time frame. This lagging indicator is typically expressed using equation 1, with n signifying the chosen time period and P_i representing the stock price during that period. In our study, we have calculated SMAs for intervals of 50, 100, and 200 periods. Another lagging indicator, EMA, illustrates the extent of stock price fluctuations within a specific timeframe. Equation 2 defines EMA, where $\mathcal{K} = 2n+1$, C_p is the current price, EMA_p is the previous EMA and n is the selected time period. In its initial calculation, the EMA_p is essentially the starting EMA, derived as an average of all prices over n periods. Our research analyses EMAs over periods of 50, 100, and 200. A crucial distinction between EMA and SMA is that EMA assigns more weightage to recent data points.

$$EMA = \mathcal{K}(C_p - EMA_p) + EMA_p \tag{2}$$

$$TR = \max\{(high - low), |high - close_p|, |low - close_p|\} \tag{3}$$

$$ATR = \frac{1}{n}[ATR_p \times (n-1) + TR] \tag{4}$$

The ATR is a price volatility indicator that shows the average price variation of stocks within a certain time period. The rise in the value of ATR is indicative of high volatility and vice versa. The prime component in the ATR formula (equation 4) is the calculation of the True Range (TR) value based on equation 3, and n is the number of periods. The Bollinger Band is a technical analysis tool defined by a set of trendlines consisting of two standard deviations (positively and negatively) away from an SMA of the same period as the standard deviations. The type of moving average can be customized. However, we have taken SMA.BOLU and BOLD as the Upper and Lower Bollinger Band, respectively (Seshu et al., 2022). The formula for the Bollinger Bands for n number of days in smoothing period and m number of standard deviations are defined by the equations 5 and 6, where $T_p = \frac{1}{3}(high + low + close)$ and $\sigma[T_p, n]$ is standard deviation over last n periods of T_p.

$$BOLU = SMA(T_p, n) + m \cdot \sigma[T_p, n] \tag{5}$$

$$BOLD = SMA(T_p, n) - m \cdot \sigma[T_p, n] \tag{6}$$

The CMF is an indicator to monitor the accumulation and distribution of a stock over a specified time period. For our analysis, a period of 14 days is considered. The indicator readings range between +1 and -1. Money flow multiplier (F_l) can be calculated using the equation 7. Equation 8 denotes the money flow volume (F_v), where V_p is the volume for a specific period. The formula for CMF is given in equation 9 based on the high, low, open, close, volume, and daily money flow. Here 21, D_f, and D_v represent a number of days in a month, daily money flow, and daily volume, respectively.

$$F_l = \frac{(close - low) - (high - close)}{high - low} \tag{7}$$

$$F_v = F_l \times V_p \tag{8}$$

$$CMF = \frac{21 - avg(D_f)}{21 - avg(D_v)} \tag{9}$$

3.3. Regime Prediction using Hidden Markov Model

The hidden Markov model (HMM) can capture and observe the hidden states of the data. An observation at time t of an HMM corresponds to a possible condition with a specific probability distribution. Baum & Petrie (1966) first introduced the mathematical foundations of HMM. After that, Baum et al. (1970) developed a maximization model that calibrates HMM parameters using a single observation. Levinson et al. (1983) designed a maximum likelihood estimation method for HMM with multiple observation training, considering all observations are independent. Li et al. (2000) introduced an HMM training for multiple statements that did not assume the observations' independence. The basic elements of HMM are:

- T: Length of observation data
- N: No. of states
- M: No. of symbols per state
- Hidden state sequence, $Q = \{Q_1, Q_2, ..., Q_T\}$
- Observation sequence, $O = \{O_1, O_2, ..., O_T\}$
- Possible values of each state, $S = \{S_1, S_2, ..., S_N\}$
- Possible symbols per state, $V = \{v_1, v_2, ..., v_M\}$
- Observation probability matrix, $B = \{b_{ik}$ where $i = 1, 2, ..., N; k = 1, 2, ..., M$ and $b_{ik} = P(O_t = v_k | q_t = S_i)$
- Vector of the initial probability of being in the state (regime) S_i at time $t = 1$, $p = \{p_1, p_2, ..., p_N\}$ where $p_i = P(q_1 = S_i)$
- Transition matrix, $A = (a_{ij})$, where $i = 1, 2, ..., N; j = 1, 2, ..., N$; and $a_{ij} = P(q_t = S_j | q_{t-1} = S_i)$

The parameters of an HMM are the matrices A, B and the vector p. For convenience, a compact notation $\lambda = \{A, B, p\}$. The observation probability for the Gaussian distribution can be defined based on the equation 10, where μ_t and σ_t are the mean and variance of the distribution corresponding to the state S_t respectively, and N is a Gaussian density function. Equation 11 represents the parameters of HMM, where μ and σ are vectors of the means and variances of the Gaussian distributions, respectively.

$$b_{ik} = N(O_t = v_k, \mu_i, \sigma_i) \tag{10}$$

$$\lambda \equiv \{A, \mu, \sigma, p\} \tag{11}$$

There are three major challenges of the hidden Markov model. The first one is to compute the probabilities of the observations $P(O|\lambda)$, given the observation data O and the model parameter $\lambda = \{A, B, p\}$. Forward (algorithm 1) and backward algorithm (algorithm 2) can be used to solve this problem. Second is to choose the best corresponding state sequence Q, given the observation data O and the model parameter $\lambda = \{A, B, p\}$. The Viterbi algorithm (algorithm 3) can be used to solve this issue. The third challenge is to calibrate HMM parameters $\lambda = \{A, B, p\}$ to maximize $P(O|\lambda)$ with the given observation data O. The Baum–Welch algorithm (Dar et al., 2022) using forward algorithm can be utilized to solve this problem.

Algorithm 1 The Forward Algorithm

```
for $i = 1, 2, ..., N$ do                                        ▷ Initialization
    $\alpha_{t=1}(i) = p_i b_i(O_1)$
end for
for $t = 2, 3, ..., T$ do                                        ▷ Recursion
    for $j = 1, 2, ..., N$ do
        $\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij}\right] b_j(O_t)$
    end for
end for
$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$                      ▷ Output
```

$$\alpha_t(i) = P(O_1, O_2, ..., O_t, q_t = S_i|\lambda) \tag{12}$$

The Forward algorithm (Eddy, 2011) is defined as based the joint probability function ($\alpha_t(i)$) (equation 12) and probability of observation ($P(O|\lambda)$. Also, the recursive backward algorithm (Ren et al., 2018) is defined based on the conditional probability $\beta_t(i)$ (equation 13), where $i = 1, 2, ..., N$.

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, ..., O_T | q_t = S_i, \lambda) \tag{13}$$

We utilized Viterbi Algorithm (Li et al., 2022; Amiens & Osamwonyi, 2022) to find the best sequence of states Q^* when (O, λ) is given. To solve the most likely state q_t (equation 15) at time t, we used $\delta_t(i)$ (equation 14), where $1 \leq t \leq T$.

Algorithm 2 The Backward Algorithm

```
for $i = 1, 2, ..., N$ do                                    ▷ Initialization
    $\beta_t(i) = 1$
end for
for $t = T - 1, T - 2, ..., 1$ do                             ▷ Recursion
    for $i = 1, 2, ..., N$ do
        $\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$
    end for
end for
$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$                   ▷ Output
```

$$\delta_t(i) = \max_{q_1, q_2, ..., q_t} P(q_1, q_2, ..., q_t = S_i, O_1, O_2, ..., O_t | \lambda) \tag{14}$$

$$q_t = \arg \max_{1 \leq i \leq n} [\delta_t(i)] \tag{15}$$

Additionally, to find the parameters $\lambda = A, B, p$ to maximize the probability $P(O, \lambda)$ of observation data $O = \{O_1, O_2, ..., O_T\}$. Unfortunately, given observation data, there is no way to find the global maximum of $P(O, \lambda)$. However, we can choose the parameters such that $P(O|\lambda)$ is locally maximized using the Baum–Welch iterative method (Jiang et al., 2023), which uses the maximum likelihood estimator (MLE) to train the model parameters. We defined the procedure $\gamma_t(i)$ (equation 16) based on the probability of being in state S_t at time

Algorithm 3 The Viterbi Algorithm

```
for $j = 1, 2, ..., N$ do                                  ▷ Initialization
    $\delta_t(j) = p_j b_j(O_1)$
    $\phi_1(j) = 0$
end for
for $t = 2, 3, ..., T$ do                                  ▷ Recursion
    for $j = 1, 2, ..., N$ do
        $\delta_t(j) = \max_i [\delta_{t-1}(i) a_{ij}] \, b_j(O_{t+1})$
        $\phi_t(j) = \arg\max_i [\delta_{t1}(i) a_{ij}]$
    end for
end for
$q_T^* = \arg\max_i [\delta_T(i)]$                         ▷ Output
$q_t^* = \phi_{t+1}(q_{t+1}^*), t = T-1, T-2, ..., 1$
```

t.

$$\gamma_t(i) = P(q_t = S_i | O, \lambda) = \frac{\alpha_t(i)\beta_t(i)}{P(O,\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum\limits_{i=1}^{n} \alpha_t(i)\beta_t(i)} \tag{16}$$

The probability ($\epsilon_t(i,j)$) of being in state S_i at time t and S_j at time $t+1$ is defined based on the equations 17 and 18.

$$\epsilon_t(i,j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O,\lambda)} \tag{17}$$

$$\gamma_t(i) = \sum_{i=1}^{N} \varepsilon_t(i,j) \tag{18}$$

3.4. Gaussian Mixture Model (GMM) Clustering Algorithm

Gaussian mixture model (Klem et al., 2022) is a probabilistic clustering algorithm which uses soft clustering approach to cluster the datapoints. In one dimensional space, probability density function of a Gaussian distribution (Normal distribution) for an input vector x is defined by the equation 19, where μ and σ are mean and variance of x respectively. For Gaussian mixture model

Algorithm 4 The Baum–Welch Algorithm

```
Input: parameter $\lambda$, the tolerance $tol$, a real number $\Delta$
Output: parameter updated $\lambda$
while $\Delta < tol$ do
    Calculate $P(O, \lambda)$                                ▷ Using algorithm 1
    for $i = 1, 2, ..., N$ do
        $p_i^* = \gamma_1(i)$
        for $j = 1, 2, ..., N$ do
            $a_{ij}^* = \frac{\sum_{t=1}^{T-1} \varepsilon_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$
        end for
        $b_{ik}^* = \frac{\sum_{t=1, O_t = v_k}^{T} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)}$
    end for
    $\Delta = |P(O, \lambda^*) - P(O, \lambda)|$
    $\lambda = \lambda^*$
end while
```

with K components probability estimate model is defined by the equation 20, where φ_i is the weight of component i and $\sum_{i=1}^{K} \varphi_i = 1$.

$$f(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{19}$$

$$p(x) = \sum_{i=1}^{K} \varphi_i f(\mu_i, \sigma_i^2) \tag{20}$$

The parameters of the above model can be estimated using the expectation maximization technique. Expectation maximization is an iterative algorithm which maximizes the likelihood the data. The model is trained using maximum likelihood estimation technique which maximizes the probability given the model parameters. Using Bayes theorem and the estimated model parameters using expectation maximization, the posteriori component assignment probability can be estimated. Cluster assignment is determined by the most likely component assignment (Dai & Jayantha, 2022). The probability that a data point x belongs to cluster C_i given the parameters of univariate model is defined by the equation 21 using Bayes theorem. The datapoint is labeled with cluster which has maximum $p(x)$. The bag of stocks is clustered into K clusters using the GMM clustering algorithm (Bagirov et al., 2022).

$$p(x) = \frac{p(C_i)p(x|C_i)}{\sum\limits_{j=1}^{K} p(C_j)p(x|C_j)} = \frac{\varphi_i f(\mu_i, \sigma_i^2)}{\sum\limits_{j=1}^{K} \varphi_j f(\mu_j, \sigma_j^2)} \tag{21}$$

Under the assumption of GMM model, it is assumed that the data which is given as an input is generated by a mixture of several Gaussian distributions. The subset of the data having similar stock price movement will be clustered together. Apart from open, high, low and close features, several other technical indicators which are discussed above are considered for GMM clustering (van den Berg, 2022). Before fitting the data into GMM model, dataset passed through preprocessing step. The length of the dataset is N, where each data point is represented by a d-dimensional feature vector $X_i = (x_{i1}, x_{i2}, ..., x_{id})$ where d represents the number of features of the dataset and $i = \{1, 2, ..., n\}$.

For this study, the dataset D will be clustered in K clusters. The probability density function (pdf) of a GMM is defined as a mixture of K Gaussian distributions, where K is the number of clusters (components) that we want to identify in the data. The pdf of the GMM for K clusters ID defined by the equation 22, where π_k is the mixture weight of the k-th component, representing the prior probability of a data point belonging to the k-th cluster, Σ_k represents covariance matrix which is a probability of a data point X_i belonging to the k-th cluster, $0 \leq \pi_k \leq 1$, $\sum_{k=1}^{1} \pi_k = 1$, and $N(X_i|\mu_k, \Sigma_k)$ can be defined by the equation 23.

$$P(X_i) = \sum_{k=1}^{K} \pi_k \cdot N(X_1|\mu_k, \Sigma_k) \tag{22}$$

$$N(X_1|\mu_k, \Sigma_k) = (2\pi)^{-d/2}|\Sigma_k|^{-1/2}e^{-0.5(X_i-\mu_k)}\Sigma_k^{-1(}X_i - \mu_k) \tag{23}$$

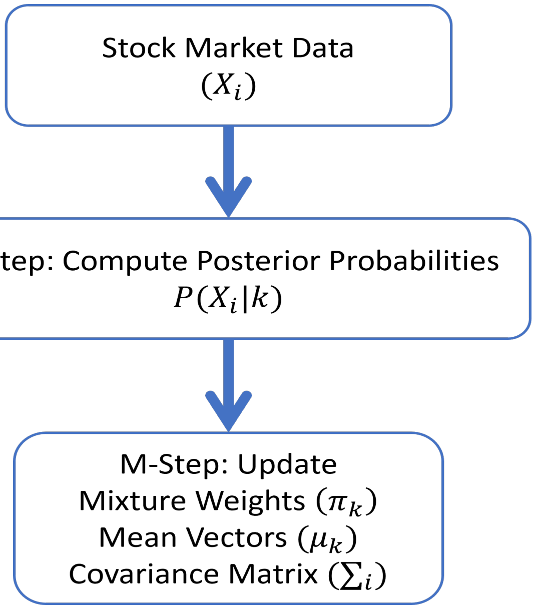

Stock Market Data
(X_i)
E-Step: Compute Posterior Probabilities
$P(X_i|k)$
M-Step: Update
Mixture Weights (π_k)
Mean Vectors (μ_k)
Covariance Matrix $(\sum_i)$

Figure 1: Flow chart of EM algorithm in GMM

The GMM has three sets of parameters that need to be estimated from the data. The mixture weights π_k which represent the prior probabilities of each cluster. These can be estimated as the sample means of the data points in each cluster. The covariance matrices Σ_k, which represent the shape and orientation of each cluster. These can be estimated as the sample covariances of the data points in each cluster. The parameters of the GMM can be estimated using the Expectation-Maximization (EM) algorithm, which is an iterative algorithm that alternates between the E-step and M-step.

- E-step: Compute the posterior probabilities of each data point belonging to each cluster, given the current estimates of the parameters. This can be done using Bayes' theorem and the pdf of the GMM.

- M-step: Update the estimates of the parameters using the computed posterior probabilities and the data points. This can be done using maximum likelihood estimation (MLE) or other optimization techniques.

The EM algorithm is repeated until convergence, typically based on a convergence criterion such as the change in the log-likelihood of the data or the change in the estimated parameters. The model working is briefly explained in figure 1.

3.5. Probability Prediction

For each cluster K_i, predictive classification model is framed. The output of each model will be the probability of price increment. Dataset with newly engineered features will serve as the input. We used tree-based ensemble Random Forest (RF) for predicting the stock price signal prediction. Random Forest is a tree based supervised machine learning algorithm (Ghosh et al., 2022). Random Forest combines decision tree predictors based on the bagging classification. Random Forest combines decision tree predictors based on the bagging classification. Before training, the target variable (signal) is engineered (equation 24), where $close_t$ is close price of the stock at a trading date and $close_{t-7}$ is close price of the stock in previous week of the trading date.

$$Signal = \begin{cases} 1 & \text{for} \quad close_t - close_{t-7} > 0 \\ 0 & \text{for} \quad close_t - close_{t-7} \leq 0 \end{cases} \tag{24}$$

In this paper, the K-RF signal prediction models are framed. The data for each cluster is fitted into random forest model. K random forest models are framed to better predict the trend of stock model. Each model is according used to predict the probability of stock increment (Wetzel & Hamel, 2023). Let the RF ensemble model (algorithm 5) for cluster k be γ_k which consists of N decision

Algorithm 5 The K-RF Algorithm

```
for $i = 1, 2, ..., K$ do                                ▷ To generate $K$ classifiers
    Select data $D_i$ based on the stocks present in $K_i$
    Randomly sample $D_i$ to $D_{train}$
    Create a root nodes $N_i$ contained $D_{train}$
    Call BuildTrees ($N_i$)
end for
```

trees. We have two classes, class $1(C_1)$ for price increment and class $0(C_2)$ for price decrement. For a given data point x, the prediction of the RF ensemble γ_k for class C_k is denoted by $P(Signal = C_k|x, \gamma_k)$, where signal is the target class variable. This represents the posterior probability or confidence of the prediction for class C_k, given the input data point x and the random forest ensemble γ_k. The posterior probability or confidence of the prediction for class C_k can be estimated as the proportion of decision trees in the ensemble γ_k that predict class C_k for the input data point x, out of the total number of decision trees N. This can be mathematically formulated based on the equation 25, where $Signal_i$ represents the predicted class by the i-th decision tree in the ensemble for the input data point x, and $I(Signal_i = C_k)$ is an indicator function that takes the value 1 if $Signal_i = C_k$, and 0 otherwise.

Algorithm 6 BuildTrees (N_i)

```
Randomly select certain percentage of spitting features in $N$
Select feature with highest information gain to spit on
Create $f$ child nodes of $N(N_1, N_2, ..., N_f)$, where $F$ has $f$ possible values $F_1, F_2, ..., F_f$
for $i = 1, 2, ..., f$ do
    Set the contains of $N_i$ to $D_{train}$, where $D_{train}$ is all instances in $N$ that match $F_i$
    Call BuildTrees ($N_i$)
end for
```

$$P(Signal = C_k|x, \gamma_k) = \frac{1}{N} \sum_{i=1}^{N} I(Signal_i = C_k) \tag{25}$$

$$confidence\,(P(Signal = C_k|x, \gamma_k)) = \frac{1}{N} \sum_{i=1}^{N} P(Signal = C_k|x, tree_i) \tag{26}$$

Alternatively, if the random forest implementation provides a continuous confidence score, the confidence of the prediction for class C_k can be calculated based on the equation 26, where $P(Signal = C_k|x, tree_i)$ is the estimated probability of class C_k predicted by the i-th decision tree in the ensemble for the

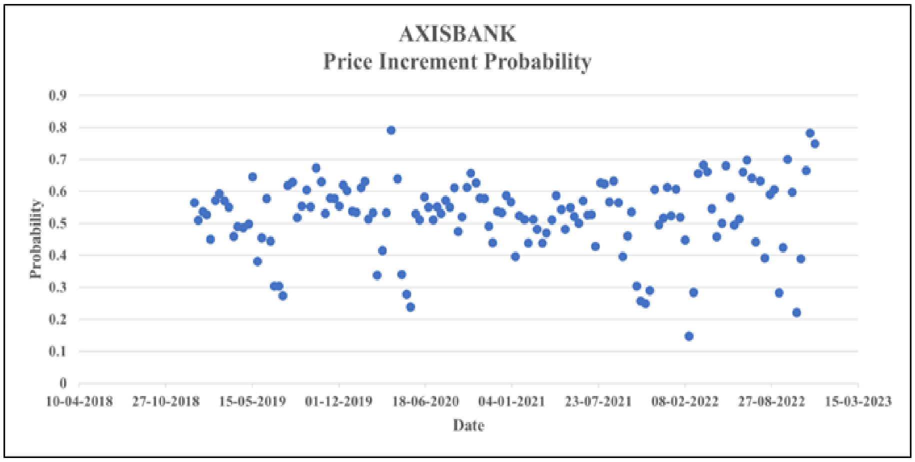


Figure 2: Sample Probability variation for AXISBANK

input data point x, and $tree_i$ represents the i-th decision tree in the ensemble (Olkhov, 2023). The class probability $p_{i,k} = P(Y = k|X = x_i)$ is also calculated based on the voting mechanism. For testing dataset, $\hat{p}_{i,k}^{prox}$ is the proportion of the trees predicting class x_i after voting. Using the proximities (equation 27), the most probable class label is predicted, and the class probability can also be calculated, where the $prox(i, j)$ for n_{tree} trees is given by the formula 28 with indicator function I (Zhang et al., 2022).

$$\hat{p}_{i,k}^{prox} = \frac{\sum\limits_{j \in A, j \neq i} prox(i, j) I(y_j = k)}{\sum\limits_{j \in A, j \neq i} prox(i, j)} \tag{27}$$

$$prox(i, j) = \frac{1}{n_{tree}} \sum_{i=0}^{n_{tree}} I\left(q_t(i) - q_t(j)\right) \tag{28}$$

In an RF model, the posterior probability represents the estimated probability of a particular class given an input data point and the predictions of the ensemble. The sample Probability distribution for AXISBANK is shown in figure 2.

3.6. Model Description

The trading system discussed in this paper is designed to trade stocks and consists of three layers: Regime Prediction, Stock Clustering and Probability

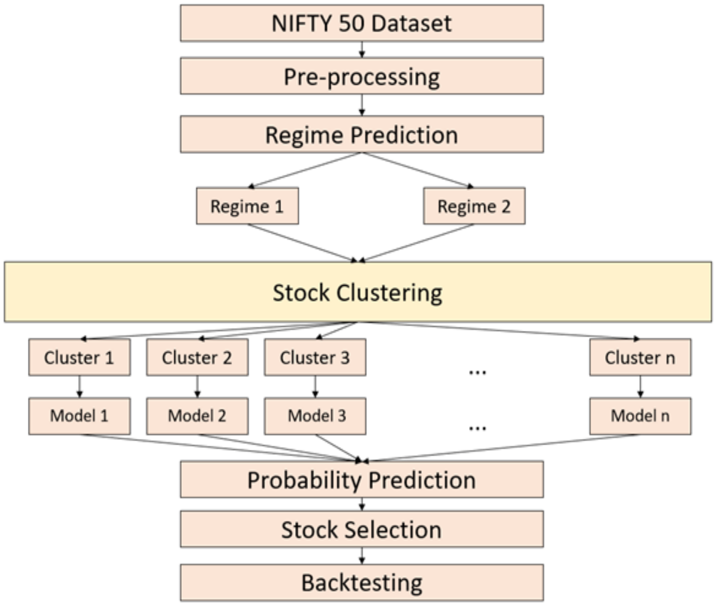


Figure 3: Framework of the model

Prediction. While each of these layers is discussed in greater detail below, an overview of the system is given in figure 3. The backtesting period is January 2019 to December 2022. In this period, we select five stocks to be added to a portfolio of equal weight. We rebalance the portfolio every seven days.

Regime Prediction

The hidden Markov model has been utilized extensively in the field of financial mathematics to forecast economic regimes and stock prices. We choose one of the common benchmarks for the Indian stock market, the NIFTY50 index to implement our model. In this step, we try to find market regime of NIFTY50 index. We start by taking a training data from 1 January 2007 to 1 January 2019.We take number of states N=2. We use a block of length (T) 12 years. For the second calibration, we move the 12-year data upward one week, we have a new data set from 8 January 2007 to 8 January 2019 and use the calibrated parameters from the first calibration as initial parameters. We repeat the process for the backtesting time which is January 2019 to December 2022. After training the HMM model, on each trading day we find the regime of previous day's market according to its features (Open, High, Low, Close). After getting the regime, the next step is stock clustering. The training data for the next step is divided into 2 parts according to its regime. If the current regime is 'regime_1', then the historical data of dates categorized in 'regime_1' is used for clustering and probability prediction.

Stock Clustering

In this step we create 10 clusters ($k = 10$) out of 50 stocks of NIFTY50. The GMM clustering algorithm puts stocks with similar price movement behavior in one cluster. This step is performed to get a better result in the next step, which is probability prediction. We assume that clustering similar stocks in one group would increase the accuracy of prediction model. For this, we divide the training dataset into 10 parts, according to result of GMM algorithm. This is then passed on to next step.

Probability Prediction

The role of the prediction layer is to generate a model every 7 days to predict the price increment probability of a particular stock. On each trading day(D), we start with initializing 10 different Random Forest models. The training data for each model comes from step stock clustering. For training this model, we used the data from 1 January 2007 to day(D-1). We take data that is filtered in regime prediction. We again divide this data into 10 parts according to 10 clusters made in step stock clustering (Puspita & Wulandhari, 2022). After training, we evaluate the price increment probability of all stocks.

Stock Selection and Portfolio Rebalancing

This is the last step in this trading algorithm. After probability prediction, we arrange the list of stocks according to its probability of price increment. Five stocks with highest probability of price increment are chosen for the portfolio (Yang et al., 2022; Ozcalici & Bumin, 2022). At the end of backtesting, we calculate various evaluation matrices.

Matrix	**Formula**
Annual Returns	$\left(\frac{\text{Portfolio ending value}}{\text{Portfolio strating value}}\right)^{\frac{1}{\text{No of years}}} - 1$
Cumulative Returns	$\left(\frac{\text{Portfolio ending value}-\text{Portfolio strating value}}{\text{Portfolio strating value}}\right) \times 100$
Annual Volatility	$\text{Standard deviation } \times \sqrt{252}$
Sharpe Ratio	$\frac{\text{Portfolio returns}-\text{Risk Free Rate}}{\text{Standard deviation}}$
Calmar Ratio	$\frac{\text{Annual returns}}{\text{Maximum drawdown}}$
Stability	$1 - \frac{\text{Sum of squares of residuals (Cumulative return)}}{\text{Total sum of squares (Cumulative return)}}$
Max Drawdown	$\frac{\text{Trough Value — Peak Value}}{\text{Peak Value}}$
Sortino Ratio	$\frac{\text{Portfolio returns}-\text{Risk Free Rate}}{\text{Standard Deviation of negative returns}}$
Skew	$\frac{\sum(X_i-\bar{X})^3}{(N-1)\sigma^3}$
Kurtosis	$\frac{\mu_4}{\sigma^4}$

Table 3: Evaluation Matrices

4. Results and Analysis

There are three distinct approaches that we have taken into consideration for back testing. In conclusion, we performed an analysis and evaluate the outcomes for each technique based on the matrices prasented in table 3, and then we compare those outcomes to a benchmark index. We decided to use the NIFTY50 index as a standard instead. The three methodologies are referred to as Stock Clustering, Regime Prediction, and Combined Stock Clustering and Regime Prediction. The technique that is under consideration in this research is referred to as the ”Combined Stock Clustering and regime Prediction” strategy. Following this, we will go over two additional strategies.

4.1. Stock Clustering

Only stock clustering is taken into consideration in this method, which eliminates the regime prediction phase from the overall strategy design. We begin by training a GMM clustering model with N = 10 clusters for each trading day (D). The training data we use spans from the first day of January 2007 to the first day of trading (D-1). Based on the outcome of this algorithm, we partition the training data into ten parts and then train ten Random Forest algorithms for the purpose of probability prediction. For the purpose of selecting the top five stocks, the outcomes of all prediction models are ordered using an ascending order.

4.2. Regime Prediction

In this approach, we exclude the clustering phase from our technique and solely focus on regime prediction. At the beginning of each trading day (D), we collect training data from 1 January 2007 to the previous day (D-1) and use it to train a Hidden Markov Model (HMM) with N = 2 states. The algorithm yields the following outcome: the training data is divided into two sections, and two Random Forest algorithms are trained to forecast probabilities. The outcomes of all forecasting models are organised in ascending order to choose the top 5 stocks. We evaluate three stock selection strategies: a) selecting stocks based solely on regimes, b) selecting stocks based solely on clustering, and c) selecting stocks based on both regimes and clustering. We evaluate the outcomes of these tactics by comparing them to a benchmark, specifically the NIFTY50 index. Table 4 provides a concise summary of the outcomes following the implementation of the three techniques. The integration of regimes prediction with clustering yielded superior results, with an annual return of 42%, as opposed to 30% for clustering alone and 20.2% for regimes prediction alone. The benchmark index provided a return of 13.9%. The technique that combines clustering and regimes outperforms the other two strategies in terms of Sharpe ratio. The Sharpe ratio for the combination of both clustering and regime prediction is 1.56. In comparison, the Sharpe ratio was 1.17 for clustering alone, 0.86 for regimes only, and 0.73 for the benchmark. Figure 4 displays the total accumulated returns of all the techniques.

The combination of clustering and regime prediction generates a return of 298%, whereas clustering alone achieves returns of 181.6% and regime prediction solo achieves returns of 106.3%. The NIFTY50 index yielded a return of 67%.

	Stock Clustering	Regime Prediction	Combined Regime prediction+ stock Clustering	Benchmark: Nifty50 Index
Annual Returns	30.0%	20.2%	42.0%	13.9%
Cumulative Returns	181.6%	106.3%	298.0%	67.0%
Annual Volatility	25.3%	24.9%	24.5%	20.8%
Sharpe Ratio	1.17	0.86	1.56	0.73
Calmar Ratio	0.79	0.43	1.01	0.35
Stability	0.86	0.72	0.91	0.74
Max Drawdown	-38.0%	-47.2%	-41.5%	-39.7%
Omega Ratio	1.24	1.18	1.34	1.15
Sortino Ratio	1.65	1.18	2.25	1.00
Skew	-0.82	-1.27	-0.94	-1.18
Kurtosis	10.24	14.39	11.89	16.26
Tail Ratio	1.10	1.03	1.19	0.95

Table 4: Comparison of the results for three strategies.

The cumulative returns span a period of 4 years, specifically from January 1, 2019 to January 1, 2023. The technique that incorporated clustering with regimes prediction yielded an average yearly return of 42%, while the strategy that solely relied on clustering achieved a return of 30%.

Another measure we utilise is Sharpe ratio and rolling Sharpe ratio for six months. The Sharpe Ratio is a prevalent financial indicator that quantifies the success of an investment or portfolio by taking into account the level of risk

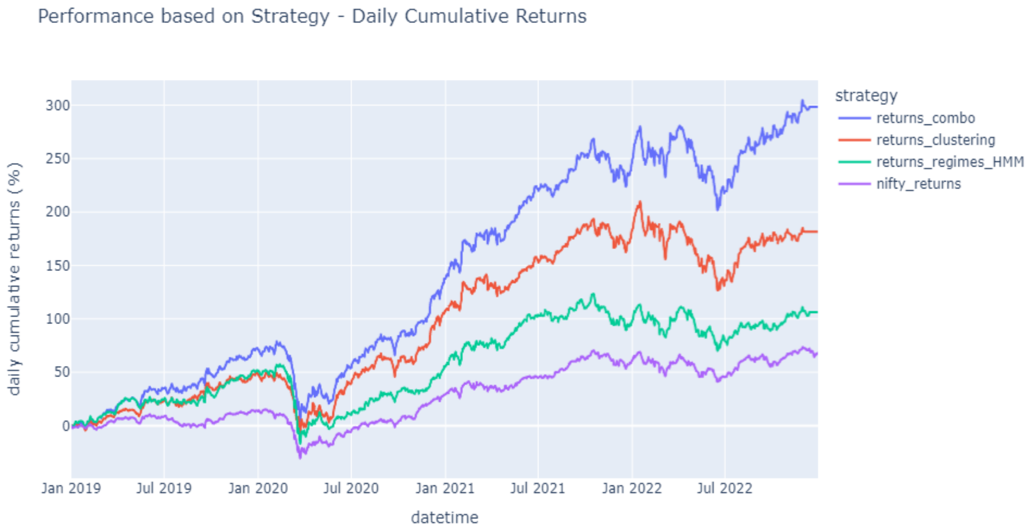
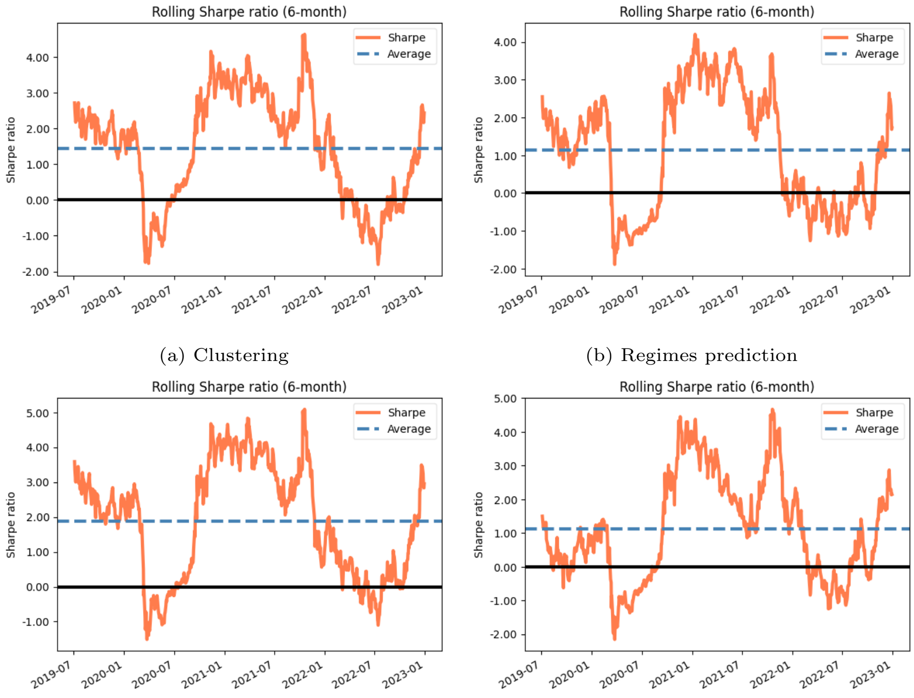


Figure 4: Cumulative returns and comparison of returns for all strategies

involved. To calculate the Sharpe ratio, divide the return difference between an investment and a risk-free rate by the return standard deviation of the portfolio. A dynamic version of the Sharpe Ratio that is calculated over a sliding time period, like six months, is called the Rolling Sharpe Ratio. Over time, changes in investment performance and market conditions are taken into account, and the metric provides a dynamic assessment of performance that takes into account both risk and market volatility. The premise that financial markets and investment returns are dynamic and susceptible to constant change is the foundation for the calculation of the rolling Sharpe Ratio. Variations in market conditions have the ability to quickly change an investment's risk and return characteristics. Investors can assess an investment's performance in light of changing market conditions by implementing a rolling window methodology to capture these swings. Using the last six months of data, the 6-month rolling Sharpe Ratio calculates the Sharpe Ratio and updates it frequently, perhaps daily, weekly, or monthly. Investors can monitor the performance of an investment over a very brief period, taking into account recent market dynamics.

An advantage of utilising a rolling Sharpe Ratio is that it offers a more dynamic assessment of performance adjusted for risk, in contrast to a fixed-time static Sharpe Ratio. Through the utilisation of a rolling window, investors can discern alterations in the risk and return attributes of an investment over time, which may not be evident in a fixed Sharpe Ratio. Figure 5 displays the rolling Sharpe ratios with 6-months period over the years for clustering (figure 5a), regimes prediction (figure 5b), both clustering and regimes prediction (figure 5c) and the NIFTY50 index (figure 5d). The strategy that combines regimes and clustering has the highest average Sharpe ratio (1.56), followed by clustering alone (1.17) and then regime prediction (0.79). In comparison, the benchmark index has a Sharpe ratio of 0.73. Strategies that solely rely on predicting market regimes tend to have a higher frequency of negative Sharpe ratios. Another



(a) Clustering

(b) Regimes prediction

(c) Clustering and regimes prediction

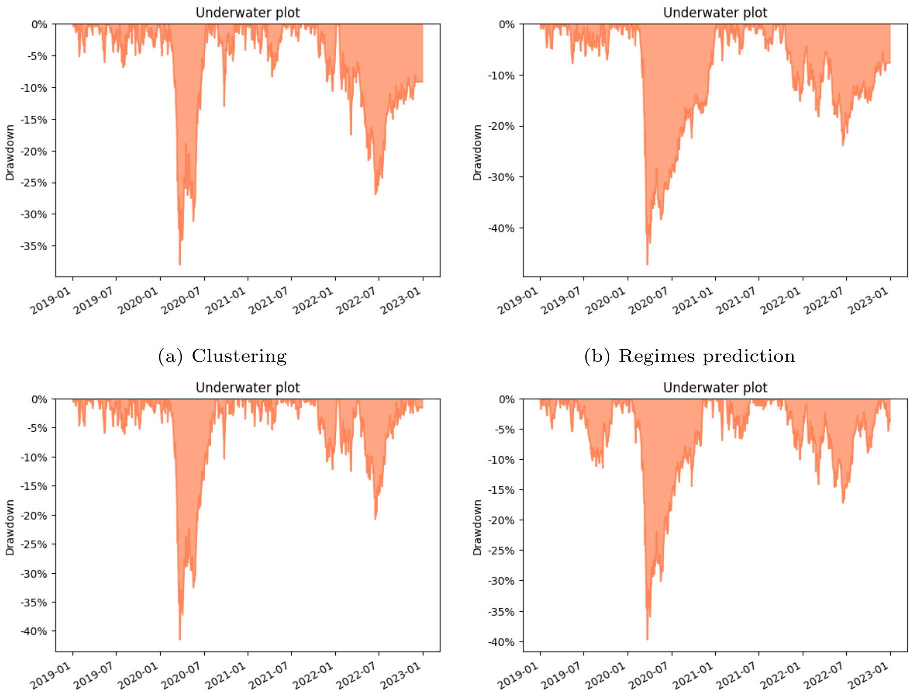
(d) NIFTY50 benchmark index

Figure 5: Rolling Sharpe ratios for different strategies

metric utilised for backtesting is maximum drawdown and underwater plot. An underwater plot, sometimes referred to as an equity curve plot or a drawdown plot, is a visual depiction of the success of an investment or trading strategy over time. It especially illustrates the percentage decrease from a previous highest point. Drawdown is the proportional decrease in the value of an investment or trading strategy from its highest point. It quantifies the degree to which an investment has decreased in value from its peak. Trading and investing will always involve drawdowns because all investments will occasionally face financial losses. The practice of risk management in the domains of trading and investing is closely related to the underwater plot. As significant drawdowns can significantly impact the success of an investment or trading strategy, managing them well is an essential part of risk management. By analysing the amount and duration of losses, traders and investors can determine the degree of risk connected with their investment or strategy by looking at the hidden storyline. Additionally, the psychological aspect of trading and investing is reflected in the underwater plot. An investor or trader may experience emotions of dread, worry, or annoyance during periods of financial losses, which can test their emotional resilience. Understanding the underlying narrative and having the necessary tools for downturns can help traders and investors control their emotions and make wise decisions.

Using an underwater plot is a useful tool for evaluating how well a trade or investment strategy is working. It provides a graphical representation of the strategy's or investment's variations, allowing for a detailed assessment of its risk-reward characteristics. Traders and investors can find the advantages and disadvantages of their investment or plan by examining the underwater plot and making the necessary adjustments.

The magnitude of drawdowns, as exemplified by the underwater maps presented in Figure 5, is an essential element in evaluating a strategy's efficacy. Higher drawdowns indicate higher risk because they represent a higher percentage decline from earlier peaks. Assessing the maximum drawdown, which refers to the greatest percentage decrease from a peak to a trough, can offer valuable information about the potential negative risk associated with the approach. The clustering approach exhibits a maximum drawdown of -38%, the regimes prediction strategy has a maximum drawdown of -47.2%, the combined strategy of clustering and regimes prediction has a maximum drawdown of -41.5%, and the benchmark index has a maximum drawdown of -39.7%. The length of drawdowns, as depicted in the underwater plot in Figure 6, is an additional crucial criterion for evaluating the performance of a strategy. Figure 6 displays the underwater plots with 6-months period over the years for clustering (figure 6a), regimes prediction (figure 6b), both clustering and regimes prediction (figure 6c) and the NIFTY50 index (figure 6d). Extended drawdown durations may suggest lengthy periods of subpar performance, which can affect the overall efficacy of the strategy. The technique that solely relies on regime prediction and



(a) Clustering

(b) Regimes prediction

(c) Clustering and regimes prediction

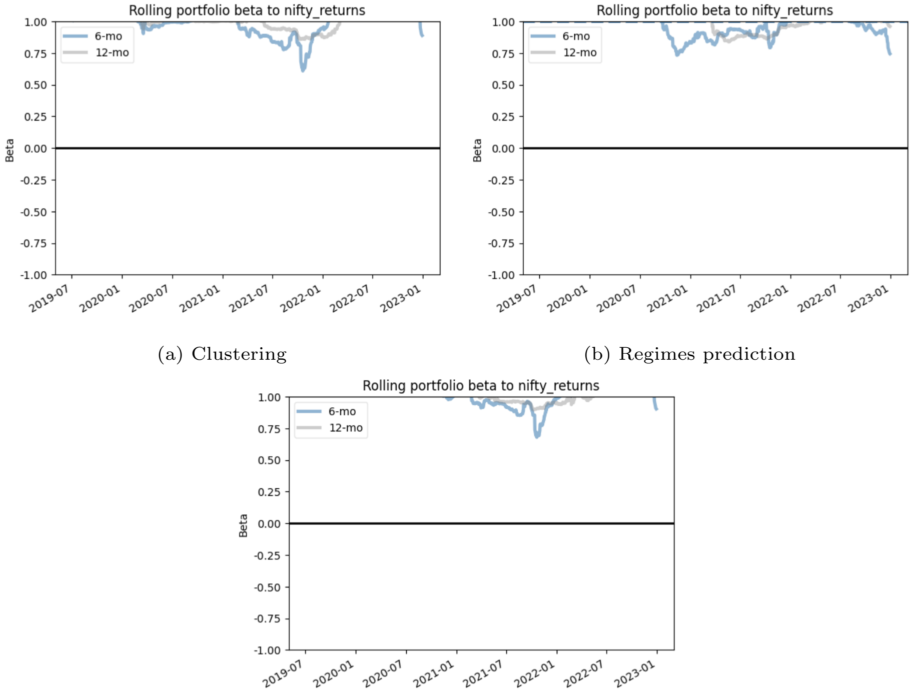
(d) NIFTY50 benchmark index

Figure 6: Underwater plots for different strategies

clustering (figure 6c) exhibits extended periods of subpar performance, which ultimately impacts the overall performance. Assessing the duration, it takes for the strategy to recover from drawdowns and achieve new peaks in equity can offer valuable information about the strategy's resilience and potential for future performance, as demonstrated by the strategy's utilisation of regimes and clustering.

One important performance parameter is the time it takes for the strategy to recover and emerge from drawdown, as seen in the underwater plot in figure 6. Extended recuperation times could indicate a reduced ability to heal quickly and could impact the method's overall efficacy. Analysing how quickly and consistently a strategy recovers from setbacks might reveal information about its capacity to recover from losses and produce profitable returns. While a volatile and erratic equity curve with frequent and big drawdowns may imply increased risk and potential concerns with the strategy's performance, a steadily increasing equity curve with smaller and shorter drawdowns may indicate a more stable and robust plan. Understanding the strategy's performance in various market conditions may be gained by examining the underwater plot's historical performance over various time periods. It is possible to determine the strategy's advantages and disadvantages in different market conditions by evaluating the strategy's performance in bull markets, bear markets, or other market stages. We validated this in the regime prediction technique. Evaluation of the strategy's success can also be aided by contrasting its underwater plot performance with the NIFTY50 benchmark index. Analysing the strategy's performance in relation to a benchmark might reveal whether it can outperform or underperform the market or a particular benchmark.

Within the framework of financial market analysis, beta is a metric that expresses the degree of risk associated with an overall market or benchmark index and measures the systematic risk of a stock. The comparison of a stock's returns to the market as a whole or to a benchmark index yields beta. The systematic risk of the entire market is assessed by the market beta, often known as the beta of the market. Regression analysis is used in the computation to compare the market or benchmark index returns to the returns of a risk-free asset, like Treasury bills. The regression line's slope indicates the market beta. When a stock's market beta is 1, it means that its returns are comparable to those of the benchmark index or the whole market. The volatility of the stock returns is higher than that of the market when the market beta is greater than 1. This suggests that the stock is more sensitive to changes in the market, which causes its returns to fluctuate more than those of the market as an entire entity. If the market beta is below 1, the stock's results exhibit lower volatility compared to the market. Consequently, the stock exhibits lower market sensitivity, resulting in relatively smaller changes in its returns compared to the overall market's returns when responding to market swings. The figure 7 displays the beta values with 6-months period over the years using distinct strategies including cluster-



(a) Clustering

(b) Regimes prediction

(c) Clustering and regimes prediction

Figure 7: Beta plots for different strategies

ing (figure 7a), regimes prediction (figure 7b), and both clustering and regimes prediction (figure 7c). A portfolio with a high market beta is more volatile than a portfolio with a low market beta, as it is more responsive to changes in the market. Consequently, a portfolio with a high market beta exhibits a larger likelihood of experiencing both substantial gains and significant losses. A rolling beta plot is a tool used by investors and portfolio managers to assess the risk and return attributes of a company or portfolio across different time periods.

Using a rolling window of historical data, the rolling beta plot is created by calculating the beta of the stock or portfolio. A predefined number of historical periods, such as days, weeks, or months, are frequently included in the rolling window. Let's look at a rolling beta plot with a rolling window of six and twelve months as an example. Next, we will calculate the stock or portfolio's beta for the last sixty days. The stock or portfolio may have seen an increase in volatility and a higher possibility for both returns and losses if the rolling beta plot shows an upward trend in the beta of the stock or portfolio. If the rolling beta plot demonstrates a declining trend in the beta of the stock or portfolio, it suggests that the stock or portfolio has experienced a reduction in volatility and possesses a diminished potential for generating both significant returns and losses.

5. Conclusion

Numerous studies have been conducted on stock clustering and predicting regimes. We address the unaddressed research gap by employing a mix of Hidden Markov Models (HMM) for regime prediction and stock clustering based on various technical indicators. We have created a stock selection framework that surpasses two strategies: stocks selection using stock market clustering and stock market regimes prediction. In addition, we have selected the standard stock clustering and regimes prediction models as our competing models to assess the predictability of the combination of regimes and clustering model. A distinct prediction model is created for each cluster inside each regime. Regimes indicate periods of market conditions characterised by either a strong upward trend (bullish) or a strong downward trend (bearish). Empirical findings demonstrate that our new model, which incorporates regime-switching dynamics and clusters stocks based on technical indicators, performs exceptionally well in terms of both statistical significance and economic value. Specifically, the combination model yields the most favourable outcomes in terms of annual returns. In addition, the integrated model demonstrates superior performance compared to other models. We utilise many indicators like as the Sharpe ratio, Calmar ratio, maximum drawdown, omega ratio, and others to evaluate and validate our outcomes through backtesting. Significantly, we see that the model exhibited superior performance not only in relation to yearly gains, but also in regard to risk metrics such as the Sharpe ratio and maximum drawdown. It is imperative to examine the mechanism of combining prediction and clustering methods for portfolio optimization.

The study focused on a portfolio that assigned equal weight to each asset for the purpose of backtesting. Employing various strategies to optimise a portfolio has the potential to enhance both returns and other performance indicators. The study also provides valuable insights into the function of probability in selecting stocks from a pool of options to achieve superior outcomes. We employ a random forest ensemble model to forecast the probability of an increase in stock prices. The studies were conducted using a dataset that includes companies from the NIFTY50 index traded in the Indian stock market between July 2007 and January 2023. The method of selecting stocks involved utilising the probability forecast generated by a random forest model for a particular cluster within a specified regime. Therefore, only stocks that were forecasted to have a high probability of increasing in stock price were taken into consideration.

Assumptions of the study are as follows: any number of purchases and sales can be made at the closing price of the day, the portfolio will always have equally weighted stocks (since we have selected 5 stocks i.e., each stock has 20% weight), tax calculations are ignored, dividend payments are ignored, the transaction cost is ignored, the possibility of short selling is ignored. Also, the training of GMM, HMM, and random forest models was designed to consider the computational power of the computer used in the analysis. In future studies, it is very important to consider the transaction costs in order to enhance the practicality of the model. In addition to technical indicators, incorporating additional fundamental indicators as input before clustering can yield improved clustering results. In future research, the utilization of ratios or financial statement items as inputs can enhance the predictive accuracy of price increases. One can employ several portfolio optimization approaches to enhance investment results. The proposed methodology can be used for the stock market data of exchanges in different countries. Additionally, this method can be utilized on various stock markets or financial time series that are appropriate for trading, such as commodity prices, option prices, and exchange rates.

Declaration of competing interest

No potential conflict of interest was reported by the author(s).

Acknowledgments

The authors acknowledge the support provided by the Indian Institute of Technology Kharagpur, India, and the Indian Institute of Management Mumbai, India.

References

Alamdari, M. K., Esfahanipour, A., & Dastkhan, H. (2023). A portfolio trading system using a novel pixel graph network for stock selection and a mean-cdar optimization for portfolio rebalancing. *Applied Soft Computing*, (p. 111213).

Alzaman, C. (2024). Deep learning in stock portfolio selection and predictions. *Expert Systems with Applications*, *237*, 121404.

Amiens, E. O., & Osamwonyi, I. O. (2022). Stock price forecasting using hidden markov model. *International Journal of Information and Decision Sciences*, *14*, 39–59.

Bagirov, S., Cavicchia, C., & Koning, N. (2022). Improving the extended ultrametric covariance structure (eucovs) for gaussian mixture model clustering: Application on financial ratios, .

Banerjee, P., & Nayak, R. (2024). Recommendations on financial models for stock price prediction. *SN Computer Science*, *5*, 178.

Baum, L. E., & Petrie, T. (1966). Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, *37*, 1554–1563.

Baum, L. E., Petrie, T., Soules, G., & Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, *41*, 164–171.

van den Berg, H. (2022). Temporal clustering of commodity market trading days: Clarifying volatility regimes with the gaussian mixture model with extended ultrametric covariance structure, .

Caporale, G. M., & Spagnolo, N. (2004). Modelling east asian exchange rates: a markov-switching approach. *Applied Financial Economics*, *14*, 233–242.

Chaudhari, K., & Thakkar, A. (2023). Neural network systems with an integrated coefficient of variation-based feature selection for stock price and trend prediction. *Expert Systems with Applications*, *219*, 119527.

Chauhan, A., Shivaprakash, S., Sabireen, H., Md, A. Q., & Venkataraman, N. (2023). Stock price forecasting using pso hypertuned neural nets and ensembling. *Applied Soft Computing*, *147*, 110835.

Chen, S., Gao, T., He, Y., & Jin, Y. (2019). Predicting the stock price movement by social media analysis. *Journal of Data Analysis and Information Processing*, *7*, 295–305.

Dai, D., & Jayantha, A. (2022). *Using a Gaussian Mixture Model to measure transit time bimodality and its impact on inventory decisions*. Ph.D. thesis.

Dar, G. F., Padi, T. R., Rekha, S., & Dar, Q. F. (2022). Stochastic modeling for the analysis and forecasting of stock market trend using hidden markov model. *Asian Journal of Probability and Statistics*, *18*, 43–56.

Eddy, S. R. (2011). Accelerated profile hmm searches. *PLoS computational biology*, *7*, e1002195.

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The journal of Finance*, *25*, 383–417.

Fikri, N., Rida, M., Abghour, N., Moussaid, K., El Omri, A., & Myara, M. (2022). A blockchain architecture for trusted sub-ledger operations and financial audit using decentralized microservices. *IEEE Access*, *10*, 90873–90886.

Ghosh, P., Neufeld, A., & Sahoo, J. K. (2022). Forecasting directional movements of stock prices for intraday trading using lstm and random forests. *Finance Research Letters*, *46*, 102280.

Giudici, P., & Abu Hashish, I. (2020). A hidden markov model to detect regime changes in cryptoasset markets. *Quality and Reliability Engineering International*, *36*, 2057–2065.

Hassan, M. R., Nath, B., & Kirley, M. (2007). A fusion model of hmm, ann and ga for stock market forecasting. *Expert systems with Applications*, *33*, 171–180.

Jiang, J., Wu, L., Zhao, H., Zhu, H., & Zhang, W. (2023). Forecasting movements of stock time series based on hidden state guided deep learning approach. *Information Processing & Management*, *60*, 103328.

Joshi, D. (2022). Detecting regime changes in financial markets using hidden markov models and directional changes. *IJFMR-International Journal For Multidisciplinary Research*, *4*.

Kim, S., Ku, S., Chang, W., & Song, J. W. (2020). Predicting the direction of us stock prices using effective transfer entropy and machine learning techniques. *IEEE Access*, *8*, 111660–111682.

Klem, H., Hocky, G. M., & McCullagh, M. (2022). Size-and-shape space gaussian mixture models for structural clustering of molecular dynamics trajectories. *Journal of chemical theory and computation*, *18*, 3218–3230.

Levinson, S. E., Rabiner, L. R., & Sondhi, M. M. (1983). An introduction to the application of the theory of probabilistic functions of a markov process to automatic speech recognition. *Bell System Technical Journal*, *62*, 1035–1074.

Li, S., Bai, Y. et al. (2022). Deep learning and improved hmm training algorithm and its analysis in facial expression recognition of sports athletes. *Computational Intelligence and Neuroscience*, *2022*.

Li, X., Parizeau, M., & Plamondon, R. (2000). Training hidden markov models with multiple observations-a combinatorial method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*, 371–377.

Liu, Z., Khojandi, A., Li, X., Mohammed, A., Davis, R. L., & Kamaleswaran, R. (2022). A machine learning–enabled partially observable markov decision process framework for early sepsis prediction. *INFORMS Journal on Computing*, *34*, 2039–2057.

Mamplata, J., Mamon, R., & David, G. (2022). Modelling and filtering for dynamic investment in the precious-metals market. *International Journal of Computer Mathematics*, *99*, 2382–2409.

Mor, B., Garhwal, S., & Kumar, A. (2021). A systematic review of hidden markov models and their applications. *Archives of computational methods in engineering*, *28*, 1429–1448.

Nystrup, P., Madsen, H., & Lindström, E. (2018). Dynamic portfolio optimization across hidden market regimes. *Quantitative Finance*, *18*, 83–95.

Olkhov, V. (2023). The market-based probability of stock returns. *arXiv preprint arXiv:2302.07935*, .

Ozcalici, M., & Bumin, M. (2022). Optimizing filter rule parameters with genetic algorithm and stock selection with artificial neural networks for an improved trading: The case of borsa istanbul. *Expert Systems with Applications*, *208*, 118120.

Pandey, V. S., & Sharma, J. K. (2020). Predictive accuracy of neural network model with multiple train functions for stochastic stock indices. *IUP Journal of Applied Finance*, *26*.

Puspita, R., & Wulandhari, L. A. (2022). Hardware sales forecasting using clustering and machine learning approach. *IAES International Journal of Artificial Intelligence*, *11*, 1074.

Radojičić, D., Radojičić, N., & Rheinländer, T. (2024). A comparative study of the neural network models for the stock market data classification—a multicriteria optimization approach. *Expert Systems with Applications*, *238*, 122287.

Ren, S., Bertels, K., & Al-Ars, Z. (2018). Efficient acceleration of the pair-hmms forward algorithm for gatk haplotypecaller on graphics processing units. *Evolutionary Bioinformatics*, *14*, 1176934318760543.

Seerattan, D., & Spagnolo, N. (2009). Central bank intervention and foreign exchange markets. *Applied financial economics*, *19*, 1417–1432.

Seshu, V., Shanbhag, H., Rao, S. R., Venkatesh, D., Agarwal, P., & Arya, A. (2022). Performance analysis of bollinger bands and long short-term memory (lstm) models based strategies on nifty50 companies. In *2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (pp. 184–190). IEEE.

Sisodia, P. S., Gupta, A., Kumar, Y., & Ameta, G. K. (2022). Stock market analysis and prediction for nifty50 using lstm deep learning approach. In

2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM) (pp. 156–161). IEEE volume 2.

Thakkar, A., & Chaudhari, K. (2020). Predicting stock trend using an integrated term frequency–inverse document frequency-based feature weight matrix with neural networks. *Applied Soft Computing*, *96*, 106684.

Tsai, C.-F., Lin, Y.-C., Yen, D. C., & Chen, Y.-M. (2011). Predicting stock returns by classifier ensembles. *Applied Soft Computing*, *11*, 2452–2459.

Virigineni, A., Tanuj, M., Mani, A., & Subramani, R. (2022). Stock forecasting using hmm and svr. In *2022 International Conference on Communication, Computing and Internet of Things (IC3IoT)* (pp. 1–7). IEEE.

Wang, J.-H., & Leu, J.-Y. (1996). Stock market trend prediction using arima-based neural networks. In *Proceedings of International Conference on Neural Networks (ICNN'96)* (pp. 2160–2165). IEEE volume 4.

Wang, X., & Hsieh, F. (2022). Unraveling s&p500 stock volatility and networks–an encoding-and-decoding approach. *Quantitative Finance*, *22*, 997–1016.

Wetzel, C. R., & Hamel, O. S. (2023). Applying a probability harvest control rule to account for increased uncertainty in setting precautionary harvest limits from past stock assessments. *Fisheries Research*, *262*, 106659.

Woodland, P. C., & Povey, D. (2002). Large scale discriminative training of hidden markov models for speech recognition. *Computer Speech & Language*, *16*, 25–47.

Yang, F., Chen, Z., Li, J., & Tang, L. (2019). A novel hybrid stock selection method with stock prediction. *Applied Soft Computing*, *80*, 820–831.

Yang, Y., Hu, X., & Jiang, H. (2022). Group penalized logistic regressions predict up and down trends for stock prices. *The North American Journal of Economics and Finance*, *59*, 101564.

Zhang, L., Lu, S., Ding, Y., Duan, D., Wang, Y., Wang, P., Yang, L., Fan, H., & Cheng, Y. (2022). Probability prediction of short-term user-level load based on random forest and kernel density estimation. *Energy Reports*, *8*, 1130–1138.

Zhang, M., Jiang, X., Fang, Z., Zeng, Y., & Xu, K. (2019). High-order hidden markov model for trend prediction in financial time series. *Physica A: Statistical Mechanics and its Applications*, *517*, 1–12.

Zhao, A. (2022). Improving portfolio optimization using option skewness regimes, .