DSE Stock Price Prediction using Hidden Markov Model

Raihan Tanvir* §, MD Tanvir Rouf Shawon† §, Md. Golam Rabiul Alam‡
Department of Computer Science and Engineering
BRAC University
66 Mohakhali, Dhaka 1212, Bangladesh
{*raihantanvir.96, †shawontanvir95}@gmail.com, ‡rabiul.alam@bracu.ac.bd

***Abstract*—Stock market forecasting is a classic problem that has been thoroughly investigated using machine learning and artificial neural network based tools and techniques. Interesting aspects of this problem include its time reliance as well as its volatility and other complex relationships. To combine them, hidden markov models (HMMs) have been utilized to anticipate the price of stocks. We demonstrated the Maximum A Posteriori (MAP) HMM method for predicting stock prices for the next day based on previous data. An HMM is trained by analyzing the fractional change in the stock price as well as the intraday high and low values. It is then utilized to produce a MAP estimate across all possible stock prices for the next day. The approach demonstrated in our work is quite generalized and can be used to predict the stock prices for any company, given that the hmm is trained on the dataset of that company's stocks dataset. We evaluated the accuracy of our models using some extensively used accuracy metrics for regression problems and came up with a satisfactory outcome.**

***Index Terms*—Hidden Markov Models, Stock Price Forecasting, Time Series Analysis, Price Prediction.**

I. Introduction

Stock price forecasting has been one of the most difficult issues for the AI community. Typical AI research, which is primarily focused on building intelligent solutions that are meant to imitate human intelligence, has typically gone beyond the limits of forecasting research. Stock price forecasting, however, remains severely constrained because of its non-stationary, cyclic, and stochastic character. A variety of factors influence the rate of price fluctuations in such a series, including equity, the rate of interest, options, securities, warrants, mergers and acquisitions of significant financial organizations, and so on. In such market, ordinary investors can not profit regularly. As a result, an intelligent forecasting model for the stock market would be highly demanded and of significant interest to ordinary investors.

HMMs are seen to be successful in analyzing and forecasting time-dependent events. This technique has already been used for voice recognition [1], handwriting recognition [2], facial expression recognition [3], ECG analysis [4], and other purposes. Stock market forecasting is analogous to these problems in terms of its inherent relationship with time. Based on an unseen collection of states from which transitions can be made, hidden Markov models correlate each state with a probable observation. In a similar fashion, the stock market can be seen. Investors are usually unaware of the inherent factors that influence share prices. Transitions between these underlying states are influenced by business strategy, decisions, the market environment, etc. The stock's value is the observable result that reflects these. So, HMM obviously complies with this real-world setting.

The selection of features is very crucial in this method. Several attempts have been made in the past to use the volume of trade, the stock's momentum, as well as the market's correlation and volatility. We incorporate the daily fractional differences in stock values as well as the fractional deviation of intraday maximum and minimum values, as proposed in [5]. It is essential to comprehend the fractional change to generate the desired prediction. The fractional difference between the intraday high and low values is a good predictor of volatility direction.

Despite the fact that HMMs have been utilized in this area for a long time, none of the works have contributed to the Bangladesh stock market. As a response, we opted to use a HMM-based approach to serve this purpose. To employ the technique, we use stock data from companies listed on the Dhaka Stock Exchange (DSE). For each stock, a unique HMM is trained. The sole constraint that the training dataset must satisfy is significant variability in the observations. It is resolved by effectively using long spans of time (13 years) during which the stock price swings consistently, albeit significantly.

II. Previous Works

Various studies have been conducted recently in an attempt to develop a stock market forecasting model that is flawless (or close so). In most of the forecasting research, statistical time series analysis methodologies such as the auto-regression moving average (ARMA) [11] and multiple regression approaches are utilized.

A HMM based approach for stock price forecasting [5] is deeply studied. For predicting the following dayś stock value given historical data, the authors followed the Maximum a Posteriori HMM method. The continuous HMM is trained using the fractional difference in stock values and the stock's intraday highs and lows. After much deliberation, we decided

§Raihan Tanvir and MD Tanvir Rouf Shawon have equal contributions.

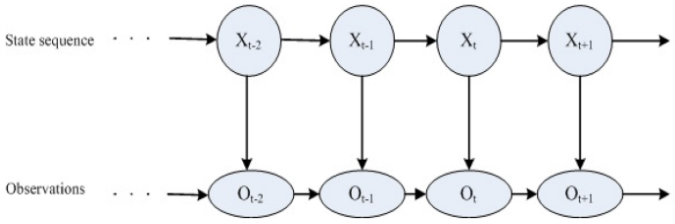

Fig. 1: Hidden Markov Model

to use the technique outlined in this paper to develop our model.

In [7], a directed-weighted chunking SVMs approach is described, where the complete training dataset is partitioned into numerous sections and support vectors for each portion are created. To create the forecast model, weighted support vector regressions are computed on the new working data set.

Md. Rafiul Hassan et al. introduced [6], an HMM-based model in which predictions are generated by interpolating the adjacent values of the dataset. The results achieved from this experiment are inspiring, and a novel framework for stock market analysis is explored.

Using autonomously generated fuzzy connections, Romahi Y et al. introduced a unique technique to dynamic financial forecasting [8]. This method has yielded promising results, but building a fuzzy system requires domain expertise.

H. Liu et al. offered a deep residual network based strategy for prediction where the stock price graph is utilized as an input [13]. This model's average accuracy was 0.40, which is higher than the stochastic indicator's average accuracy of 0.33.

Also, a naive approach demonstrated in the official documentation of the *hmmlearn* package is thoroughly investigated. In this scheme, only the difference between two consecutive closing prices of a stock is considered. Though it's a simple approach and ignores several key factors in stock price prediction, it has shown quite satisfactory results.

III. Methodology

An HMM, λ may be represented as,
$\lambda = (\pi, A, B)$,
where A is the transition matrix, the entries of which represent the likelihood of a state switching from one state to another, B is the emission matrix, that provides $b_j\left(O_t\right)$ the probability of witnessing O_t while in state j and, π represents the initial probabilities of the states at time, $t = 1$. We consider the emission probability distribution to be continuous, since the samples are a vector of continuous random variables. For simplicity, we'll consider it a multinomial Gaussian distribution having parameters, (μ and Σ)

The model's observation is a three dimensional vector representing daily stock information,

$$O_t = \left(\frac{\text{close} - \text{open}}{\text{open}}, \frac{\text{high} - \text{open}}{\text{open}}, \frac{\text{open} - \text{low}}{\text{open}}\right) := \left(frac_{Change}, frac_{high}, frac_{Low} \right) \quad (1)$$

In (1) open represents the day's opening price, close denotes the day's closing price, high and low represent the day's highest price and lowest price, respectively. To characterize the variance in stock values that stays constant throughout time, we utilize fractional changes.

After training, an approximation of the Maximum a Posteriori (MAP) technique is employed to test it. When projecting future stock prices, we assume a d day latency. As a result, having the HMM model, λ and the stock prices for previous d days $(O_1, O_2, \ldots, O_d)$, as well as the opening price for the $(d+1)^{st}$ day, the task is to calculate the closing price for the $(d+1)^{st}$ day, which is equivalent to computing the fractional difference for the $(d+1)^{st}$ day, $\frac{close-open}{open}$. This is done using the MAP approximation of the observation vector, O_{d+1}.

Lets consider $\hat{O}_{d+1}$, the MAP value of the observation on the $d+1$ day, provided the values of the previous d days.

$$\hat{O}_{d+1} = \arg\max_{o_{d+1}} P\left(O_{d+1} \mid O_1, O_2, \ldots, O_d, \lambda\right) \quad (2)$$

$$= \arg\max_{o_{d+1}} \frac{P\left(O_1, O_2, \ldots, O_d, O_{d+1} \mid \lambda\right)}{P\left(O_1, O_2, \ldots, O_d, \lambda\right)} \quad (3)$$

The observation vector (O_{d+1}) is adjusted throughout the whole range of potential values. Because the denominator is invariant with regard to (O_{d+1}) the MAP approximation reduces to (4).

$$\hat{O}_{d+1} = \arg\max_{o_{d+1}} P\left(O_1, O_2, \ldots, O_d, O_{d+1} \mid \lambda\right) \quad (4)$$

We determine the maximum probability by computing the probability across a distinct set of potential O_{d+1} values. The computational cost of determining the likelihood of a particular observation is $O(n^2 d)$, where n denotes the number of states and d denotes the latency. This process is repeated for all distinct set of potential values of O_{d+1}. We have $n = 4, d = 30$ and the number of possible values of O_{d+1} is $50 \times 10 \times 10$ (see table II). The closing price of a given day can be determined by taking the day's opening price and incorporating it with the expected fractional difference for that day as shown in (5).

$$closing\_price = opening\_price \times (1 + frac_{change}) \quad (5)$$

We also employed a naive approach, where only the differences in the closing prices of two successive days of stock are considered. The observations in this approach are the difference in closing prices for two consecutive days and the volume of stocks. After training the model, a prediction is made by computing the inner product of the transition matrix and the mean of the observations. Finally, the predicted changes are incorporated with the previous day's closing price to generate the ultimate prediction.

IV. Dataset

To put the chosen technique into practice, we selected the DSEBD[1] dataset from Kaggle. The dataset consists of annual stock data of companies registered to Dhaka Stock Exchange

[1] https://www.kaggle.com/mahmudulhaque/dsebd

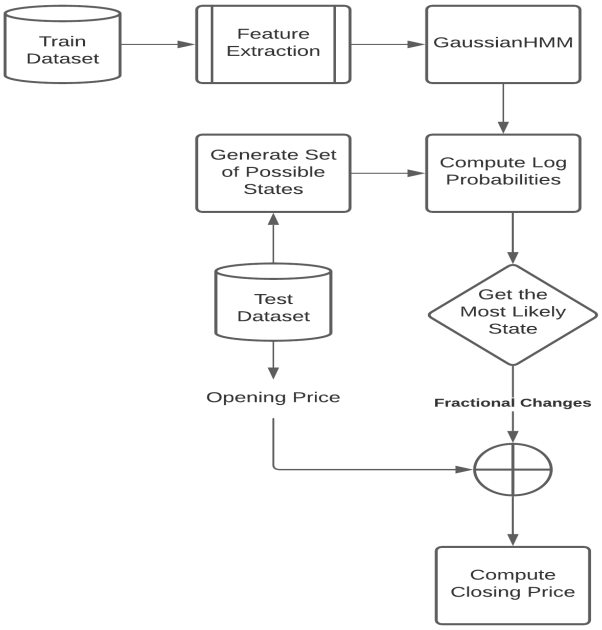

Fig. 2: Predicting Closing Price Using Fractional Changes

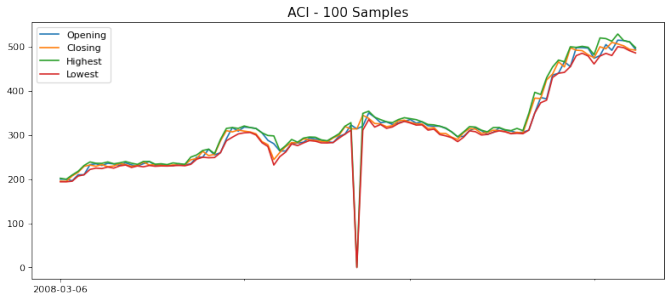

Fig. 3: Trend of opening, closing, highest and lowest prices for ACI stocks (100 samples)

(DSE) from January 2008 to December 2020 in separate JSON files. The dataset contains stock records for a total of 589 companies from 22 different sectors with 5 unique instrument types. The number of observations in the dataset is 15, 75, 134.

To fit the dataset into our scheme, the following steps of processing have been performed.

- The json files are loaded into pandas data-frame and stored into a list of data-frames
- The items of the list are concatenated into a single data-frame
- The concatenated frame is saved as csv.
- Separate Dataset is created by extracting the data for individual stocks from the concatenated frame.

The dataset is split into $0.8 : 0.2$ ratio using *train_test_split* function supplied by sklearn.model_selection module, where 80% data is used in train set and rest of the 20% is used for test set. As we are working with time-series data, we need to preserve the temporal sequence of the data. To avoid

Fig. 4: Trend of closing prices for ACI's Stocks from Jan 2008 to Dec 2020.

TABLE I: Sample data from ACI stocks dataset

date	open	high	low	close	volume	prev_close
2008-03-06	200.0	202.0	194.0	195.5	266850	198.8
2008-03-09	199.8	199.8	194.0	195.0	333600	195.5
2008-03-09	199.8	199.8	194.0	195.0	333600	195.5
2008-03-10	196.5	209.5	195.4	207.3	381650	195.0
2008-03-11	209.9	217.9	207.0	215.5	509550	207.3

random splitting of data into train and test sets, we passed *shuffle=False* as the parameter.

The trend of the stock price of **ACI** Limited is shown in the figure 3 and 4. Table I shows some sample data from **ACI** stocks dataset.

V. Implementation

As our HMM, we employed the GaussianHMM class from the hmmlearn[2] package and performed parameter approximation using the *fit* method provided by it. Because we investigated two approaches, we'll go through the implementation specifics of the first technique, which uses fractional changes in stock prices. Following that, the execution of a later strategy that takes into account successive price fluctuations in stocks is addressed.

A. HMM with Fractional Changes

1) Initialization:: The initialization of the HMM is done by setting the parameters according to following configurations:

- Quantity of hidden states, n = 4
- Dimension of observations, D = 3
- Latency, d = 30 days
- Max number of iterations, $n\_iter = 10000$
- Convergence threshold $tol = 0.001$

These values are obtained from [5], on the basis of which we're developing our model. Furthermore, [9] recommended using 4 underlying states because the dimension of observation is likewise 4. The remaining model parameters are initialized with the default values of the GaussianHMM class provided by the *hmmlearn* package.

[2]https://hmmlearn.readthedocs.io/en/latest/

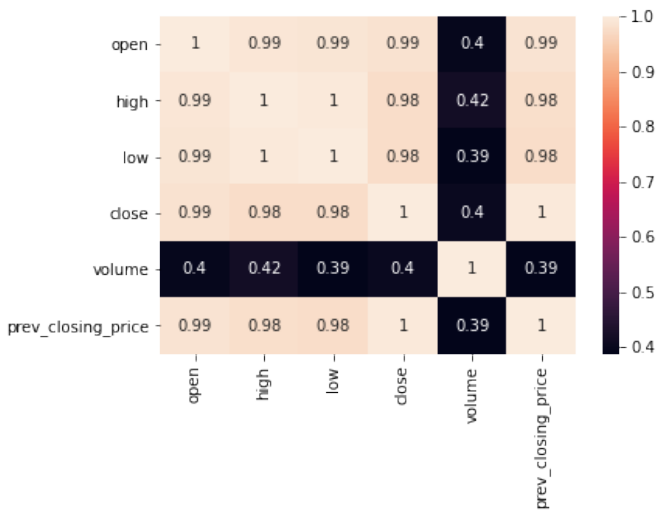

Fig. 5: Correlation matrix for ACI stocks data

2) Training:: Though our dataset has many features, not all of them have a strong influence on the closing price of a stock. From the correlation matrix in figure 5, we can see that the opening price, highest price, and lowest price have a strongly positive relationship with the closing price (exactly 1 or very close to 1). Hence, we consider these four features to be the attributes of our interest.

Now, we have relatively few characteristics for each day, primarily the starting and ending prices of the stock for that day, as well as the maximum and minimum prices of the stock. So, we utilize them to compute stock prices. Instead of directly using these values, we extracted the fractional differences in each of them that would be used to train our HMM.

Following feature extraction, some rows may include NaN, infinity, or values that are too big for dtype('float64'). A data cleansing procedure is carried out to remove such data. After performing the aforementioned two operations, the HMM is trained using the *fit* method of **GaussianHMM** class with the obtained feature vectors. Though the n_iter, was set 10000, yet the training was early stopped after 35 iterations, due to the convergence threshold, tol.

3) Prediction:: Once our model has been trained, we then estimate the stock's closing price. Provided the opening stock price for a day and the data from the previous d days, we may compute the closing stock price for that day. Our predictor would have a latency of d day. This means that if we can estimate $frac_{change}$ for a particular day, we can calculate the closing price using the (5).

The optimization of the problem would have been computationally expensive if $frac_{change}$ is considered as a continuous variable. As a result, we discretized them into values within the boundaries of two finite variables (as shown in the table II) and generated a set of fractional changes, $< frac_{change}, frac_{high}, frac_{low} >$ by deriving the cartesian products of these three variables. A higher number of steps is used for the $frac_{change}$ since these are eventually utilized for

TABLE II: Range of values for dividing fractional changes

Observation	Min Value	Max Value	Number of Steps
$frac_{change}$	-0.1	0.1	50
$frac_{high}$	0	0.1	10
$frac_{low}$	0	0.1	10

stock prediction. Then, the log probability under the model for the observation sets from the previous d days are derived using the *score* method of the GaussianHMM class, and the maximum is found. This way, the most probable outcome i.e: $frac_{change}$ is found and the closing price is computed using Equation 5. For predicting d days stock price, this method is called iteratively for those days index. Figure 6 and 7 shows the actual stock price along with the forecasted price using our trained model for **ACI Limited** and **Grameenphone Limited** respectively. The method we are following is a generalized approach and can be used to build a predictor for any company given that the HMM is trained on the dataset of that company's stock. Hence, we could use the same approach and build a separate prediction model for each of the companies listed on the Dhaka Stock Exchange.

B. HMM with Successive Fluctuations

We followed the same settings for initialization in this method as in the previously described approach. The difference is in the feature set considered for this technique.

Fig. 8: The fluctuations in closing prices for ACI stocks.

1) Training:: To train the model with this approach, we first extracted the closing prices and volumes of the stocks. Then the successive difference between stock prices is calculated and stored. Finally, the tuple of dates, consecutive differences, and volumes is stacked vertically to form the feature vectors. Finally, the model is trained with these feature vectors. It took 240 iterations for the convergence of the model.

2) Prediction:: Since our model is trained to detect the pattern of changes in the closing price, it is ready to make a prediction of the changes that would appear in the closing price of the next day, given the closing price of the previous day. To achieve this objective, the dot product of the transition matrix A and the mean of the data distribution are computed. Then we incorporated the value of the expected change with the value of the previous closing price to make our final predictions

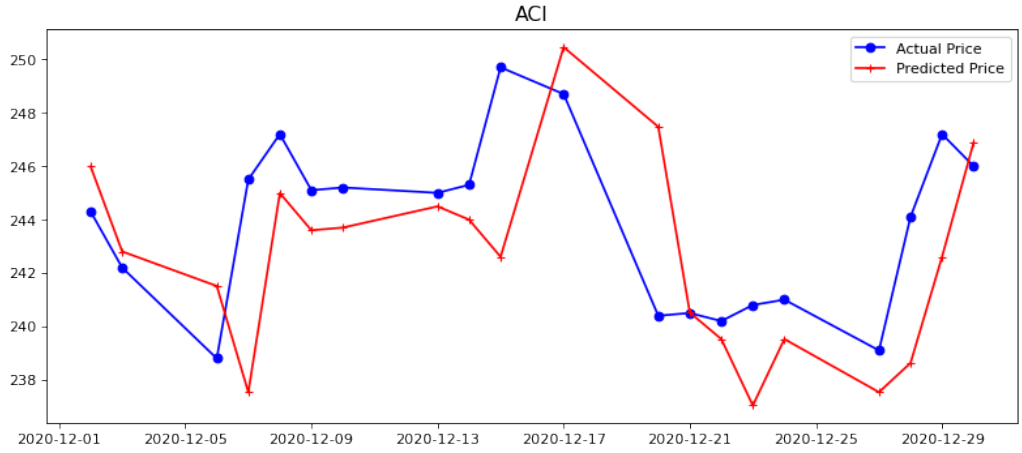

Fig. 6: Actual v/s Forecasted Value for ACI stocks from Dec 02, 2020 to Dec 30, 2020 by HMM with Fractional Changes.

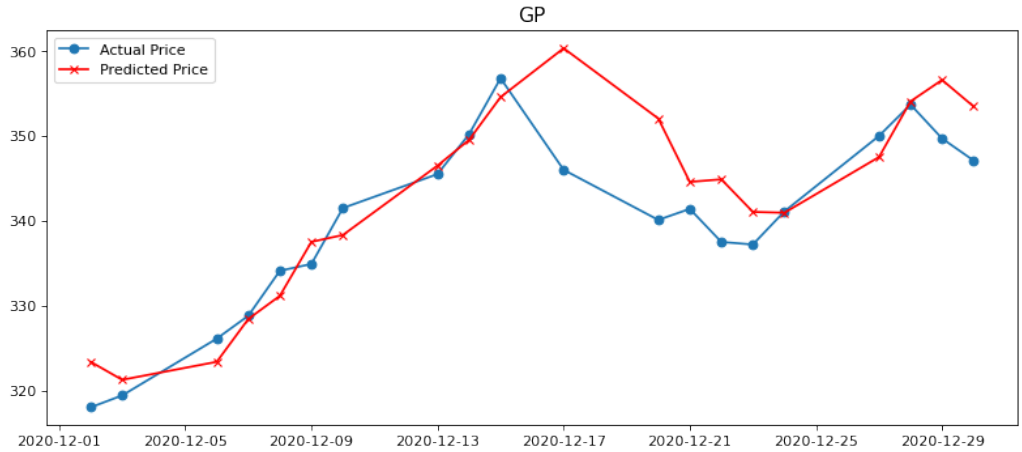

Fig. 7: Actual v/s Forecasted Value for GrameenPhone stocks from Dec 02, 2020 to Dec 30, 2020 by HMM with Fractional Changes.

on the closing price of that day. Figure 9 illustrates the *ACI Limited's* actual stock price along with the predicted price.

C. Evaluation

To assess the performance of our models, three widely accepted performance metrics for regression task have been used.

Mean Absolute Error (MAE) is a straightforward measure that computes the absolute difference between actual and projected values. It is the most robust metric for outliers.

Root Mean Squared Error (RMSE) is self-explanatory; it is the square root of the mean squared error(MSE). MSE is the mean of squared difference between the true and predicted values.

Mean Absolute Percentage Error (MAPE) is the mean absolute error between the true and forecasted values in percentage.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|p_i - y_i|}{|y_i|} \times 100\% \quad (6)$$

TABLE III: Evaluation of predictions made on ACI stocks

Metric	$HMM_{frac_{change}}$	$HMM_{succ_{fluctuation}}$
MAE	2.5064	3.3382
RMSE	3.4003	5.8141
MAPE	1.0265	1.3672

here y_i and p_i is the true and forecasted values respectively and n is the number of days for which the data are evaluated.

Table III shows the metric scores for the ACI Limiteds stocks using our trained model. Here we can see that, the *MAPE* values for the models' are 1.0265 and 1.3672. A MAPE score of less than 5 indicates that the forecast's accuracy is acceptable, according to [14]. So we can conclude that our predictors' performance is quite excellent, leaving the limitations described in the later section aside.

VI. Limitations and Future Works

We have considered a dataset that contains stock data from January 2008 to December 2020. There are many open-source scrapping tools that facilitate the retrieval of the upto date

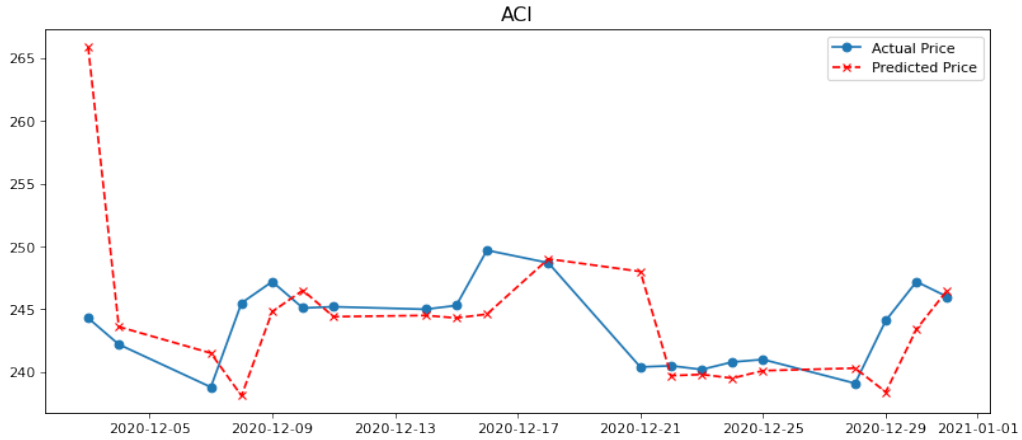

Fig. 9: Actual v/s Forecasted Value for *ACI Ltd*'s from Dec 02, 2020 to Dec 30, 2020 by HMM with Successive Fluctuations.

stock information. Due to time constraints, we could not employ them. The stock market has a volatile property, and the prices of stocks may fluctuate to a large extent. For this type of scenario, our models fail to provide accurate predictions. We are willing to overcome this limitation by using various smoothing techniques in the future.

VII. Conclusion

Artificial intelligence and machine learning techniques have been frequently used to forecast stock values in Bangladesh's stock market. To the best of our knowledge, none of them have yet exploited the efficiency of hidden markov models in their work. In our paper, we have demonstrated the capability of HMMs to predict stock prices using historical data. Two approaches have been investigated. And both of them provide satisfactory outcomes for typical cases. If we can gather more domain knowledge and apply techniques for handling the abrupt changes in stocks' values, then we may be able to achieve perfection in the accuracy of our model. Finally, we believe that our work will extend the window of research on forecasting using the Hidden Markov Model.

References

1 Negin Najkar, Farbod Razzazi, Hossein Sameti, A novel approach to HMM-based speech recognition systems using particle swarm optimization, Mathematical and Computer Modelling, Volume 52, Issues 11–12, 2010, Pages 1910-1920, ISSN 0895-7177, https://doi.org/10.1016/j.mcm.2010.03.041.

2 P. Kumawat, A. Khatri and B. Nagaria, "Comparative Analysis of Offline Handwriting Recognition Using Invariant Moments with HMM and Combined SVM-HMM Classifier," 2013 International Conference on Communication Systems and Network Technologies, 2013, pp. 140-143, doi: 10.1109/CSNT.2013.39.

3 X. Jiang, "A facial expression recognition model based on HMM," Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology, 2011, pp. 3054-3057, doi: 10.1109/EMEIT.2011.6023733.

4 R. V. Andreao, B. Dorizzi and J. Boudy, "ECG signal analysis through hidden Markov models," in IEEE Transactions on Biomedical Engineering, vol. 53, no. 8, pp. 1541-1549, Aug. 2006, doi:10.1109/TBME.2006.877103.

5 A. Gupta and B. Dhingra, "Stock market prediction using Hidden Markov Models," 2012 Students Conference on Engineering and Systems, 2012, pp. 1-4, doi: 10.1109/SCES.2012.6199099.

6 M. R. Hassan and B. Nath, "Stock market forecasting using hidden Markov model: a new approach," 5th International Conference on Intelligent Systems Design and Applications (ISDA'05), 2005, pp. 192-196, doi: 10.1109/ISDA.2005.85.

7 L. Cao and F.E.H. Tay. "Financial forecasting using support vector machines". Neural Computation and Application, pages 184–192, 2007.

8 Romahi Y and Shen Q (2000), Dynamic Financial Forecasting with Automatically Induced Fuzzy Associations, Proceedings of the 9th international conference on Fuzzy systems, pp. 493-498.

9 M.R. Hassan. "A combination of hidden markov model and fuzzy model for stock market forecasting". Journal of Neurocomputing, pages 3439– 3446, 2009

10 Haque, M. (2021, January 23). Dhaka stock exchange. Retrieved September 14, 2021, from https://www.kaggle.com/mahmudulhaque/dsebd

11 Kimoto T, Asakawa K, Yoda M and Takeoka M (1990), Stock market prediction system with modular neural networks, Proc. International Joint Conference on Neural Networks, San Diego, Vol. 1, pp. 1-6.

12 Kim S H and Chun S H (1998), Graded forecasting using an array of bipolar predictions: application of probabilistic neural networks to a stock market index. International Journal of Forecasting, Vol. 14, pp. 323-337

13 H. Liu and B. Song, "Stock Price Trend Prediction Model Based on Deep Residual Network and Stock Price Graph," 2018 11th International Symposium on Computational Intelligence and Design (ISCID), 2018, pp. 328-331, doi: 10.1109/ISCID.2018.10176.

14 Swanson, David. (2015). On the Relationship among Values of the Same Summary Measure of Error when it is used across Multiple Characteristics at the Same Point in Time: An Examination of MALPE and MAPE 1. Review of Economics and Finance. 5. 1-14.