Using a Hidden Markov Model as a Financial Advisor

Emil Lindqvist and Robert Andersson

***Abstract*—People have been trying to predict the stock market since its inception and financial investors have made it their profession. What makes predicting the stock market such a hard task is its seemingly random dependency on everything from Elon Musks tweets to future earnings. Machine learning handles this apparent randomness with ease and we will try it out by implementing a Hidden Markov Model. We will model two different stocks, Tesla, Inc. and Coca-Cola Company, and try using the forecasted prices as a template for a simple trading algorithm. We used an approach of calculating the log-likelihood of preceding observations and correlated it with the log-likelihood of all the preceding subsequences of equivalent size by turning the time window by one day in the past. The results show that modeling two stocks of different volatility is possible, but using the result as a template for trading came back inconclusive with less than 50 percent successful trades for both of the modelled stocks.**

***Sammanfattning*—Människor har försökt förutsäga aktiemarknaden sedan starten och finansiella investerare har gjort det till sitt yrke. Det som gör att förutsäga aktiemarknaden till en så svår uppgift är dess till synes slumpmässiga beroende av allt från Elon Musks tweets till framtida intäkter. Maskininlärning hanterar denna uppenbara slumpmässighet med lätthet och vi kommer att testa det genom att implementera en Hidden Markov-modell. Vi kommer att modellera två olika aktier, Tesla, Inc. och Coca-Cola Company, och försöka använda de prognostiserade priserna som bas för en enkel algoritm att handla på. Vi använde ett tillvägagångssätt för att beräkna log-sannolikheten för föregående observationer och korrelerade den med log-sannolikheten för alla föregående följder av motsvarande storlek genom att vrida tidsfönstret med en dag tidigare. Resultaten visar att det är möjligt att modellera två aktier med olika volatilitet, men att använda resultatet som en mall för handel kom tillbaka oavgörande med mindre än 50 procent framgångsrika affärer för båda modellerna.**

***Index Terms*—hidden markov models, stock market prediction**

Supervisors:
Yu Wang and Cristian Rojas

TRITA number: 145

I. Introduction

The stock market is a system that administers a platform for all large-scale economic transactions in the world at a dynamic rate labeled as stock value. Forecasting the stock value can potentially grant one with enormous profit opportunities, which have been a huge motivation for research in this area, both in academia and industry. The main hypothesis is that a probabilistically correct forecast can be extremely profitable, despite the problems like, dependence on time, seasonality and volatility. In detail, the tendency of stock market index prices point to the movement of the price index or the direction of fluctuation in the stock market index in the future. To make the correct financial decisions, the prediction of price trends is an essential tool. In spite of this essence, due to uncertainties and nonlinear factors convoluted in the data, prediction of financial time series is a tough task. In reality, a stock market is an extremely complex system, where the components who form the system, have changes in their prices without having significant patterns. Furthermore, the mood of the stock market is built upon various qualitative factors, such as natural, political and economic, which indicates non linearity and an enormous complexity to dimensionality [1]. However, investigating market behavior can lead to a better understanding of how moods alternate and thereby increases the chance of making profitable investment decisions. In general, one should seek to invest at the beginning of upward trends, namely termed as bullish markets and repel shares just in time before the prices fall again, namely termed as bearish markets.

The past years, a considerable amount of machine learning methods have been applied to the areas of financial time series prediction. There are numerous forecasting models of financial time series applying machine learning tools such as Support Vector Machines [2], Neural Networks [3] , Hidden Markov Models (HMM). HMM is a suitable approach when it comes to modeling sequential data, such as time series, based on the hypothesis of first-order Markov chain. As a matter of fact, due to the short-term and long term correlations found in empirical time series, Markov property plays an important role in financial time forecasting. In recent years, $HMMs$ have come into view as prominent tools for modeling financial time series. Hassan and Nath [4] developed an HMM for stock market forecasting, by trying to find some day in the past which is most alike with the current day in order to forecast the next day's stock price. Park and Lee [5] used continuous first-order HMM to forecast change in direction of next day's closing price. Nguyen [6] used $HMMs$ to forecast monthly closing prices and as consequence to derive an optimal trading strategy, which showed to exceed the conventional buy-and-hold strategy. All these applications establish that $HMMs$ genuinely accounts for stock market dynamics.

II. Problem formulation

The purpose of this project is to model two different stocks with *Hidden Markov Models*. We will use the models to forecast the adjusted closing price 100 days into the future. These models will be evaluated by three different evaluation methods which includes a trading analysis which uses the forecasted price as a template for a trading algorithm. We want to see if there are any difference between the two models and if it is possible to use the models for stock trading.

A. Datasets

We chose two different stocks, *Tesla, Inc.* and *the Coca-Cola Company* to model in this project. These two stocks were chosen to evaluate our models' capacity of modeling stocks of varying volatility. *Tesla, Inc.* represents the high volatility stock and *the Coca-Cola Company* represents the low volatility stock.

We used a simplified notion of stock trading where you buy and sell stocks relative closing price and therefore chose the daily adjusted closing price of both stocks as the data used in this project.

For both stocks we train our model on data between 2010-06-29 and 2018-09-29. We forecasted 100 days into the future and used the historical price of these dates to evaluate the model.

III. Hidden Markov Model

A. Theory

A Hidden Markov Model (HMM) is a stochastic process consisting of a Markov chain, which has a finite number of states X_n [7] interlinked with a stochastic process Y_n which are assumed to be dependent on the hidden process X_n. The fundamental difference between a HMM and a Markov chain is the unobservable, hidden, states which represents the underlying mood of the stock which we can not observe. The observable process Y_n is the actual stock price we can observe. These hidden states describe distinctive market moods, where each mood corresponds to its own distinct trend. The stock price can therefore be interpreted as a sequence generated by the different hidden states and thus the logical relationship between hidden states and observations can be learnt from data.

The process can be identified by a compact notation $\lambda = (A, B, \pi)$, where A is the transition matrix, whose elements $a_{ij} = P(i_{i+1} = j | i_t = i)$ are depicting the probability of a transition from one state to another. B embody the emission matrix, yielding the observation symbol probability $b_i(o_t)$, which pinpoints the probability of the observation o_t when in state i,such as $b_i(o_t) = P(o_t | i_t = i)$. In conclusion, π is the initial state distribution, such as π =$P(i_1 = i)$.

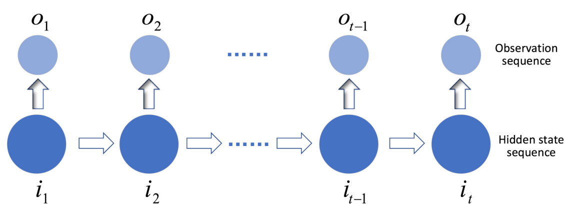


Fig. 1: Basic structure of Hidden Markov Model

Every interrelated hidden state sequence to an observation sequence $O = (o_1, ..., o_T)$ is expressed as $I = (i_1, ..., i_T)$, where $o_t = (o_t^1, ..., o_t^d)$, d is the dimension of observation value. Whereas the HMM is continuous, the emission probability is modeled as Gaussian mixture distributions,

$$b_i(o_t) = \sum_{k=1}^{K} c_{ik} g(o_t, \mu_{ik}, \Sigma_{ik}) \tag{1}$$

Where K is the number of Gaussian mixture components, c_{ik} is the mixture coefficient for the $k-th$ mixture in state i, $g(o_t, \mu_{ik}, \Sigma_{ik})$ is the multivariate Gaussian probability density function, where μ and Σ is the mean and covariance matrix for Gaussian distribution for state i [8]. Let $\alpha = \frac{1}{(\sqrt{2\pi}^d \sqrt{det(\sigma_{ik})}}$ and $\beta = exp[-\frac{1}{2}(o_t - \mu_{ik}\Sigma_{ik}^{-1}(o_t - \mu_{ik})^T]$, then

$$g(o_t, \mu_{ik}, \Sigma_{ik}) = \alpha\beta \tag{2}$$

Hence, a fully notation of the first-order HMM parameters, could be expressed as, $\lambda = \{\pi, A, c_{ik}, \Sigma_{ik}, i \in S\}$, where $S = \{0, ..., N-1\}$, note that N is the number of hidden states [9].

B. Training

For accurate and optimal use of an HMM, one have to consider the following questions:

- Considering a set of observations, what are the optimal model parameters?
- Considering a model and the corresponding set of observations, what is the optimal number of states?
- Considering a model, what is the likelihood to observe the considering set of data?

We use the $hmmlearn$ [10] package to solve the points above. The first point at issue is solved by using the Baum-Welch algorithm, which operates with Expectation-Maximization (EM) algorithm to reach for the optimal parameters for the HMM [11]. The third point at issue is solved by using the Forward algorithm [11]. For the second point at issue, we trained an arrangement of models by changing the number of states N. We varied N in a span from $[2, 15]$, where we calculated the negative log-likelihood of the training data which was in use for every of the models. The purpose of applying this was to choose the model which had the lowest value. Nonetheless, this method favors a more complex model, meaning that the number of states gravitated towards a higher number, which could result in overfitting. Afterwards, the performance of the identified sub-sequence is charted to the sub-sequence which is used for prediction. For the sake of eluding a more complex model, a penalty term to the negative log-likelihood was added. Based upon which of the penalty terms was chosen, restrictions were introduced on the model at a fluctuating degree. We analyzed two various performance measure metrics, especially, Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC). In BIC, the product of model parameters and the logarithm of the total of observation samples used was added, on the other hand, for AIC, the total of model parameters was added to the negative log-likelihood value, to attain the performance measure.

$$AIC = -2log(P(O_{trained}|\lambda)) - 2p \tag{3}$$

$$BIC = -2log(P(O_{trained}|\lambda)) - 2plog(T) \tag{4}$$

where $p = N^2 + 2N - 1$ and T is the number of observations. We have chosen the number of states, according to the performance measure of BIC.

C. Forecasting

Our implementation is based on the idea of calculating the log-likelihood of K preceding observations and correlating it with the log-likelihood of all the preceding sub-sequences of equivalent size by turning the time window by one day in the past. Note that K stands for latency and sub-sequences are sequences of equal latency as K. Next, we determine which day in the past whose log-likelihood of its K preceding observation is most correlated to the sub-sequence whose next day's price is to be predicted.

$$j = argmin_i(|P(O_t, ..., O_{t-K}|\lambda) - P(O_{t-i}, ..., O_{t-i-K}|\lambda)|) \tag{5}$$

where $i = 1, ..., \frac{d}{K}$. After that, we calculate the numerical price fluctuation from the selected day to its next day. This fluctuation is then added ongoing day's price to realize our next day's prediction.

$$O_{t+1} = O_t + (O_{t-j+1} - O_{t-j}) \tag{6}$$

Consequently, after we have attained the true observation, we incorporate it into the data set and recondition the model parameters in order to avoid model divergence. To put into perspective, the size of the sub-sequence is kept fixed, while another sub-sequence is located from the past data that displays an analogous pattern.

IV. Evaluation

Model evaluation is an essential part of the design process of a predictive model and due to the theme of this project we chose three different evaluation metrics to capture the performance of the model. We will be evaluating by statistical analysis, trend analysis with change point detection and trading analysis.

A. Statistical analysis

This part of the evaluation deals with statistical metrics as a tool to showcase the accuracy of the model. We have chosen three standard statistical metrics to evaluate the model which are seen below.

1) Mean Absolute Percentage Error (MAPE): The Mean Absolute Percentage Error (MAPE) is a statistical performance metric defined as the mean of absolute relative errors

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{7}$$

where y_i is the actual value of index i, $\hat{y}_i$ is the forecasted value of index i and n is the length of the sequence.

2) Max Error (ME): The Max Error value is the maximum residual error of the sequence. A statistical metric showcasing the worst case error between the forecasted and true value.

$$ME = max(|y_i - \hat{y}_i|) \tag{8}$$

where y_i is the actual value of index i, $\hat{y}_i$ is the forecasted value of index i and $max()$ is function returning the maximum value of the sequence.

3) Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \tag{9}$$

where y_i is the actual value of index i, $\hat{y}_i$ is the forecasted value of index i and n is the length of the sequence.

B. Trend analysis

This part of the evaluation deals with how well the model manages to capture the seasonality of the actual data. This is done by change point detection through the `python package rupture` [12].

1) Graphical analysis: We used the search method `Dynp` [13] which uses a cost function to find the minimum of the sum of the costs of all subsequences of the time series. This is done over all possible segmentation.

The cost function used in the evaluation was `CostL1` and are based on the Least Absolute Deviation.

$$c(y_i) = \sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{10}$$

where $\{y_i\}_i$ is the signal on interval I $\hat{y}_i$ is the component wise median of signal $\{y_i\}_i$

We applied `Dynp` [13] on both datasets to be able to compare the graphical representation of the result. The red highlighted areas seen on the graphs are the detected change point areas and the blue areas are areas the algorithm have not detected any significant change in regards to the cost function the algorithm is following.

2) Numerical analysis: In the numerical part of the trend analysis we use two clustering metrics, Precision and Recall and the Rand Index.

The Rand Index (RI) is a measure of similarity between two data sets and determines the accuracy of the model. In our analysis the RI will be calculated on the change points segments detected by the method `Dynp` above. The Rand Index will produce a percentage as a measure of the similarity between the two change points segments and thereby the two data sets similarity in seasonality.

$$RI = \frac{TP + TN}{TP + FP + FN + TN}$$

(11)

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives and FN is the number of false negatives.

Precision and recall is a model evaluation metric using two measurements, Precision and recall, to determine the performance of a model.

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

where TP is the number of true positives and FP is the number of false positives.

Precision measures the correctly identified change points in relation to all identified change points, correct and incorrectly labeled. This will return a percentage of how precisely the modeled identified true change points.

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

where TP is the number of true positives, FP is the number of false positives and FN is the number of false negatives.

Recall measures the correctly identified change points in relation to change points correctly identified and change points not identified. This will return a percentage of how well the model managed to recall change points.

C. Trading Analysis

This part of the evaluation deals with the topic of trading. This will be done by applying a simple trading algorithm on our forecasted data. We will later compare the predicted gain against the actual gain or loss. This will show the reliability of the model when used as a basis for trading.

We will be using *Algorithm 1* which is a simple algorithm iterating through the list of forecasted prices, buying low and selling high. The algorithm will only trade when there is a perceived gain to the investment.

V. Results

TABLE I: Results of statistical analysis

	Tesla	Coca-Cola
MAPE	4.29	0.815
ME	7.580	2.24
RMSE	1.367	0.590

Notes: A lower value indicates better performance.

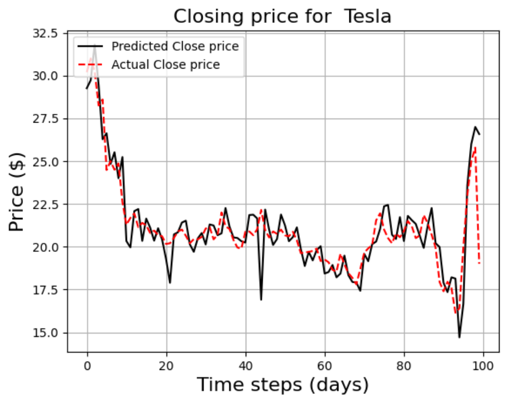
Algorithm 1 Algorithm for buying low and selling high

```
Input: list of forecasted prices
Output: calculates profit and buy/sell days
    Initialisation :
1:  $profit \leftarrow 0$
2:  $minima \leftarrow 0$
3:  $prices \leftarrow Input$
4:  for $i = 0$ to $range(prices)$ do
5:      if $(prices[i-1] > prices[i])$ then
6:          $minima = i$
7:      end if
8:      if $prices[i-1] <= price[i]$
        and $i + 1 \equiv len(price)$
        or $price[i] > price[i+1]$ then
9:          $profit += (price[i] - price[j])$
10:     end if
11: end for
12: return $profit$
```

TABLE II: Results of numerical trend analysis

	Tesla	Coca-Cola
Recall	0.6	0.8
Precision	0.6	0.8
RI	0.95	0.94

Notes: All three metrics have a value between 0 and 1. 1 indicates the datasets are equal and 0 indicates totally difference. A higher value indicates better accuracy, precision or recall.



VI. Conclusion

A. The model

One would think that the result would be greatly affected by the choice of the model, for instance the number of states in Hidden Markov model, however that was not the case. When we implemented models with a higher number of hidden states the results didn't show an analogous change in the results. In conclusion, a more complex model with a higher number of states doesn't necessarily imply a better model.

B. Statistical analysis

The results from the statistical evaluation highlighted one of the major hypothesize we had in the beginning of the project

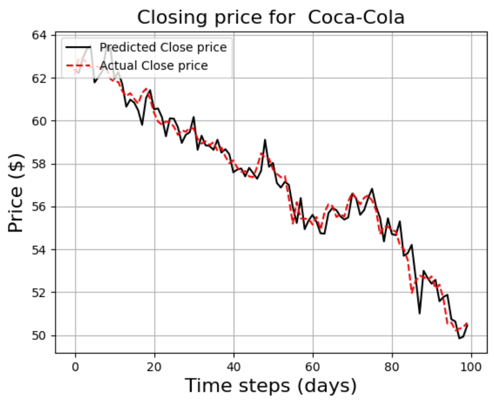


Fig. 2: Predicted closing price of Coca-Cola and Tesla

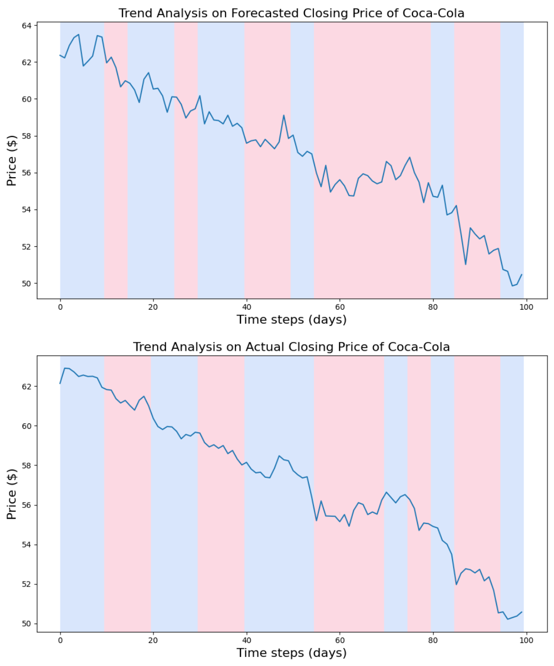


Fig. 3: The upper/lower graph shows how Algorithm 1 acted on the forecasted/actual price for Coca-Cola Company. The blue circle shows a buy and the green circle shows a sell. Every buy is followed by a sell and is considered a pair.

– would the volatility of the stock matter when modelling it? This question seems to be arbitrary, but due to the inherent risk of trading stocks, especially with volatile stock, it is crucial to evaluate the limitations of the model. As we can see from both *Figure 2* and *Table I* there are a tangible deviation in the *MAPE* and *RMSE* scores between the two models. The model based on the stock chosen as the high volatility choice, Tesla, Inc., have a higher metric on both counts and when considering the models are optimize regarding the models' parameters and uses the same time interval for the training dataset, it might indicate that the volatility of the stock does matter and could be used as a basis for future work.

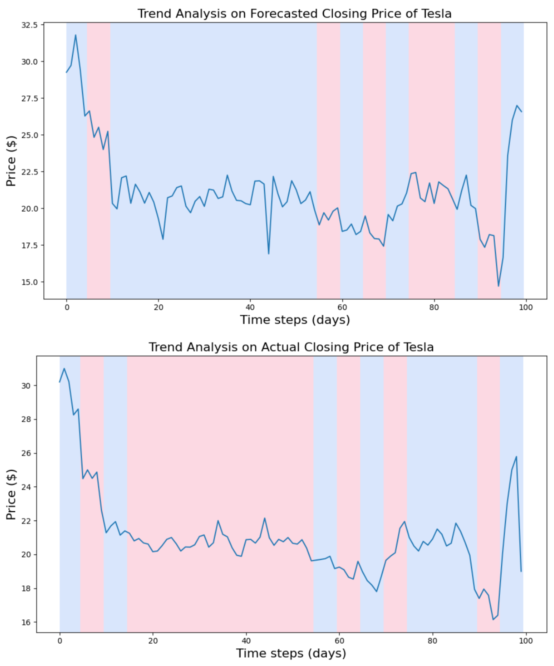


Fig. 4: The upper/lower graph shows how Algorithm 1 acted on the forecasted/actual price for Tesla, Inc. The blue circle shows a buy and the green circle shows a sell. Every buy is followed by a sell and is considered a pair.

C. Trend analysis

The results from the trend analysis seems to further strengthen the hypothesis of *the Coca-Cola Company model* being the better model in terms of statistical accuracy.

When discerning *Figure 3* we can see that the change point detection graphs of the *Coca-Cola Company* have detected areas of change points overlapping between the forecasted and actual price data. This should indicate that that the model is performing well and by looking at the numerical part of the trend analysis in *Table II* we see a RI score of *0,94* and Precision and Recall of *0,8*. These results indicates that the model managed to capture the inherent change point characteristics of the *Coca-Cola Company stock*. When discerning *Figure 4* there a major area of change points detected which are not capture by the model. This should indicate that the model in some ways have not been able to capture the inherent change point characteristics of the *Tesla, Inc.* stock. When looking at the numerical part of the trend analysis in *Table II* we see a RI score of *0,95* and Precision and Recall of *0,6*.

The Precision and Recall score of *0,6* looks reasonable considering the graphical representation of the trend analysis in *Figure 4*, but the RI score of *0,95* seems odd when you compare it to the score of the *Coca-Cola Company model*, which is smaller. This is what makes these two metrics needed together – even though the accuracy of the model is seemingly high, the Precision and Recall metrics shows that a more nuanced picture where the precision and recall are not as high as the *Coca-Cola Company model*. We have not found a good explanation for this instance and with this in mind we are

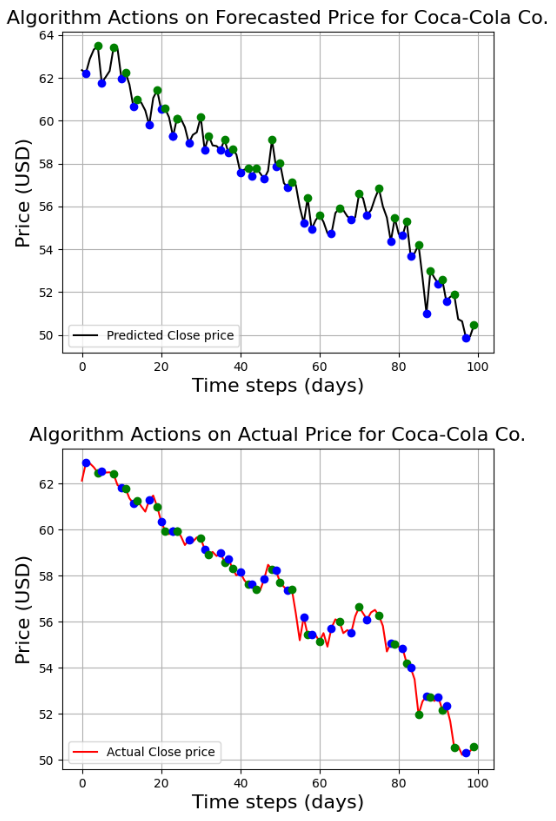


Fig. 5: The upper/lower graph shows how Algorithm 1 acted on the forecasted/actual price for Coca-Cola Co. The blue circle shows a buy and the green circle shows a sell. Every buy is followed by a sell and is considered a pair.

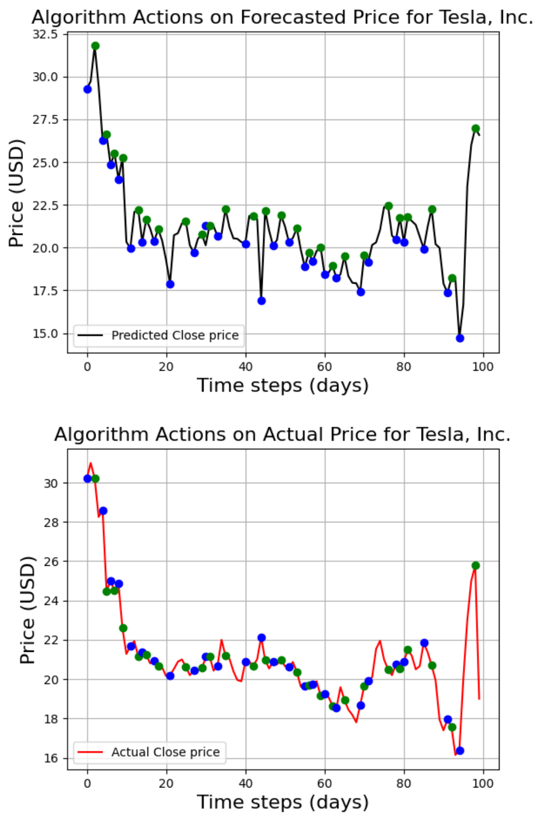


Fig. 6: The upper/lower graph shows how Algorithm 1 acted on the forecasted/actual price for Tesla, Inc. The blue circle shows a buy and the green circle shows a sell. Every buy is followed by a sell and is considered a pair.

not dismissing the RI score as a measure of accuracy, but it is needed to be accompanied with the Precision and Recall scores.

D. Trading analysis

In *Table III* we can see how *Algorithm 1* fares on the forecasted price of the *Tesla, Inc.* stock. It managed 11 trades with an actual gain out of the 25 trades with a perceived gain. That is a *44 percent success rate* when forecasting a trade with a perceived gain. The biggest discrepancy between a perceived gain and an actual gain is *26,4 percent*, meanwhile the biggest discrepancy between a perceived gain and actual loss is *36,5 percent*. The model did however manage to forecast Trade 25 in *Table III* which resulted in a trade with an actual gain of *57,3 percent*. Out of the 5 trades with the biggest perceived gain 4 trades were an actual gain, but out of these actual trade gains there were an average discrepancy of *16,4 (16,375) USD* between the perceived and actual gain.

In *Table VI* we can see how *Algorithm 1* fares on the forecasted price of the *Coca-Cola Company* stock. It managed 8 trades with an actual gain out of 28 trades with a perceived gain. That is a *29 percent success rate* when forecasting a trade. The biggest discrepancy between a perceived gain and an actual gain is *2,5 percent*, meanwhile the biggest discrepancy between a perceived gain and actual loss is *4,7 percent*.

The suitableness of *Algorithm 1* can be questioned and it might be the case that another algorithm would have fared better. That is a possibility and makes it difficult to draw any conclusion from this evaluation concerning the fitness of these models in stock forecasting. The simplicity of how *Algorithm 1* operates makes it a valuable indicator if the model is accurate. When buying low and selling high with no restrictions an accurate model is needed.

The discrepancy between the performance of the *Coca-Cola Company model* and the *Tesla, Inc.* model in the statistical and trend analysis and the trading analysis. One would have thought a model performing well in the prior two evaluation methods would also perform well in the third – this further highlight the inconclusiveness of the trading analysis.

E. Summary

As we can see from the results it is possible to model both stocks and as we hypothesized the stock of low volatility had better results on the statistical and trend analysis metrics. The results from the trading analysis is inconclusive and makes the hunt for a better performing *Hidden Markov Model* a future endeavour.

TABLE III: Results from trading analysis
Tesla

	Perceived gain	Actual gain/loss
Trade 1	+8,6	±0
Trade 2	+1,3	-14,4
Trade 3	+2,7	-2,0
Trade 4	+5,2	-9,3
Trade 5	+11,2	-12,5
Trade 6	+6,4	-0,7
Trade 7	+3,6	-1,2
Trade 8	+20,3	+2,2
Trade 9	+5,6	+0,6
Trade 10	+5,8	+0,4
Trade 11	+7,7	+2,4
Trade 12	+8,0	-1,0
Trade 13	+31,2	-5,3
Trade 14	+8,8	+0,5
Trade 15	+4,0	-1,3
Trade 16	+4,4	+0,2
Trade 17	+2,7	-3,0
Trade 18	+7,0	-3,2
Trade 19	+7,5	+2,2
Trade 20	+17,2	+5,1
Trade 21	+6,3	+3,0
Trade 22	+7,2	-1,1
Trade 23	+11,7	+2,8
Trade 24	+5,0	-5,3
Trade 25	+83,7	+57,3

Notes: The trades 1-25 consists of a buy and sell pair visible in Figure 5. Green numericals with a '+' in front means a gain, red numericals with a '-' in front means a loss and ±0 means approximately no gain nor loss.

TABLE IV: Results from trading analysis
Coca-Cola

	Perceived gain	Actual gain/loss
Trade 1	+2,0	-1,0
Trade 2	+2,7	-0,3
Trade 3	+0,5	-0,1
Trade 4	+0,5	+0,2
Trade 5	+2,7	-0,5
Trade 6	+0,06	-0,7
Trade 7	+1,4	±0
Trade 8	+2,1	+0,1
Trade 9	+1,1	-0,4
Trade 10	+0,8	-0,7
Trade 11	+0,3	-0,7
Trade 12	+0,3	-0,9
Trade 13	+0,7	-0,5
Trade 14	+3,2	+0,7
Trade 15	+0,3	-0,9
Trade 16	+0,4	+0,1
Trade 17	+2,1	-1,4
Trade 18	+1,2	-0,5
Trade 19	+2,2	+0,5
Trade 20	+2,2	+2.0
Trade 21	+2,2	+0,3
Trade 22	+2,0	±0
Trade 23	+1,2	-1,5
Trade 24	+0,9	-3,8
Trade 25	+3,9	±0
Trade 26	+0,3	-1,1
Trade 27	+0,6	-3,5
Trade 28	+1,2	+0,5

Notes: The trades 1-28 consists of a buy and sell pair visible in Figure 5. Green numericals with a '+' in front means a gain, red numericals with a '-' in front means a loss and ±0 means approximately no gain nor loss.

VII. Future Work

We would like to further investigate the hierarchical Hidden Markov Model which uses a hierarchical structure of multiple Hidden Markov Models. This model have been shown to successfully model stocks with a high accuracy when using different scales, coarse and fine, of time intervals in the different layers. It would be interesting to apply the hierarchical structure to the same trading analysis as in this project to see if it would perform better.

VIII. Acknowledgement

We want to thank our supervisor Yu Wang for putting up with all our questions, rescheduling and mishaps.

References

[1] T. Cazenave and S. B. Hamida, "Forecasting financial volatility using nested monte carlo expression discovery," in *2015 IEEE Symposium Series on Computational Intelligence*, 2015, pp. 726–733.
[2] Y. Lin, H. Guo, and J. Hu, "An svm-based approach for stock market trend prediction," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, 2013, pp. 1–7.
[3] D. M. Q. Nelson, A. C. M. Pereira, and R. A. de Oliveira, "Stock market's price movement prediction with lstm neural networks," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 1419–1426.
[4] M. R. Hassan and B. Nath, "Stock market forecasting using hidden markov model: a new approach," in *5th International Conference on Intelligent Systems Design and Applications (ISDA'05)*, 2005, pp. 192–196.
[5] S.-H. Park, J.-H. Lee, J.-W. Song, and T.-S. Park, "Forecasting change directions for financial time series using hidden markov model," in *Rough Sets and Knowledge Technology*, P. Wen, Y. Li, L. Polkowski, Y. Yao, S. Tsumoto, and G. Wang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 184–191.
[6] N. Nguyen, "Hidden markov model for stock trading," *International Journal of Financial Studies*, vol. 6, no. 2, 2018. [Online]. Available: https://www.mdpi.com/2227-7072/6/2/36
[7] Z. Lu, "Hidden markov models for time series: An introduction using r, 2nd edition, by walter zucchini, iain l. macdonald, and roland langrock. monographs on statistics and applied probability 150, published by crc press, 2016. total number of pages: 28+370. isbn: 978-1-4822-5383-2 (hardback)," *Journal of Time Series Analysis*, vol. 39, 09 2017.
[8] D. C. Scott L. Miller. (2012) Probability and random processes (second edition). [Online]. Available: https://www.sciencedirect.com/topics/mathematics/gaussian-probability-density-function/
[9] M. Zhang, X. Jiang, Z. Fang, Y. Zeng, and K. Xu, "High-order hidden markov model for trend prediction in financial time series," *Physica A: Statistical Mechanics and its Applications*, vol. 517, pp. 1–12, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378437118314018
[10] D. Tsai, "Unsupervised learning and inference of hidden markov models," [Online]. Available from: https://hmmlearn.readthedocs.io/en/latest/tutorial.html, May 25 2021.
[11] N. Nguyen, "An analysis and implementation of the hidden markov model to technology stock prediction," *Risks*, vol. 5, no. 4, 2017. [Online]. Available: https://www.mdpi.com/2227-9091/5/4/62
[12] C. Truong, L. Oudre, and N. Vayatis, "Selective review of offline change point detection methods," *Signal Processing*, vol. 167, p. 107299, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165168419303494
[13] E. Paris-Saclay. (2017) Exact segmentation: dynamic programming. [Online]. Available: https://centre-borelli.github.io/ruptures-docs/user-guide/detection/dynp/