J. Indones. Math. Soc.
Vol. 32, No. 3 (2026), pp. 1–18.

Modelling The US Dollar Index Using Continuous Hidden Markov

David Vijanarco Martal[1], Berlian Setiawaty[1*], Retno Budiarti[1]

[1]School of Data Science, Mathematics and Informatics, IPB University, Indonesia

Abstract. This study employs the continuous hidden Markov model (HMM) to model the index data of the US Dollar index from 2018 to 2024. HMM is used to predict and analyze hidden patterns that generated the data. The data is modelled using a continuous HMM with 11 hidden states and lognormal distributions with different parameters for each hidden state. The accuracy of the continuous HMM is measured by MAPE. The MAPE value for both training and testing data is very low, less than 4%. This means that the continuous HMM can be used to model the data accurately. The plot shows accurate predictions between simulated data and real data, and furthermore, the model can capture the fluctuations of the data.

Key words and Phrases: US Dollar index, Continuous hidden Markov model, price prediction.

1. INTRODUCTION

The US Dollar index (USDX) is an indicator that reflects the price of US Dollar futures contracts against several major world currencies, such as the Euro (EUR), Japanese Yen (JPY), British Pound Sterling (GBP), Canadian Dollar (CAD), Swedish Krona (SEK), and Swiss Franc (CHF). The fluctuations in the value of the US Dollar index can provide clues about the strength of the US economy, Federal Reserve monetary policy, and market sentiment towards the currency. As a crucial variable in international finance, exchange rates have a significant impact on decision-making in international financial markets. For example, the crude oil trade is denominated in the US Dollar. The changes in the US Dollar index have a direct impact on crude oil prices [1]. Initially, futures contracts were commonly used by hedgers or traders to hedge against the commodities they traded. However, over time, the high volatility in commodity futures markets has been exploited by speculators, also known as traders, to gain financial profits. Futures contracts in

*Corresponding author : berlianse@apps.ipb.ac.id
2020 Mathematics Subject Classification: 60J20, 62P05, 91G15.
Received: 22-03-2025, accepted: 14-02-2026.

futures exchanges are alternative investment instruments that investors can use to gain profits and minimize risks when they receive new information [2]. Information from currency futures markets, including the US Dollar index, can be used to forecast exchange rate movements and their implications in economic decision-making [3].

Research on this index is important because exchange rate fluctuations could influence economic activity on global financial markets and international trade activities [4]. Modelling the US Dollar index is crucial for the financial market, as it provides valuable insights for stakeholders in managing currency risk, making informed investment decisions, and planning effective business strategies. By understanding the dynamics and predicting the movement of the US Dollar, market participants can anticipate currency volatility, manage currency risk exposure, and make smarter asset allocation decisions. The US Dollar index was recently researched to see its impact during the Covid-19 crisis [5]. Previous research has also shown a relationship between the US Dollar Index and other financial factors, such as the S&P 500 index [6]. The US Dollar index can be modelled using historical data by examining the factors influencing it. One model that can be used is the hidden Markov model (HMM). This model uses hidden states to represent market conditions. In this model, a sequence of observations is inferred using sequences of hidden states.

Besides HMM-based approaches, several other statistical and machine learning methods have been applied to model or forecast the US Dollar index. Time series models, such as autoregressive integrated moving average (ARIMA) and generalized autoregressive conditional heteroskedasticity (GARCH), have been used to study future trends formed by the Dollar Index [7]. In addition, various machine learning methods, such as random forest, support vector machines, and artificial neural networks, have been utilized and compared in predicting the US Dollar Index [8]. Moreover, hybrid models that combine statistical and deep learning methods have shown improved predictive performance in modelling complex financial indicators, such as the US Dollar index [9]. These diverse approaches demonstrate that various modelling frameworks can be utilized to understand and forecast USDX dynamics, depending on the research objective and data characteristics. However, the utilization of trend changes based on probabilistic approaches has not been extensively explored.

The hidden Markov model has been widely used in the past, such as in measuring earning quality [10], sleep monitoring for physical conditions [11], electrocardiogram (ECG) analysis [12], and seismic analysis [13]. In biochemistry, HMM has been used to model diploid plant crossing [14]. Another application of HMM includes researching the progression of dementia [15]. In finance, HMM has been used to construct portfolios and asset allocation [16]. In [17], HMM has been systematically reviewed in terms of its applications and development in various fields. This approach can also be applied to other historical data, such as US dollar index data.

A hidden Markov model is based on a set of unobserved (hidden) states which form a Markov chain and each state is associated with a set of possible observations. Hidden Markov models are categorised into two types: discrete HMMs, which are used for discrete observation sequences, and continuous HMMs, which are used for continuous observation sequences. This study models the data of the US Dollar index using a continuous HMM. This choice is made because the USDX data are continuous-valued time series, making the continuous HMM framework more appropriate for capturing the probabilistic structure of such data. The data used consists of the closing index of the US Dollar index from January 1, 2018, to February 29, 2024.

This paper presents the development of a continuous hidden Markov model (HMM) in detail. The modelling process involves several algorithms whose formulations and implementation steps are also described. The analysis focuses on how the resulting model captures the movement patterns derived from the data and how the best-performing continuous HMM is identified for specific test data ranges. Furthermore, the results and discussion sections provide an in-depth examination of the model's performance and behaviour based on the simulated data. Finally, the paper concludes with a summary of the main findings, an overview of the study's limitations, and potential directions for future research and improvement.

2. CONTINUOUS HIDDEN MARKOV MODELS

According to [18], a hidden Markov model (HMM) consist of a pair of discrete-time stochastic processes $\{(X_t, O_t) : t \in \mathbb{N}\}$. $\{X_t\}$ which has a state space $S_X = \{1, 2, \ldots, N\}$, represents a collection of unobserved states that form a Markov chain. So $\{X_t\}$ is hidden behind an observation process $\{O_t\}$. Based on the type of the observations, hidden Markov models are divided into two types, discrete HMMs and continuous HMMs.

For a continuous hidden Markov model, O_t is a continuous random variable and conditional distributions of $O_t \mid X_t = i$, for $i = 1, 2, \ldots, N$ come from a certain family of parametric distributions. A continuous hidden Markov model is characterized by the parameter $\lambda = (A, \theta, \pi)$ [14] which has the following properties.

(a) $A = [a_{ij}]_{N\times N}$ is a transition probability matrix, with $a_{ij} = P(X_{t+1} = j \mid X_t = i)$, for $i, j = 1, 2, \ldots, N$ and $\sum_{j=1}^{N} a_{ij} = 1$, for $i = 1, 2, \ldots, N$.
(b) $\theta = [f(o_t \mid \theta_i)]_{N\times 1}$ is a probability density function matrix, with $f(o_t \mid \theta_i)$ is the conditional probability density function of $O_t \mid X_t = i$, for $i = 1, 2, \ldots, N$.
(c) $\pi = [\pi_i]_{N\times 1}$ is an initial probability matrix, with $\pi_i = P(X_1 = i)$ for $i = 1, 2, \ldots, N$ and $\sum_{i=1}^{N} \pi_i = 1$.

To estimate the parameters $\lambda = (A, \theta, \pi)$ from observations $o_1, o_2, \ldots, o_T$, the forward-backward algorithm, the Viterbi algorithm, and the Baum-Welch algorithm are used [18] .

2.1. **The Forward-Backward Algorithm.**

Based on the characteristic of hidden Markov model, the joint probability density function of $o_1, o_2, \ldots, o_T$ given $\lambda = (A, \theta, \pi)$ is

$$p(o_1, o_2, \ldots, o_T \mid \lambda) = \sum_{x_1, x_2, \ldots, x_T = 1}^{N} \left[\prod_{i=2}^{T} f(o_i \mid \theta_{x_i}) a_{x_{i-1} x_i} \right] f(o_1 \mid \theta_{x_1}) \pi_{x_1}. \quad (1)$$

For calculating this joint probability, [19] defines a forward variable for $t = 1, 2, \ldots, T$,

$$\alpha_t(x_t) = p(o_1, o_2, \ldots, o_T, x_t \mid \lambda), \quad x_t = 1, 2, \ldots, N. \quad (2)$$

According to [19], these are the steps of the forward-backward algorithm.

(1) Initialization

$$\alpha_1(x_1) = p(o_1, x_1 \mid \lambda) = \pi_{x_1} f(o_1 \mid \theta_{x_1}), \quad x_1 = 1, 2, \ldots, N.$$

(2) Induction for $t = 1, 2, \ldots, T-1$,

$$\alpha_{t+1}(x_{t+1}) = \left[\sum_{x_t=1}^{N} \alpha_t(x_t) a_{x_t x_{t+1}} \right] f(o_{t+1} \mid \theta_{x_{t+1}}), \quad x_{t+1} = 1, 2, \ldots, N.$$

(3) Termination

$$\alpha_T(x_T) = p(o_1, o_2, \ldots, o_T, x_T \mid \lambda), \quad x_T = 1, 2, \ldots, N,$$

so that

$$p(o_1, o_2, \ldots, o_T \mid \lambda) = \sum_{x_T=1}^{N} \alpha_T(x_T).$$

To calculate the joint probablity in equation 1, we can also use a backward algorithm. According to [20], for $t = T-1, T-2, \ldots, 0$, define a backward variable

$$\beta_t(x_t) = p(o_{t+1}, o_{t+2}, \ldots, o_T \mid x_t, \lambda), \quad x_t = 1, 2, \ldots, N. \quad (3)$$

From [21], these are the steps that need to be taken in using backward algorithm.

(1) Initialization

$$\beta_T(x_T) = 1, \quad x_T = 1, 2, \ldots, N.$$

(2) induction

$$\beta_t(x_t) = \sum_{x_{t+1}=1}^{N} a_{x_t x_{t+1}} f(o_{t+1} \mid \theta_{x_{t+1}}) \beta_{t+1}(x_{t+1}), \quad t = T-1, T-2, \ldots, 1.$$

(3) Termination

$$p(o_1, o_2, \ldots, o_T \mid \lambda) = \sum_{x_1=1}^{N} f(o_1 \mid \theta_{x_1}) \pi_{x_1} \beta_1(x_1).$$

Using the forward-backward variables in equation (2) and (3), the joint probability of $o_1, o_2, \ldots, o_T$ given λ can be calculated as

$$\begin{aligned} p(o_1, o_2, \ldots, o_T \mid \lambda) &= \sum_{x_t=1}^{N} p(o_1, o_2, \ldots, o_t, x_t, o_{t+1}, o_{t+2}, \ldots, o_T \mid \lambda) \\ &= \sum_{x_t=1}^{N} p(o_1, o_2, \ldots, o_t \mid x_t, \lambda) p(o_{t+1}, o_{t+2}, \ldots, o_T \mid x_t, \lambda) p(x_t \mid \lambda) \\ &= \sum_{x_t=1}^{N} p(o_1, o_2, \ldots, o_t, x_t \mid \lambda) p(o_{t+1}, o_{t+2}, \ldots, o_T \mid x_t, \lambda) \\ &= \sum_{x_t=1}^{N} \alpha_t(x_t)\beta_t(x_t). \end{aligned} \tag{4}$$

2.2. **The Viterbi Algorithm.**

The Viterbi algorithm is used to determine the most optimal hidden state sequence $x_1^*, x_2^*, \ldots, x_T^*$ wich generated observations $o_1, o_2, \ldots, o_T$. In other words

$$p(o_1, o_2, \ldots, o_T, x_1^*, x_2^*, \ldots, x_T^* \mid \lambda) = \max_{x_1, x_2, \ldots, x_T} p(o_1, o_2, \ldots, o_T, x_1, x_2, \ldots, x_T \mid \lambda).$$

The Viterbi algorithm keeps track of states in every step to predict most probable state sequence. The algorithm is as follows.

According to [22], defined $\delta_t(x_t)$ for $t = 1, 2, \ldots, T$ as

$$\delta_t(x_t) = \max_{x_1, x_2, \ldots, x_{t-1}} p(o_1, o_2, \ldots, o_t, x_1, x_2, \ldots, x_t \mid \lambda), \quad x_t = 1, 2, \ldots, N. \tag{5}$$

Using induction, $\delta_{t+1}(x_t)$ will be follow equation below.

$$\delta_{t+1}(x_{t+1}) = \max_{x_1, x_2, \ldots, x_t} [\delta_t(x_t) a_{x_t x_{t+1}}] f(o_{t+1} \mid \theta_{x_{t+1}}), \quad x_{t+1} = 1, 2, \ldots, N. \tag{6}$$

Below are the steps that need to be taken [22].

(1) Initialization

$$\delta_1(x_1) = \pi_{x_1} f(o_1 \mid \theta_{x_1}), \quad x_1 = 1, 2, \ldots, N.$$

$$\psi_1(x_1) = 0, \quad x_1 = 1, 2, \ldots, N.$$

(2) Recursion for $t = 1, 2, \ldots, T-1$ with

$$\delta_{t+1}(x_{t+1}) = \max_{x_t = 1, 2, \ldots, N} [\delta_t(x_t) a_{x_t x_{t+1}}] f(o_{t+1} \mid \theta_{x_{t+1}}), \quad x_{t+1} = 1, 2, \ldots, N.$$

$$\psi_{t+1}(x_{t+1}) = \arg \max_{x_t = 1, 2, \ldots, N} \delta_t(x_t) a_{x_t x_{t+1}}, \quad x_{t+1} = 1, 2, \ldots, N.$$

(3) Termination

$$p^* = \max_{x_T = 1, 2, \ldots, N} [\delta_T(x_T)].$$

$$x_T^* = \arg \max_{x_T = 1, 2, \ldots, N} [\delta_T(x_T)].$$

(4) Backtracking for $t = T, T-1, \ldots, 2$,

$$x^*_{t-1} = \psi_t(x^*_t).$$

From backtracing $\{x^*_T, x^*_{T-1}, \ldots, x^*_1\}$ is obtained

2.3. **The Baum-Welch Algorithm.**

The last algorithm is the Baum-Welch algorithm. For the sequence of observation $o_1, o_2, \ldots, o_T$ and $\lambda = (A, \theta, \pi)$, we will find $\hat{\lambda} = (\hat{A}, \hat{\theta}, \hat{\pi})$ such that

$$p(o_1, o_2, \ldots, o_T \mid \hat{\lambda}) \geq p(o_1, o_2, \ldots, o_T \mid \lambda).$$

To estimate $\hat{\lambda}$ using the Baum-Welch algorithm, the expectation maximization (EM) approach is used [23].

(1) E step

On this stage, define two functions G and H in Λ which is a set of all possible parameter for HMM. For all $\lambda' = (A', \theta', \pi') \in \Lambda$, define $G(\lambda, \lambda')$, $H(\lambda, \lambda')$, and $l(\lambda')$ as follow.

$$\begin{aligned} G(\lambda, \lambda') &= \sum_{x_T, x_{T-1}, \ldots, x_1 = 1}^{N} p(x_1, x_2, \ldots, x_T \mid o_1, o_2, \ldots, o_T, \lambda) \\ &\quad \times \log\left[p(o_1, o_2, \ldots, o_T, x_1, x_2, \ldots, x_T \mid \lambda')\right]. \qquad (7) \\ H(\lambda, \lambda') &= \sum_{x_T, x_{T-1}, \ldots, x_1 = 1}^{N} p(x_1, x_2, \ldots, x_T \mid o_1, o_2, \ldots, o_T, \lambda) \\ &\quad \times \log\left[p(x_1, x_2, \ldots, x_T \mid o_1, o_2, \ldots, o_T, \lambda')\right]. \qquad (8) \\ l(\lambda') &= \log\left[p(o_1, o_2, \ldots, o_T \mid \lambda')\right] = G(\lambda, \lambda') - H(\lambda, \lambda'). \qquad (9) \end{aligned}$$

(2) M step

On this stage, we will find $\hat{\lambda} \in \Lambda$ such that

$$G(\lambda, \hat{\lambda}) = \max_{\lambda' \in \Lambda} G(\lambda, \lambda').$$

[22] showed that

$$\begin{aligned} l(\hat{\lambda}) - l(\lambda) &= G(\lambda, \hat{\lambda}) - H(\lambda, \hat{\lambda}) - (G(\lambda, \lambda) - H(\lambda, \lambda)) \\ &= (G(\lambda, \hat{\lambda}) - G(\lambda, \lambda)) + (H(\lambda, \lambda) - H(\lambda, \hat{\lambda})) \end{aligned}$$

[24] proved that

$$H(\lambda, \lambda) \geq H(\lambda, \hat{\lambda}) \quad \hat{\lambda} \in \Lambda.$$

So that

$$l(\hat{\lambda}) - l(\lambda) \geq G(\lambda, \hat{\lambda}) - G(\lambda, \lambda) \geq 0.$$

This result prove that

$$l(\hat{\lambda}) \geq l(\lambda).$$

Because logarithmic is an increasing function, it can be concluded that

$$p(o_1, o_2, \ldots, o_T \mid \hat{\lambda}) \geq p(o_1, o_2, \ldots, o_T \mid \lambda).$$

Now, the optimization problem becomes finding the parameter $\hat{\lambda} = (\hat{A}, \hat{\theta}, \hat{\pi})$ that maximize G with two constrains. The problem is

$$\max_{\lambda' \in \Lambda} G(\lambda, \lambda')$$

with $G(\lambda, \lambda')$ in equation (7) satisfies the following equation.

$$\begin{aligned} G(\lambda, \lambda') = & \sum_{x_t=1}^{N} p(x_t \mid o_1, o_2, \ldots, o_T, \lambda) \log \pi'_{x_t} \\ & + \sum_{x_t=1}^{N} \sum_{x_{t+1}=1}^{N} \sum_{t=1}^{T-1} p(x_t \mid o_1, o_2, \ldots, o_T, \lambda) \log a'_{x_t x_{t+1}} \\ & + \sum_{x_t=1}^{N} \sum_{t=1}^{T} p(x_t \mid o_1, o_2, \ldots, o_T, \lambda) \log f'(o_t \mid \theta_{x_t}) \end{aligned} \tag{10}$$

with constrains

$$\begin{aligned} \sum_{x_t=1}^{N} \pi'_{x_t} &= 1 \\ \sum_{x_{t+1}=1}^{N} a'_{x_t x_{t+1}} &= 1, \quad x_t = 1, 2, \ldots, N. \end{aligned}$$

This problem can be solved using the multiplier Lagrange. From (10), define the Lagrange equation for $\lambda' \in \Lambda$ as

$$\begin{aligned} L(\lambda') = G(\lambda, \lambda') & + \eta_1 \left(\sum_{x_t=1}^{N} \pi'_{x_t} - 1 \right) \\ & + \sum_{x_t=1}^{N} \eta_2 \left(\sum_{x_{t+1}=1}^{N} a'_{x_t x_{t+1}} - 1 \right) \end{aligned} \tag{11}$$

with η_1 and η_2, are Lagrange multiplier variables. By solving three equations below

$$\frac{\partial L}{\partial \pi'_{x_t}} = 0, \quad \frac{\partial L}{\partial a'_{x_t x_{t+1}}} = 0, \quad \frac{\partial L}{\partial \theta'_{x_t}} = 0$$

[20] obtained the estimate parameter $\hat{A}$ and $\hat{\pi}$ as follows .

$$\hat{a}_{x_t x_{t+1}} = \frac{\sum_{t=1}^{T-1} \xi_t(x_t, x_{t+1})}{\sum_{t=1}^{T-1} \gamma_t(x_t)}, \quad \hat{\pi} = \gamma_1(x_t).$$

where

$$\xi_t(x_t, x_{t+1}) = p(x_t, x_{t+1} \mid o_1, o_2, \ldots, o_T, \lambda). \tag{12}$$

and

$$\gamma_t(x_t) = p(x_t \mid o_1, o_2, \ldots, o_T, \lambda). \tag{13}$$

These probabilities can be calculated using the forward and backward variables in equation (2), (3), and (4) as follow.

$$\xi_t(x_t, x_{t+1}) = \frac{\alpha_t(x_t) a_{x_t x_{t+1}} f(o_{t+1} \mid \theta_{x_{t+1}}) \beta_{t+1}(x_{t+1})}{\sum_{x_t=1}^{N} \alpha_t(x_t)\beta_t(x_t)}, \quad t = 1, 2, \ldots, T. \tag{14}$$

and

$$\gamma_t(x_t) = \frac{\alpha_t(x_t)\beta_{t+1}(x_{t+1})}{\sum_{x_t=1}^{N} \alpha_t(x_t)\beta_t(x_t)}, \quad t = 1, 2, \ldots, T. \tag{15}$$

Based on equation (14) and (15), both variable has relationship as follow [25].

$$\gamma_t(x_t) = \sum_{x_{t+1}=1}^{N} \xi_t(x_t, x_{t+1}), \quad t = 1, 2, \ldots, T. \tag{16}$$

For parameter $\hat{\theta}$, the result depends on distribution of $f(o_t \mid \theta_{x_t})$. In this research, the distributions used was selected from family continuous distribution such as Weibull, gamma and lognormal distributions.

3. MODELLING THE US DOLLAR INDEX

This research uses data obtained from investing.com sites. The data used are daily closing data for the US Dollar index for the period from January 1, 2018, to February 29, 2024. The number of data is 1593. The data is divided into training and testing sets, with 1274 (80%) training data and 319 (20%) testing data. This division is used to compare the accuracy of prediction results over the modelling and testing time range. A graph of the daily closing price can be seen in Figure 1.

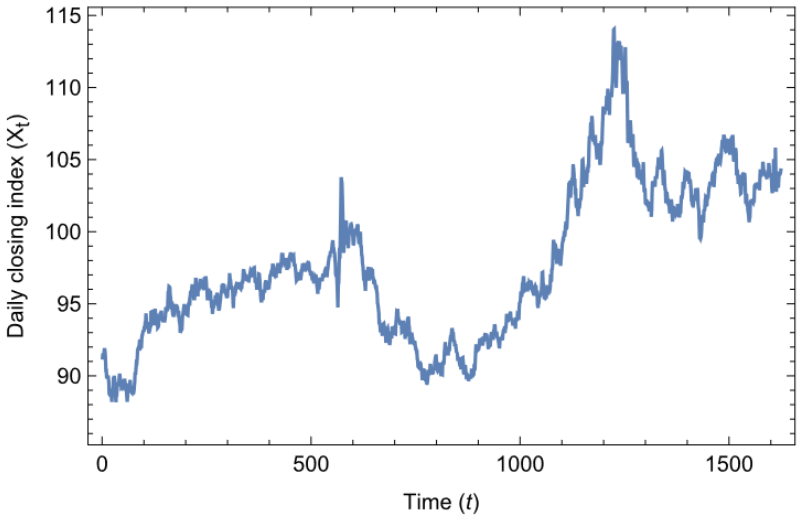

FIGURE 1. Daily closing index of US Dollar index

Figure 1 shows that daily closing data for the US Dollar index moves fluctuatingly and does not have a tendency to form a particular pattern. The data has a range of 88.170 to 114.047, with a mean of 97.638. The first step in data modelling is to determine the optimal number of hidden states and their corresponding distribution. One method for determining model selection is to utilise the Akaike information criterion (AIC).

The Akaike information criterion can be calculated using the following equation.

$$AIC = -2 \log L + 2k \tag{17}$$

where L is the likelihood value of the data and k is called the penalty term [26]. In this research, the penalty term is calculated using the following formula.

$$k = N^2 + pN - 1 \tag{18}$$

with N is the number of states in the Markov chain and p is the number of parameters from the distribution used to create the model. The likelihood of the HMM can be calculated using equation (9). Using equation (17) and (18), the results of the AIC values for several types of continuous distribution and different numbers of states can be seen in Table 1 below.

TABLE 1. AIC value for several types of distribution and different number of states

Number of states (N)	Weibull	Gamma	Lognormal
2	6366.78	7706.04	6431.47
3	5079.10	7720.04	4974.31
4	4443.11	6568.98	4261.26
5	4054.90	7760.04	3911.06
6	3742.01	7786.04	3646.61
7	3424.81	7816.04	3321.88
8	3311.56	5524.89	3196.90
9	2994.28	5086.25	2906.07
10	2918.02	5128.14	2766.96
11	2833.39	5172.04	2647.12
12	2749.49	5228.27	2559.63
13	2590.39	5262.23	2452.02
14	2540.32	5079.10	2425.92
15	2485.60	4805.12	2376.33
16	2462.89	4834.50	2347.56
17	2431.35	3668.91	2314.54
18	2467.47	3304.49	2348.06
19	2481.36	3332.27	2372.03
20	2515.42	3399.94	2406.39

From Table 1 above, it can be seen that the AIC value get smaller as N increases. We find that minimum AIC value for each state always reached by

lognormal distribution. To determine the optimal value for N, we define

$$\Delta AIC_N = \frac{\mid AIC_{N+1} - AIC_N \mid}{AIC_N}, \quad N = 2, \ldots 19 \tag{19}$$

where AIC_N is the AIC value of N-state HMM. The plot of ΔAIC_N for lognormal distributionscan be seen in Figure 2.

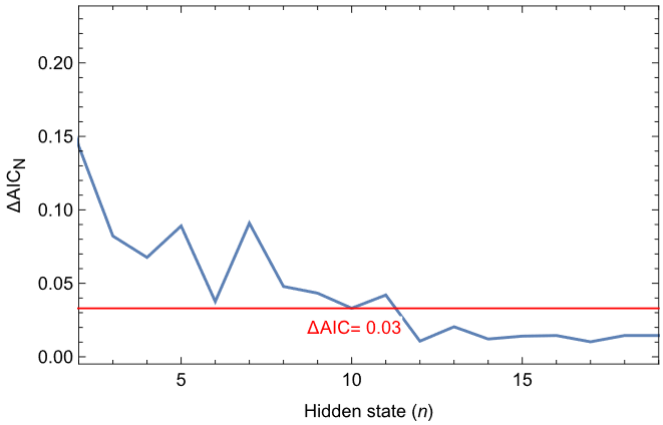

FIGURE 2. Plot of ΔAIC_N for lognormal distribution

In this research, $\Delta AIC_N < 0.03$ is set as a benchmark for the optimal value of N. The Figure 2 shows that ΔAIC_N tend to decrease as N increase. $\Delta AIC_N \leq 0.03$ occurs in state 11 and 12. The AIC value difference in state 11 and 12 is 0.033052 or almost zero, so 11 is chosen to be best number of state for HMM with lognormal distribution.

Using the Baum-Welch algorithm for $N = 11$ and lognormal distribution, the estimated parameters of the hidden Markov model consisting of the transition probability matrix between states $\hat{A}$ are obtained , the probability density function matrix $\hat{\theta}$, and the initial probability vector $\hat{\pi}$ can be calculated. Using Mathematica 12.3 software, we obtain the estimated parameter $\hat{\lambda} = (\hat{A}, \hat{\theta}, \hat{\pi})$ as follows.

$$\hat{A} = \begin{bmatrix} 0.90 & 0.03 & 0. & 0.07 & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0.03 & 0.96 & 0.01 & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0.02 & 0.98 & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0.04 & 0. & 0. & 0.90 & 0.05 & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0.05 & 0.90 & 0.04 & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0.04 & 0.85 & 0.10 & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0.06 & 0.89 & 0.05 & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0.05 & 0.92 & 0.03 & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0.03 & 0.95 & 0.02 & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0.01 & 0.98 & 0.01 \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0.02 & 0.98 \end{bmatrix}$$

$$\hat{\theta} = \begin{bmatrix} \text{lognormal}[4.522, 0.004] \\ \text{lognormal}[4.505, 0.005] \\ \text{lognormal}[4.491, 0.005] \\ \text{lognormal}[4.533, 0.003] \\ \text{lognormal}[4.543, 0.003] \\ \text{lognormal}[4.556, 0.004] \\ \text{lognormal}[4.566, 0.003] \\ \text{lognormal}[4.576, 0.004] \\ \text{lognormal}[4.596, 0.009] \\ \text{lognormal}[4.650, 0.018] \\ \text{lognormal}[4.706, 0.016] \end{bmatrix} \quad \hat{\pi} = \begin{bmatrix} 1. \\ 0. \\ 0. \\ 0. \\ 0. \\ 0. \\ 0. \\ 0. \\ 0. \\ 0. \\ 0. \end{bmatrix}$$

The matrix $\hat{A}$ shows the transition probability between hidden states. Each state tends to remain in its current state, with a slight chance of moving to a nearby state. Matrix $\hat{\theta}$ shows the parameter of the lognormal distribution for data generated by each state. The matrix $\hat{\pi}$ shows that the model started with hidden state 1.

Using the Viterbi algorithm, we can estimate the parameter $\hat{\lambda} = (\hat{A}, \hat{\theta}, \hat{\pi})$ and determine the sequence of hidden states that generated the data. These hidden states are visualized in 11 different colors. The data and the estimated hidden states are shown in Figure 3 as follows.

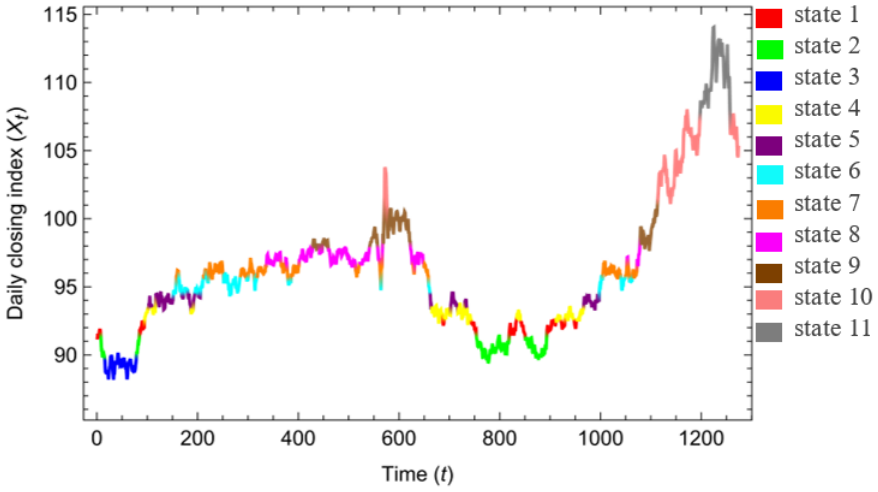

FIGURE 3. Visualization of the data and its hidden state

Figure 3 shows how hidden state moves from one state to another. This figure also shows range of the values of the data generated by each hidden state. From Figure 3, a certain amount of data is generated a hidden state. To make sure that the data for each hidden state is generated by lognormal distribution, we do the Kolmogorov-Smirnov test with the hypothesis:

H_0: that the data distributed lognormally.

H_1: that the data does not distributed lognormally. For each state, the amount of data, distribution parameters that describes the data, and the p-value result for KS-test can be seen in Table 2 below.

TABLE 2. A summary of the data for each continuous HMM hidden state with lognormal distribution

Hidden states	Number of data generated	Distribution	p-value
1	85	lognormal$[4.522, 0.004]$	0.813
2	122	lognormal$[4.505, 0.005]$	0.106
3	63	lognormal$[4.491, 0.005]$	0.627
4	128	lognormal$[4.533, 0.003]$	0.183
5	111	lognormal$[4.543, 0.003]$	0.281
6	111	lognormal$[4.556, 0.004]$	0.460
7	186	lognormal$[4.566, 0.003]$	0.384
8	176	lognormal$[4.576, 0.004]$	0.107
9	125	lognormal$[4.596, 0.009]$	0.421
10	107	lognormal$[4.650, 0.018]$	0.050
11	60	lognormal$[4.706, 0.016]$	0.300

Based on Table 2, it can be seen that the 1274 data is divided into 11 states. The amount of data for each state formed varies. The p-value for each data point for each state is greater than 0.05, so it can be concluded that there is not enough evidence to reject H_0, meaning that all data points for each state come from a lognormal distribution with the parameters that have been obtained. A comparison of the mean and variance between theoretical and actual data is presented in Table 3 below.

TABLE 3. Comparison between theoritical and actual data

States	Actual Mean	Theoritical Mean	Actual Variance	Theoritical Variance
1	92.0291	92.0202	0.1053	0.1355
2	90.4289	90.4695	0.2190	0.2046
3	89.1938	89.2117	0.2152	0.1989
4	93.0088	93.0377	0.1013	0.0779
5	93.9839	93.9727	0.0958	0.0795
6	95.1452	95.2027	0.1508	0.1450
7	96.1421	96.1591	0.0664	0.0832
8	96.9906	97.1259	0.0878	0.1509
9	97.9837	99.0912	0.1577	0.7954
10	99.6662	104.6020	0.3754	3.5456
11	103.1150	110.6230	1.1329	3.1332

Based on Table 3, it can be concluded that the actual data and the estimated distribution have almost the same mean and variance. This also suggests that

the distribution can effectively represent the original data. Even though there are differences in mean and variances values in the data range that are too high.

The HMM $\hat{\lambda} = (\hat{A}, \hat{\theta}, \hat{\pi})$ that has been obtained is then used to generate 1274 simulation data as much as the training data used. Data is generated to find the best seeds to make predictions on daily closing data for the US Dollar index. Simulations to identify the optimal seeds were conducted using Mathematica 12.3 software. To determine the accuracy of the model, this research utilises the mean absolute percentage error (MAPE) to calculate it. The MAPE can be calculated using the formula [27] as follows.

$$MAPE = \frac{1}{n} \sum_{t=1}^{n} \frac{| X_t - \hat{X}_t |}{X_t} \times 100\%, \tag{20}$$

where X_t is actual data at time t, $\hat{X}_t$ is prediction data at time t, and n is the number of data.

Based on the simulation results, the continuous HMM training yielded a MAPE of 1.39819%. Visualization of the training data and the results of continuous HMM training, with a MAPE of 1.39819% can be seen in Figure 4 below. Figure 4 shows that the model simulation results closely follow the ups and downs of the original data.

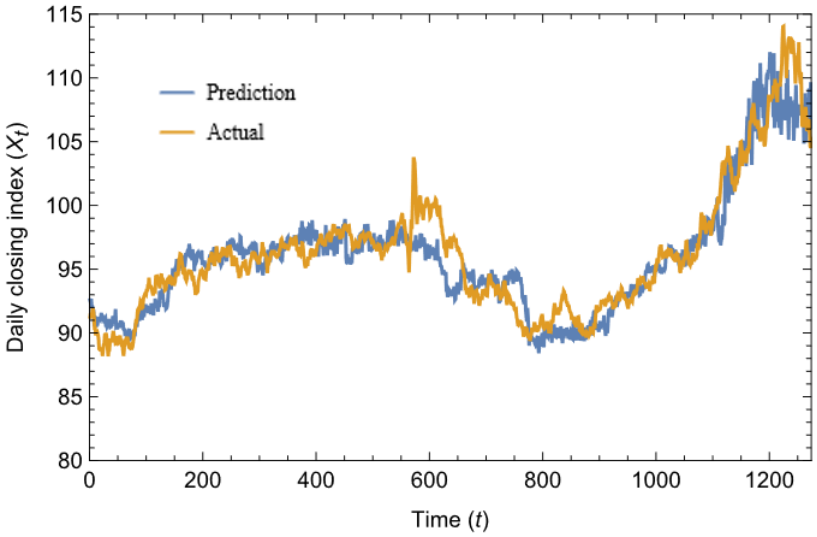

FIGURE 4. Plot training data simulation results with MAPE 1.39819%

Predictions are made on testing data using the HMM $\hat{\lambda} = (\hat{A}, \hat{\theta}, \hat{\pi})$ that has been estimated from training data. Simulations were carried out using Mathematica 12.3 software. MAPE summary results for various testing data ranges can be seen in Table 4 below.

TABLE 4. MAPE values for simulated testing data

Prediction time (days)	MAPE (%)
20	2.63494
40	2.82689
60	3.46179
80	3.94428
100	4.19032
120	4.38700
140	4.26673
160	4.52996
180	4.61455
200	4.29501
220	4.17964
240	4.15056
260	3.90854
280	3.79830
300	3.71390
319	3.66477

From Table 4, the MAPE results of simulated test data for data all range prediction times are smaller than 4%. This means that the HMM $\hat{\lambda} = (\hat{A}, \hat{\theta}, \hat{\pi})$ provides very accurate forecasts. Table 4 also show that the MAPE gets bigger if the prediction time span longer. Plot of simulated data (prediction) and actual data from continuous HMM with MAPE 3.66477% can be seen in Figure 5 below.

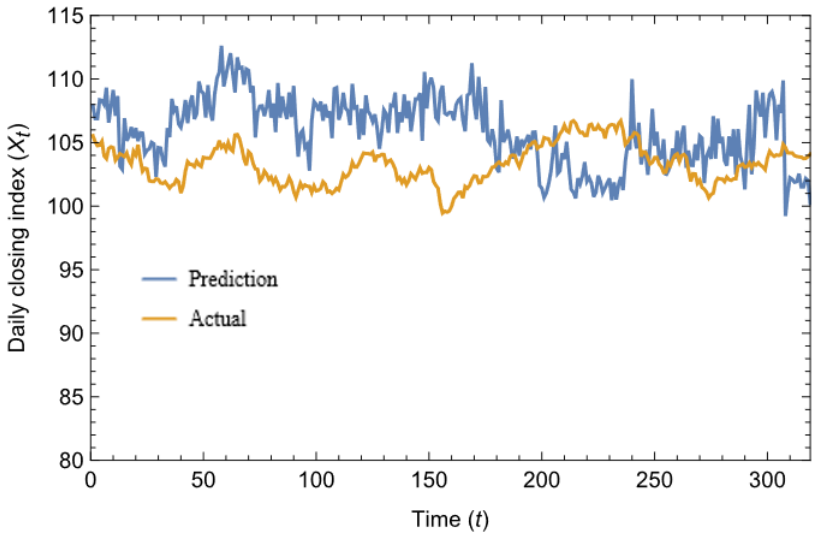

FIGURE 5. Prediction results on testing data with MAPE 3.66477%

From the Figure 5, it can be seen that the prediction results of the testing data are accurate. There is still error in this prediction, but the difference is not very significant. So we come to the coclusion that the continuous HMM can well predict the up and down of data. In Figure 6, the visualisation of the combined training and testing data is shown, illustrating the movement of the simulated data across both datasets as generated by the continuous HMM model.

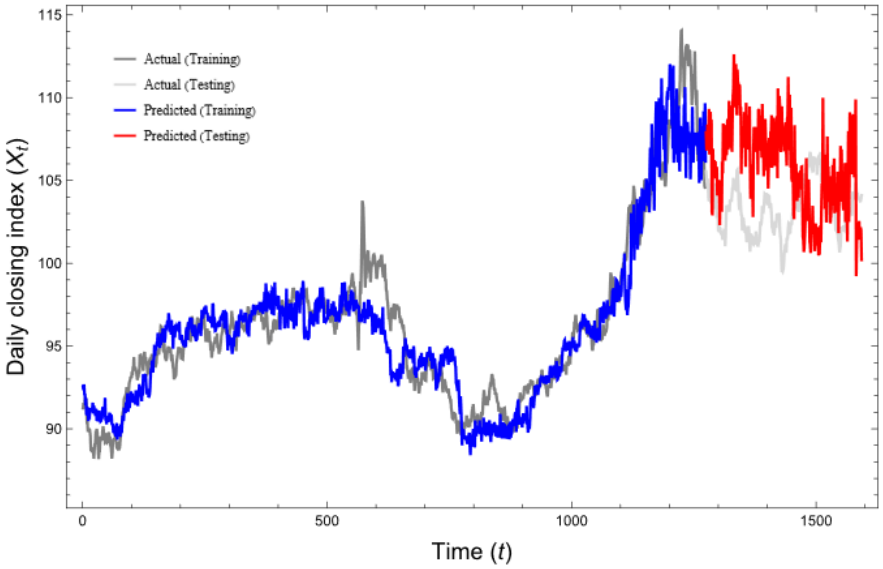

FIGURE 6. Visualization of the combined training and testing data

Figure 6 illustrates that the continuous Hidden Markov Model effectively captures the primary dynamics of the time series, including both short-term fluctuations and longer-term trends, in both the training and testing datasets. The simulated values closely follow the actual data, indicating good generalisation. However, the model still struggles to detect sudden or extreme price changes caused by rapid market shifts. Overall, these results suggest that the continuous HMM can model complex time-series patterns and provide useful forecasts for financial indicators such as the US Dollar index.

4. CONCLUDING REMARKS

Daily closing data on the US Dollar index from 2018 to 2024 can be modelled using a continuous HMM with 11 hidden states and lognormal distributions with different parameters for each hidden state. MAPE measures the accuracy of the model. The MAPE value for both training and testing data is very low, which

is smaller than 4%. This indicates that the continuous HMM can model the data very accurately. The plot shows strong agreement between the simulated and actual data, and the model effectively captures the upward and downward fluctuations of the US Dollar index.

However, this study has several limitations. The model utilises only univariate USDX time-series data and does not incorporate external factors, such as interest rates, inflation, or commodity prices, that may influence index movements. Although the use of 11 hidden states performs well empirically, it may not fully represent actual market regimes and could be improved with a more systematic selection process. Additionally, the assumption of constant parameters within each state may not accurately capture structural changes in global financial conditions.

Future research can address these limitations by incorporating multivariate inputs or relevant macroeconomic indicators. Hybrid models that combine HMM with deep learning methods, such as LSTM, may help capture more complex temporal patterns. Additionally, using dynamic HMMs or regime-switching models with time-varying parameters could better reflect changes in currency market behaviour. These improvements may enhance both the interpretability and predictive accuracy of the model for the US Dollar index and other financial time series.

Data Availability Statement. The dataset analyzed during the current study is publicly available in the investing website, accessible at: https://www.investing.com/currencies/us-dollar-index.

Conflicts of Interest. The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Author Contributions.
David Vijanarco Martal: Conceptualization, methodology, software development, formal analysis, data curation, writing original draft preparation, and visualization. Berlian Setiawaty: Validation, investigation, writing, review, editing, and supervision. Retno Budiarti: Methodology support, statistical analysis, interpretation of results, project administration, and final manuscript review. All authors contributed to refining the manuscript, discussed the results, and approved the final version.

Funding. This research was funded by School of Data Science, Mathematics, and Informatics, IPB University.

Acknowledgement. Thanks for the support of School of Data Science, Mathematics and Informatics, IPB University, in financing the publication of this article.

REFERENCES

[1] J. Zhou, M. Sun, D. Han, and C. Gao, "Analysis of oil price fluctuation under the influence of crude oil stocks and us dollar index-based on time series network model," *Physica A: Statistical mechanics and its applications*, vol. 582, p. 126218, 2021. https://doi.org/10.1016/j.physa.2021.126218.

[2] I. I. Laduni, "The influence of crude oil derivative instruments, us dollar index, leading stock index, us fed interest rates and inflation on gold futures prices: Analysis of the 2012-2021 period," *Contemporary Studies in Economic, Finance, and Banking*, vol. 4, no. 1, pp. 710–724, 2022. https://doi.org/10.21776/csefb.2022.01.4.14.

[3] C. Engel, "Exchange rates, interest rates, and the risk premium," *American Economic Review*, vol. 106, no. 2, pp. 436–474, 2016. https://doi.org/10.1257/aer.20121365.

[4] S. Avdjiev, V. Bruno, C. Koch, and H. S. Shin, "The dollar exchange rate as a global risk factor: Evidence from investment," *IMF Economic Review*, vol. 67, pp. 151–173, 2019. https://doi.org/10.1057/s41308-019-00074-4.

[5] J. J. A. Kumar and R. Robiyanto, "The impact of gold price and us dollar index: The volatile case of shanghai stock exchange and bombay stock exchange during the crisis of covid-19," *Journal of Finance and Banking*, vol. 25, no. 3, pp. 508–531, 2021. https://doi.org/10.26905/jkdp.v25i3.5142.

[6] S. D. G. F. Hau, "The impact of the us dollar value and interest rates on the s&p 500 returns," *Scientific Journal of Economics and Management Research*, vol. 6, no. 2, pp. 1–7, 2024. http://www.sjemr.org/download/SJEMR-6-2-1-7.pdf.

[7] Z. Liu and Y. Lv, "A study of the usdx predication based on arima and garch models," in *2011 Fourth International Conference on Business Intelligence and Financial Engineering*, pp. 82–85, Institute of Electrical and Electronics Engineers, 2011. https://doi.org/10.1109/BIFE.2011.9.

[8] M. F. Yürük, "Machine learning-based a comparative analysis for usa dollar index prediction," in *Finansal Piyasaların Evrimi-II*, pp. 33–52, Özgür Yayın Dağıtım Ltd. Şti., 2023. https://doi.org/10.58830/ozgur.pub105.c739.

[9] A. S. M. Tahir and F. M. Jassim, "Comparison of the two hybrid models, wavelet-arima and wavelet-es, to predict the prices of the us dollar index," *Periodicals of Engineering and Natural Sciences*, vol. 10, no. 2, pp. 219–230, 2022. https://doi.org/10.21533/pen.v10.i2.600.

[10] K. Du, S. Huddart, L. Xue, and Y. Zhang, "Using a hidden markov model to measure earnings quality," *Journal of Accounting and Economics*, vol. 69, no. 2, p. 101281, 2020. https://doi.org/10.1016/j.jacceco.2019.101281.

[11] L. Peng, A. Yin, W. Song, W. Yao, H. Ren, and L. Yang, "Sleep monitoring with hidden markov model for physical conditions tracking," *IEEE Sensors Journal*, vol. 21, no. 13, pp. 14232–14239, 2021. https://doi.org/10.1109/JSEN.2020.3007153.

[12] A. S. A. Huque, K. I. Ahmed, M. A. Mukit, and R. Mostafa, "Hmm-based supervised machine learning framework for the detection of fecg r-r peak locations," *IRBM*, vol. 40, no. 3, pp. 157–166, 2019. https://doi.org/10.1016/j.irbm.2019.04.004.

[13] D. V. Lindberg and H. Omre, "Inference of the transition matrix in convolved hidden markov models and the generalized baum-welch algorithm," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 12, pp. 6443–6456, 2015. https://doi.org/10.1109/TGRS.2015.2440415.

[14] N. Hayati, B. Setiawaty, and I. G. P. Purnaba, "The application of discrete hidden markov model on crosses of diploid plant," *Barekeng: Journal of Mathematics and Its Applications*, vol. 17, no. 3, pp. 1449–1462, 2023. https://doi.org/10.30598/barekengvol17iss3pp1449-1462.

[15] J. P. Williams, C. B. Storlie, T. M. Therneau, C. R. J. Jr, and J. Hannig, "A bayesian approach to multistate hidden markov models: Application to dementia progression," *Journal of the American Statistical Association*, vol. 115, no. 529, pp. 16–31, 2020. https://doi.org/10.1080/01621459.2019.1594831.

[16] E. Fons, P. Dawson, J. Yau, X.-j. Zeng, and J. Keane, "A novel dynamic asset allocation system using feature saliency hidden markov models for smart beta investing," *Expert Systems with Applications*, vol. 163, p. 113720, 2021. https://doi.org/10.1016/j.eswa.2020.113720.

[17] B. Mor, S. Garhwal, and A. Kumar, "A systematic review of hidden markov models and their applications," *Archives of Computational Methods in Engineering*, vol. 28, pp. 1429–1448, 2020. https://doi.org/10.1007/s11831-020-09422-4.

[18] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989. https://doi.org/10.1109/5.18626.

[19] J. C. Joshi, T. Kumar, S. Srivastava, and D. Sachdeva, "Optimisation of hidden markov model using baum-welch algorithm for prediction of maximum and minimum temperature over indian himalaya," *Journal of Earth System Science*, vol. 126, no. 1, pp. 1–9, 2017. https://doi.org/10.1007/s12040-016-0780-0.

[20] J. Li, J.-Y. Lee, and L. Liao, "A new algorithm to train hidden markov models for biological sequences with partial labels," *BMC Bioinformatics*, vol. 22, no. 162, pp. 1–12, 2021. https://doi.org/10.1186/s12859-021-04080-0.

[21] M. Firmasyah, B. Setiawaty, and I. G. P. Purnaba, "Markov models - training and evaluation of hidden markov models," *Nature Methods*, vol. 17, pp. 121–122, 2020. https://doi.org/10.1038/s41592-019-0702-6.

[22] B. Setiawaty, A. Y. Lestari, D. C. Lesmana, and N. K. K. Ardana, "Predicting the conditions of stock market using hidden markov models," vol. 3201, p. 020013, AIP Publishing, 2024. https://doi.org/10.1063/5.0230595.

[23] Q. Wang and W. B., "An expectation-maximization algorithm for continuous-time hidden markov models," *arXiv*, no. 2103.16810, pp. 1–34, 2021. https://doi.org/10.48550/arXiv.2103.16810.

[24] M. Firmasyah, B. Setiawaty, and I. G. P. Purnaba, "Parameter estimation of exponential hidden markov model and convergence of its parameter estimator sequence," *International Journal of Applied Mathematics*, vol. 31, no. 1, pp. 53–62, 2018. https://doi.org/10.12732/ijam.v31i1.5.

[25] M. El Annas, B. Benyacoub, and M. Ouzineb, "Semi-supervised adapted hmms for p2p credit scoring systems with reject inference," *Computational Statistics*, vol. 38, no. 1, pp. 149–169, 2023. https://doi.org/10.1007/s00180-022-01220-9.

[26] J. E. Cavanaugh and A. A. Neath, "The akaike information criterion: Background, derivation, properties, application, interpretation, and refinements," *WIREs Computational Statistics*, vol. 11, no. 3, pp. 1–12, 2019. https://doi.org/10.1002/wics.1460.

[27] S. S. Roy, P. Samui, I. Nagtode, H. Jain, V. Shivaramakrishnan, and B. Mohammadi-ivatloo, "Forecasting heating and cooling loads of buildings: a comparative performance analysis," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 1253–1264, 2020. https://doi.org/10.1007/s12652-019-01317-y.