




Article

Long- and Medium-Term Financial Strategies on Equities Using Dynamic Bayesian Networks

Karl Lewis [†], Mark Anthony Caruana [†] and David Paul Suda ^{*,†}

Department of Statistics and Operations Research, Faculty of Science, University of Malta, 2080 Msida, Malta; karl.lewis.17@um.edu.mt (K.L.); mark.caruana@um.edu.mt (M.A.C.)

* Correspondence: david.suda@um.edu.mt; Tel.: +356-9986-9884

† These authors contributed equally to this work.

Abstract: Devising a financial trading strategy that allows for long-term gains is a very common problem in finance. This paper aims to formulate a mathematically rigorous framework for the problem and compare and contrast the results obtained. The main approach considered is based on Dynamic Bayesian Networks (DBNs). Within the DBN setting, a long-term as well as a short-term trading strategy are considered and applied on twelve equities obtained from developed and developing markets. It is concluded that both the long-term and the medium-term strategies proposed in this paper outperform the benchmark buy-and-hold (B&H) trading strategy. Despite the clear advantages of the former trading strategies, the limitations of this model are discussed along with possible improvements.

Keywords: finance; dynamic Bayesian networks; trading strategies; equities



Citation: Lewis, K.; Caruana, M.A.; Suda D. Long- and Medium-Term Financial Strategies on Equities Using Dynamic Bayesian Networks. *AppliedMath* **2024**, *4*, 843–855. <https://doi.org/10.3390/appliedmath4030045>

Academic Editor: Tommi Sottinen

Received: 28 April 2024

Revised: 21 June 2024

Accepted: 24 June 2024

Published: 3 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In this paper, Dynamic Bayesian Networks (DBNs) are used to study the problem of obtaining and testing a financial strategy whose return is higher than the buy and hold strategy for a given equity. A Bayesian Network (BN) is a graphical and compact representation of a joint probability density function (PDF) that makes use of conditional independence and can be used to model a system under one time instance. DBNs extend the BN to more than one time period, which is to say that DBNs are a temporally driven extension of BNs.

Some benefits of using DBNs over other models, such as basic time series models, are the following. Firstly, DBNs have the capacity to incorporate what are known as hidden (or latent) variables, which either govern, or are thought to govern, the observations of the observable random variable. These hidden variables are random variables whose true value cannot be measured directly, sometimes due to instrumental inadequacies, and other times simply because the existence of these variables is hypothesized. Examples of such hidden variables are the intelligence of an individual (which cannot be measured precisely using any instrument) or the state (bull or bear) of a financial market (whose existence itself is hypothetical). Secondly, since they are an offshoot of BNs, DBNs allow for compact representations of an otherwise possibly cumbersome joint distribution. Russell and Norvig [1] argue that due to conditional independence, representing a joint distribution as a DBN may eliminate any redundant terms, which, in turn, also makes parameter estimation simpler. Murphy [2] state that another benefit of DBNs is the ease with which variations to the model at hand can be introduced. DBNs can be used to alternatively represent both basic Hidden Markov Models (HMMs), and HMMs with variations, such as HMMs with a mixture-of-Gaussian outputs, Auto-regressive HMMs, Input–Output HMMs and Hierarchical HMMs. DBNs can also be used to represent Kalman Filter Models where, contrary to HMMs, the hidden state is continuous rather than discrete. Lastly, DBNs also have the capacity to answer questions about different types of reasoning or inference, such

as *diagnostic* inference (from effects to causes), *causal* inference (from causes to effects), *intercausal* inference (between causes of the same effect), and mixed inference (a mixture of any two of the above types of inference).

In this research paper, we use the price–earnings (PE) ratio in the models discussed. In the literature, various papers can be found which apply the said PE ratio to examine investment strategies. Most notably, one can mention Basu [3]. This ratio has also been used in a vast range of literature including that of Lleo [4] and Angelini [5]. Chang and Tian [6] also use BNs to model the qualitative and quantitative relationships between several variables that affect the dynamics of the S&P 500 stock index, with the aim of optimizing trading decisions, namely when to open a short position and when to invest in a long position. Damiano et al. [7] base their work on a previous study, that of Tayal [8], which employs HHMMs to analyze financial data. The results obtained by Tayal are reproduced in Damiano et al.’s paper, and further insight is given. Damiano et al. [7] conclude that probabilistic inference allows the identification of two distinct states in high-frequency data that are mainly marked by buying and selling pressure.

Historically, Paul Dagum is thought to have kick-started the development of DBNs in the early 1990s, namely, in the work by Dagum et al. [9] in 1992 and by Dagum et al. [10] in 1995. Ever since, DBNs have been used in various areas, namely, in finance and economics. However, other areas include (but are not limited to) speech recognition (Murphy [2], Zweig, Russell [11], and Nefian et al. [12]), biology (Yao et al. [13], Raval et al. [14], Murphy and Mian [15]), robotics (Prembida et al. [16], Patel et al. [17]), image processing (Delage et al. [18]), and fault detection (Cozar et al. [19]). Parameter estimation techniques of DBNs are discussed by Benhamou et al. [20], whereas Murphy [21] works on the theoretical foundations of DBNs.

The rest of this paper is structured as follows. Section 2 introduces the theoretical underpinnings of DBNs. In Section 3, DBNs are applied to twelve equities from both developed and developing markets, and the return on investment is presented. The reason for considering both developed and developing markets is to test model robustness under different economic conditions and growth patterns. Finally, in the concluding section, the key points and results obtained are highlighted.

2. Theoretical Framework

This model is based on the work by Wang [22] and exploits two behavioral finance phenomena—*behavioral volatility* and *mean reversion*. Behavioral volatility refers to the phenomenon whereby market-related events such as irrational trades executed by inexperienced traders cause the trading price of a stock to deviate away from its ‘true’ value. Events like these continually affect market price either until the effects caused cancel each other out, or until rational investors balance these effects out with their rational trades. This phenomenon is known as *mean reversion*. The theory of mean reversion is applicable not only to the price of a stock, but to any price-related metric, such as the price–earnings (PE) ratio (share price divided by the earnings per share). As a result, one can decide to buy (or sell) a stock only when some metric that follows the mean reversion theory is far away from the mean, knowing that the ‘true’ value of the metric will be returned to at some point in the future.

A *Directed Graphical Model* (DGM) is a representation of a probabilistic model that uses conditional independence assumptions through a directed graph, where the vertices (note that the terms “state”, “random variable”, “node” and “vertex” may be used interchangeably) of the graph represent random variables and the arcs represent conditional dependence between pairs of random variables. An arc from vertex *A* to vertex *B* represents the statement “*A* causes *B*” (Murphy [2]). DGMs are convenient due to their compactness in exhibiting some joint probability distribution—for *N* binary random variables, the general closed form of the joint probability distribution of these random variables may need $O(2^N)$ parameters, whereas the graphical model may give the same information using fewer parameters due to the omission of terms via conditional independence statements.

A *Bayesian Network* is a DGM that represents a set of random variables X_1, \dots, X_n and their conditional dependencies through a directed *acyclic* graph (DAG), denoted by $G = (\mathcal{V}, \mathcal{E})$. Each vertex in \mathcal{V} represents a random variable X_i , and each directed edge in \mathcal{E} represents the conditional dependence a random variable has on another. A conditional probability distribution is associated to each node X_i , and the joint probability distribution on the vertex set \mathcal{V} is given by $\mathbb{P}[\mathcal{V}] = \prod_{i=1}^n \mathbb{P}[X_i \mid \text{parents}(X_i)]$. The generalization of a BN to multiple time slices gives rise to the definition of a DBN. A *Dynamic Bayesian Network* is defined to be a pair (B_0, B_{\rightarrow}) , where B_0 is the BN representing the *prior probability* (or the initial state probability), that is, the probability distribution of the random variables at time 0, and B_{\rightarrow} is a two-slice temporal BN which describes *transition probabilities* from time $t - 1$ to time t for any node X in the vertex set \mathcal{V} of the DAG $G = (\mathcal{V}, \mathcal{E})$, denoted by $\mathbb{P}[X_t \mid X_{t-1}]$. The joint probability distribution of the vertex set \mathcal{V} over all time slices is given by:

$$\mathbb{P}[\mathcal{V}] = \mathbb{P}[\mathcal{V}_0] \mathbb{P}[\mathcal{V}_{1:T}] = \prod_{X_i \in \mathcal{V}} \mathbb{P}[X_{i_0} \mid \text{parents}(X_{i_0})] \prod_{t=1}^T \prod_{X_i \in \mathcal{V}} \mathbb{P}[X_{i_t} \mid \text{parents}(X_{i_t})],$$

where $\mathcal{V}_{t_1:t_2}$ refers to the set of all vertices indexed from time t_1 up to time t_2 . A DBN can be parametrized by its transition matrix $\mathbb{A}_t = \mathbb{P}[X_t = j \mid X_{t-1} = i]$, and its prior distribution $\pi(i) = \mathbb{P}[X_1 = i]$. If the transition probabilities are assumed to be constant for all time slices ($\mathbb{A}_t = \mathbb{A}$), then they are said to be *homogeneous* and have a much more compact joint distribution function. When representing a DBN pictorially, two or three time slices are typically shown—the initial time slice and the subsequent one or two—since its structure is assumed to replicate throughout time.

An important property of DAGs is that “nodes can be ordered such that parents come before children. This is called a *topological ordering*, and it can be constructed from any DAG.” (Murphy [23]). Given such an ordering, the *Ordered Markov property* (or *Local Markov property*) is defined to be the “assumption that a node only depends on its parents, not on all its predecessors in the ordering” (Murphy [23]). This assumption is in fact a generalization of the Markov property for Markov chains.

2.1. Inference for DBNs

The objective in inference is to infer the value of the latent states given the observations of the observable states, that is, inferring the marginals $\mathbb{P}[X_t = i \mid y_{1:\tau}]$. If $\tau = t$, the process is known as *filtering* (or ‘*now-casting*’); if $\tau > t$, then this is *smoothing*; and if $\tau < t$, then one would be performing *prediction* (or *forecasting*). A commonly used inference algorithm for DBNs is the *forward-backward algorithm*. In this algorithm, dynamic programming is implemented through two steps, known as *passes*, that run in a counter-directional manner—one runs forward in time, whilst the other runs backward. Note that it is assumed that the transition probability matrix, emission probability matrix, and prior probabilities are all known. In the *forward pass*, the value of $\alpha_t(i) := \mathbb{P}[X_t = i \mid y_{1:t}]$ is found in a recursive manner. In the *backward pass*, the value of $\beta_t(i) := \mathbb{P}[y_{t+1:T} \mid X_t = i]$ is also found recursively but moving in counter chronological order (from time T to time 2). After the forward and backward passes are complete, the value for $\gamma_t(i) := \mathbb{P}[X_t = i \mid y_{1:T}]$ can finally be obtained:

$$\gamma_t(i) = \mathbb{P}[X_t = i \mid y_{1:T}] = \frac{\mathbb{P}[y_{t+1:T} \mid X_t = i] \mathbb{P}[X_t = i \mid y_{1:t}]}{\mathbb{P}[y_{1:T}]}$$

where $\mathbb{P}[y_{1:T}] = \prod_{t=1}^T c_t$.

2.2. Learning DBNs

In this context, learning refers to the parameter estimation process. In parameter estimation, learning can be tackled either through a Maximum Likelihood (ML) approach or through the Maximum A Posteriori (MAP) approach. If using ML, then the data are used to obtain parameter estimates via the solution of the optimization problem:

$\theta_{ML}^* = \arg \max_{\theta} \left\{ \mathbb{P}[Y | \theta] \right\} = \arg \max_{\theta} \left\{ \log \mathbb{P}[Y | \theta] \right\}$, where θ is the set of parameters to be estimated. Typically, θ contains the transition matrix and parameters pertaining to the probability distribution used in the emission matrix. On the other hand, if using the MAP method, the optimization problem above is adjusted slightly to become $\theta_{MAP}^* = \arg \max_{\theta} \left\{ \log \mathbb{P}[Y | \theta] + \log \mathbb{P}[\theta] \right\}$, where $\mathbb{P}[\theta]$ is the parameter prior distribution. The approach to solving the optimization problems above is through an adaptation of the Expectation–Maximization (EM) algorithm known as the *Baum–Welch* algorithm, proven to give a local optimum to the optimization problems above (Baum et al. [24], Dempster et al. [25]). It uses the forward–backward algorithm as a subroutine. Hence, in cases where the model parameters are unknown, the Baum–Welch algorithm is first used to estimate (or learn) the model parameters. Then, the forward–backward algorithm is used to infer the posterior marginals.

2.3. Application of Theory

The main hypothesis of the model is that the stock price of a firm is not always equal to the firm’s ‘true’ intrinsic value. Utilizing the phenomena of behavioral volatility and mean reversion, temporary effects that cause stock metrics to deviate from their ‘true’ value are classified into two: short-term effects (length of a few days) and medium-term effects (length of several weeks).

It is hypothesized that the fundamental value of a company i at time t , denoted by $P_{i,t}^*$, is directly proportional to its annual earnings at time t , denoted by $E_{i,t}$, with the *fundamental PE ratio*, denoted by $PE_{i,t}^*$, acting as the constant of proportionality:

$$P_{i,t}^* = (PE_{i,t}^*)(E_{i,t}).$$

On the other hand, the *observed PE ratio* (openly available on the public domain), denoted by $PE_{i,t}$, is given by $PE_{i,t} = \frac{P_{i,t}}{E_{i,t}}$ where $P_{i,t}$ denotes the actual trading price of a company i at time t .

Note that from this point onward, the index i is dropped, as the analysis on stocks is performed univariately. The model equation that results from the above is given by:

$$\begin{aligned} P_t = P_t^*(1 + Z_t)(1 + \varepsilon_t) &\implies y_t = \ln \left[PE^*(1 + Z_t) \right] + \ln \left[1 + \varepsilon_t \right] \\ &\implies y_t = \ln \left[PE^*(1 + Z_t) \right] + \varepsilon_t \end{aligned} \tag{1}$$

where $y_t = \ln \left[\frac{P_t}{E_t} \right]$, $\{Z_t\}_{t \in \mathbb{N}}$ is a discrete-time Markov chain modeling the medium-term noise effects, and $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$ is a random variable modeling the short-term noise effects. Due to the above model equation, the model used is only applicable for firms that have positive earnings throughout the period under study. Furthermore, one must note that y_t is an observable quantity (since both P_t and E_t are); however, PE^* and Z_t are not. It is assumed that both Z_t and PE^* are discrete-valued; hence, $Z_t \in \{a_1, \dots, a_M\}$ and $PE^* \in \{b_1, \dots, b_N\}$ where M and N represent the number of possible latent states of Z_t and PE^* , respectively.

The conditional independence assumptions used in this model are presented next:

$$\forall t \geq 1, \forall i, j \geq 1,$$

$$Z_t \perp PE^* \mid \emptyset. \tag{2}$$

$$Z_i \perp Z_j \mid Z_k \quad \exists k \in \{i + 1, \dots, j - 1\} \tag{3}$$

$$y_i \perp Z_j \mid Z_k, PE^* \quad \exists k \in \{i, \dots, j - 1\} \text{ or } k \in \{j + 1, \dots, i\}. \tag{4}$$

$$y_i \perp y_j \mid Z_k, PE^* \quad \exists k \in \{i, \dots, j\}. \tag{5}$$

For $t \geq 2$ and $r, s \in \{1, \dots, M\}$, the matrix $\mathbb{W} := [w_{rs}]_{M \times M}$ is defined to be the **transition probability matrix**, where

$$w_{rs} := \mathbb{P}[Z_t = a_s | Z_{t-1} = a_r].$$

Note that $w_{rs} \in [0, 1]$ and $\sum_{s=1}^M w_{rs} = 1$. For $t \geq 1, m \in \{1, \dots, M\}$ and $n \in \{1, \dots, N\}$, the matrix $\mathbb{D}_t := [d_{mn}(y_t)]_{M \times N}$ is defined to be the **emission probability matrix** at time t , where

$$d_{mn}(y_t) := \mathbb{P}[y_t | Z_t = a_m, PE^* = b_n].$$

where $d_{mn}(y_t) \sim \mathcal{N}(\ln[b_n(1 + a_m)], \sigma^2)$. For $m \in \{1, \dots, M\}$ and $n \in \{1, \dots, N\}$, the vectors $\mathbf{u} := (u_m)_{m \in \mathbb{N}}$ and $\mathbf{v} := (v_n)_{n \in \mathbb{N}}$ are defined to be the **initial probability vectors** that contain the initial probability distributions, where

$$u_m := \mathbb{P}[Z_1 = a_m]; \quad v_n := \mathbb{P}[PE^* = b_n].$$

Note that $u_m, v_n \in [0, 1], \sum_{m=1}^M u_m = 1$ and $\sum_{n=1}^N v_n = 1$. These prior distributions serve to incorporate any expert knowledge that the researcher may have available. A graphical representation of the model described above can be found in Figure 1.

Having laid out the principles needed for inference, the aim of the analysis now makes itself clearer—that of inferring the value of PE^* so as to estimate the fundamental price of a stock and formulate a trading strategy based on this knowledge. Furthermore, inferring the value of Z_t is also useful, as it can be used to test an alternative trading strategy. In this context, the data used will be split into a training and a test set. The set of model parameters is given by $\theta = \{\mathbb{W}, \mathbf{u}, \mathbf{v}, \sigma^2\}$. The space of possible parameters is given by the set:

$$\Theta = \left\{ \theta : u_m, v_n, w_{rs} \in [0, 1], \sum_{m=1}^M u_m = 1, \sum_{n=1}^N v_n = 1, \sum_{s=1}^M w_{rs} = 1, \sigma^2 > 0 \right\}. \quad (6)$$

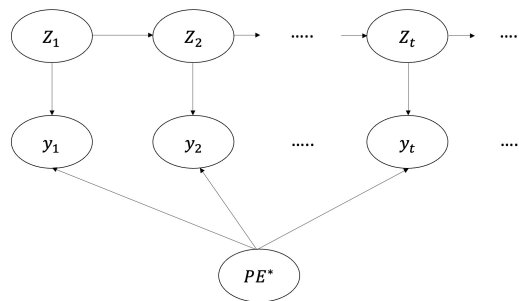


Figure 1. Representation of the model described above using a DBN, where y_t is observable and PE^* and Z_t are latent, discrete variables.

Learning and inference then follow. Since the parameters in θ are unknown, they will first need to be estimated (learning procedure). As mentioned, the algorithm used to obtain the MAP estimates is the Baum–Welch algorithm. After the unknown model parameters are estimated, the forward–backward algorithm is used on the parameter estimates to infer the constant value of PE^* for the training set and test set, and to infer the value of Z_t through smoothing for the training set, and through filtering for the test set:

- Filtering probabilities: $\mathbb{P}[Z_T, PE^* | y_{1:T}, \theta]$;
- Smoothing probabilities: $\mathbb{P}[Z_t, PE^* | y_{1:T}, \theta]$, where $t \in \{1, \dots, T - 1\}$.

2.4. Inference with Known Parameters

As per the forward pass of the forward–backward algorithm, the following definitions are made and equations derived for the filtering probabilities:

$$\forall t \in \{1, \dots, T\}, m \in \{1, \dots, M\}, n \in \{1, \dots, N\},$$

$$\begin{aligned} \alpha_{tmn} &:= \mathbb{P}[Z_t = a_m, PE^* = b_n \mid y_{1:t}, \theta]; \\ \alpha_{1mn} &= \frac{d_{mn}(y_1)u_m v_n}{\sum_{m'=1}^M \sum_{n'=1}^N d_{m'n'}(y_1)u_{m'} v_{n'}}; \\ \alpha_{tmn} &= \frac{d_{mn}(y_t) \sum_{i=1}^M \mathbb{P}[Z_{t-1} = a_i, PE^* = b_n \mid y_{1:t-1}] w_{mi}}{c_t}; \\ c_t &:= \sum_{m=1}^M \sum_{n=1}^N d_{mn}(y_t) \sum_{i=1}^M \mathbb{P}[Z_{t-1} = a_i, PE^* = b_n \mid y_{1:t-1}] w_{mi}. \end{aligned}$$

After defining the filtering probabilities, the smoothing probabilities are given through the estimate denoted by γ_{tmn} :

$$\begin{aligned} \forall t \in \{1, \dots, T-1\}, m \in \{1, \dots, M\}, n \in \{1, \dots, N\}, \\ \gamma_{tmn} &:= \mathbb{P}[Z_t = a_m, PE^* = b_n \mid y_{1:T}]. \end{aligned}$$

Next, as per the backward pass of the forward-backward algorithm, the definition of β_{tmn} is to be given so as to be able to obtain the values for γ_{tmn} :

$$\beta_{tmn} := \frac{\mathbb{P}[y_{t+1:T} \mid Z_t = a_m, PE^* = b_n]}{\mathbb{P}[y_{t+1:T} \mid y_{1:t}]} = \frac{\mathbb{P}[y_{t+1:T} \mid Z_t = a_m, PE^* = b_n]}{\prod_{t'=t+1}^T c_{t'}}$$

Therefore, as per the forward-backward algorithm, one has:

$$\gamma_{tmn} = \alpha_{tmn} \beta_{tmn}.$$

What remains to be derived are the expressions for $\beta_{tmn} \forall t \in \{1, \dots, T-1\}$:

$$\beta_{(T-1)mn} = \frac{\beta'_{(T-1)mn}}{c_T} = \frac{\sum_{i=1}^M d_{in}(y_T) w_{im}}{\sum_{m=1}^M \sum_{n=1}^N c_T}.$$

For $\beta_{tmn} \forall t \in \{1, \dots, T-2\}, m \in \{1, \dots, M\}, n \in \{1, \dots, N\}$:

$$\beta'_{tmn} = \sum_{i=1}^M \beta'_{(t+1)in} d_{in}(y_{t+1}) w_{im}; \quad \beta_{tmn} = \frac{\sum_{i=1}^M \beta_{(t+1)in} d_{in}(y_{t+1}) w_{im}}{c_{t+1}}.$$

With expressions found for both the smoothing and filtering probabilities, the most probable values of the latent variables PE^* and Z_t are found through marginalization:

$$\widehat{PE^*} = \arg \max_{b_n} \left\{ \sum_{m=1}^M \mathbb{P}[Z_t = a_m, PE^* = b_n \mid y_{1:T}] \right\} = \arg \max_{b_n} \left\{ \sum_{m=1}^M \gamma_{tmn} \right\}$$

To find the estimate \widehat{Z}_t for the latent state Z_t , smoothing is used on the training set whilst filtering is applied on the test set:

$$\begin{aligned} \widehat{Z}_t &= \arg \max_{a_m} \left\{ \sum_{n=1}^N \gamma_{tmn} \right\} \quad \forall t \in \{t : y_t \in \mathbb{X}_{train}\}; \\ \widehat{Z}_t &= \arg \max_{a_m} \left\{ \sum_{n=1}^N \alpha_{tmn} \right\} \quad \forall t \in \{t : y_t \in \mathbb{X}_{test}\}. \end{aligned}$$

where \mathbb{X}_{train} and \mathbb{X}_{test} denote the training set and test set, respectively.

2.5. Learning Unknown Parameters

The optimization problem in learning is given by $\widehat{\theta}_{MAP} = \arg \max_{\theta \in \Theta} \{ \mathbb{P}[\theta | y_1^T] \}$ and is solved using the Baum–Welch algorithm (note that in the forthcoming expressions, θ should technically be written with a hat superscript ($\widehat{\cdot}$) since it is a set of parameter estimates):

1. Set $j = 1$.
2. Set $\theta^{(1)} - \theta^{(0)} > \delta$
3. **while** $\theta^{(j)} - \theta^{(j-1)} > \delta$ **do**
 - (i) Calculate probabilities $\mathbb{P}[Z_t = a_m, PE^* = b_n | y_{1:T}, \theta^{(j)}] \forall t, m, n$
 - (ii) Solve the constrained maximization problem

$$\widehat{\theta}^{(j+1)} = \arg \max_{\theta \in \Theta} \{ Q(\theta; \theta^{(j)}) + \ln \mathbb{P}[\theta] \}$$

where $Q(\theta; \theta^{(j)}) = \mathbb{E}_{Z_{1:T}, PE^* | y_{1:T}, \theta^{(j)}} [\ln \mathbb{P}[y_{1:T}, Z_{1:T}, PE^* | \theta]]$.

- (iii) Increment j

An expression in closed form can be obtained for $Q(\theta; \theta^{(j)})$ by using the smoothing probabilities:

$$\begin{aligned} Q(\theta; \theta^{(j)}) &= \sum_{t=1}^{T-1} \sum_{m=1}^M \sum_{n=1}^N \alpha_{tmn}^{(j-1)} \beta_{tmn}^{(j-1)} \ln[d_{mn}(y_t)] + \sum_{m=1}^M \sum_{n=1}^N \alpha_{Tmn}^{(j-1)} \ln[d_{mn}(y_T)] \\ &+ \sum_{t'=2}^{T-1} \sum_{i,m=1}^M \sum_{n=1}^N \beta_{tmn}^{(j-1)} \left(\frac{d_{mn}^{(j-1)}(y_t)}{c_t^{(j-1)}} \right) w_{mi}^{(j-1)} \alpha_{(t-1)in}^{(j-1)} \ln[w_{mi}] \\ &+ \sum_{i,m=1}^M \sum_{n=1}^N \left(\frac{d_{mn}^{(j-1)}(y_T)}{c_T^{(j-1)}} \right) w_{mi}^{(j-1)} \alpha_{T-1,in}^{(j-1)} \ln[w_{mi}] \\ &+ \sum_{m=1}^M \sum_{n=1}^N \alpha_{1mn}^{(j-1)} \beta_{1mn}^{(j-1)} (\ln[u_m] + \ln[v_n]). \end{aligned}$$

In the argument of the maximization problem above, $\ln \mathbb{P}[\theta]$ is the logarithm of the prior distribution $\mathbb{P}[\theta]$, where $\mathbb{P}[\theta] = \mathbb{P}[\mathbf{u}]\mathbb{P}[\mathbf{v}]\mathbb{P}[\mathbb{W}]\mathbb{P}[\sigma^2]$ since the independence of priors is assumed. Within this prior distribution, two types of expert knowledge can be included—prior knowledge of the ballpark value of PE^* and prior knowledge of the persistence of the medium-term noise effects, which are encoded through the prior $\mathbb{P}[\mathbf{v}]$ and the prior $\mathbb{P}[\mathbb{W}]$, respectively.

The prior for the vector \mathbf{v} is represented by the Dirichlet distribution:

$$f(\mathbf{v}) = \frac{\Gamma(k_1 + k_2 + \dots + k_N)}{\Gamma(k_1)\Gamma(k_2) \dots \Gamma(k_N)} \prod_{n=1}^N v_n^{k_n-1}. \tag{7}$$

The values k_n for $n \in \{1, \dots, N\}$ intuitively correspond to the degree of belief an expert has on the event that b_n is the ‘true’ value for PE^* . For this analysis, $k_n = 1 \forall n$. The prior for the matrix \mathbb{W} is also derived from the Dirichlet distribution $f(\mathbb{W}) = \prod_{m=1}^M f(\mathbf{w}_m)$, where $\mathbf{w}_m = (w_{im})_{i=1, \dots, M}$ and

$$f(\mathbf{w}_m) = \frac{\Gamma(k_{1m} + k_{2m} + \dots + k_{Mm})}{\Gamma(k_{1m})\Gamma(k_{2m}) \dots \Gamma(k_{Mm})} \prod_{i=1}^M w_{im}^{k_{im}-1}.$$

Since \mathbb{W} is the transition matrix for the hidden Markov chain $\{Z_t\}_{t \in \mathbb{N}}$, then the diagonal entries of \mathbb{W} represent the probability that a particular state persists (stays as is in the next time point). The greater the value of Z_t (or, correspondingly, a_m), the greater that

state’s persistence. In this analysis, the off-diagonal entries (the reader is suggested to refer to Wang [22] for more information on the values of the off-diagonals) are set to 0.

Taking all the above into consideration, the logarithm of the prior $\mathbb{P}[\theta]$ becomes

$$\ln \mathbb{P}[\theta] = \ln \mathbb{P}[\mathbb{W}] + \ln \mathbb{P}[\mathbf{v}] = \sum_{m=1}^M \sum_{i=1}^M (k_{im} - 1) \ln[w_{im}] + \sum_{n=1}^N (k_n - 1) \ln[v_n] + s_1,$$

where $s_1 = \ln \left[\frac{\Gamma(k_1+k_2+\dots+k_N)}{\Gamma(k_1)\Gamma(k_2)\dots\Gamma(k_N)} \right] + \ln \left[\frac{\Gamma(k_{1m}+k_{2m}+\dots+k_{Mm})}{\Gamma(k_{1m})\Gamma(k_{2m})\dots\Gamma(k_{Mm})} \right]$.

The constant s_1 is only included for completeness’ sake—it is rendered irrelevant when maximizing in the Baum–Welch algorithm.

With the priors set, the constrained optimization problem is now fully defined. Note that only the equality constraints in (6) will be considered, as the inequality constraints will end up being satisfied still. Therefore, the method of Lagrange multipliers can be used to solve this optimization problem. The expressions for the estimators of the four variables in question are given below:

$$\widehat{u}_m^{(j)} = \sum_{n=1}^N \alpha_{1mn}^{(j-1)} \beta_{1mn}^{(j-1)}; \tag{8}$$

$$\widehat{v}_n^{(j)} = \frac{\sum_{m=1}^M q_{1mn}^{(j)}(Z_1, PE^*) + (k_n - 1)}{1 + \sum_{n'=1}^N (k_{n'} - 1)} = \frac{\sum_{m=1}^M \alpha_{1mn}^{(j-1)} \beta_{1mn}^{(j-1)} + (k_n - 1)}{1 + \sum_{n'=1}^N (k_{n'} - 1)}; \tag{9}$$

$$\begin{aligned} \widehat{w}_{mi}^{(j)} &= - \frac{\sum_{t'=2}^T \sum_{n=1}^N q_{t',m,i,n}^{(j)}(Z_{t'}, Z_{t'-1}, PE^*) + (k_{mi} - 1)}{\sum_{t'=2}^T \sum_{m=1}^M \sum_{n=1}^N q_{t',m,i,n}^{(j)}(Z_{t'}, Z_{t'-1}, PE^*) + \sum_{m'=1}^M (k_{m'i} - 1)} \\ &= \frac{\omega_{t',m,i,n}^{(j)}}{\sum_{m'=1}^M \omega_{t',m,i,n}^{(j)}}, \end{aligned} \tag{10}$$

where

$$\begin{aligned} \omega_{t',m,i,n}^{(j)} &= \sum_{t'=2}^{T-1} \sum_{n=1}^N \beta_{t'mn}^{(j-1)} \left(\frac{d_{mn}^{(j-1)}(y_{t'})}{c_{t'}^{(j-1)}} \right) w_{mi}^{(j-1)} \alpha_{t'-1,i,n}^{(j-1)} \\ &\quad + \sum_{n=1}^N \left(\frac{d_{mn}^{(j-1)}(y_T)}{c_T^{(j-1)}} \right) w_{mi}^{(j-1)} \alpha_{(T-1)in}^{(j-1)} + (k_{mi} - 1) \\ \widehat{\sigma}^{2(j-1)} &= \frac{\sum_{t=1}^{T-1} \sum_{m=1}^M \sum_{n=1}^N \alpha_{tmn}^{(j-1)} \beta_{tmn}^{(j-1)} (y_t - \ln[b_n(1 + a_m)])^2}{\sum_{t=1}^{T-1} \sum_{m=1}^M \sum_{n=1}^N \alpha_{tmn}^{(j-1)} \beta_{tmn}^{(j-1)} + \sum_{m=1}^M \sum_{n=1}^N \alpha_{Tmn}^{(j-1)}} \\ &\quad + \frac{\sum_{m=1}^M \sum_{n=1}^N \alpha_{Tmn}^{(j-1)} (y_T - \ln[b_n(1 + a_m)])^2}{\sum_{t=1}^{T-1} \sum_{m=1}^M \sum_{n=1}^N \alpha_{tmn}^{(j-1)} \beta_{tmn}^{(j-1)} + \sum_{m=1}^M \sum_{n=1}^N \alpha_{Tmn}^{(j-1)}}. \end{aligned} \tag{11}$$

3. Methodology of Analysis and Results

Twelve equities were chosen to be analyzed in this paper (see Table 1)—nine from a developed market (US) and three from emerging markets (Brazil and China). The training set contains data from the 1st of January 2011 to the 31st of December 2019 whilst the test set contains data from the 1st of January 2020 to the 30st of September 2020. Using a 12-month rolling Sharpe ratio as a point of reference, all equities displayed average or above average returns (with respect to an S&P benchmark) in the test period, with the exception of the two underperforming Brazilian equities [26]. Two datasets were collected for each stock for the above-mentioned period—daily price data and quarterly earnings per share data.

Table 1. Table of results, showing the return on investment as a percentage of the sum invested. Values in brackets are negative.

Stock	Thresholds for LT Strategy				Thresholds for MT Strategy				Profit from B&H
	0.05	0.10	0.15	0.20	0.03	0.05	0.07	0.10	
AAPL (US)	55.30%	55.30%	55.30%	55.30%	69.51%	57.82%	48.06%	76.02%	55.30%
ADBE (US)	25.46%	37.86%	44.45%	0%	46.65%	46.65%	46.65%	46.65%	46.65%
BLK (US)	12.98%	12.98%	12.98%	12.98%	33.92%	34.66%	20.85%	20.85%	12.98%
COST (US)	22.59%	22.59%	22.59%	22.59%	50.12%	42.85%	28.21%	0%	22.59%
HD (US)	28.71%	28.71%	28.71%	28.71%	57.76%	51.31%	37.99%	47.35%	28.71%
ITUB (BZ)	(39.86%)	(29.36%)	(29.36%)	(27.57%)	9.10%	7.07%	7.07%	(9.59%)	(55.85%)
MA (US)	27.17%	29.34%	61.09%	60.41%	47.70%	32.57%	24.66%	34.39%	11.93%
MCD (US)	11.38%	11.38%	11.38%	11.38%	18.94%	18.94%	30.63%	30.63%	11.38%
NTES (CHN)	47.89%	50.84%	57.35%	39.54%	39.54%	39.54%	39.54%	39.54%	39.54%
NVDA (US)	125.90%	125.90%	125.90%	125.90%	149.00%	156.79%	125.90%	125.90%	125.90%
SAN (BZ)	(2.08%)	(14.35%)	(17.78%)	(17.78%)	(57.17%)	(57.17%)	(57.17%)	(57.17%)	(57.17%)
UNH (US)	7.95%	7.95%	15.90%	19.88%	72.41%	21.83%	38.58%	2.87%	7.95%
Average	26.12%	25.07%	32.38%	27.61%	44.79%	37.74%	32.60%	29.79%	20.83%
W/D/L	4/7/1	4/7/1	5/6/1	4/7/1	9/3/0	9/3/0	7/4/1	6/4/2	
Totals			17/27/4				31/14/3		

After the preliminary data cleaning and preparation phase, the initial values of vectors $\mathbf{a} = (a_m)$ and $\mathbf{b} = (b_n)$ are set next. Priors of the vector \mathbf{v} and matrix \mathbb{W} are set as discussed earlier; the prior for \mathbf{u} is set to be the discrete uniform distribution, and the initial value of σ^2 is set to 5. Parameter learning is then performed before inference of the latent states \widehat{PE}^* and \widehat{Z}_t . Simulation of the trading strategies follows. The trading strategies proposed by Wang [22] and used in this paper will be compared to the benchmark B&H strategy. For both the proposed trading strategies, let I_t denote the amount of cash available at time t ; let N_t denote the units of a security held at time t ; let T_{train} represent the size of the training set; and let T represent the size of the dataset (sum of sizes of training and test sets). Both the long-term and the medium-term trading strategies can be described by three possible courses of action (labeled (i), (ii) and (iii) below) at time t ; that is, courses of action (i) through (ii) are common to both trading strategies:

(i) If $PE_t \leq A_t(1 - Tr)$ and $I_t > 0$, then buy the security using all the available cash I_t . Hence, $N_{t+1} = \frac{I_t}{P_t}$ and $I_{t+1} = 0$.

(ii) If $PE_t \geq A_t(1 + Tr)$ and $I_t = 0$, then sell all the units held N_t . Hence, $N_{t+1} = 0$ and $I_{t+1} = P_t N_t$.

(iii) If neither (i) or (ii) are satisfied, do not execute any trades. Hence, $N_{t+1} = N_t$ and $I_{t+1} = I_t$.

Note that A_t is considered to be a baseline and depends on the trading strategy. The threshold value $Tr \in (0, 1)$ acts as a sensitivity gauge defining how much the investor wants to allow $PE_t = \frac{P_t}{E_t}$ to deviate from the baseline A_t before triggering a particular course of action in the trading strategies. This is clear from how the courses of action are defined. The total profit at the end of the trading period is given by $I_T + P_T N_T - I_{T_{train}+1}$.

The first trading strategy is the so-called ‘long-term strategy’, where trading is performed with respect to the constant value of \widehat{PE}^* , so $A_t = \widehat{PE}^*$. The alternative strategy is the ‘medium-term strategy’, where trading is performed with respect to the dynamic values of $\widehat{PE}^*(1 + \widehat{Z}_t)$, where each \widehat{Z}_t is dynamically estimated through filtering. Therefore, for the medium-term strategy, $A_t = \widehat{PE}^*(1 + \widehat{Z}_t)$.

The results on the BLK and ITUB stocks are presented in graphical detail in this paper. Firstly, the long-term strategy on the BLK stock data suggests that the investor buys the stock at time point $2265 = T_{train} + 1$ and holds the stock for the rest of the period. Clearly, this strategy coincides with the B&H strategy and, as a result, profit from the long-term-strategy would be equal to profit from the B&H strategy which is equal to USD1298.04, equaling a 12.98% return on the initial investment of USD 10,000. On the other hand, the

medium-term strategy suggests that the investor buys the stock at time points 2265, 2301 and 2437, sells it at time points 2289 and 2431, and holds it for the rest of the time points. This would yield a profit of USD 3391.85; equalling a 33.92% return on the initial investment of USD 10,000 in the nine-month period that the testing set covers. This means that the medium-term strategy beats the B&H strategy by 20.94%. Figure 2 illustrates the long-term strategy and medium-term strategy.

Next, we discuss the ITUB stock data. The long-term strategy suggests that the investor buys the stock at time point 2308 and holds the stock for the rest of the period. Implementing this strategy would yield a loss of USD 3986.16, which equates to a 39.86% loss on the initial investment of USD 10,000. The buy-and-hold strategy, however, would yield a greater loss of USD 5584.67. In contrast, the medium-term strategy suggests that the investor buys the stock at time points 2313, 2315 and 2376, sells it at time points 2314 and 2372, and holds it for the rest of the time points. This would actually yield a profit of USD 909.81. Although this is only a 9.1% return on the initial investment of USD 10,000, the medium-term strategy provides the investor with a strategy whereby he or she can make a profit in a period when the stock is actually crashing. Figure 3 illustrates the long-term strategy and medium-term strategy, respectively.

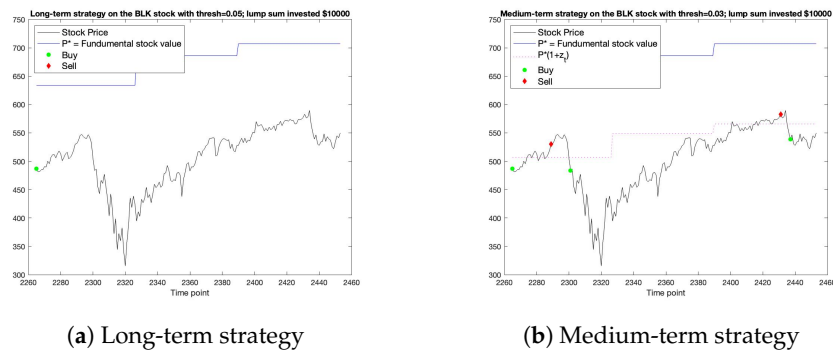


Figure 2. Simulated strategies for the BLK stock.

More generally, we see in Table 1 that the medium strategy has been consistently superior (for various thresholds) for BLK, COST, HD, ITUB, MA, MCD, NVDA and UNH. For ITUB, for all but one threshold, the strategy turns a slight profit even though a loss is registered for other strategies. The long-term strategy has been consistently superior for NTES and SAN. Neither strategy has offered any advantages on ADBE, while for AAPL, the success of the medium-term strategy depends on the choice of threshold. Some further analysis on these results will be given in the conclusion.

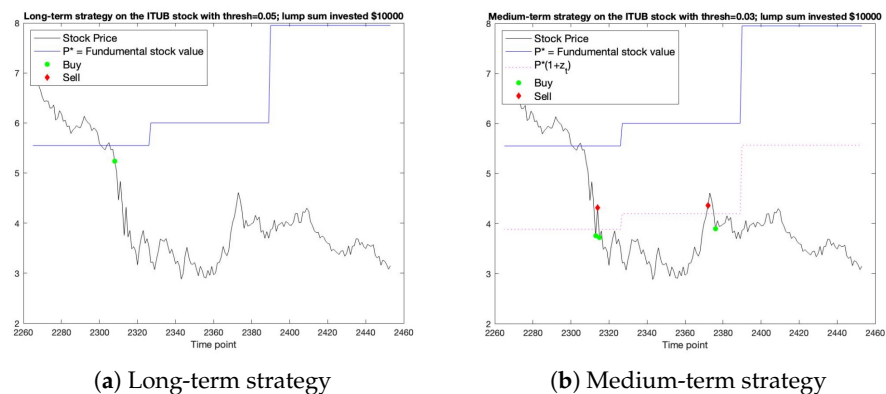


Figure 3. Simulated strategies for the IUTB stock.

4. Conclusions

Through the use of DBNs, the model for stock movement by Wang [22] is built for our existing equity dataset. This model includes two latent states—one modeling the medium-term noise effects, and the other modeling the true fundamental PE ratio of a firm, the latter assumed to be constant throughout the period under study. The forward–backward algorithm and the Baum–Welch algorithm (variant of the EM algorithm) are used to perform parameter learning and inference. Based on this fitted model, the long- and medium-term strategies are applied to the twelve stocks studied here, with nine of these stocks trading on a developed market and the rest on an emerging one. Overall, both the long-term and medium-term strategies outperform the benchmark B&H trading strategy 17 and 31 times, respectively, out of a total of 48 experiment runs for each strategy (four for each of the twelve stocks). The strategies proposed by Wang only lose out to the B&H four and three times, respectively. Furthermore, the outperformance of these trading strategies is substantial. Whereas the average profit over all stocks when using the B&H is 20.83%, the average profit over all stocks and over all thresholds for the long-term strategy is 27.78% and that for the medium-term strategy is 36.23%. Lastly, it results that these trading strategies provide the investor with trading suggestions that, in certain cases, can even turn a loss under the B&H strategy into a profit as was shown to be the case for the ITUB stock. This stock was on a downfall, but the medium-term strategy still yielded a profit on the sum invested. All these results are as displayed in Table 1.

As with any statistical model, the model implemented in this work has room for improvement. The first and most significant limitation is the fact that only stocks that had a positive EPS during the period under study can be modeled by the model. This is due to the left-hand side of the model equation, that is, Equation (1). In certain times such as during pandemics or during recessions, it may be considerably difficult to find a firm who has not registered a loss in at least one quarter for the period under study. A second limitation is the lack of use of expert knowledge. In real-world investing, expertise in the field is considered highly valuable, and only seasoned investors are generally advised to trade actively. Since the model in this paper allows for the incorporation of expert knowledge, it is indeed a limitation that such knowledge is not made use of. Another limitation related to expert knowledge is the limitation that no trading fees or commissions are taken into account in this analysis. Although the proposed model allows for commissions to be considered in the trading strategies, no knowledge on the actual values of fees or commissions charged was available at the time of writing, and hence, such expenses could not be taken into account. It is well known that certain trading fees can sometimes tally up when executing numerous trades in such a way that a profit can turn into a loss when these fees are brought to the fore, which would make the buy-and-hold strategy more profitable.

Improvements on the above-mentioned limitations can add value to the model and should be considered in future works. In addition, another improvement that makes the model here more applicable in real-life scenarios is the incorporation of some variable that measures the liquidity of a stock. This is because the suggestions provided by the trading strategies on when to buy and sell a stock are rendered useless if the stock itself is not liquid. Prices may change by the time the stock becomes liquid enough for an actual trading opportunity to arise, and if the price changes significantly, that obviously renders the suggestion itself useless. In this study, liquidity was not envisaged to be of concern since all 12 stocks considered are large-cap stocks.

The choice of historical data used is always an important decision due to the fact that the length of the time series plays a major role. In principle, longer time series are preferred to shorter ones, but if the historical data contain changes in regime, this may inhibit the model in its forecasting performance. Furthermore, future regime changes that may occur in the test period may also impact the model's forecasting performance.

Finally, other improvements that can be implemented in future studies are further and deeper experimentation and testing, and the possibility of short-selling stock. To begin with, experimentation at a portfolio level can be implemented with the stocks studied here

to understand how the profits change when a group of stocks are considered together—for instance, samples of nine stocks from the twelve studied here can be taken to form portfolios, and the ‘optimal’ portfolio can be identified. Apart from this, more time can be spent on cross-validation in future studies to further improve on the suggestions provided by the trading strategies in this work. Also, the properties of all the estimators used could be derived in future works, to better understand their behavior. Finally, this strategy does not allow the short-selling of stock; extensions to the model which allow for this could be proposed.

Author Contributions: All the three authors worked equally hard to write this paper. K.L. took care of the coding in MATLAB and the results section, M.A.C. and D.P.S. took care of the methodology and theoretical content of the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All the data is openly available from Yahoo Finance at <https://finance.yahoo.com/>.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BN	Bayesian Network
DAG	Directed Acyclic Graph
DBN	Dynamic Bayesian Network
DGM	Directed Graphical Model
EM	Expectation–Maximization
HNN	Hidden Markov Model
PDF	Probability Density Function
MAP	Maximum A Posteriori
ML	Maximum Likelihood

References

1. Russell, S.; Norvig, P. *Artificial Intelligence: A Modern Approach*, 2nd ed.; Prentice Hall: Upper Saddle River, NJ, USA, 1995.
2. Murphy, K. *Dynamic Bayesian Networks: Representation, Inference and Learning*. Ph.D. Dissertation, University of California, Berkeley, CA, USA, 2002.
3. Basu, S. Investment Performance of common stocks in relation to their price-earnings ratios: A Test of the efficient market hypothesis. *J. Financ.* **1977**, *23*, 3. 663–673.
4. Lleo, S.; Ziemba, W.T. A Tale of Two Indexes: Predicting Equity Market Downturns in China. 2020. [CrossRef]
5. Angelini, N.; Bormetti, G.; Marmi, S.; Nardini, F. Value Matters: Predictability of Stock Index Returns. *SSRN* **2012**. [CrossRef]
6. Chang, K.C.; Tian, Z. Market analysis and trading strategies with Bayesian networks. In Proceedings of the 18th International Conference on Information Fusion, FUSION 2015, Washington, DC, USA, 6–9 July 2015; pp. 1922–1929.
7. Damiano, L.; Peterson, B.; Weylandt, M. Regime Switching and Technical Trading with Dynamic Bayesian Networks in High-Frequency Stock Markets, 2017.
8. Tayal, A. *Regime Switching and Technical Trading with Dynamic Bayesian Networks in High-Frequency Stock Markets*; University of Waterloo: Waterloo, ON, Canada, 2009.
9. Dagum, P.; Galper, A.; Horvitz, E. Dynamic Network Models for Forecasting. In Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence, Stanford, CA, USA, 17–19 July 1992; pp. 41–48.
10. Dagum, P.; Galper, A.; Horvitz, E. Uncertain Reasoning and Forecasting. *Int. J. Forecast.* **1995**, *11*, 73–87. [CrossRef]
11. Zweig, Z.; Russell, S. Speech Recognition with Dynamic Bayesian Networks. 1998. Available online: <http://www.cse.msu.edu/~cse841/papers/zweig.pdf> (accessed on 28 April 2024).
12. Nefian, A.; Liang, L.; Pi, X.; Liu, X.; Murphy, K. Dynamic Bayesian Networks for Audio-Visual Speech Recognition. *EURASIP J. Adv. Signal Process.* **2002**, *11*, 783042. [CrossRef]
13. Yao, X.Q.; Zhu, H.; She, Z. A dynamic Bayesian network approach to protein secondary structure prediction. *BMC Bioinform.* **2008**, *9*, 49. [CrossRef] [PubMed]
14. Raval, A.; Ghahramani, Z.; Wild, D.L. A Bayesian network model for protein fold and remote homologue recognition. *Bioinformatics* **2002**, *18*, 788–801. [CrossRef] [PubMed]

15. Murphy, K.; Mian, S. Modelling Gene Expression Data Using Dynamic Bayesian Networks. 1996. Available online: <https://www.cs.ubc.ca/~murphyk/Papers/ismb99.pdf> (accessed on 28 April 2024).
16. Prembida, C.; Souza, F.A.A.; Faria, D.R. Dynamic Bayesian network for semantic place classification in mobile robotics. *Auton. Robot.* **2016**, *5*, 1161–1172. [[CrossRef](#)]
17. Patel, M.; Miro, J.V.; Dissanayake, G. Dynamic Bayesian Networks for Learning Interactions Between Assistive Robotic Walker and Human Users. 2010. Available online: <https://opus.lib.uts.edu.au/bitstream/10453/16294/1/2010000382.pdf> (accessed on 28 April 2024).
18. Delage, E.; Lee, H.A.Y. A Dynamic Bayesian Network Model for Autonomous 3D Reconstruction from a Single Indoor Image. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006; Volume 2, pp. 2418–2428. [[CrossRef](#)]
19. Cozar, J.; Puerta J.; Gamez, J.A. An Application of Dynamic Bayesian Networks to Condition Monitoring and Fault Prediction in a Sensored System: A Case Study. *Int. J. Comput. Intell. Syst.* **2017**, *10*, 176. [[CrossRef](#)]
20. Benhamou, E.; Atif, J.; Laraki, R. A New Approach to Learning in Dynamic Bayesian Networks (DBNs). *SSRN Electron. J.* **2018**. [[CrossRef](#)]
21. Murphy, K. An introduction to Graphical Models. 2001. Available online: <https://www.cs.ubc.ca/~murphyk/Bayes/bnintro.html> (accessed on 28 April 2024).
22. Wang, H. Bayesian Probabilistic Inference on Firm-Level Stock Price Dynamics. Ph.D. Dissertation, Suranaree University of Technology, Nakhon Ratchasima, Thailand, 2016.
23. Murphy, K. *Machine Learning: A Probabilistic Perspective*; MIT Press: Cambridge, MA, USA, 2012.
24. Baum, L.E.; Petries, T.; Soules, G.; Weiss, N. A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *Ann. Math. Stat.* **1970**, *41*, 1, 167–171. [[CrossRef](#)]
25. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum Likelihood from Incomplete Data Via the EM Algorithm. *J. R. Stat. Soc. Ser. B* **1977**, *39*, 1–22. [[CrossRef](#)]
26. PortfoliosLab. Available online: <https://portfolioslab.com/> (accessed on 28 April 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.