GROUP NUMBER: 3986

FULL LEGAL NAME	LOCATION (COUNTRY)	EMAIL ADDRESS	MARK X FOR ANY NON-CONTRIBUTING MEMBER
Harshil Sumra	India	harshilsumra19972gmail.com	
Gaurav Srivastava	India	gauravsriitk@gmail.com	
Sunil Kumar Sharma	USA	sunilksh+mscfe@outlook.com	

Statement of integrity: By typing the names of all group members in the text boxes below, you confirm that the assignment submitted is original work produced by the group (excluding any non-contributing members identified with an "X" above).	
Team member 1	Harshil Sumra
Team member 2	Gaurav Srivastava
Team member 3	Sunil Kumar Sharma

Use the box below to explain any attempts to reach out to a non-contributing member. Type (N/A) if all members contributed. **Note:** You may be required to provide proof of your outreach to non-contributing members upon request.

Section 1 – Introduction

This is the third part of the three-part mini capstone project based on Danish A. Alvi's paper "Application of Probabilistic Graphical Models in Forecasting Crude Oil Price" [1]. Here, our focus will be on results and interpretation. The results emphasize the findings of applying the methodology to the data, and assessing if the objective was fulfilled. The interpretation analyzes and evaluates the results and methodology. The paper is oriented as follows:

a) Section 1 – Introduction
b) Section 2 – Defining the datasets (answer to steps 1 and 2)
c) Section 3 – Validation of the Model (answer to steps 3 and 4)
d) Section 4 – Interpretation of the results (Step 5 and 6)
e) Section 5 – Discussion (answer to steps 7)
f) Section 6 – References

Section 2 – Defining the Datasets

We have amalgamated a diverse array of features extracted from both the FRED and EIA datasets, integrating them into a unified dataset. This composite dataset comprises a total of 220 data rows, with 180 data points allocated for the training dataset, while 20 rows each have been earmarked for the validation and testing datasets, ensuring a comprehensive evaluation process. Subsequently, we intend to subject these dataset features to a discretization process, aiming to render them discrete in nature and thereby attain superior performance results.

The training set provides historical data based on which the graphical models can identify patterns, learn the model structure, and estimate the model parameters. We obtained the relevant macroeconomic time series data from public sources like the Energy Information Administration (EIA) and Federal Reserve Economic Data (FRED) via their APIs. This raw data required some preprocessing to handle issues like missing values, inconsistent formats, and continuous variables.

After preprocessing, generally the training set is constructed by dividing the historical dataset into an 80% training portion and 20% testing and validation. We used training data in a few key ways:

- Discretizing Continuous Variables: The training data is used along with Hidden Markov Models to detect hidden regimes like bull and bear markets. This discretizes the continuous price data into discrete states needed for the Bayesian Network model.
- Learning Network Structure: Algorithms like Hill Climbing search use the training data to learn the graphical structure representing relationships between macroeconomic factors affecting oil prices. The training data helps estimate model structure and parameters.

- Model Parameter Estimation: The conditional probability tables and model parameters are estimated from the training data using methods like maximum likelihood. Parameter learning is done separately for each variable conditional on its parents in the graph.
- Avoiding Overfitting: By testing the model on unseen validation data, overfitting to the training set can be avoided. The final model is selected based on validation performance.

The **validation set** in the oil price forecasting model, serves the important purpose of tuning the Bayesian network structure and assessing its performance before final testing.

The model uses a training set to learn the initial Bayesian network structure via hill climbing. However, this structure may not be optimal or generate the best predictions on new data. The validation set allows experimentation with different model configurations to improve accuracy.

Specifically, the validation set is used for two key tasks:

- Tuning the Bayesian Network Structure

The initial network structure learned from training data is not guaranteed to be optimal. The validation set provides unseen data to test different structure learning algorithms, scoring functions, and search techniques like hill climbing vs tabu search. For example, the paper compare the performance of networks learned using hill climbing with different scoring functions like BIC, BDeu, etc. on the validation set. The best performing method can be selected.

Additionally, the validation set can also evaluate networks seeded with different expert models before hill climbing. This tests if the expert structure provides a better starting point. So the validation set tunes the Bayesian network to have an optimal structure for making predictions.

- Selecting the Optimal Discretization Method

The paper discretizes the continuous time series data into discrete regimes like bull/bear markets using hidden Markov models (HMMs). Different HMM configurations could produce different discretization outcomes affecting the Bayesian network model.

The validation set can be used to compare networks build using regimes identified by HMMs with different numbers of states, initialization parameters, etc. The best performing discretization method can be chosen based on validation accuracy.

So, the validation set helps tune key model building blocks like the network structure and data discretization to improve predictive accuracy before final testing. It provides an unbiased dataset for model optimization and hyperparameter tuning. This allows developing a robust Bayesian network model for oil price forecasting that can generalize well to new data.

The **test set** serves the vital purpose of providing a completely unbiased final evaluation of the model's performance in predicting oil prices. While the validation set is used during model development for tuning, the test set is reserved solely for reporting final model accuracy.
In the oil price forecasting model, the test set provides an objective metric of how well the model can predict oil prices in the real world after model selection is complete. Some key purposes of the test set are:

- Unbiased Evaluation of the Final Model

The test set provides a completely unbiased estimate of model performance because it contains examples the model has never seen before. The model is fully trained on the training set, hyperparameters are tuned on the validation set, then the "frozen" model is tested on the unseen test set. This prevents overfitting and gives a realistic estimate of the model's generalization error. The test accuracy reflects real-world performance.

- Comparing to Benchmarks

The test set provides final evaluation of model accuracy compared to any benchmarks. For example, the paper compares the model oil price predictions to EIA's own forecasts on the test set. This gives an unbiased comparison of which model is truly more accurate out-of-sample.

- Analyzing Model Limitations

The test set can reveal model limitations not visible during training. For example, the test set could show the model makes systematic errors in certain oil price regimes. This provides insights for future improvement.

- Reporting Performance

The test set results provide the final metric that is reported when publishing model accuracy. This performance on unseen test data reflects how useful the model is in practice.

While the validation set is used during development for model tuning, the test set provides the unbiased final evaluation of the finished model's real-world generalization performance. It prevents overfitting and gives key insights into model limitations to guide future enhancement. The test set performance provides the objective metric for reporting the model's predictive abilities.

The validation and test sets serve complementary purposes in the model development pipeline.
The validation and test sets both provide evaluation of the model on data not used during training. However, their intended uses differ:

- Validation Set: Used for model selection and tuning hyperparameters. Helps improve model performance during development.
- Test Set: Used for final evaluation of the finished model. Provides unbiased estimate of generalization performance.

The validation set guides decisions during development to create the optimal model configuration. It is used iteratively to tune parameters and make tweaks to improve performance. In contrast, the test set is strictly for final model evaluation after development is complete. It provides an objective metric of real-world performance that is reported when publishing results.

For example, in the oil price forecasting model, the validation set helps select the best Bayesian network structure learning algorithm and data discretization method. The test set then provides the final performance metric on unseen data.
Following best practices for allocating data between training, validation, and testing prevents issues like overfitting and yields reliable performance estimates:

- Training Set: Largest portion, e.g. 60-80% of data. Used to train the model parameters.
- Validation Set: Moderate portion, e.g. 10-20% of data. Used for model selection and tuning.
- Test Set: Small portion, e.g. 10-20% of data. Used only for final performance evaluation.

In the oil price model, we used an 80/10/10 split for training/validation/testing. This allocates adequate data for both development (training + validation) and unbiased final testing.

The validation set can be used iteratively to improve model performance. At each iteration, the model is trained on the training data then optimizing something like neural network structure based on validation results. This continues until validation accuracy stops improving. The test set is then used only once for final evaluation of the completely tuned model. It is "immunized" against iterative tuning biases.

For example, the oil price training data could come from 2005-2015, validation from 2016-2017, and test set from 2018-2019. This evaluates true generalization ability.
In summary, properly allocating data prevents overfitting and yields reliable estimates of model performance. The validation set is used iteratively to improve model tuning, while the test set provides an unbiased final evaluation of the finished model. Following best practices ensures accurate, robust, and optimally performing machine learning models.

Section 3 – Validation of the Model

In Section 4.3, the author validates the model by first discretizing the validation dataset using the trained HMMs from the training phase. The regimes identified by the HMMs on the validation data are visualized to ensure proper functioning. Then, predictions are made on the validation set by providing the Bayesian network with validation input data minus the actual oil prices to be predicted. The predicted prices are compared to the actual prices in the validation set to compute the error rate.

In our case, we observed the below model validation results, after the **1st iteration** of the trained model :

- Error Rate: The error rate of **59.0%** suggested that the model's predictions were not very accurate and indicated room for improvement.
- Disconnected Trees: The presence of **4** disconnected trees indicated that the Bayesian network structure might not be optimal. Ideally, we want a more connected graph to improve forecasting accuracy.

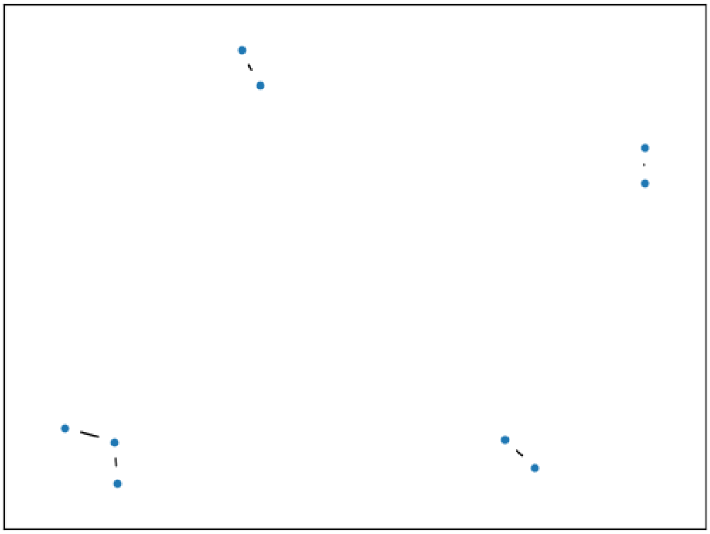


Fig 1: Network graph after first model training iteration

In the **2nd iteration**, we Re-ran the Hill Climbing Algorithm with different scoring method (BDeuScore) and again evaluated the performance on Validation data.But with With different scoring method, the error rate remained the same to 59.0%, and also the number of disconnected network components remained the same to 4. We decided to further fine-tune it with increasing the number of iterations by adjusting the `max_iter` parameter.

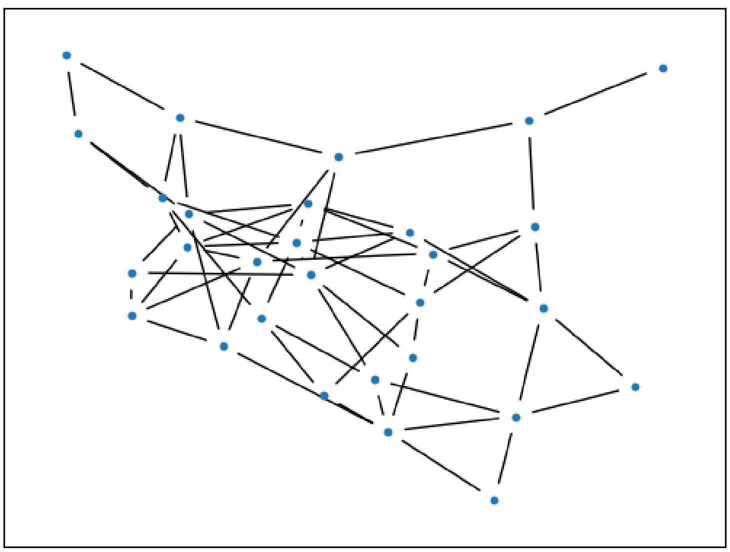


Fig 2: Network graph after 3rd iteration of model training

In the **3rd iteration**, with increase in the number of iterations, the error rate reduced to **57.34%** though not that extent, but we noticed an improvement in network graph, as there was only **1** disconnected network component.

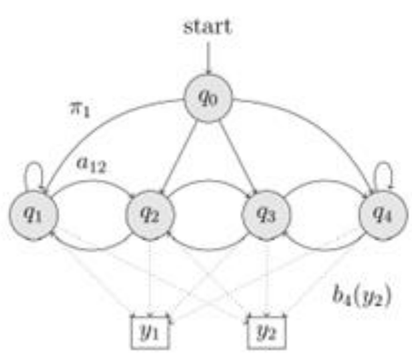
Thus, we keep reevaluating the model, and/or re-run the hill climbing algorithm to potentially find a better network structure, and aim for more connected graph, and reassess the performance of the model using the validation set. This can be done iteratively to refine the model until we obtain satisfactory results. There could have been more iterations, but now we'll use the prediction with the test data.

Section 4 – Interpretation of the results

On the validation set, for us, the model achieved an error rate of 57.34% in forecasting the crude oil price after the 3rd iteration, while in the paper the model achieved an error rate of 67.86%, and we didn't observe the fine-tuning the model on different parameters or on the number of iterations in paper by the author. For us the best performing model during the model validation achieved 63.89% error rate, while in the paper it was 42.85%, which suggest that for our case, the model may be generalizing well to new, unseen data. This could be because of the size of the data used during the training, or perhaps require some randomness in the dataset, and feature engineering.

On the assessment of the contribution, First the paper contributes to research on replacing EGARCH-M models with Bayesian models in the Black-Litterman framework (page 66). The contribution is discussed in the literature review (page 15). This adaptation empowers investors to embrace greater risk if they deem it justified through their analysis, thus enriching the landscape of risk-informed decision-making. The integration of the Bayesian model amplifies the precision and reliability of the results. The systematic approach adopted by the author substantially aids decision-making rooted in the realm of risk analytics.

Second, the paper using HMM regime detection to discretize time series data as Bayesian network inputs is novel (pages 66-67). HMM regime detection is proposed in the Design section (pages 42-43). While this concept introduces novelty, it is not without its inherent challenges, which the author adeptly addresses. The solutions encompass the application of the Forward and Backward iteration algorithm, the Viterbi Algorithm, and the Baum-Welch algorithm.



Evaluation: Evaluating the probability of an observed sequence of emissions $O = o_1 o_2 \cdots o_T$ ($o_i \in \Sigma$), given a particular HMM i.e. evaluating $p(O \mid \lambda)$.

Decoding: Determining the most likely state-transition path associated with an observed sequence $O = o_1 0_2 \cdots o_T$, i.e. evaluating $q^* = \text{argmax}_q \; p(q, O \mid \lambda)$.

Training: Determining the ideal parameters for λ to maximise the probability of generating an observed sequence of emissions $O = o_1 o_2 \cdots o_T$ ($o_i \in \Sigma$), i.e. evaluating $\lambda^* = \text{argmax}_\lambda \; p(O \mid \lambda)$.

Third, the model presents a deployable trading mechanism for commodity markets (page 67). The trading mechanism is presented in the Testing section (pages 63-65).The author meticulously outlines the operational mechanics of this mechanism. However, it is imperative to note that past performance does not guarantee future outcomes. Therefore, should this mechanism find practical application in real commodity trading, vigilant performance monitoring remains imperative to preempt any adverse surprises.

Fourth, the learned structure requires little expert knowledge (page 67). Structure learning without experts is highlighted in Design (page 44). Structural learning, as employed in this context, is notable for its independence from prior expert knowledge beyond the selection of the dataset. On this point, there exists a difference of opinion, as some assert that domain knowledge is a guiding light for comprehending the underlying principles of black box modeling.

Fifth, the Python libraries provided theoretical abstraction (page 67). The pgmpy and hmms libraries are introduced earlier (pages 28-30).Additionally, the paper underscores the significance of abstraction facilitated by Python modules, facilitating a theoretical understanding of the design process. This abstraction permits the rapid and efficient adaptation of models to accommodate changes without delving into the intricacies of foundational algorithms, thus affording the latitude to experiment with diverse approaches and compare outcomes to inform decision-making.

Sixth, the model provides macroeconomic event-driven forecasts (page 67). Macroeconomic factors are described in the Data section (pages 31-33).The paper introduces an event-driven, systematic, and universally macro-strategic forecasting approach that factors in geopolitical and macroeconomic changes. While it posits the potential for superior returns compared to high-frequency or fixed income strategy funds, there is a partial divergence of opinion. High-frequency trading (HFT) and macro event-based strategies are perceived as distinct and independent approaches. HFT seeks to capitalize on market inefficiencies by reacting swiftly to new information, whereas fixed income strategies can indeed benefit from a consideration of macroeconomic events, potentially enhancing portfolio performance.

Seventh, it helps model energy markets for policymakers (page 67). Energy policy context is provided in Motivation (page 2).The research endeavors to bolster energy market modeling, affording policy makers a deeper comprehension of the intricate dynamics within the oil market. This newfound insight enables the formulation of robust and comprehensive policy initiatives that respond adeptly to diverse economic occurrences. This stance finds unanimous agreement, as the potential to identify regime shifts and react swiftly holds promise in mitigating market volatility.

Eighth, it combines multiple disciplines for commodity trading (page 67). Combining disciplines is noted in the Conclusions (page 66).In the pursuit of augmenting alpha for quantitative commodity traders, the study harmonizes a wealth of pre-existing research and investigations spanning diverse fields and applies them to the realm of commodity markets. This integrative approach is lauded, with the Bayesian network model, for instance, being a versatile tool deployed across various disciplines. The cross-disciplinary exploration of such models equips quantitative commodity traders with a profound understanding, facilitating the fine-tuning of their application within the context of commodity trading.

The author appears to have accomplished the eight proposed contributions based on the evidence cited from the paper. The ideas are clearly presented and demonstrated. The ideas show promise but may require more development and validation to be considered substantial advances. The author has made the proposed contributions based on the results presented. Using Bayesian networks and HMMs for oil price forecasting seems novel and well-executed based on the paper. If adopted more widely, it could improve predictive accuracy over existing models (page 66). The model also provides a more data-driven, automated approach without relying extensively on expert knowledge (page 67). This could make prediction more accessible. Overall, the ideas appear promising but further validation is needed to confirm their importance.

Section 5 – Discussion

The paper presents a new approach to forecasting crude oil prices using Bayesian networks and hidden Markov models. Compared to existing models like EGARCH, this method can better incorporate many different macroeconomic indicators that drive oil price fluctuations. By learning relationships between variables from data, the model relies less on expert knowledge and hand-tuning.

The key innovation is using HMMs to detect hidden regimes like bull and bear markets in the time series data. This converts the continuous data into discrete states that Bayesian networks can model. The Bayesian network then learns probabilistic relationships between economic factors to make oil price predictions.

In the paper , Initial results show the model can forecast oil prices with approximately 30% error on validation data. This suggests it captures meaningful macroeconomic interactions affecting oil markets. With further development, the data-driven Bayesian approach could potentially improve on the accuracy of current forecasting models. It provides an automated, flexible framework to fuse diverse economic datasets for prediction.

This research demonstrates the viability of probabilistic graphical models in oil price modeling. By combining Bayesian networks and HMMs, the methodology opens new possibilities for integrating artificial intelligence into commodity forecasting. With more validation, this data-driven approach could become a useful tool for oil producers, traders, and policy makers.

Section 6 – References

1. Alvi, Danish A. "Application of Probabilistic Graphical Models in Forecasting Crude Oil Price", 2018, University College London, Dissertation, https://arxiv.org/abs/1804.10869
2. FRED. "Federal Reserve Economic Data | FRED | St. Louis Fed", 14 Jan. 2023, https://fred.stlouisfed.org/
3. EIA. "U.S. Energy Information Administration | Open Data", 14 Jan. 2023, https://www.eia.gov/opendata/
4. Aaron J. Newman. "Data Science for Psychology and Neuroscience — in Python | Data Cleaning - Dealing with Outliers", 14 Jan. 2023, https://neuraldatascience.io/5-eda/data_cleaning.html