***Abstract* —** This report presents a comparative analysis of three models, including Linear Regression, Gated Recurrent Unit (GRU), Autoregressive Integrated Moving Average (ARIMA), Long Short-Term Memory (LSTM) for predicting gold prices. The models are evaluated using MAE, MAPE, and RMSE metrics on historical gold price indicators data. The model with the lowest MAE, MAPE, and RMSE is recommended for gold price forecasting, contributing to improved understanding and accurate predictions in the gold market.

***Key words: gold price, forecasting, GRU, ARIMA, LSTM***

1. Introduction

A key factor influencing the prices of precious metals is the balance of supply and demand. Demand for these metals is affected by numerous variables, including economic conditions, geopolitical tensions, inflation, and shifts in consumer behavior. Historically, precious metals have been regarded as reliable hedges against inflation and economic instability. When inflation rises, investors often seek out these metals to safeguard their wealth, which increases demand and drives up prices. Conversely, during periods of low inflation, demand for precious metals tends to fall, leading to lower prices. A deep understanding of supply and demand dynamics is crucial for forecasting future market trends and making informed investment choices. This research focuses on evaluating the effectiveness of various statistical and machine learning models for predicting the future prices of gold, silver, and platinum.

Several studies have been conducted on predicting the future prices of precious metals using statistical models and machine learning algorithms. For example, the study by Tripathy, N. (2017) [1] used an autoregressive integrated moving average (ARIMA) model to predict the prices of gold. Manjula, K. A. et al. (2019) [2] used an ensemble model combining multiple machine learning algorithms to predict the prices of gold and silver. Another study by Boongasame, L. et al. **(**2022**)** [3] used long short-term memory and the association rule to forecast the gold price. These studies have shown promising results in predicting the future prices of precious metals using statistical models and machine learning algorithms.

The goal of this research is to compare the performance of various statistical and machine learning models in predicting the future prices of gold, silver, and platinum. We will use Gold, Platinum, and Silver daily datasets from December 2017 to June 2023 to make predictions using a variety of models such as linear regression, K-Nearest Neighbors, and neural networks. By doing so, we aim to provide insights into which models are most effective in predicting the future prices of these precious metals. This research can be beneficial for investors and traders who want to make informed decisions about buying or selling these metals based on predicted future prices.

1. RELATED WORK

We also found out Ho Thanh Tri’s paper [5] about using ARIMA (5,1,5) model to forecast gold price in Viet Nam and the forecast error in this case is about 3.46%.

The performance GRU and LSTM have also been investigated for gold price prediction, as detailed in a research paper by Yurtsever, M. (2010) [6] by predict the gold price using LSTM and GRU, with RMSE of 61.728 and 87.425 respectively. This indicates that the model is good enough to make predictions.

*A. ARIMA*

ARIMA is a statistical model that uses time series data to understand the dataset or predict future trends.

Autoregressive models assume that the future will resemble the past and predict future prices based on past performance, but it can be inaccurate during certain market conditions. ARIMA analyzes the strength of one dependent variable compared to other changing variables and aims to predict future market moves by examining differences between values in the series.[12]

*1) Integrated*

𝐷𝑖𝑓𝑓𝑒𝑟𝑒𝑛𝑡𝑎𝑡𝑖𝑜𝑛 1 : ∆𝑦𝑡 = 𝑦𝑡 − 𝑦𝑡−1 𝐷𝑖𝑓𝑓𝑒𝑟𝑒𝑛𝑡𝑎𝑡𝑖𝑜𝑛 2 : ∆𝑦𝑡 = (𝑦𝑡 − 𝑦𝑡−1) – (𝑦𝑡−1 − 𝑦𝑡−2)

Where,

𝑦𝑡 : is the value at time t

𝑦𝑡−1 : is the value of the 1 unit of time ago

𝑦𝑡−2 : is the value of the 2 units of time ago

*2) Auto regression*

𝑦𝑡 = 𝑎0 + 𝑎1𝑦𝑡−1 + 𝑎2𝑦𝑡−2 + ⋯ + 𝑎𝑝𝑦𝑡−𝑝 + 𝜀𝑡

Where,

𝑦𝑡 : is the value of a dependent variable at time t

𝑎0 : is a constant term

𝑦𝑡−1, 𝑦𝑡−2, 𝑦𝑡−𝑝 : the dependent variables in the past

𝑎1,𝑎2*,* 𝑎𝑝 : are the coefficients of the lagged values of the dependent variables respectively

𝜀𝑡 : is the error term at time t

*3) Moving average*

𝑦𝑡 = 𝛽0 + 𝛽1𝜀𝑡−1 + 𝛽2𝜀𝑡−2 + ⋯ + 𝛽𝑞𝜀𝑡−𝑞 + 𝜇𝑡

Where,

𝑦𝑡: is the value of a dependent variable at time t

𝛽0: is a constant term

𝜀𝑡−1, 𝜀𝑡−2, 𝜀𝑡−𝑞: the error terms in the past

𝛽1,𝛽2*,* 𝛽𝑞: are the coefficients of the lagged values of the error term respectively

𝜇𝑡: is the current error term or residual at time t

*4) ARIMA*

𝐴𝑅𝐼𝑀𝐴 (𝑝, 𝑑, 𝑞) = 𝐴𝑅(𝑝) + 𝐼(𝑑) + 𝑀𝐴(𝑞)[13]

Where,

𝑝, 𝑑, 𝑞 : are non-negative integers representing the order of autoregression, differencing, and moving average respectively.

𝐴𝑅(𝑝) : Autoregression of order p refers to the regression of the variable on its own lagged values up to p lags. The AR(p) component captures the linear dependence between the current value and its past values.

𝐼(𝑑) : Integration of order d refers to differing the time series d times until it becomes stationary. The I(d) component removes the trend from the time series.

𝑀𝐴(𝑞) : Moving average of order q refers to the regression of the variable on the past errors up to q lags. The MA(q) component captures the linear dependence between the current value and its past errors.

*B. LSTM*

LSTM (Long Short-Term Memory) is a deep learning, sequential neural network that overcomes the vanishing gradient problem faced by RNN. It allows information to persist and works similarly to an RNN cell.

A diagram of a computer program

Description automatically generated with medium confidence*Figure 2. LSTM Architecture*

An LSTM has a hidden state for short-term memory and a cell state for long-term memory. The cell state remembers important information over time while filtering out irrelevant data, allowing LSTMs to make accurate predictions.

A white rectangular object with a black border

Description automatically generated

*Figure 3. Long and Short Term Memory*

LSTM has three gates that control the flow of information in and out of the memory cell : Forget, Input, and Output.

First, the Forget gate chooses whether the information coming from the previous timestamp is to be remembered or is irrelevant and can be forgotten.

1

𝜎 = 1 + 𝑒−𝑥

𝑓𝑡 = 𝜎(𝑊ℎℎ(𝑓) ℎ𝑡−1 + 𝑊ℎ𝑥(𝑓) 𝑥𝑡)

It determines how much information from the previous cell state 𝐶𝑡−1 should be retained for the current time step. It does this by taking a weighted sum of the previous hidden state ℎ𝑡−1 and current input 𝑥𝑡, with weights given by matrices 𝑊ℎℎ(𝑓)

and 𝑊ℎ𝑥(𝑓), respectively.

The result is passed through a sigmoid activation function σ, which squashes the values between 0 and 1. A value of 0 means that all information from the previous cell state will be forgotten, while a value of 1 means that all information will be retained.

Next, the Input gate tries to learn new information from the input to this cell and determines how much to take from the intermediate cell state.

𝑖𝑡 = 𝜎(𝑊ℎℎ(𝑖) ℎ𝑡−1 + 𝑊ℎ𝑥(𝑖) 𝑥𝑡)

In contract, the Input gate identifies how much information from the current input 𝑥𝑡 and previous hidden state ℎ𝑡−1 should be used to update the cell state 𝐶𝑡, with weights given by matrices 𝑊ℎℎ(𝑖) and 𝑊ℎ𝑥(𝑖), respectively.

Again, we have applied the sigmoid function over it. As a result, the value of 𝑖𝑡 will be between 0 and 1.

𝐶̃𝑡 = 𝑡𝑎𝑛ℎ(𝑊ℎℎ(𝑐) ℎ𝑡−1 + 𝑊ℎ𝑥(𝑐) 𝑥𝑡)

LSTM may not take everything from the intermediate cell state, the 𝐶̃𝑡 is there to distinguish between different weight matrices. Due to the 𝑡𝑎𝑛ℎ function, the value of new information will be between -1 and 1. If the value of 𝐶̃𝑡 is negative, the information is subtracted from the cell state, and if the value is positive, the information is added to the cell state at the current timestamp. However, the 𝐶̃𝑡 won’t be added directly to the cell state.

Here comes the updated equation :

𝐶̃𝑡 = (𝑓𝑡 𝐶𝑡−1 + 𝑖𝑡 𝐶̃𝑡)

The Output gate passes the updated information from the current timestamp to the next timestamp. Together, they function as a layer of neurons with hidden layers and a current state.

𝑜𝑡 = 𝜎(𝑊ℎℎ(𝑜) ℎ𝑡−1 + 𝑊ℎ𝑥(𝑜) 𝑥𝑡)

Its value will also lie between 0 and 1 because of this sigmoid function. Now to calculate the current hidden state, we will use

𝑜𝑡 and tanh of the updated cell state. [21]

ℎ𝑡 = 𝑜𝑡 ∗ tanh(𝐶𝑡)

*H. GRU*

GRU (Gated Recurrent Unit) is a type of recurrent neural network that addresses the vanishing gradient problem. It is similar to LSTM and can produce comparable results.

To solve the vanishing gradient problem of a standard RNN, GRU uses, Update gate and Reset gate. Basically, these are two vectors which decide what information should be passed to the output. The special thing about them is that they can be trained to keep information from long ago, without washing it through time or removing information which is irrelevant to the prediction.

A computer screen shot of a diagram

Description automatically generated

*Figure 5. GRU Model*

The update gate helps the model to determine how much of the past information (from previous time steps) needs to be passed along to the future and avoid the vanishing gradient problem. The formula using for Update gate :

𝑧𝑡 = 𝜎(𝑊ℎℎ (𝑧) ℎ𝑡−1 + 𝑊ℎ𝑥(𝑧) 𝑥𝑡)

When 𝑥𝑡 is plugged into the network unit, it is multiplied by its own weight 𝑊ℎ𝑥(𝑧). The same goes for ℎ𝑡−1 which holds the information for the previous t - 1 units and is multiplied by its own weight 𝑊ℎℎ(𝑧). Both results are added together, and a sigmoid activation function is applied to squash the result between 0 and 1.

Reset gate is used from the model to decide how much of the past information to forget. To calculate it, we use :

𝑟𝑡 = 𝜎(𝑊ℎℎ(𝑟) ℎ𝑡−1 + 𝑊ℎ𝑥(𝑟) 𝑥𝑡)

This formula is the same as the one for the update gate. We plug in ℎ𝑡−1 — blue line and 𝑥𝑡 — purple line, multiply them with their corresponding weights, sum the results and apply the sigmoid function. The difference comes in the weights and the gate’s usage.

To determine what to remove from the previous time steps, we will calculate current memory content by using reset gate to store relevant information from the past.

ℎ′𝑡 = 𝑡𝑎𝑛ℎ(𝑟𝑡 ∗ 𝑊ℎℎ ℎ𝑡−1 + 𝑊ℎ𝑥 𝑥𝑡)

To process input 𝑥𝑡 and ℎ𝑡−1, we multiply them by weights 𝑊ℎ𝑥 and 𝑊ℎℎ respectively. The reset gate 𝑟𝑡 is used to determine what information to remove from previous time steps by calculating the Hadamard product with (𝑊ℎℎ ℎ𝑡−1). The results of steps 1 and 2 are summed up and passed through the tanh activation function.

A screenshot of a computer

Description automatically generated

*Figure 6. Current memory content*

Finally, the network needs to calculate ℎ𝑡 — vector which holds information for the current unit and passes it down to the network. In order to do that the update gate is needed. It determines what to collect from the current memory content — ℎ′𝑡 and what from the previous steps — ℎ𝑡−1. That is done as follows :

ℎ𝑡 = 𝑧𝑡 ∗ ℎ𝑡−1 + (1 − 𝑧𝑡) ∗ ℎ′𝑡

A computer screen shot of a diagram

Description automatically generated

*Figure 7. Final memory content*

The model can learn to set the vector 𝑧𝑡 close to 1 and keep most of the previous information by apply elementwise multiplication to the update gate 𝑧𝑡 and ℎ𝑡−1. Since 𝑧𝑡 will be close to 1 at this time step, (1 − 𝑧𝑡) will be close to 0 and ignore big portion of the current content, which is irrelevant for our prediction by apply element-wise multiplication to (1 − 𝑧𝑡) and ℎ′𝑡. Final, we sum the results.[23]

IV. METHOD

To perform prediction in machine learning, it is essential to ensure that the input data is properly formatted and preprocessed.

Step 1: Data Preprocessing

First, we must clean and preprocess the collected data by removing duplicates, handling missing values, and transforming the data into a format that can be used by machine learning algorithms. This may also involve feature engineering, where you create new features from existing ones to improve model performance.

Step 2: Data Splitting

Splitting the preprocessed data into training, testing and validate sets. The training set is used to train the machine learning model, while the testing set is used to evaluate the performance of the trained model. The validation set is used to evaluate the model prediction using the last set of the data to assets the model precision in the future.

Step 3: Model Training

Train the selected machine learning model using the training data. This involves feeding the data into the model, adjusting the model parameters, and evaluating the model performance.

Step 4: Model evaluation

Evaluate the performance of the trained model using testing and validate data. This can be done by calculating various metrics such as accuracy, precision, recall, and F1 score.

Step 5: Make predictions

Once the model has been trained and evaluated, it can be used to make predictions in the next 30 days.

1. IMPLEMENTATION

*A. Dataset*

*1) Description*

The dataset includes daily prices for gold, silver and platinum from January 1st, 2020, to November 9th, 2024. Each row represents a single day and contains the following information:

* Date: The date of the trading day.
* Price: The closing price of the metal on that day.
* Open: The opening price of the metal on that day.
* High: The highest price that the metal reached during the trading day.
* Low: The lowest price that the metal reached during the trading day.
* Vol.: The volume of the metal that was traded during the day.
* Change %: The percentage change in price from the

previous day's closing price.

The dataset was collected from investing.com [25] and has been preprocessed to remove any missing or erroneous data points.

1. *Descriptive Statistical*
   1. *Gold data*

*A table with numbers and a price

Description automatically generated*

*Figure 10. Descriptive statistics for gold data*

*b) Silver data*

*Figure 11. Descriptive statistics for silver data*

*c) Platinum data*

*A table with numbers and a price

Description automatically generated*

*Figure 12. Descriptive statistics for platinum data*

1. *Visualization*
   1. *Gold data*

*A graph showing the price of gold

Description automatically generated*

*Figure 13. Visualize gold data*

*b) Silver data*

*A graph showing the price of silver prices

Description automatically generated*

*Figure 14. Visualize silver data*

*c) Platinum data*

*Figure 15. Visualize platinum data*

*B. Technology*

In this paper, Google Collab will be used for data analysis. It’s a free cloud-based platform that offers powerful hardware, seamless integration with Google Drive, collaboration through shared notebooks, making it a user-friendly choice.

Library used: Pandas, mathplotlib, Tensorflow, keras, sklearn, statsmodels…

*C. Spliting data*

We will split the dataset into 3 ratios (Train: Test: Validate): 7:2:1, 6:3:1, 5:3:2. To build a better machine learning model that generalizes well on new data and avoids overfitting.

*D. Evaluation*

Evaluating a model is crucial to ensure that it is accurate, reliable, and generalizes well to new data. The following metrics will be used: MAE, MAPE and RMSE.

*1) MAE*

MAE (Mean Absolute Error) is a measure of the average size of the mistakes in a collection of predictions, without taking their direction into account. It is measured as the average absolute difference between the predicted values and the actual values and is used to assess the effectiveness of a model.



Where,

𝑛: is the number of observations in the dataset

𝐴𝑡: is the actual value at time t

𝐹𝑡: is the predicted value at time t

*2) MAPE*

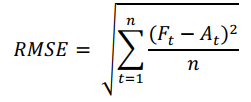
MAPE (Mean absolute percentage error) is a metric that defines the accuracy of a forecasting method. It represents the average of the absolute percentage errors of each entry in a dataset to calculate how accurate the forecasted quantities were in comparison with the actual quantities. MAPE is often effective for analyzing large sets of data and requires the use of dataset values other than zero.

A mathematical equation with numbers and symbols

Description automatically generated

*3) RMSE*

The root mean square error (RMSE) measures the average difference between a statistical model’s predicted values and the actual values. Mathematically, it is the standard deviation of the residuals. Residuals represent the distance between the regression line and the data points.



1. *Result*
   1. *Gold data*
   2. *Silver data*
   3. *Platinum data*
   4. *Remark*

Based on the result of evaluating 9 models, we have GRU, LSTM, KNN and RNN appear to be the best performing models based on their low RMSE, MAE, and MAPE values for both testing and validation sets. The KNN model had the lowest RMSE and MAE values among all models for the testing set, indicating that it had the smallest average error in predicting actual values.

However, in the long run, KNN model cannot perform well when predicting for longer periods, such as 30 days, due to its reliance on nearest neighbors in the training data. As the prediction horizon increases, there may not be enough historical observations available to accurately predict future values.

On the other hand, LSTM, GRU, RNN models have been shown to perform better in predicting longer horizons due to their ability to capture temporal dependencies and patterns in the input data. The recurrent connections in the RNN allow the model to remember past information and use it to make predictions for future time steps. These models had relatively low RMSE and MAE values for both testing and validation sets.

VI. CONCLUSION

In conclusion, the use of statistical models and machine learning algorithms has proven to be an effective method for predicting daily prices of gold, silver, and platinum. The results obtained from this study show that **LSTM, GRU and RNN models** can provide accurate predictions of precious metalprices based on historical data and other relevant factors.

Though, predicting financial markets is a challenging task for our group and requires continuous monitoring, updating models to account for new trends and changes in market conditions. The toughness of this problem cannot be overstated, but with the right approach and tools, we have overcome this problem perfectly.

In future, there is much potential for our team to do further research in this area. As more data becomes available, it will be possible to develop even more sophisticated models that can capture subtle patterns and relationships in the market. Additionally, advances in computing power and machine learning techniques will continue to enhance the accuracy and efficiency of these models. Overall, the future looks bright for those working to improve the prediction of precious metal prices using statistical modeling and machine learning.

VII. ACKNOWLEDGEMENTS

This paper and the research behind it would not have been possible without the exceptional support of our supervisor, **Assoc. Prof. Dr. Nguyễn Đình Thuân**. His enthusiasm,knowledge and exacting attention to detail have been an inspiration for the final draft of this paper. We are grateful for his time, effort, and dedication in helping us to achieve our goals, and we could not have completed this work without his unwavering support and encouragement.

We would also like to extend my heartfelt thanks to **Mr. Nguyễn Minh Nhựt** for his invaluable contribution to thispaper. His expertise in the technical aspects of this work has been invaluable, and his insights and feedback have helped us to refine our ideas and arguments. He has been a patient and supportive mentor, always willing to share his knowledge and experience, and we are grateful for his guidance and advice throughout this process.

Finally, we would like to express my gratitude for the invaluable support and assistance provided by **Ms. Nguyễn Thị Viết Hương** throughout the development of this paper. She hasbeen a dependable and helpful mentor, readily offering feedback and suggestions that have been instrumental in shaping the direction and content of this work.

REFERENCES

1. N. Tripathy, “Forecasting Gold Price with Auto Regressive Integrated Moving Average Model,” *Int. J. Econ. Financ. Issues*, vol. 7, pp. 324–329, Aug. 2017.
2. K. A. Manjula and P. Karthikeyan, “Gold Price

Prediction using Ensemble based Machine Learning

Techniques,” in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India: IEEE, Apr. 2019, pp. 1360–1364. doi: 10.1109/ICOEI.2019.8862557.

1. L. Boongasame, P. Viriyaphol, K. Tassanavipas, and P. Temdee, “Gold-Price Forecasting Method Using Long Short-Term Memory and the Association Rule,” *J. Mob. Multimed.*, pp. 165–186, 2023, doi: 10.13052/jmm1550-4646.1919.
2. A. Azzutti, “Forecasting Gold Price: A Comparative

Study,” Feb. 2016. doi: 10.13140/RG.2.1.4206.5686.

1. “(PDF) Application of ARIMA Model to Forecast Gold

Price in Vietnam.” https://www.researchgate.net/publication/324226832\_A pplication\_of\_ARIMA\_Model\_to\_Forecast\_Gold\_Price \_in\_Vietnam (accessed Jun. 21, 2023).

1. M. Yurtsever, “Gold Price Forecasting Using LSTM, Bi-

LSTM and GRU,” *Eur. J. Sci. Technol.*, Dec. 2021, doi: 10.31590/ejosat.959405.

1. R. K. Singh, D. Ray, and B. C. Sarkar, “Recurrent neural network approach to mineral deposit modelling,” in *2018 4th International Conference on Recent Advances in Information Technology (RAIT)*, Dhanbad: IEEE, Mar. 2018, pp. 1–5. doi: 10.1109/RAIT.2018.8389063.
2. “Risks | Free Full-Text | An Analysis and Implementation of the Hidden Markov Model to Technology Stock

Prediction.” https://www.mdpi.com/2227-9091/5/4/62 (accessed Jun. 21, 2023).

1. G. S. K. Ranjan, A. Verma, and R. Sudha, “K-Nearest Neighbors and Grid Search CV Based Real Time Fault

Monitoring System for Industries,” Mar. 2019, pp. 1–5. doi: 10.1109/I2CT45611.2019.9033691.

1. “About Linear Regression | IBM.” https://www.ibm.com/topics/linear-regression (accessed Jun. 21, 2023).
2. “Linear regression - Wikipedia.” https://en.wikipedia.org/wiki/Linear\_regression ( accessed Jun. 21, 2023).
3. “Autoregressive Integrated Moving Average (ARIMA)

Prediction Model,” *Investopedia*.

https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp (accessed May 18, 2023).

[13] “Khoa học dữ liệu.”

https://phamdinhkhanh.github.io/2019/12/12/ARIMAm odel.html (accessed Jun. 21, 2023).

1. F. Sohil, M. U. Sohali, and J. Shabbir, “An introduction to statistical learning with applications in R: by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, New York, Springer Science and Business Media, 2013, $41.98, eISBN: 978-1-4614- 7137-7,” *Stat.* *Theory Relat. Fields*, vol. 6, no. 1, pp. 87–87, Jan. 2022, doi: 10.1080/24754269.2021.1980261.
2. T. Stasik, “Trevor Stasik: Blog (Archive 2007-2019):

Time-Series Forecasting: Exponential Smoothing Part

1,” *Trevor Stasik*, Oct. 05, 2007. https://trevorstasik.blogspot.com/2007/10/time-series-forecasting-exponential.html (accessed May 18, 2023).

1. “Smoothing 5: Holt’s exponential smoothing - YouTube.” https://www.youtube.com/watch?v=DUyZl-abnNM (accessed Jun. 21, 2023).
2. *7.3 Holt-Winters’ seasonal method | Forecasting:*

*Principles and Practice (2nd ed)*. Accessed: May 18,

2023. [Online]. Available: https://otexts.com/fpp2/holt-winters.html

1. “What is the k-nearest neighbors algorithm? | IBM.” https://www.ibm.com/topics/knn (accessed Jun. 21, 2023).
2. “K-Nearest Neighbor (KNN) Algorithm for Machine Learning - Javatpoint.” https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning (accessed Jun. 21, 2023).
3. “Introduction to Recurrent Neural Network - GeeksforGeeks.” https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/ (accessed Jun. 21, 2023).
4. “LSTM | Introduction to LSTM | Long Short-Term

Memory Algorithms.” https://www.analyticsvidhya.com/blog/2021/03/introduc tion-to-long-short-term-memory-lstm/ (accessed Jun. 21, 2023).

1. P. Radhakrishnan, “Sequence to Sequence Learning,”

*Medium*, Oct. 10, 2017. https://towardsdatascience.com/sequence-to-sequence-learning-e0709eb9482d (accessed Jun. 21, 2023).

1. S. Kostadinov, “Understanding GRU Networks,”

*Medium*, Nov. 10, 2019. https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be (accessed Jun. 21, 2023).

1. “Hidden Markov Model in Machine learning - GeeksforGeeks.” https://www.geeksforgeeks.org/hidden-markov-model-in-machine-learning/ (accessed Jun. 21, 2023).
2. “Investing.com - Stock Market Quotes & Financial

News.” https://www.investing.com/ (accessed Jun. 21,

2023).