International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2025: 7.089

Are Silver Prices Predictable? A Machine-Learning Study with High Predictive Accuracy

Sanjay Bharath

Heartfulness International School - Omega branch
Email: *sanjay.mbharath[at]gmail.com*

Abstract: ***This study examines the predictability of silver prices using time series and machine learning models. Daily silver price data from 2000 to 2025, comprising 6,360 observations, were analysed using ARIMA, Random Forest, and XGBoost models. The dataset was split chronologically into 70% training and 30% testing sets. Model performance was evaluated using RMSE, MAE, and MAPE. Results show that the ARIMA model achieved the lowest prediction errors, indicating strong performance in capturing short term price dynamics. Machine learning models were able to follow general trends but showed reduced accuracy during periods of high volatility. These findings suggest that traditional time series models remain effective for short term forecasting of silver prices, although incorporating additional explanatory variables may improve machine learning performance. The study highlights the challenges of forecasting volatile commodity markets and suggests directions for future research.***

Keywords: Predictive accuracy, ARIMA, XGBoost, Random Forest, Decision Trees, Time series forecasting, Volatility modelling,

1. Introduction

Silver is an important asset that acts as an economic indicator and an extremely valuable investment. It is a precious metal used widely in jewellery that also plays a major role in cultures of different communities, including India, where it is traditionally associated with offerings to deities. Silver, from a financial aspect, acts as a portfolio diversifier; investors use it as a store of value, holding it as a hedge/protection against inflation or during times of economic uncertainty (Goel et al., 2022). Silver also plays a crucial role as an industrial input due to its extremely high thermal and electrical conductivity. It is used widely in electronics, semiconductors, circuits, medical devices, and renewable energy technologies like photovoltaic cells (Cattaneo et al., 2026). This makes silver a widely used commodity across consumer and industrial applications like phones, laptops, TVs, and jewellery. Due to its wide usage, the price of silver will in turn affect the price of the common items in which silver is used. Fluctuations in silver prices, therefore, influence not only financial markets but also industrial production costs, consumer technology prices, and renewable energy-driven sectors.

Silver prices, however, are extremely volatile, exhibiting short- and long-term fluctuations (Ergin et al., 2024). Industrial demand and supply chain disruptions play a major role in quantifying silver's price and influencing silver price dynamics. Macroeconomic factors like exchange rates and inflation also play a critical role in characterising silver prices (Ergin et al., 2024). Several factors, including supply conditions, rate of mining and the rate of its consumption, geopolitical events and crises, volatility of market and equity indexes, affect trends in silver's price. Oil and gold prices are two variables that also affect the price dynamics of silver through inter-commodity relations (Sadorsky, 2022). The price of silver, therefore, depends on a complex web of relations between multiple variables, which leads to a nonlinear and constantly changing price. This makes forecasting the silver price inherently challenging and complex. However, due to silver's important role as both a safe investment asset and a crucial industrial input, forecasting silver prices can lead to better investor decision-making, industrial production and planning, and overall informed decision-making across sectors.

The non-linear and non-stationary nature of silver prices makes forecasting extremely complex and difficult using traditional linear econometric models. Silver price prediction is multivariate or univariate in nature, depending on multiple macroeconomic and industrial indicators, thus making it difficult to model using traditional econometric approaches. In recent years, however, machine learning has emerged as a powerful alternative in the field of financial time series forecasting. Traditional econometric models often rely on assumptions such as linearity, stationarity, and fixed relations, which can lead to faults and errors in a constantly changing and volatile commodity market (Ergin et al., 2024). Silver price markets are highly dependent on industrial and economic conditions and multiple variables, thus requiring models that can adapt and are multivariate in nature rather than univariate. Machine learning models and techniques are capable of modelling multiple variables, which are linear or non-linear in nature, that can adapt to changing conditions and that do not rely on strict parametric assumptions (Rouaba, 2025).

Prior studies and research have shown how traditional econometric models, like the Autoregressive Integrated Moving Average (ARIMA) framework, provide good performance in linear and stable price dynamics, while machine learning models and algorithms like Random Forests, Support Vector Machines, and Gradient Boosting models perform well in nonlinear and complex silver price dynamics conditions (Rouaba, 2025; Ergin et al., 2024). Due to these advantages, machine learning models have the potential to provide more accurate predictions supporting decision-making for investors, industrial stakeholders, systems, and production.

The first objective is to examine how predictable silver prices are using time series and machine learning techniques. The study investigates whether past price movements and selected economic variables can help forecast future silver prices.

Variables such as historical price values, trading volume, and other related financial indicators are included in the analysis. The study builds prediction models using different algorithms and evaluates their performance by measuring forecast accuracy using error metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). The second objective is to systematically compare different forecasting models. In this paper, three main models are used and compared: the ARIMA model, which is a traditional statistical time series model, the Random Forest model, and the XGBoost model, which are machine learning methods. By comparing these models, the study aims to determine which approach performs better in predicting silver prices.

The findings of this study aim to contribute to better understanding of silver price dynamics. The results may help investors, industrial stakeholders involved in manufacturing and supply chain planning, and consumers make more informed decisions. Beyond financial forecasting, this study has broader implications. Silver is a critical industrial input in the renewable energy sector, particularly in solar photovoltaic cells, due to silver's extremely high electrical conductivity. Thus, fluctuations in silver's price will affect the renewable energy sector and will influence the affordability of renewable energy. Rising silver prices will affect the solar value chain and thus will affect the value of renewable energy (Sadorsky, 2022). Due to silver's widespread use in the electronics sector, fluctuations in silver's price will ultimately affect the electronics sector as well, and thus impact the prices of everyday electronic devices, ultimately affecting consumers in turn (Goel et al., 2022). Accurate prediction will also help industries plan procurement, allotment, and optimize resource use. Thus, this paper demonstrates the sustainability implications associated with silver price dynamics, with implications for industries, the renewable energy sector, and consumers.

2. Literature Review

Statistical, traditional econometric, and ML models and algorithms in recent years have been widely used in the field of financial time series forecasting. A growing body of literature is exploring various models, algorithms, and techniques for the accurate prediction of precious metals like gold and silver due to their investment, industrial, and economic advantages and sensitivity to changing economic conditions. This section reviews recent studies that applied statistical, ML, and Hybrid models in financial time series forecasting, particularly for precious metals like silver.

In the paper by Mohammed (2025), daily silver prices were predicted using ARIMA and XGBoost models. The 2 models were compared, and their performance was evaluated using standard error metrics like MAE, MSE, and RMSE. The author found that the ARIMA model performed better than the XGBoost model in terms of prediction accuracy, achieving lower MAE, MSE, and RMSE values. The XGBoost model, however, was able to capture nonlinear trends in large datasets. The paper concluded that traditional statistical models like ARIMA perform better in short-term forecasting, while more advanced models like XGBoost are able to capture nonlinear large dataset trends.

In the work by authors Kulkarni et al. (2025), a linear regression model forecasting gold and silver prices named SOONAR was proposed. The results of the model made the authors conclude that SOONAR achieved 92% accuracy in the predictions made, closely tracking market trends.

In the paper by Sadorsky (2022), tree-based machine learning models were used to predict the direction in which solar stock prices would go depending on silver and oil price volatility. The findings demonstrate the superiority of using tree-based ML algorithms like random forest, extremely randomized trees, and support vector machines as compared to using LOGIT-based models. The models used in the paper were able to detect the direction of solar stock prices, with silver price volatility as a variable, with 85% accuracy for the first 8 days of prediction and 90% accuracy for the first 15 days. On the other hand, the predictions made by the LOGIT model never went above 65%. The paper concluded that silver price volatility is an important factor affecting solar stock price, while showing the high prediction accuracy of tree-based ML models.

In the work by Ergin et al. (2024), 4 different ML models were used for forecasting silver prices, and the accuracy of the various models used was also compared with each other to find out which model works the best. The 4 models used in this work include linear regression, support vector regression (SVM), k-nearest neighbours (KNN), and random forest (RF). The study concluded that the RF and KNN models had higher performance due to higher correlation coefficients, lower error rates, and could capture nonlinear trends as compared to the linear models. The paper also concluded that KNN and RF models had a lower rate of performance decline and thus can be used in long-term predictions. The paper by Goel et al. (2022) evaluated the use of hybrid deep learning models like CNN (Convolutional Neural Network) and CNN-RNN (CNN with Recurrent Neural Network) in silver and gold price forecasting, comparing their performance with a simple RNN model. The paper concluded that the simple RNN model outperformed the hybrid CNN - RNN model in terms of prediction. The paper, however, did not decline the importance and usage of the hybrid models.

The interconnectedness of metal markets plays a vital role in affecting their daily prices. Arı, Y. (2021) used GARCH models and Engle - Granger cointegration to analyse and examine volatility spillovers (when a sharp increase or decrease due to instability in one market crosses over to another market, therefore affecting it). The results of the study show a significant long-term volatility transmission relationship between silver and gold, showing how silver price dynamics are often interconnected with other markets.

Overall, although previous studies demonstrate the effectiveness of advanced machine learning models in capturing nonlinear patterns and improving prediction accuracy, there is still a lack of clear comparison with traditional time series models, such as ARIMA, particularly for silver price prediction.

3. Methods

This section will give a brief explanation about each of the models used to test the gathered data in this study.

3.1. ARIMA

The ARIMA model, developed under the Box-Jenkins methodology, is a widely used statistical model that uses time series data in predictive forecasting. Time series data refers to data of the same type observed at successive points in time. For instance, Figure 1, which we will be discussing later in the Data section of this manuscript, shows a time series plot of the closing price of silver against its time of observation. ARIMA works with data that are often measured at equal intervals, for example, daily, weekly, monthly, or yearly. Time series data, unlike independent data, are often dependent on their past values, and so past behaviour affects current and future behaviour.

The ARIMA model does accurate forecasting and estimation under the key assumption that the time series data is stationary. Stationary time series data means the data recorded does not depend on the time it was observed. The statistical properties of the data, like its mean and variance, remain constant throughout the observation period. Time series data with trends and seasonality are often not stationary, as this would mean the trend affects the value of the data. Since most real-world time series data, like silver prices, often show trends or exhibit seasonality, converting the non-stationary time series data into a stationary time series data is a crucial first step. If X_t it is a stationary time series, then the data points $X_t, \ldots, X_{t+s}$ do not depend on the time of observation t (Box et al. 2015).

Differencing is a process that converts non-stationary data points to a stationary time series, and it is the primary method used in ARIMA models. Differencing is done by taking the difference of two consecutive data points, removing the linear trends, and therefore making the mean remain constant. First-order differencing refers to taking the difference of two consecutive data points (Box et al. 2015).

$$\Delta Y_t = Y_t - Y_{t-1},$$

where, Y_t is the data recorded at time t. Differencing can be performed multiple times if first-order differencing does not yield a constant mean. Second-order differencing is the subtraction of the first-order difference of Y_t and Y_{t-1}.

$$\Delta Y_t'' = Y_t' - Y_{t-1}',$$

where Y_t is the data recorded at time t and Y_t' is the first order difference of Y_t.

Furthermore, $d = x$ represents the order of differencing of a time series data in an ARIMA model, where x is the order. Thus, if $d = 1$, the data has been different once in order to make the data stationary.

ARIMA model stands for AutoRegressive Integrated Moving Average model. It is a model that forecasts future values based on current values. The ARIMA model has three parameters. The Autoregressive (AR) parameter represented by p, the Integrated or the differencing parameter represented by d, and the Moving Average (MA) parameter represented by q. The ARIMA model is also represented as ARIMA(p, d, q). The AR(p) component is the parameter of ARIMA that establishes the current value of interest, as a linear combination of past values. Specifically, p represents the number of past terms the current data is dependent on, so if $p = 2$,

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2},$$

where Y_t is value, at time t. Further, ϕ_n is the coefficient MA(q) which is the parameter of ARIMA, which models how the predictions change based on past errors. So, if the value of Y_{t-1} is affected due to a random shock or event, the difference between the predicted value $Y_{predicted}$ and Y_{t-1} gives the error value represented by ϵ_t, which is also called white noise (Box et al. 2015).

The model takes into account the presence of these random events and their effect on the current value, and so, by past errors, the model can predict the value of interest accurately. Further, q here represents the number of past errors, the current value is dependent on. So, if $q = 1$, then $Y_t = \varepsilon_t + \theta_1 \varepsilon_{t-1}$, where Y_t is value, at time t and θ_m the coefficients. The last parameter of the model is the integrated or differencing model, which converts non-stationary data into stationary time series data. This is done by predicting the ΔY_t which is the change in price rather than the actual price itself, the actual price can then be calculated using the relation $\Delta Y_t = Y_t - Y_{t-1}$.

The complete ARIMA(p, d, q) model then becomes:

$$\Delta Y_t^d = c + \phi_1 \Delta Y_{t-1}^d + \phi_2 \Delta Y_{t-2}^d + \ldots + \phi_n \Delta Y_{t-p}^d + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_m \varepsilon_{t-q}$$

where, ΔY_t is d times difference data, c is the constant term which is called drift, Φ_n is AR coefficients, p is order of AR parameter, ϵ_t is white noise error at time t, θ_m is the MA coefficients, q is the order of MA parameter and d is the order of differencing. Moreover, drift is a constant term that represents the average change per period of the data. The components form the ARIMA (p, d, q) model, which uses past values and errors of the differenced data to estimate coefficients and generate forecasts (Rouaba, 2025).

3.2. Random Forest

Random Forest is an ensemble machine learning model that forms multiple decision trees, combining their outputs to give a final prediction. Unlike linear models, the Random Forest model can capture nonlinear trends and relationships within datasets and has low variance in predictions across datasets, reducing the high variance of a single decision tree.

A decision tree is a learning model that is used for classification and regression tasks. For this research paper, regression decision trees will be used because the study is based on a time-series dataset and financial forecasting with

a continuous target variable. A decision tree is a hierarchical structure of criteria (decision rules) represented in a tree-based structure consisting of nodes and branches. The root node contains all data, which is split into child nodes based on decision rules, and leaf nodes represent final predictions. The criteria and rules partition the dataset into smaller subsets through a process called splitting. Leaf nodes represent final predictions and subsets that can no longer be further divided and are considered to be homogeneous. The model works by recursively partitioning the data into smaller homogeneous subsets. The regression tree model splits the data to minimize the variance of the target variable within subsets and maximise variance reduction at each step. (Salman et al. 2024).

Decision trees are highly sensitive to the dataset employed, where small changes can lead to different decision rules, leading to high variance in predictions. They also follow a greedy splitting approach, which may not produce a globally optimal tree.

Splits in decision trees aim to reduce impurity within the data. In classification trees, impurity is measured using entropy, which quantifies how mixed the data is within a node:

$$H(s) = -\sum_{i=1}^{k} p_i \log_2(pi),$$

where $H(s)$ is the entropy of the dataset S, p_i is the probability of class i in dataset S, and k is the number of child nodes splits that the dataset underwent, is the mathematical model for entropy. A lower entropy value indicates more homogeneous data (Sadorsky, 2022).

Information gain measures the reduction in entropy after a split and is given by:

$$IG = H(Parent) - \sum_{i=1}^{k} \frac{n_i}{n} H(child)_i,$$

where $H(Parent)$ represents the entropy of the parent node, $H(child)_i$ represents the entropy of the $i'th$ child node, n is the size of the original dataset, n_i is the size of the $i'th$ split dataset, $\frac{n_i}{n}$ represents the proportion of data in each child node and k is the number of child nodes created by the split. Higher information gain indicates a better split. While these measures are used in classification trees, regression trees instead use variance to measure impurity.

Variance measures how spread out the data points are from the mean of the dataset. Variance reflects the sensitivity of the model to fluctuations in the datasets, with high variance values indicating how predictions change with small changes to the parent dataset. In the context of financial forecasting, variance reflects the volatility in price movements, with greater variance indicating greater volatility. Mathematically, variance is modelled by the formula

$$Var(S) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2,$$

where $Var(S)$ represents the variance in the dataset S, y_i is the actual value of the variable, $\bar{y}$ is the mean of the target variable, and n is the size of the dataset S. To get pure nodes, the data in each node must be similar to each other, and so the variance of the data must be minimised, and thus variance reduction must be maximised. Splits and the decision rules in regression trees are chosen in such a way as to maximize variance reduction after each step. Variance reduction quantifies how much the variance of the target variable is reduced after each step (Sadorsky, 2022).

The best split is chosen such that variance reduction (VR) is maximised.

$$VR = Var(Parent) - \sum_{i=1}^{k} \frac{n_i}{n} Var(child)_i$$

where $Var(Parent)$ represents the variance of the parent node, $Var(child)_i$ represents the variance of the $i'th$ child node, n is the size of the original dataset, n_i is the size of the i^{th} split dataset, and k is the number of child nodes created by the split. This formula can be understood as subtracting the weighted variance of all child nodes from the variance of the parent dataset. The higher the VR value, the greater the similarity of data points in the split subsets. Thus, by recursively minimising variance at each split, regression trees construct partitioned subsets, with homogeneous data points in each set, contributing to greater predictive performance of the model.

The Random Forest (RF) model is an ensemble ML model that is based on multiple decision trees. The random forest model builds a collection of decision trees and combines their outputs to improve predictive accuracy. Each tree is trained independently on created subsets of the main dataset. Due to random forest being based on decision trees, the RF model is capable of capturing nonlinear and complex trends, which is further attributable to the process of piecewise splitting of data based on a variety of decision rules that form the basis of a decision tree.

The RF model is based on the process of Bagging (Bootstrapping + Aggregation). Bagging is a 2-step process that involves bootstrapping and aggregation. Bootstrapping is the process by which the random forest model generates multiple subsets of the original dataset using sampling with replacement and trains decision trees on these subsets. In addition to bootstrapping, random feature selection is used, wherein only a subset of features is considered; thus, if the price of silver is dependent on gold, oil, and the price one day ago, random feature selection will take any one or two of these features while creating the subsets of the data. Thus, bootstrapping and random feature selection together enable the creation of n subsets from the original dataset and n decision trees trained on these subsets. This ensures that each decision tree sees a different dataset and learns different patterns, helping reduce the correlation and covariance between trees. Aggregation is the process through which the random forest model combines the predictions of the trained decision trees to get one final prediction value (Salman et al. 2024). In a regression random forest model, this is done by taking the average/mean of all predictions of the decision trees.

$$\hat{y} = \frac{1}{n}\sum_{i=1}^{n} T_i$$

where T_i is the prediction of tree i, and n is the number of trees, representing the mathematical expression for the aggregation process of a regression random forest model. This averaging helps reduce the variance in the final prediction of the RF model. This works as individual trees can fluctuate a lot, and averaging helps cancel these fluctuations, reducing the overall variation of the prediction. Thus, bootstrapping and random feature selection ensure a reduction in correlation and covariance between trees. This reduced correlation strengthens the reduction in variance after aggregation, as trees with low correlation move differently. Since the trees are weakly correlated, their errors are less likely to occur in the same direction, making averaging more effective. Averaging also reduces the impact of outliers, producing smoother predictions and improving model accuracy. Thus, random forests are effective in handling nonlinear data while reducing overfitting through the aggregation of multiple low-correlated decision trees.

3.3. Extreme Gradient Boosting (XGBoost)

XGBoost is an ensemble learning algorithm that is based on the gradient boosting method, while also incorporating aspects from the random forests model. The XGBoost model uses multiple decision trees sequentially to improve prediction accuracy and minimize error. It follows a gradient boosting approach where each step improves accuracy and lowers error. This boosting approach makes it effective in capturing complex and nonlinear relationships in financial time series datasets. It is therefore widely used in financial forecasting and time series prediction.

XGBoost works by combining predictions from multiple decision trees, minimising error while increasing accuracy after each step. The model first generates an initial decision tree that provides baseline predictions from the training dataset, after which the errors (residuals) are calculated. A new decision tree is then trained on these residual errors. Thus, when applied to the testing dataset, an initial tree provides a baseline prediction, while subsequent trees predict the errors of previous predictions based on patterns learned from the training dataset. The final prediction is calculated by summing the baseline prediction with the predicted errors. This process is repeated, with each subsequent tree predicting errors, resulting in a final prediction with minimised error. The XGBoost model therefore generates trees sequentially, reducing error at each step and improving overall accuracy. (Rouaba, 2025).

The XGBoost model works by minimising a loss function iteratively at each step. The objective function relates accuracy and complexity of the model, measuring how wrong a prediction for a specific data point is while taking into account the complexity of the generated trees. Mathematically, it is modelled as

$$\iota = \frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

where ι represents the loss after each step, the first term $\sum_{i=1}^{n} L(y_i, \hat{y}_i)$ represents the sum of errors overall n data points, and $\sum_{k=1}^{K} \Omega(f_k)$ represents the sum of the penalties for complex models (regularization) that each of the K generated decision trees receives. Thus, this function ensures that the XGBoost model has high predictive accuracy while keeping the model simple in order to reduce overfitting (Chen and Guestrin, 2016).

The goal of XGBoost is to minimise the loss at each step while ensuring that the model is not too complex, so as not to overfit predictions. Since real-world data contains both patterns and noise, if models become too complex, they start learning the noise as well. This makes the model learn the training set very well, but if the dataset changes slightly, prediction accuracy will decrease due to the noise patterns learnt. Thus, complex trees start learning noise along with the true pattern, which is called overfitting. Overfitting leads to high performance in training but low performance while testing with new, unseen data. The gradient boosting approach ensures that the model moves in the direction that reduces error fastest, showing the model how much to change so that the prediction is accurate. The XGBoost model, through regularization, penalizes complex trees and removes unnecessary splits (the second term in the objective function) (Bentéjac et al. 2019), ensuring that overfitting is avoided and the model performs well in both training and testing. The model also ensures that the error at each step reduces gradually, resulting in stable learning through small updates. The model follows a greedy split selection, where decision trees evaluate possible splits and choose the split that maximizes change in loss, resulting in a minimum loss value. Therefore, these mechanisms ensure that the XGBoost model is high in predictive accuracy while preventing overfitting.

Unlike random forest, which generates decision trees independently of each other, XGBoost generates trees sequentially, with each tree learning from the errors of the previous trees, resulting in higher prediction accuracy. XGBoost is also capable of capturing trends in complex datasets with nonlinear relationships while taking into account multiple features, unlike the ARIMA model, which captures linear trends using past values of the variable. Thus, XGBoost is an effective Machine Learning algorithm that combines decision trees with gradient boosting to ensure iterative improvement through reduction in error at each step. Its ability to reduce error through sequential learning while controlling overfitting makes it suitable for complex datasets such as financial forecasting and time series prediction.

4. Data

For the analyses in this study, we used silver price data sourced from Yahoo Finance. The data consists of 6,360 daily observations of six different variables. The six variables include the date of the observation, the opening, closing, highest, and lowest prices of silver on that day, all of which were in United States Dollars (USD) per ounce, and volume of silver traded on each day, measured in silver units. The data under study uses data from August 2000 to December 2025. Summary statistics were calculated for the main numerical variables: Open, High, Low, Close prices, and trading Volume to understand the general patterns of the data under investigation.

Over the time period, the price of silver varied substantially, from the lowest closing price of 4.026 to a maximum of 77.374 per ounce. The mean of the closing price was 17.961,

and the median was 16.990. The mean being higher than the median indicates and indicates how there were periods of unusually high prices that pulled the average upward. The large difference between the lowest and highest closing prices indicates how silver experienced high volatility. The closing price performance over the study period is shown in Figure 1, in which the major volatility is clearly observed in different time periods, for instance, between 2008 and 2012, and more recently, the sharp price increase.

The daily opening prices exhibited similar behaviour to the daily closing prices. The opening prices varied from a range of $4.03 to $79.70. These prices had a mean of 17.967, which is equal to the mean of the daily closing prices, indicating consistency in the long-term pricing of silver.

The highest and lowest price variables represent the highest and lowest prices that silver was traded at in one trading session/day. The minimum highest price that silver reached in a day was 4.026, and the maximum highest price was 79.700. The lowest price that silver reached throughout the 25 years was 4.026, and the maximum lowest price silver was traded at was 73.735$ per ounce. These high and low variables help capture intraday trading volatility and thus, quantify short-term market volatility.

The interquartile range describes the middle 50% of the values in the dataset. 25% of closing prices were below approximately $12.14, which was the 1st quartile of the dataset. 50% of closing prices were below approximately $16.99, which is the median of the dataset, and 75% of closing prices were below approximately $23.5, which was the 3rd quartile of the dataset. This demonstrates how, for the dataset period, silver was traded between 12$ and 23$ per ounce with some extremely high spikes in between that lead to an increase in silver's mean.

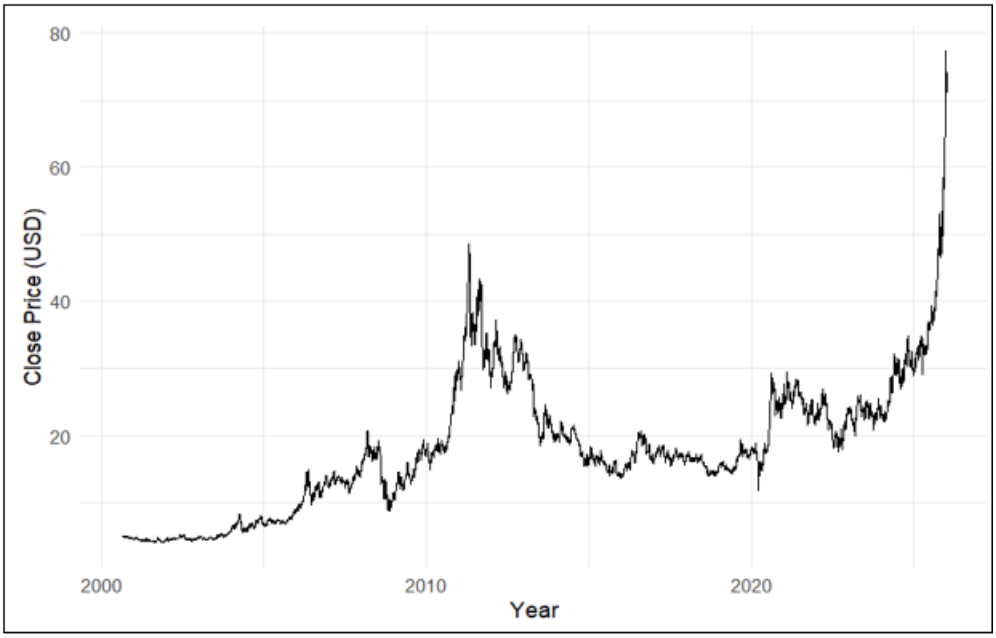

Figure 1: Time series of historical silver closing prices in USD for the period August 2000 to December 2025. The series shows long-term growth, volatility spikes in the period 2008-2012, and a sharp increase in recent periods, reaching new highs near 80 USD.

5. Results

5.1 ARIMA findings

An ARIMA (3,1,3) model with a constant drift value was estimated to model the closing prices of silver for the time series data. The ARIMA model has p=3, d=1 and q=3, showing how the value of interest depends on the consecutive past 3 values, how the series was different once to achieve stationarity, and how the value of interest depends on the past 3 consecutive white noise errors, respectively. The estimated model is given by:

$$\Delta Y_t = 0.01 - 0.35\Delta Y_{t-1} - 0.37\Delta Y_{t-2} - 0.89\Delta Y_{t-3} + \varepsilon_t + 0.27\varepsilon_{t-1} + 0.42\varepsilon_{t-2} + 0.83\varepsilon_{t-3}$$

where, $\Delta Y_t = Y_t - Y_{t-1}$ with Y_t representing the price of silver at price t. The estimated positive drift value is 0.01, indicating a small but persistent upward movement of prices over time. The negative AR coefficients -0.35, -0.37, and -0.89 suggest a strong dependency on past values, while the MA coefficients of 0.27, 0.42, and 0.83 indicate how the value of interest is substantially affected by past shocks. The estimated AR and MA coefficients indicate that past changes and past shocks significantly affect current price movements, suggesting short-term dependence in the data.

Three standard error (uncertainty) metrics were used to evaluate the performance of the estimated ARIMA model. The error metrics used were the Root Mean Squared Error (RMSE), the Mean Absolute Error (MAE), and the Mean Absolute Percentage Error (MAPE). For this particular model, the metrics were found as RMSE = 0.47, MAE = 0.27, and MAPE = 1.39%.

In more detail, RMSE measures the square root of the average squared difference between the actual and predicted values, because of the squaring, large errors become much bigger, thus penalizing large outliers. A low RMSE value indicates

how a model is more precise, having smaller deviations from the actual prediction. Thus, the RMSE value of 0.47 indicates how the model's prediction deviates from actual values by 47 cents per ounce. Moreover, MAE represents the average of the absolute difference between the actual and predicted prices. This error metric treats all errors equally, giving equal weight to all errors. It is a clear representation of average forecasting accuracy. The model received an MAE of 0.27, indicating how the model's prediction is off by 27 cents per ounce. Given that the RMSE value is almost double the MAE value, we have an indication that there are periods where the silver price is extremely high or low, causing large forecasting errors during periods of extreme price fluctuations and high volatility. This means that the model would struggle to predict silver prices during high volatility periods very accurately, which is a limitation of such methods and such markets, because exogenous factors affect the actual prices. Finally, MAPE measures the average absolute error in forecasting as a percentage of the actual values. It measures errors in relative terms, making the errors easy to interpret. An MAPE value of 1.39% indicates how the model's predictions are, on average, 1.39% away from the actual price of silver. This low MAPE value demonstrates how the errors are relatively low in magnitude, indicating high accuracy and strong predictive performance. The MAPE value is a very important indicator because it is not based on absolute value terms of the variable under study (which are affected by the practical relevance for the specific investigation), but a percentage measure.

Following the interpretation of the fitted ARIMA model, a 365-day ahead forecast was generated using the estimations. The forecast exhibits a continued upward trend with wide confidence intervals, as seen in Figure 2, consistent with the positive drift constant value of 0.01. The wide confidence intervals show how uncertainty in the predicted value is large and how this uncertainty is increasing with time, which is consistent with the strong MA coefficients indicating how the persistence of shocks increases the uncertainty over time. This expanding uncertainty band highlights the volatility of silver prices and how uncertainty accumulates from previous predicted values. To increase our confidence around the point estimates, as a future research plan, we can increase our sample size or get more informative estimates by increasing the number of factors in the study.

Overall, the estimated ARIMA(3,1,3) model predictions with a positive drift value show a strong fit to the dataset due to a small MAPE value capturing the short-term dynamics of silver prices, although over a long period, the uncertainty band may increase with time due to the inherent volatility in the silver price market.

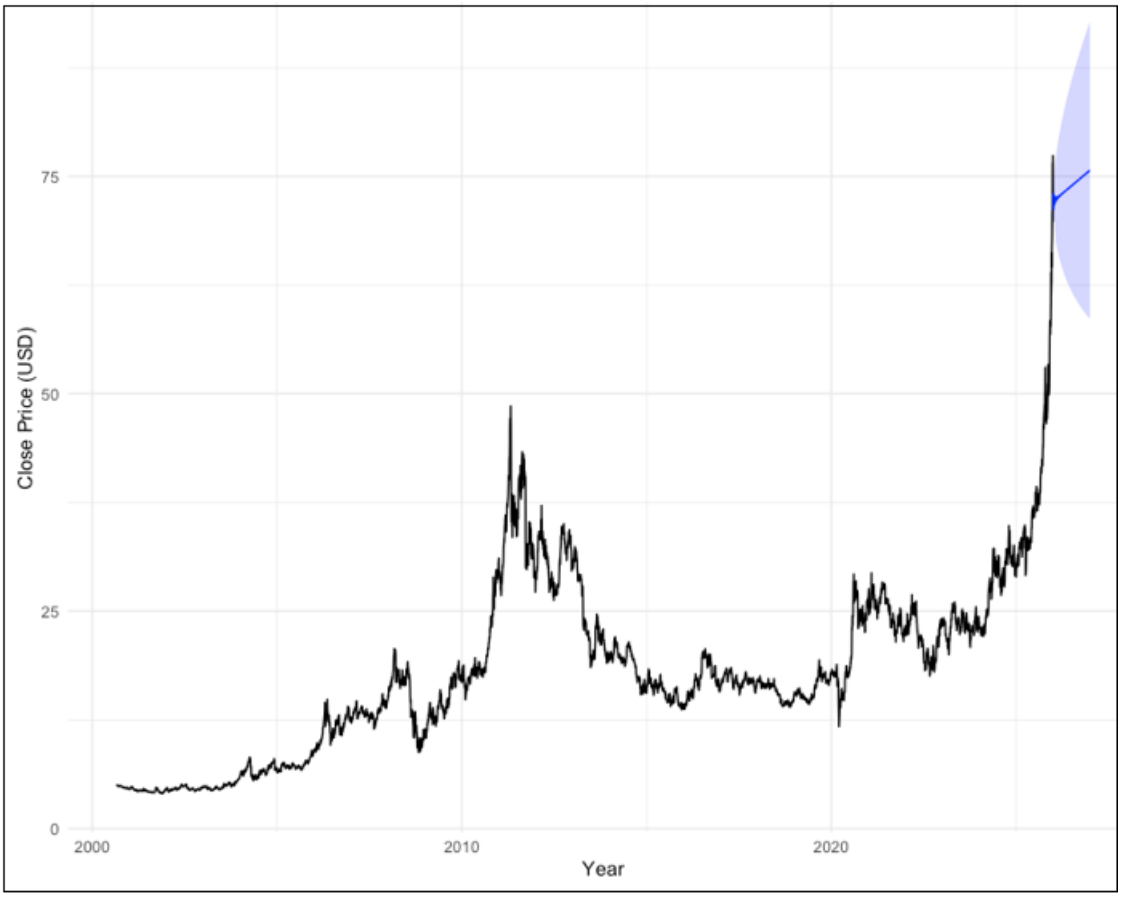

Figure 2: One-year forecast of silver futures prices using the ARIMA model. The shaded areas represent the confidence intervals, showing that prediction uncertainty increases as the forecast moves further into the future.

5.2 Random Forest findings

A random forest regression model was trained to predict daily silver closing prices. Unlike traditional time series regression models like ARIMA, random forests learn patterns between input feature variables and the output target variable. To create and train the model, the original dataset of 6360 observations and 6 variables was transformed into a new

dataset with 4 additional variables, acting as features for the model. The 4 additional variables added to the dataset are lag1, lag3, lag5, and MA3. lag-i represents the closing price i days ago, so here lag1, lag3, and lag5 represent the closing price of silver 1, 3, and 5 days ago, respectively. The lag-i term helps the model learn the relationship of the target variable with its history, creating a pattern between the price today and the price i days ago. MA3 represents the moving average of silver's closing prices for the past 3 days, and so MA3 is the average of the past 3 days of silver's closing price. MA3 helps capture the general trend over the past 3 days for the target variable, developing a pattern between the present price and the general trend observed for the past 3 days. Thus, lag- i for $i = 1,3,5$, and MA3 help the model learn relationships between recent price history and help capture the recent local trend.

The random forest model was trained on these 4 new variables: lag1, lag3, lag5, and MA3. The model thus used the predictor variables lag1, lag3, lag5, and MA3 as inputs to predict the target output variable (closing price of silver). All predictor variables used were only of past known data, ensuring no future information is used in the prediction process, preventing data leakage. The random forest model learns a function, given recent prices, to predict today's price.

$$\begin{aligned} Lag\ 1 &= Close_{t-1} \\ Lag\ 3 &= Close_{t-3} \\ Lag\ 5 &= Close_{t-5} \\ MA3_t &= \frac{Close_{t-1}+Close_{t-2}+Close_{t-3}}{3}, \end{aligned}$$

where $Close_t$ represents the closing price of silver on the t observation, represents lag1, lag3, lag5, and MA3. Thus, the random forest model can be mathematically represented as

$$Close_t = f(lag1, la3, lag5, MA3) = f(Close_{t-1}, Close_{t-3}, Close_{t-5}, \frac{Close_{t-1}+Close_{t-2}+Close_{t-3}}{3}),$$

where $Close_t$ is represented as a function of the input predictor variables.

The new dataset with the predictor variables was also chronologically split, such that 70% of the data was used to train the model, and the remaining 30% of the data was used to test the predictive performance of the model. The random forest model formed was of 500 trained decision trees, with each tree learning different patterns from different subsets of the data, with the final prediction being an average of the 500 decision trees' predictions, reducing variance and giving a stable model (bagging). The model is doing one-step-ahead prediction using known lag and moving average values, where the closing price of silver at day t is predicted using known past values. Figure 3 shows a graph representing the actual price of silver (blue line) and the model's predicted price of silver for the 30% testing dataset (red line).

The model's performance was evaluated using RMSE, MAE, and MAPE. The model achieved the following results on the test data: RMSE = 2.37, MAE = 0.60, and MAPE = 1.72%. The MAE indicates that the predictions were, on average, 60 cents per ounce off from the actual silver price. The MAPE value of 1.72% shows that the average prediction error was approximately 1.72% of the true price. However, the high RMSE value of 2.37 compared to MAE shows the presence of large prediction errors (RMSE gives more weight to large errors), likely due to high volatility. The model was able to accurately capture the trend in silver prices; however, during periods of rapid price change, the model predictions were less accurate (shown by low MAE but high RMSE value). This could be explained as the random forest model relies heavily on past data and trends, struggling to predict during times of sudden market changes.

The performance of the random forest model can be compared to the performance of the ARIMA model used earlier in the study by comparing the evaluation metrics received by both models. The ARIMA model achieved a lower MAE of 0.27 compared to the MAE of 0.60 for the random forest model, and also a slightly lower MAPE 1.39% when compared with a MAPE of 1.72% for the random forest model. This suggests that the ARIMA model provided more accurate predictions than the random forest model. This may be explained by the fact that ARIMA is designed for time series data and can capture patterns such as trends and dependencies over time more effectively than machine learning models like the random forest model, which is highly dependent on the features chosen and may not be able to capture the trend completely. The lower RMSE value achieved by ARIMA also indicates how the ARIMA model performs better during times of volatility.

Figure 3: Actual (blue) versus predicted (red) silver prices using the Random Forest model. The model closely follows overall price trends but underestimates sharp increases in price towards the end of the test period.

5.3 Extreme Gradient Boosting (XGBoost) findings

An XGBoost model was trained to predict the daily silver closing prices. The features used in the dataset were the same as the features used for the Random Forest model, which are lag1, lag3, lag5, and MA3. The chronological splitting of the dataset for the training and testing sets for the XGBoost model was also 70% and 30%, the same as the Random Forest model, allowing fair comparison between the two models. The number of sequential decision trees used in the model was 100 trees, with each tree reducing the error of prediction from the previous tree. Mathematically, the XGBoost model can be represented by

$$\hat{y} = \sum_{k=1}^{100} f_k(x)$$

where $\hat{y}$ is the predicted value, $x = (lag1, lag3, lag5, MA3)$ are the input features, and $f_k(x)$ is the output of the kth decision tree. The function can be interpreted as summing the outputs of all decision trees generated, with each decision tree mapping the input feature variables, i.e., lag1, lag3, lag5, and MA3, to an output value.

The first decision tree provides the baseline prediction, learning relationships between the input features and the target variable, and the subsequent decision trees predict residual errors for previous predictions, using the same features, in turn improving predictive accuracy. The final prediction is obtained by adding the baseline prediction to all the residual errors, with the objective of each decision tree generated being to minimise the loss function. It can also be understood mathematically by

$$\hat{y} = f_1(x) + f_2(x) + f_3(x) + \ldots + f_{100}(x)$$

where $f_1(x)$ represents the first decision tree's baseline predictions, with subsequent trees $f_2(x)$ to $f_{100}(x)$ representing the residual errors and $\hat{y}$ representing the final prediction. This model thus learns historical patterns through the lag features and the short-term trends through the moving average feature, while reducing overall error to give accurate predictions.

The model's performance was evaluated using the same performance metrics as ARIMA and Random Forest, i.e., RMSE, MAE, and MAPE. The model achieved the following performance metric values: RMSE - 2.321, MAE - 0.617, and MAPE - 1.758%. The MAE value of 0.617 demonstrates how the model's predictions were, on average, 0.62 cents away from the actual value. The MAPE value indicates that the predictions were, on average, about 1.76% different from the true silver price. The high RMSE value of 2.321 compared to the MAE shows the presence of large errors, showing how some predictions in the model have high deviation from the actual value, which could be due to the influence of the volatility of the price of silver. The relatively low values of RMSE, MAE, and MAPE show how the model performs well on unseen data, suggesting that overfitting is limited. Therefore, the model is able to capture the trend accurately, modelling patterns in time series data effectively.

6. Conclusion

This study compared ARIMA, Random Forest, and XGBoost models for forecasting silver prices using daily data from 2000 to 2025. The results show that ARIMA achieved the lowest error values across RMSE, MAE, and MAPE, indicating stronger performance in short term forecasting. Machine learning models captured overall trends but were less effective during periods of high volatility. These findings suggest that traditional time series approaches remain reliable for short term commodity forecasting, while machine learning models may benefit from additional explanatory variables. Future research should incorporate macroeconomic indicators and external market factors to improve predictive performance and better capture sudden price movements.

References

[1] Arı, Y. (2021). Engle-Granger cointegration analysis between Garch-type volatilities of gold and silver returns. *Alanya Akademik Bakış*, *5*(2), 589-618.

[2] Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2019). A comparative analysis of XGBoost. *arXiv preprint arXiv:1911.01914*.

[3] Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.

[4] Cattaneo, V., Mast, J., Hackenhaar, I., Nardone, S., Scheerlinck, S., Mertens, J., & Dewulf, J. (2026). Forecasting silver demand and supply by 2030: Impact of silver-intensive photovoltaic cells and sectoral competition. *Resources, Conservation and Recycling*, *224*, 108562.

[5] Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).

[6] Ergin, E., & Eren, B. S. (2024). Assessing the Effectiveness of Machine Learning Techniques for Silver Price Prediction: A Comparative Study. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, *13*(4), 1293-1303.

[7] Goel, S., Saxena, M., Sarangi, P. K., & Rani, L. (2022, November). Gold and silver price prediction using hybrid machine learning models. In *2022 Seventh International Conference on Parallel, Distributed and Grid Computing (PDGC)* (pp. 390-395). IEEE.

[8] Kulkarni, S., Patil, P., Mankar, A., Thamke, P., & Pathak, G. (2025). Gold and Silver Price Forecasting System using Linear Regression Model (SOONAR). *Journal of Scientific advances*, *2*(01), 85-96.

[9] Rouaba, M. (2025). Forecasting Daily Silver Prices Using ARIMA and XGBoost: A Comparative Analysis. *مجلة إقتصاد المال والأعمال*, *10*(2), 1075-1088.

[10] Sadorsky, P. (2022). Forecasting solar stock prices using tree-based machine learning classification: How important are silver prices? *The North American Journal of Economics and Finance*, *61*, 101705.

[11] Salman, H. A., Kalakech, A., & Steiti, A. (2024). Random forest algorithm overview. *Babylonian Journal of Machine Learning,* 2024, 69-79.