AKDENİZ İİBF DERGİSİ
Akdeniz İİBF Journal
https://dergipark.org.tr/tr/pub/auiibfd

e-ISSN 2667-7229 http://dx.doi.org/10.25294/auiibfd.1404173

Comparative Analysis of Deep Learning Models for Silver Price Prediction: CNN, LSTM, GRU and Hybrid Approach

Derin Öğrenme Modellerinin Gümüş Fiyat Tahmininde Karşılaştırmalı Analizi: CNN, LSTM, GRU ve Hibrit Yaklaşım

Yunus Emre GÜR[a]

ARTICLE INFO

Article History	
Received	13 December 2023
Accepted	23 December 2023
Available Online	8 February 2024
Article Type	Research Article

Keywords

Silver Price Prediction,
Deep Learning,
LSTM,
CNN-LSTM-GRU,
Financial Analysis.

ABSTRACT

In this study, the performance of different deep learning algorithms to predict silver prices was evaluated. It was focused on the use of deep learning models such as CNN, LSTM, and GRU for the prediction process, as well as a new hybrid model based on combining these models. Each algorithm was trained on historical silver price data and compared its performance in price prediction using this data. This approach aims to achieve more comprehensive and accurate forecasts by combining the strengths of each model. It also makes a unique contribution to the literature in this area by addressing a specialized area such as the silver market, which is often neglected in financial forecasting. The study presents an innovative approach to financial forecasting and analysis methodologies, highlighting the advantages and potential of deep learning models for time-series data processing. The results compare the ability of these algorithms to analyze silver prices based on historical data only and to assess past trends. The study showed that these algorithms exhibit different performances in analyzing historical data. In conclusion, this study compared the performance of different deep learning algorithms for predicting silver prices based on historical data and found that the CNN-LSTM-GRU hybrid model has the potential to make better predictions. These results can provide guidance to researchers working on financial analysis and forecasting.

MAKALE BİLGİSİ

Makale Geçmişi	
Başvuru	13 Aralık 2023
Kabul	23 Aralık 2023
Yayın	8 Şubat 2024
Makale Türü	Araştırma Makalesi

Anahtar Kelimeler

Gümüş Fiyat Tahmini,
Derin Öğrenme,
LSTM,
CNN-LSTM-GRU,
Finansal Analiz.

ÖZ

Bu çalışmada, gümüş fiyatlarını tahmin etmek amacıyla farklı derin öğrenme algoritmalarının performansını değerlendirilmiştir. Tahmin işlemi için CNN, LSTM ve GRU gibi derin öğrenme modellerinin kullanımı ile birlikte bu modellerin birleştirilmesi üzerine yeni bir hibrit model üzerine odaklanılmıştır. Her bir algoritma, geçmiş gümüş fiyat verileri üzerinde eğitilmiş ve bu verileri kullanarak fiyat tahminlerindeki performansları karşılaştırılmıştır. Bu yaklaşım, her bir modelin güçlü yönlerini birleştirerek daha kapsamlı ve hassas tahminler elde etmeyi hedefler. Ayrıca, finansal tahminlerde sıklıkla göz ardı edilen gümüş piyasası gibi özel bir alanı ele alarak, bu alandaki literatüre özgün bir katkı sağlamaktadır. Çalışma, zaman serisi verilerinin işlenmesi konusunda derin öğrenme modellerinin avantajlarını ve potansiyelini vurgulayarak, finansal tahmin ve analiz metodolojilerinde yenilikçi bir yaklaşım sunmaktadır. Sonuçlar, bu algoritmaların sadece geçmiş verilere dayalı olarak gümüş fiyatlarını analiz etme ve geçmiş trendleri değerlendirme yeteneklerini karşılaştırmıştır. Çalışma, bu algoritmaların geçmiş verilere dayalı analizlerde farklı performanslar sergilediğini göstermiştir. Sonuç olarak, bu çalışma, gümüş fiyatlarının geçmiş verileri üzerinden tahmin edilmesi için farklı derin öğrenme algoritmalarının performansını karşılaştırmış ve CNN-LSTM-GRU hibrit modelinin daha iyi tahminler yapma potansiyeli taşıdığını ortaya koymuştur. Bu sonuçlar, finansal analiz ve tahmin konularında çalışan araştırmacılara yol gösterici olabilir.

1. Introduction

One unique metal that is very significant to economies is silver. The metal's industrial use and strong demand as a jewelry metal make trade in this metal a crucial area to concentrate on. A lot of investors hoard silver as a hedge against an impending economic collapse, and the volume is high. The pandemic has had a significant negative impact on the financial sector, and silver has become more and more popular as an investment in recent years. A major factor influencing economic development is the demand for

[a] Research Assist., Fırat University, Faculty of Economics, Administrative and Social Sciences, Management Information Systems, Elazığ, E-Posta: yegur@firat.edu.tr, ORCID: https://orcid.org/0000-0001-6530-0598

silver and how quickly its prices change. All the same, silver investments have evolved into a tool for financial management, as investors have begun to recognize of late. Silver continues to be a significant player in the financial markets and often serves as both an industrial metal and an investment product. Accurate silver price forecasting is crucial for economic growth, nevertheless, because of the recent economic impact of the pandemic and fluctuations in silver prices (Goel et al., 2022: 390).

The price of silver is influenced by a number of variables, including the enactment of new laws, the health of the world economy, political events, investor psychology, etc. With these factors influencing price swings, it is difficult to determine the price of silver with any degree of precision. Forecasting time series data has been done using conventional machine learning techniques including genetic algorithms, decision trees, and support vector machines. All of these methods, however, have drawbacks, including inadequate handling of unique values in time series data and inadequate non-linear data fitting capabilities. An increasing number of deep learning techniques are being used for time series forecasting as technology advances. The alterations in nonlinear time series data may be more effectively accommodated by deep learning algorithms (Wang et al., 2023: 1-2).

Silver market analysis is an important area of financial forecasting, and deep learning (DL) models are receiving increasing attention in this field. Existing literature addresses the use of DL models in a variety of financial applications, including exchange rate, stock market, and oil price forecasting. In particular, models such as LSTM and CNN are popularly used in such forecasts, and complex neural networks have the potential to achieve high accuracy. By focusing on the application of DL models for price forecasting in the silver market, this study aims to extend the existing work in this area and provide a more in-depth analysis. In this way, we aim to better understand the potential of deep learning in financial forecasting and analysis.

The deep learning algorithms used in this study have different capabilities in analyzing and predicting the silver market, and each makes a unique contribution. CNNs are powerful at recognizing complex data patterns because they have the capacity to automatically learn meaningful features from high-dimensional data. On the other hand, RNN models such as LSTM and GRU are effective in modeling sequential features of time series data because they can capture dependencies over time and learn relationships within the data stream. These features make both types of models suitable for use in time series forecasting, but each offers advantages in different ways. CNNs are effective in recognizing and learning complex patterns within the data, while LSTM and GRU are powerful in modeling time dependencies between the data (Wibawa et al., 2022). Therefore, this study focuses on the combination of these three algorithms and explores how each of them plays different roles in analyzing the silver market. The aim of this approach is to utilize the complementary features of the algorithms to obtain more comprehensive and accurate forecasts. This information will provide readers with a better understanding of whether these algorithms are alternatives to each other throughout the paper.

The main objective of this study is to objectively compare the performance of various deep learning algorithms using historical silver price data. There are many reasons for using deep learning algorithms in this study. While traditional methods such as XGBoost offer fast training times and efficient performances, deep learning models are better at modeling advanced structures, especially time series data and sequence data. Deep learning methods are generally better at time series prediction than machine learning methods. They can more effectively model the complex structures of time series, which improves forecasting performance (Elsayed et al., 2021; Lara-Benítez et al., 2021). For these reasons, deep learning algorithms are preferred in this study. However, evaluations based on historical data show how well each algorithm evaluates previous patterns. The study uses daily data on silver prices per gram. This data is used to evaluate how well deep learning systems such as CNN, LSTM, and GRU predict these data. The paper also proposes a new hybrid CNN-LSTM-GRU model. This model outperforms other algorithms in terms of silver price prediction. For academics and investors interested in understanding the complexity of silver prices and improving their ability to predict future price fluctuations, these findings provide an important discovery.

2. Literature Review

In a research, Vidya and Hari (2020) stress how crucial it is to predict the trajectory of gold prices, beginning with the fact that gold is one of the greatest investment possibilities and is always in demand. Planning for finances and investments requires these estimates since gold prices are not linear. An exponential curve may be used to represent changes in the price of gold. Convolutional neural networks (CNN) are among the finest methods for resolving the nonlinear features of data; furthermore, recurrent neural networks (RNN) are the most appropriate for time series forecasting and assessment. The suggested model is among the most effective techniques for financial forecasting, according to the findings of the research conducted using the World Gold Council's dataset.

Kong et al. (2021) were able to more accurately portray current data variations than the MSE approach by employing the k-AMSE parameter to indicate the price fluctuation of a spot sample. Data on Amazon spot prices going back 90 days was retrieved from the Amazon Cloud and the researchers analyzed the resulting price dispersion. Next, a prediction model based on the GRU network is presented, and the factors that affect price volatility are discussed. The RMSE is used to evaluate the proposed approach in comparison to other methods. The experimental results show that the GRU network method can achieve higher accuracy (1.58e-3%).

Hamayel and Owda (2021) proposed three separate RNN techniques for forecasting the prices of three various cryptocurrencies, including Ethereum (ETH), Litecoin (LTC), and Bitcoin (BTC). Models employ the MAPE statistic to make extremely accurate projections. The results demonstrated that, across all bitcoin kinds, the GRU model performed better than the other two models, LSTM and bi-LSTM. GRU was regarded as the best algorithm as a result. For Bitcoin, Ethereum, and Litecoin, GRU's forecasts were the most accurate (0.2454%, 0.8267%, and 0.2116%

MAPE, respectively). The lowest performance in predicting Bitcoin, Ethereum, and Litecoin (MAPE percentages: 5.990%, 6.85%, and 2.332%, respectively) was achieved by the Bi-LSTM algorithm. In general, the study's prediction models produced fair estimates of future cryptocurrency values.

Due to their importance in both the financial and industrial sectors, Lin et al. (2022) set out to develop a method to reliably predict the price of precious metals. In order to increase the accuracy of price predictions for precious metals, this research integrates a long short-term memory neural network (LSTM) with the modified ensemble empirical mode decomposition (MEEMD) method. According to studies using multiscale permutation entropy (MPE), MEEMD has a greater decomposition effect than ensemble empirical mode decomposition (EEMD). After feeding each IMF into the LSTM that MEEMD had generated, a forecast was created. The aggregate forecast was then calculated by aggregating the individual IMF forecasts. When compared to traditional forecasting models such as those based on multilayer perceptron neural networks (MLP), support vector regression (SVR), and super learners (SL), MEEMD-LSTM scored better in both single- and multi-step forward predictions. The multi-horizon model confidence set (MCS) test provides robust and statistical evidence that MEEMD-LSTM has the best forecasting performance. This research also shown that the model's predictive accuracy increases throughout a range of training-to-test set ratios and stages of the economic cycle.

The research by Malik et al. (2022) tackles a persistent pattern in which financial trading in the stock market is determined by time series forecasting. Value forecasting is done in the research using deep learning methods like LSTM. The long-term reliance issues have been resolved by the LSTM approach. Additionally, the unit cell and forgetting gate shielded the LSTM from gradient fading and made it possible for the algorithms to efficiently store and analyze over a thousand data points. The actual stock values and the anticipated stock values are shown in the experimental analysis findings.

Goel et al. (2022) conducted a study to determine the efficacy of machine learning models in predicting the prices of gold and silver on the Indian market. This prompted the exploration of CNN and CNN-RNN-based hybrid machine learning models. The experimental phases of the project used daily trading data from January 2021 through August 2022. The MAPE was used to measure the quality of the models. The results showed that the RNN model performed best only when predicting the price of gold, whereas the other models performed almost as well.

In their research, Patel et al. (2022) defined cryptocurrencies as digital trade instruments that are secured by secure hashing algorithms (SHA-256 and MD5) and function in a decentralized way. Since their introduction, the values of cryptocurrencies have fluctuated greatly and seen sharp increases, particularly during the COVID-19 epidemic. Because of this, it has grown in popularity as a choice among investors who want to get big profits quickly. Due of these significant price swings, investors and experts have begun to forecast bitcoin values. Various machine learning and deep learning algorithms, including as GRU, LSTM, and ARIMAX, have been applied to the task of forecasting cryptocurrency prices and investigating the factors that affect them. Though the research now in publication concentrates on predicting cryptocurrency values, it often overlooks the coin's dependency on other cryptocurrencies. This article presents a paradigm for predicting cryptocurrency prices by taking Dash Coin's reliance on other cryptocurrencies, such Bitcoin and Litecoin, into account. This method takes into consideration the hierarchical relationship between various cryptocurrencies in order to attain improved forecast accuracy. The expected price of the Dash currency is c to demonstrate this idea. This method takes into consideration the hierarchical relationship between various cryptocurrencies in order to attain improved forecast accuracy. In order to demonstrate this idea, the price prediction of Dash coins is taken into consideration. The findings demonstrate that the suggested model accomplishes price predictions with minimal loss and high accuracy.

A number of machine learning models, including as ANN, LSTM, and SVR, were studied by Yang et al. (2022) in order to forecast gold prices based on commodities, conventional indices, emergent indicators, and gold history time series. Models for predicting gold prices are built using ANN, LSTM, and SVR, three machine learning techniques. The research uses a time series that begins on January 1, 2017 and ends on December 31, 2020 as its dataset. It covers the S&P 500 and the DJI, along with Bitcoin and Ethereum, silver and crude oil, the USD index (which tracks the value of the US Dollar in comparison to the Euro) and gold price data (both historical and volatility). MAE, RMSE, and MAPE are used as measures of effectiveness. In the first step, we compared the three models side by side. In the second part, we see an assessment of how the models are affected by cryptocurrency. The findings of the study indicate that the Support Vector Regression (SVR) model had superior performance compared to the other two models. Furthermore, the inclusion of supplementary data pertaining to cryptocurrencies had a beneficial effect on all three models.

Li et al. (2023) selected 11 influencing factors to be used as explanatory variables for the fluctuations in copper prices after analyzing the qualitative relationships between the variations in copper prices and variables like supply and demand, energy costs, alternative metals, global macroeconomic conditions, and national policies. These variables are merged with monthly time series data for copper price forecasts to create a two-dimensional multivariate time series that is integrated into a CNN-LSTM network. Experimental results show that the suggested strategy performs better than other existing techniques because to its ability to extract features on both the temporal side of LSTMs and the attribute space side of CNNs.

Rao et al. (2023) mentioned that industry-specific stock price swings are a big cause for worry in the market and that as more players enter the market, the capacity to correctly forecast stock price movements becomes increasingly valuable. The authors claim that historically, a great deal of study has been done on the subject of stock market forecasting utilizing machine learning techniques and technologies. Interesting features that contribute to the complexity of this modeling include time dependency, volatility, and similar complicated relationships. The approach to predicting stock prices suggested in this study uses a hybrid technique based on deep learning to get around this problem. Following input data processing, preprocessing is carried out to increase accuracy before MPSOA characteristics are chosen. Lastly, the MC-GRU model training employs this technique. The suggested approach outperforms the CNN and GRU models in terms of performance.

According to a study by Sulistio et al. (2023), one way to reduce the risks involved in stock investing—which is especially popular among inexperienced investors—is to use deep learning to estimate stock closing prices. When six different deep learning algorithms were compared for their ability to predict stock closing prices, the CNN-LSTM-GRU hybrid algorithm combination fared better than the other methods. The RMSE went down by 1,100 units, or 14%, the MAE went down by 0.798 units, or 13%, and the R squared went up by 0.957 units, or 3.9%, as measured by these metrics. For forecasting energy stocks on the Indonesian Stock Exchange, in particular, the authors believe that the CNN-LSTM-GRU combination is preferable than employing a single algorithm.

Chen et al. (2023) did a research to forecast and explain the factors influencing the Bitcoin price, with the goal of obtaining the Bitcoin price for the next day using a highly accurate algorithm model. The authors note that there is a substantial body of earlier work on the issue of Bitcoin price forecasting research and that the two most prevalent techniques are the ARMA model and the LSTM algorithm. Even though random forest regression produces predictions with lower RMSE and MAPE than LSTM, the Diebold-Mariano test is unable to show that random forest regression predicts more accurately than LSTM. Random forest regression was also used to obtain shifts in the factors that determined the Bitcoin price throughout the various time periods. Between 2015 and 2018, the price of Bitcoin was affected by the price of oil, the price of Ethereum, and the three major NASDAQ, DJI, and S&P500 indexes. Two important variables since 2018 have been the price of ETH and the JP225 index of Japanese equities. In terms of the relationship between the accuracy and the number of periods in which the model incorporates explanatory variables, the model that uses a single lag to anticipate the price of Bitcoin the next day has the highest prediction accuracy.

In this literature review, it is important to explain in detail the limitations and improvements of the methods used by the studies described for price forecasting. Vidya and Hari (2020) on gold prices and Kong et al. (2021) on Amazon spot prices focus on specific datasets. However, these studies lack generalizability across different market conditions and data types. This study aims to extend the applicability of deep learning models to more diverse datasets and market conditions. However, Hamayel and Owda's (2021) study on cryptocurrencies and Lin et al.'s (2022) study on precious metals illustrate limitations in terms of the complexity and performance of specific models. While the studies evaluate the performance of a single model, they ignore the potential advantages of combining different models. In this study, the focus is on exploring the synergies that can be achieved by hybridizing these models. In the existing literature, the focus is often on a particular algorithm or model, but the potential of combining these models and hybrid approaches has not been sufficiently explored. Moreover, the inability to adequately model the complexity of time series data and market fluctuations is a common shortcoming. This paper aims to fill these gaps and presents an advanced hybrid model that addresses the challenges that existing models in the literature cannot overcome.

3. Proposed Methodology

Daily silver price forecasts were generated using four procedures: data initialization, preprocessing, the forecasting model, and assessment. Investing.com was mined for its XAGg/TRY gram silver price data to kick off the data cleaning procedure. There are a total of 1645 trading days' worth of information here, spanning the previous six years (01.08.2017-29.09.2023). The Excel file format was used to get these records. The silver market makes use of the closing price and the starting price as defining criteria. A dataset of 1D array data frame type was created by reading and merging the excel files.

The Python programming language is used for the estimation process, and popular deep learning libraries such as numpy, pandas, keras, sklearn, tensorflow, and matplotlib are imported. This information will help readers understand how the algorithms are implemented and bring more transparency to the methodology. During preprocessing, a dataset of input pricing data is created by converting the silver data from the dataname type to the numpy type. In addition, MinMaxScaler (Equation 1) is used to normalize the dataset, making it possible to rapidly compute the value of all features by transforming them to a common scale. After data standardization, the data set was divided into two parts: a 70% training set and a 30% test set. In Vrigazova's (2021) research, it was suggested that splitting the data set in the ratio of 70/30 can further improve model performance. Accordingly, this ratio was preferred for data splitting. A specific random state (random_state) was used to split the dataset. The "random_state" is used to ensure that the dataset is randomly partitioned in the same way each time. This guarantees repeatability. In other words, the same data split is produced each time using the same "random_state" value. When testing hyperparameter settings or evaluating the performance of the model, this preserves the comparability of the findings. Given that the prediction performance of different models will be compared in this study, it is imperative to ensure that the results are comparable by using the same data split. This will make it possible to understand which model performs better. Throughout the optimization process, the evaluation dataset was used to compare the prediction errors introduced by the deep learning algorithm models. The research concludes by estimating and evaluating the performance of the model using the test dataset.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

The calculation of the CNN layer is the first step in the training process that underlies the prediction process, which makes use of a sliding window dataset. The data is processed via a convolution layer, then a pooling layer, in that order. After the input dataset has been processed by the CNN, it is sent to the LSTM layer, where input gates look for updated cell values. After the output gate has checked the information included in the cell, the data then travels via the forget gates, which examine the information contained in the cell to find constant values. Finally, the LSTM layer produces the data. The information then passes via a reset gate before entering the GRU layer, where it will be merged with previously received information. The information will next go via an update gate, which will calculate how much previous data must be kept. In addition, the GRU layer output data will be produced by this gate. The information will then go via each interconnected layer, culminating in a predicted value for one gram of silver.

Evaluating several approaches to forecasting the closing price of a gram of silver yields useful information. In this study, a technical evaluation is made by comparing the training and testing performance of traditional CNNs, LSTMs, GRUs, and a hybrid model consisting of all three. The error rate of the prediction outcomes derived from these models was computed using MAPE, RMSE, MAE, R^2, MASE, and SMAPE. Equations 2, 3, 4, 5, 6, and 7 are a few examples of formulas that may be used to compute this statistical data.

$$\mathrm{MAPE} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \frac{|\widehat{\mathrm{X}_\mathrm{i}} - \mathrm{X}_\mathrm{i}|}{\mathrm{X}_\mathrm{i}} \tag{2}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{n}} \sum_{\mathrm{i}}^{\mathrm{n}} (\mathrm{X}_\mathrm{i} - \widehat{\mathrm{X}_\mathrm{i}})^2} \tag{3}$$

$$\mathrm{MAE} = \frac{1}{\mathrm{n}} \sum_{\mathrm{i}}^{\mathrm{n}} \left|\mathrm{X}_\mathrm{i} - \widehat{\mathrm{X}_\mathrm{i}}\right| \tag{4}$$

$$\mathrm{R}^2 = 1 - \frac{\sum_\mathrm{i}(\mathrm{y}_\mathrm{i} - \hat{\mathrm{y}}_\mathrm{i})^2}{\sum_\mathrm{i}(\mathrm{y}_\mathrm{i} - \mu)^2} \tag{5}$$

$$\mathrm{MASE} = \frac{\sum_{t=1}^{n} |F_t - A_t|}{\frac{n}{n-1} \sum_{t=2}^{n} |A_t - A_{t-1}|} \tag{6}$$

$$SMAPE = \frac{100\%}{n} \sum_{t=1}^{n} \frac{|F_t - A_t|}{(|A_t| + |F_t|)/2} \tag{7}$$

The main motivation for using a combination of CNN, LSTM, and GRU algorithms in this study is the potential for stronger and more accurate forecasts by combining the unique advantages of each algorithm. "Hybrid Machine Learning Models for Gold and Silver Price Forecasting" by Goel et al. (2022) showed that hybrid models can outperform single models. However, in the work of Sulistio et al. (2023), the proposed hybrid model is a combination of CNN, LSTM, and GRU algorithms. This model was used in energy sector stock price forecasting and obtained superior results compared to the use of single algorithms. Moreover, the study also compared the combination of two hybrid models, CNN-GRU-LSTM and CNN-LSTM-GRU. As a result, the performance of the hybrid models is remarkable, especially the increase in R^2 values. This indicates that the hybrid models better capture the datasets and increase their predictive power. In particular, the CNN-LSTM-GRU hybrid model significantly reduced the MAE and RMSE values and increased the R^2 value. These findings provide an important reference for the construction of the CNN-LSTM-GRU hybrid model created in this study and support the methodology. The results of this study reinforce the results of this study.

The CNN-LSTM-GRU ranking used in this study is based on the potential to make more accurate predictions by combining the strengths of each algorithm. CNN captures spatial features of the data; LSTM captures long-term dependencies; and GRU learns functions similar to LSTM quickly with fewer parameters. The combination of these three enables in-depth analysis of the data in both time and feature dimensions. An alternative ranking, LSTM-GRU-CNN, could theoretically produce different results, but the ranking chosen in this study is the most appropriate because it can be thought of as prioritizing the power of CNN in feature extraction and then applying these features to time series analysis with LSTM and GRU. The effects of this ranking on performance are supported by the results in Sulistio et al. (2023) and are consistent with the results obtained in this study. In the next section of the paper, the models used for the forecasting process will be introduced.

However, the hybrid approach used in this study provides the necessary depth and flexibility for complex and volatile data sets, such as the silver market. Despite the limited number of observations, this approach aims to more effectively model the complex patterns and time-series characteristics of the data set. Hybrid modeling approaches offer a deeper and more flexible analysis, especially for complex and volatile market data sets. For example, the combination of CNN and GRU can combine the serial dependence of financial time series and the correlation properties of different financial market time series in the same model. This approach improves explanatory ability and forecasting accuracy by extracting short-term attention features and the long-term effects of time series data. Hybrid models have the potential to provide more accurate and reliable forecasts, regardless of the size of the data set (Song et al., 2023). Therefore, the use of the hybrid model was considered an appropriate and effective methodology for such data sets.

On the other hand, different hyperparameter optimization methods were used for all algorithms used in this study. The reason for not using a single optimization method for all algorithms is due to the need to optimize the different structural and functional properties of different algorithms. For example, the Bayesian optimization used in the study may be effective in certain situations, while for other algorithms, different optimization techniques (e.g., PSO or grid search) may be more appropriate. Using optimization methods that best meet the unique needs and structures of each algorithm plays a critical role in improving overall model performance and accuracy. The motivation for choosing Bayesian optimization for the CNN algorithm is important for two reasons. First, the time cost can be controlled and fixed according to the needs. Second, the balance of exploitation and exploration is well maintained compared to other existing models, making the search more effective (Zulfiqar et al., 2022). However, the manual selection of LSTM hyperparameters has a significant impact on the results. The prediction performance of models trained with different parameters varies greatly; therefore, it is very important to select appropriate model parameters (Xu et al., 2022). The particle swarm optimization method is used for the LSTM algorithm, and its success in time series forecasting is well known (Kumar et al., 2022; Pranolo et al., 2022; Xu et al., 2022). In addition, Brownlee (2020) examined the effectiveness of the Grid Search optimization method for the GRU algorithm in time series forecasting. The researcher described the development of a grid search test rig that can evaluate a set of hyperparameters for different neural network models. This method involves splitting time series data into training and test sets, developing models that can handle time-varying data, and summarizing performance through repeated evaluation. This information provides evidence to support the claim that the grid search optimization method for the GRU model is effective in time series forecasting. In line with all this information, separate optimization methods were chosen for each algorithm. Lastly, the training of all models is carried out on a laptop computer with Intel Core i7-7700HG CPU, NVIDIA GeForce RTX 3050 4 GB graphics card and 16 GB RAM.

3.1. Convolutional Neural Networks (CNN)

The CNN model is reportedly trained using the Adam variation of the stochastic gradient descent optimization method, as stated by Hsieh et al. (2020). When training an Adam neural network, its parameters are optimized by reducing the loss function, which is a measure of how much predictions deviate from the actual data. Equation (8) may be used to determine that CNN employs the cross-entropy loss function $\mathrm{J}(\mathrm{x},\mathrm{y},\theta)$. By reducing $\mathrm{J}(\mathrm{x},\mathrm{y},\theta)$ with regard to the set of network parameters, the network learns.

$$\mathrm{J}(\mathrm{x},\mathrm{y},\theta) = -\sum_{\mathrm{i}=1}^{\mathrm{K}} \mathrm{x_i} \log \mathrm{y_i}, \tag{8}$$

Given a minibatch size of K, let's say that x and y stand for the truth and the predicted CNN output, respectively. The CNN's parameter set, which is as follows, was acquired through iteratively updating backpropagation training:

$$\theta_\mathrm{t} \leftarrow \theta_{\mathrm{t}-1} - \frac{\eta \widehat{\mathrm{m}}_\mathrm{t}}{\sqrt{\theta_\mathrm{t}}+\varepsilon} \tag{9}$$

The values of $\widehat{\mathrm{m}}_\mathrm{t}$ and θ_t, which reflect the first-moment estimate with bias correction and the second-moment computation with bias adjustment, respectively, are very small constants.

In this study, the Bayesian optimization hyperparameter algorithm and the Trainbr training algorithm were used to create the 2D CNN model. The CNN model consists of convolutional layer, MaxPooling Layer, flatten layer, and dense layer.

The convolutional layer is a widely used layer, especially in image processing applications. This layer extracts features by making local connections to the input data. The convolution is done by scrolling over the entire input using a small window (kernel or filter). During each shift, an element-wise multiplication is performed between the data under the window and the filter, and the results are summed:

$$a_{ij}^{l} = \sigma\left(\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} W_{mn}^{l} \cdot x_{(i+m)(j+n)} + b^{l}\right) \tag{10}$$

Where, a_{ij}^{l} is an element of the activation matrix in layer l. W_{mn}^{l} is an element of the weight matrix in layer l. $x_{(i+m)(j+n)}$ is an element of the input matrix. b^{l} is the bias in layer l, σ is the activation function, and $M \times N$ is the kernel (filter) dimensions.

The MaxPooling layer is usually used after convolutional layers, and its purpose is to reduce the size of feature maps. This is done by taking the maximum value within a given window size. MaxPooling reduces the computational burden of the model while at the same time providing some resistance to overfitting. It also increases the robustness of the model to small displacements and transformations.

$$p_{ij} = max_{0 \le m < M, 0 \le n < N} x_{(i+m)(j+n)} \tag{11}$$

Where, p_{ij} is an element of the output of the pooling layer. $x_{(i+m)(j+n)}$ is an element of the input matrix. MxN are the pooling window dimensions.

Flattening converts a multidimensional input into a one-dimensional vector. For example, an input of size $m \times n \times k$ is transformed into a vector of size $m \times n \times k$.

A dense layer is an artificial neural network layer where each input is connected to each neuron of the layer. This layer combines information from previous layers to extract higher-level features. It is usually located in the final layers of neural networks and generates the outputs for tasks such as classification or regression. The dense layer multiplies the input vector by weights, adds bias, and passes it through an activation function. This process allows the network to learn complex functions.

$$a^{l} = \sigma(W^{l} \cdot a^{l-1} + b^{l}) \tag{12}$$

Where a^{l} is the activation vector in layer l and "wl" is the weight matrix. a^{l-1} is the activation vector of the previous layer. b^{l} is the bias vector in layer l. σ is the activation function.

Bayesian hyperparameter optimization is used in the CNN model. In Bayesian hyperparameter optimization, a prior distribution is first determined about the performance of the hyperparameters. This is an estimate of which values of the hyperparameters are likely to give good performance. The model is trained with the chosen set of hyperparameters, and the performance of the model (i.e., the error rate on the validation set) is measured. This new data point is added to the prior model, and the prior model is updated with this new observation. The updated model gives a probabilistic distribution of the performance of each combination of hyperparameters. This is known as the posterior distribution. Using the posterior distribution, the optimization process selects new sets of hyperparameters that can give the best result. This selection is usually done with Acquisition Functions (Doke et al., 2020).

The model is based on Gaussian Process (GP) regression:

$$f(x) \sim GP(m(x), k(x, x')) \tag{13}$$

Where $f(x)$ is the function that models the performance of the hyperparameters, $m(x)$ is the mean function, and $k(x, x')$ is the kernel function.

Flow Functions:

Expected Improvement (EI):

$$EI(x) = \mathrm{E}[\max(f(x) - f(x^{+}), 0)] \tag{14}$$

Where x is the new set of hyperparameters, $f(x)$ is the performance of this set and x^{+} is the best performance achieved so far. Bayesian optimization uses these mathematical models to decide which hyperparameters to try and thus efficiently explore the search space. This method is particularly advantageous when the hyperparameter search space is large and the cost of evaluation is high.

The CNN model is trained using the backpropagation algorithm. This algorithm includes forward propagation, error calculation and backpropagation stages. During forward propagation, the outputs are calculated for the neurons in each layer of the network. This is done by applying the activation functions of the neurons in each layer starting from the input:

$$a^l = \sigma(W^l a^{l-1} + b^l) \tag{15}$$

Here, a^l represents the activation at layer l and W^l represents the weight and bias. σ is the activation function. The error between the actual values in the training data set and the predicted values of the model is then calculated using an error function (loss function).

During back propagation, the error term is back propagated from the output to the input of the network and how to update the weights and biases in each layer is calculated.

Weight Update:

$$W^l = W^l - a\frac{\partial E}{\partial W^l} \tag{16}$$

$$b^l = b^l - a\frac{\partial E}{\partial b^l} \tag{17}$$

Here, a is the learning rate. $\frac{\partial E}{\partial W^l}$ and $a\frac{\partial E}{\partial b^l}$ are the derivatives of the error function with respect to weight and bias.

The error derivative is then calculated for each layer using the chain rule:

$$a\frac{\partial E}{\partial W^l} = \frac{\partial E}{\partial a^l}\cdot\frac{\partial a^l}{\partial Z^l}\cdot\frac{\partial Z^l}{\partial W^l} \tag{18}$$

Where $Z^l = W^l a^{l-1} + b^l$ is the input to the neurons in layer l.

The best hyperparameter settings obtained by the Bayesian Optimization hyperparameter algorithm in the proposed CNN model are shown in Table 1.

Table 1. Best Hyperparameters Determined by BO Algorithm in CNN Model

CNN (Bayesian Optimization)	Number of Layers: 1
	Number of Filters (for Convolutional Layers): 32
	Kernel Size: 3
	Activation Function: ReLU
	Pooling Size (for MaxPooling): 2
	Dropout Rate: 0.2
	Learning Rate: 0.000601
	Batch Size: 64
	Number of Epochs: 300

3.2. Long and Short Term Memory (LSTM)

LSTM, a relatively new kind of neural network design, has a stellar reputation for its ability to classify sequential data. An integral feature distinguishing LSTM from traditional neural networks is a memory gate that facilitates the retention of critical information. Additionally, LSTM incorporates a forgetting gate, allowing it to discard irrelevant data (Alshaikhdeeb ve Cheah, 2023: 546).

The LSTM mechanism's forget gate specifically functions to eliminate the cell state data from the preceding sequence. The time series' current input is denoted by x_t, while its previous hidden state is symbolized by h_{t-1}. Both of these values are processed by the activation function σ_g, resulting in the generation of the output vector f_t, which is linked to the forget gate. This relationship can be expressed using Equation (19), wherein the bias coefficient is denoted as b_f, the forget gates are represented as W_f and U_f, and the activation function is symbolized as σ_g.

$$f_t = \sigma_g\,(W_f x_t + U_f h_{t-1} + b_f) \tag{19}$$

The coefficients i_t and C'_t within this gate are calculated using the current data point in the time series input, denoted as x_t, together with the hidden state from the previous time step, denoted as h_{t-1}. Via the activation function, these coefficients are calculated. σ_g and σ_c stand for the acronyms for the activation function, whereas W_i, U_i, W_c, and U_c stand for the weight coefficients.

$$i_t = \sigma_g\,(W_i x_t + U_i h_{t-1} + b_i) \tag{20}$$

$$C'_t = \sigma_c\,(W_c x_t + U_c h_{t-1} + b_c) \tag{21}$$

The cell state, represented by the symbol C_t, is a component of Equation 22 update process. After the forget gate's output, f_t, is multiplied by the previous cell state, C_{t-1}, the cell candidate data, C'_t, is added to the input gate's output, i_t. The modified cell state, C_t, is described by this calculation.

$$C_t = f_t \text{ x } C_{t-1} + i_t \text{ x } C'_t \tag{22}$$

The creation of the output vector o_t is shown in equation (23) and is accomplished by applying the activation function σ_g to the input vectors h_{t-1} and x_t. W_o and U_o, the weight coefficients for the cell state, and the bias coefficient b_0 are related to the input gate.

$$o_t = \sigma_g\,(W_o x_t + U_o h_{t-1} + b_o) \tag{23}$$

$$h_t = o_t \text{ x } \tanh\,(C_t) \tag{24}$$

Following generation, the current sequential cell state C_t is multiplied by the output value o_t. Equation (24) shows how the activation function tanh generates the buried layer's output.

However, in this study, particle swarm optimization (PSO) hyperparameter optimization, which is referenced in the work of Cansu et al. (2023), was used in the LSTM model, which is the deep learning model used for the prediction process. The PSO method was used to find the best combination of hyperparameters. In PSO, the position and velocity of each particle are updated with the following formulas:

Speed Update:

$$v_{id}^{new} = w\cdot v_{id} + c_1\cdot r_1\cdot(pbest_{id} - x_{id}) + c_2\cdot r_2\cdot (gbest_d - x_{id}) \tag{25}$$

Where v_{id} is the velocity of the particle, w is the inertial weight, c_1 and c_2 are learning factors, r_1 and r_2 are random numbers, $pbest_{id}$ is the best position of the particle, $gbest_d$ is the global best position, and x_{id} is the current position.

Position Update:

$$x_{id}^{new} = x_{id} + v_{id}^{new} \tag{26}$$

In the proposed LSTM model, the Adam optimizer is used during training, and this optimizer updates the weights automatically. The Adam (Adaptive Moment Estimation) optimization algorithm is based on gradient descent and updates the weights more efficiently by using momentum and second moment (RMSprop) terms. Training algorithms commonly used in neural network training utilize gradient descent as the basis for weight updates. The Adam (Adaptive Moment Estimation) optimization algorithm builds on gradient descent and improves it by including momentum and second moment (similar to RMSprop) terms for more efficient weight updates. Here are the basic formulations of the Adam optimization algorithm:

Gradient Descent:

Gradient descent updates weights using the gradient. The gradient of the network's loss function is computed, and this gradient is used to update the weights.

$$\theta_{new} = \theta_{old} - a \times \nabla L(\theta_{old}) \tag{27}$$

Where θ_{new} is the new weight value, θ_{old} is the current weight value, a is the learning rate, and $\nabla L(\theta_{old})$ is the gradient of the loss function in the current weights.

Momentum:

Momentum speeds up weight updates by adding a momentum term that considers the previous updates of the gradient.

$$v_{new} = \beta \times v_{old} + (1-\beta) \times \nabla L(\theta_{old}) \tag{28}$$

Where v_{new} is the new momentum value, v_{old} is the current momentum value, β is the momentum term (a value between 0 and 1).

RMSprop adaptively adjusts weight updates by taking the average of the squares of the gradients. This approach automatically adjusts the learning rate based on the magnitude of the gradient values.

$$S_{new} = \gamma \times S_{old} + (1-\gamma) \times (\,\nabla L(\theta\_old\,))^2 \tag{29}$$

Here, S_{new} is the new root mean square momentum value, S_{old} is the current root mean square momentum value, and γ is the RMSprop term (a value between 0 and 1). However, Adam combines momentum and RMSprop terms for weight updates.

Momentum Update:

$$m_{new} = \beta_1 \times m_{old} + (1-\beta_1) \times \nabla L(\theta_{old}) \tag{30}$$

RMSprop Update:

$$S_{new} = \beta_2 \times S_{old} + (1-\beta_2) \times \left(\nabla L(\theta_{old})\right)^2 \tag{31}$$

Bias correction fort the first moment:

$$\widehat{m} = \frac{m_{new}}{1-\beta_1^t} \tag{32}$$

Bias correction fort the second moment:

$$\hat{S} = \frac{S_{new}}{1-\beta_2^t} \tag{33}$$

Weight Update:

$$\theta_{new} = \theta_{old} - a \times \frac{\widehat{m}}{\sqrt{\hat{S}}+\varepsilon} \tag{34}$$

Where, θ_{new} New weight value, a Learning rate, t Update step, β_1, β_2 Momentum terms for the first and second moments respectively, ε a small value for numerical stability. Adam is an effective optimization algorithm for training large and complex neural networks, automatically adjusting the learning rate (Cansu et al., 2023). The best hyperparameter settings obtained by the PSO hyperparameter algorithm in the proposed LSTM model are shown in Table 2.

Table 2. Best Hyperparameters Determined by PSO Algorithm in LSTM Model

LSTM (Particle Swarm Optimization)	Number of Layers:1
	Number of First Layer Neurons:100
	Activation Function: Hyperbolic Tangent
	Batch Size: 64
	Learning Rate: 0.0013
	Epoch Value: 500

3.3. Gated Recurrent Unit (GRU)

Originally created by Cho et al. (2014), the GRU is a variant of the Recurrent Neural Network (RNN). Long-range input is difficult for recurrent neural networks (RNNs) to capture and process correctly; this problem is solved by adding a gating component. GRU just contains the update gate (z_t) and reset gate (r_t), but LSTM has a more complex structure. The update gate (or input gate) of a Gated Recurrent Unit (GRU) is critical because it decides what fraction of the current input (x_t) and previous output (h_{t-1}) should be sent on to the next cell.

In contrast, the reset gate determines how much weight should be given to previously acquired data. Based on the weight W, the information currently stored in memory may be used to send just the necessary details to the next iteration. By solving equations 35 and 36, the primary operations of the Gated Recurrent Unit (GRU) are defined.

Update Gate:

$$z_t = \sigma(W_z * [h_{t-1}, x_t)] \tag{35}$$

Reset Gate:

$$r_t = \sigma(W_r * [h_{t-1}, x_t)] \tag{36}$$

GRU is particularly notable for its ability to retain information for a long time, especially in long sequence data, and to learn temporal connections. These features make GRU a popular choice for language modeling, text generation, time series analysis, and more (Ayzel and Heistermann, 2021). In addition, grid search is a method for tuning model hyperparameters in machine learning. This method aims to find the parameter combinations that will make the model perform best by systematically trying all possible combinations within a given set of hyperparameters. Especially in complex models, choosing the right hyperparameters can greatly affect the performance of the model (Buslim et al., 2021). In this study, this method was preferred to provide the best hyperparameter settings for the GRU model. On the other hand, Adam Optimizer was used as the training algorithm in this model as in the LSTM model, and the LSTM process was applied in this model as well.

In this model created with Grid Research hyperparameter optimization, there are n different hyperparameters to be optimized:

$$H = \{h_1, h_2, \ldots, h_n\} \tag{37}$$

For each hyperparameter h_i, let D_i be the range of values or set of candidate values. For all h_i, all possible combinations over D_i are generated. The combinations are expressed by the product set "$D_1 \times D_2 \times ... \times D_n$".

If $D_1 = \{d_{11}, d_{12}\}$ and $D_2 = \{d_{21}, d_{22}\}$, the combinations to be formed are $\{(d_{11}, d_{21}), (d_{11}, d_{22}), (d_{12}, d_{21}), (d_{12}, d_{22})\}$. Then, for each combination C, the model is trained with hyperparameters C and its performance is evaluated by $P(C)$. The best of the performance values obtained for all combinations is selected:

Max performance:

$$C^* = \arg max_C P(C) \tag{38}$$

Or minumum performance:

$$C^* = \arg min_C P(C) \tag{39}$$

Finally train the model with the best found hyperparameter combination C^*. The basic logic of Grid Search is to find the best performing combination by trying all possible combinations of the specified hyperparameters. This process can be computationally intensive as it usually involves a large number of combinations. The formulas provide a mathematical representation of this process. The best hyperparameter settings obtained by the Grid Search hyperparameter algorithm in the proposed GRU model are shown in Table 3.

Table 3. Best Hyperparameters Determined by Grid Search Algorithm in GRU Model

GRU (Grid Search Optimization)	Number of Layers:1 Number of First Layer Neurons:80 Activation Function: Sigmoid Batch Size: 64 Learning Rate: 0.0088 Epoch Value: 500

3.4. Hybrid CNN-LSTM-GRU Model

The research approach was followed in the initialization and pre-processing of the data to produce sliding window data consisting of training, assessment, and testing datasets. Furthermore, the architecture and hyperparameters used in this study are applied to the prediction process of this dataset. The architecture of the CNN-LSTM-GRU model and the hyperparameter settings used in this study are based on the work of Sulistio et al. (2023). The CNN-LSTM-GRU model used in this study is composed of two convolutional layers, two pooling layers, one LSTM layer, one GRU layer, one smoothing layer, three dense layers, and two dropout layers. The GRU layer has 192 neurons, whereas the LSTM layer consists of 128 neurons. One of the improved hyperparameters in this study is the number of layers. In the layer, the relay function serves as the activation function. Because there is just one possible output value—the estimated value of kilos of silver—the thick layer consists of a single neuron. The layer has a linear activation function. This function is put to use due of the reliability with which it makes predictions.

To create a mathematical representation of this architecture, we need to detail the operations and transformations at each layer, as they process the input data. Here is a breakdown:

Input Data: Let's assume the input data is a sequence of vectors, each of size D.

Table 4. Settings for the Hybrid CNN-LSTM-GRU Model Hyperparameters

Layer (type)	Output Shape
Conv1D	(None, 15, 64)
MaxPooling1D	(None, 14, 64)
Conv1D	(None, 14, 32)
MaxPooling 1D	(None, 13, 32)
LSTM	(None, 13, 128)
GRU	(None, 13, 192)
Flatten	(None, 2496)
Dense	(None, 128)
Dropout	(None, 128)
dense_1 (Dense)	(None, 32)
dropout_1 (Dropout)	(None, 32)
dense_2 (Dense)	(None, 1)

However, the architecture of the CNN-LSTM-GRU hybrid model used is shown in Figure 1.

Conv1D Layers: These layers apply convolution operations. The first Conv1D layer transforms the input data to a shape of (15, 64), and the second Conv1D layer further processes it to a shape of (14, 32). The mathematical operation for a convolutional layer can be represented as:

$$\mathrm{Conv1D}(x) = \mathrm{ReLU}(\mathrm{W} \cdot x + b) \tag{40}$$

where W and b are the weights and biases of the convolutional filters, $*$ denotes the convolution operation, and ReLU is the activation function.

MaxPooling1D Layers: These reduce the dimensionality of the data after each Conv1D layer. The operation is:

$$\mathrm{MaxPooling1D}(x) = \max\,(window\ segments\ of\ x) \tag{41}$$

The LSTM layer processes the sequence data and has 128 neurons. Its mathematical operation involves a complex interaction of gates (input, output, and forget) and can be represented as:

$$\mathrm{LSTM} = (x_t, h_{t-1}, c_{t-1}) = h_t, c_t \tag{42}$$

where x_t is the input at time t, h_{t-1} and c_{t-1} are the previous hidden state and cell state, respectively, and h_t and c_t are the current hidden state and cell state.

The GRU Layer is similar to the LSTM but slightly simpler with 192 neurons. Its operation is as follows:

$$GRU = (x_t, h_{t-1}) = h_t \tag{43}$$

Flatten layer, converts the output from the GRU layer into a flat vector. However, Dense layers are fully connected layers. The first has 128 neurons, the second 32, and the third just 1 neuron, representing the estimated value of kilos of silver. The mathematical representation is:

$$Dense(x) = \sigma(Wx + b) \tag{44}$$

where W and b are the layer's weights and biases, respectively, and σ is either a ReLU or a linear activation function. On the other hand, Dropout layers randomly set a fraction of the input units to 0 at each update during training, which helps prevent overfitting. The dropout operation is not directly mathematical but is an operation applied during training. The last layer Output produces a single value, the estimated value of silver. The CNN-LSTM-GRU was run on a training dataset after the hyperparameter adjustments given in Table 4 were made. In addition, the model developed via this algorithmic method was utilized to make predictions on the test dataset.

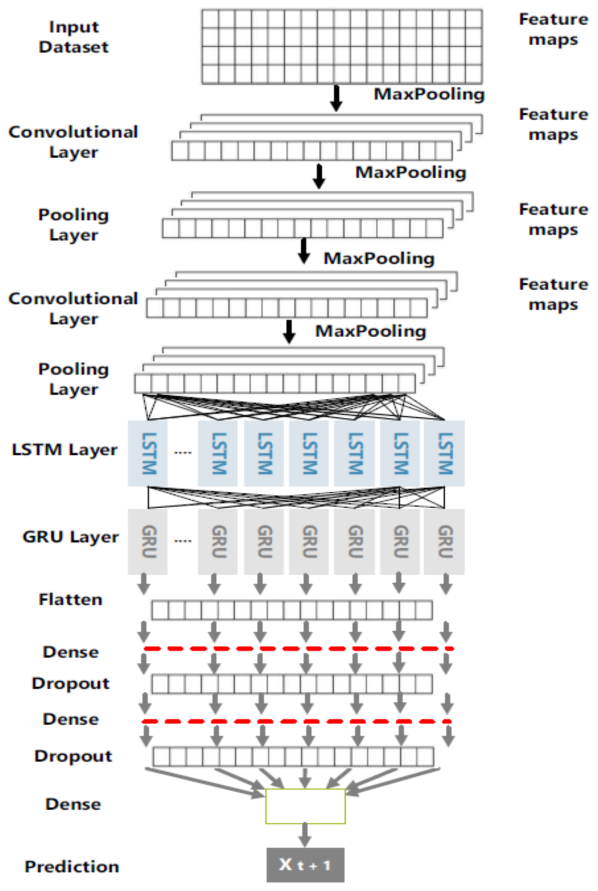

Figure 1. The Architecture of Rhe CNN-LSTM-GRU Hybrid Model

4. Results and Discussion

In this study, LSTM, CNN, GRU, and a new hybrid model, CNN-LSTM-GRU, were used to predict the XAGg/TRY-Gram silver price. Data from the website investing.com was gathered for the research. The data set has a time interval of 1645 days, starting on August 1, 2017 and ending on September 29, 2023. The silver price data specifications used in the study are the closing price and the opening price. A total of five different error statistics were used in the study. These error coefficients are RMSE, MAE, MAPE, and R^2. Each model was developed, trained, and then evaluated using a specific data set. The training performances of the models are presented in Table 5.

Table 5. Training Results of CNN, LSTM, GRU and Hybrid CNN-LSTM-GRU Models

Model	RMSE	MAE	MAPE	R^2	MASE	SMAPE
CNN	1.4780	8.3435%	0.0200	0.8983	1.6550	2.1172
LSTM	0.9120	5.6782%	0.0220	0.9014	1.1743	1.9510
GRU	2.077	11.3543%	0.0351	0.8651	2.2781	2.9783
CNN-LSTM-GRU	0.1356	1.5011%	0.0189	0.9890	0.9790	0.9520

According to these training results, the hybrid CNN-LSTM-GRU model performed significantly better than the other models. With the lowest RMSE (root mean square error), MAE (mean absolute error), and MAPE (mean absolute percentage error) values, this model predicts the data most accurately. Moreover, this model with the highest R^2 (R-squared) value shows that the variance of the data is best explained. On the other hand, the CNN and LSTM models performed moderately, while the GRU model showed the lowest performance on these measures. MASE (mean absolute scaled error) and SMAPE (symmetric mean absolute percentage error) values also show that the hybrid model outperforms the other models. These results suggest that the hybrid model is more effective in modeling complex data sets. However, Table 6 shows the error coefficients calculated from the test results of the models.

Table 6. Test Results of CNN, LSTM, GRU, and Hybrid CNN-LSTM-GRU Models

Model	RMSE	MAE	MAPE	R^2	MASE	SMAPE
CNN	1.3255	6.1482%	0.0191	0.9092	1.7124	2.0564
LSTM	0.9673	4.3551%	0.0174	0.9132	1.1678	1.8434
GRU	2.1776	10.2953%	0.0292	0.8713	2.1743	2.8692
CNN-LSTM-GRU	0.1089	1.4789%	0.0170	0.9913	0.9846	0.9678

When compared to the other three prediction techniques (CNN, LSTM, and GRU), the CNN-LSTM-GRU strategy was shown to have the greatest predicted value matching rate and to be closest to the true value. The better the forecast, the lower the MAE number should be. The more precise a prediction is, the lower the RMSE number should be. Values of R-squared (R^2) might be anything from zero to one. Higher accuracy in predictions is shown by reduced margins of error (MAE) and RMSE (the difference between the expected and actual values). When R^2 is near to 1, it indicates that the values are very close to each other (Sulistio et al., 2023: 180-181).

On the other hand, Figure 2 displays the empirical test results for each of the models. The dashed blue line displays the actual closing price of silver for each of the forecasting models, while the red line indicates the projected value for each model.

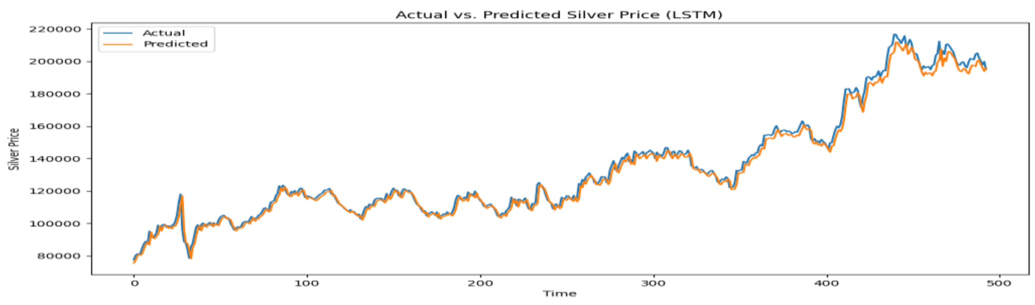

Figure 2(a). Test Outcomes of the LSTM Model

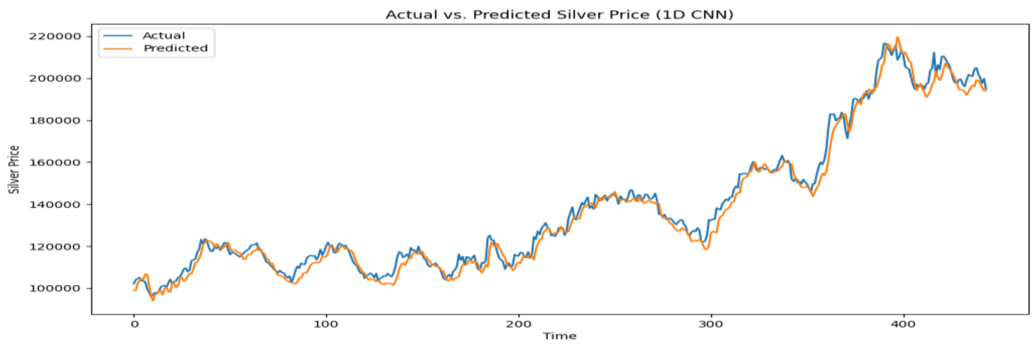

Figure 2(b). Test Outcomes of the CNN Model

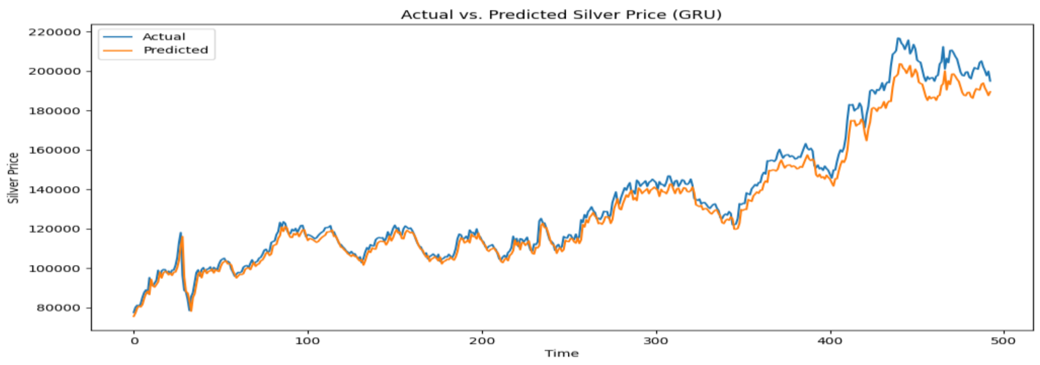

Figure 2(c). Test Outcomes of the GRU Model

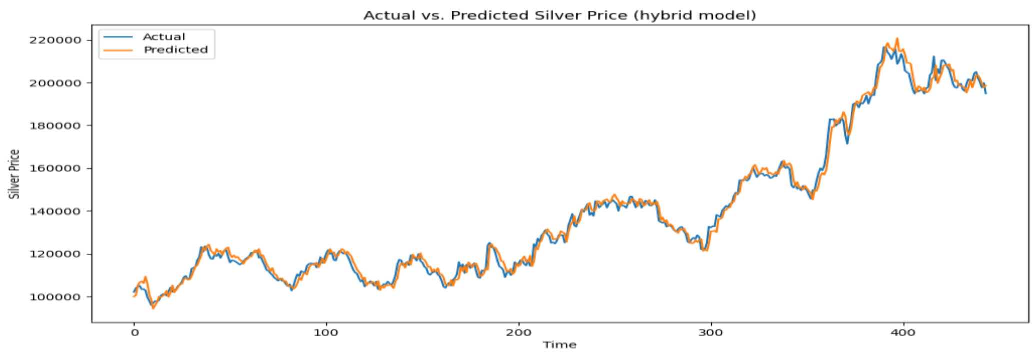

Figure 2(d). Test Outcomes of the Hybrid CNN-LSTM-GRU Model

In order to estimate the closing price of silver, four algorithms were used in the experiment, and the results were compared to determine the real value that is shown in Table 2 and Figure 1. Compared to CNN or GRU algorithms, the LSTM method that yields the lowest MAE value (4.3551%) and the best RMSE value (0.9673%) is the one that is used exclusively. An R^2 score of 0.91, or almost 1, is obtained for LSTM. When compared to the application of the CNN and GRU algorithms, this value is likewise the best. LSTM performs rather well on its own, but it can perform much better when combined with other algorithms to create a hybrid algorithm.

The CNN-LSTM-GRU hybrid technique yields the lowest and most accurate results, with mean absolute error (MAE) and root mean squared error (RMSE) values of 1.4789 and 0.1089, respectively, and an R^2 value of 0.9991, which is extremely close to 1. When comparing the CNN-LSTM-GRU hybrid algorithm to the LSTM method, we find that the MAE, RMSE, and MAPE values decrease while the R^2 value increases. These numbers demonstrate that using this hybrid approach is more beneficial than using other techniques.

In this study, MASE and SMAPE error calculations were also performed. MASE scales the forecast errors relative to the errors of a simple time series model. A MASE less than 1 indicates that the model outperforms a simple, naive forecasting model. This metric is often used in time series forecasting and is useful for comparing model performance with naive approaches. On the other hand, SMAPE expresses as a percentage how close the forecasts are to the true values. Values close to 0 indicate better forecasts. This metric is especially used in time series forecasting and financial analysis (Sbrana and Silvestrini, 2022). The MASE value of the CNN model is 1.7124. This indicates that the model performs slightly better than a simple, naive forecast, but there is still room for improvement. In the LSTM model, this value is 1.1678. The LSTM model outperforms the CNN model and has significantly better results than a naive forecast. In the GRU model, the MASE value is 2.1743. This indicates that the GRU model performs worse than a naive forecast, especially compared to the other models. On the other hand, in the CNN-LSTM-GRU model, the MASE value is 0.9846. This shows that the hybrid model performs almost the same as a naive forecast and performs the best relative to the other three models.

However, the SMAPE value for the CNN model is 2.0564%, indicating that the closeness of the model's predictions to the actual values is moderate. In the LSTM model, the SMAPE value is 1.8434%. The LSTM model shows better closeness than CNN. In the GRU model, the SMAPE value is 2.8692%. This shows that the GRU model makes the farthest predictions from the true values. Finally, the SMAPE value of the CNN-LSTM-GRU model is 0.9678%. This indicates that the predictions of the hybrid model are closest to the true values and perform the best at this scale. In general, the CNN-LSTM-GRU hybrid model outperformed the other three models in terms of both MASE and SMAPE. On the other hand, while the LSTM model also gives good results, the performance of the GRU model is found to be the lowest in these metrics.

The CNN-LSTM-GRU hybrid algorithm outperforms conventional forecasting methods that depend only on the CNN or LSTM algorithm when it comes to predicting the closing price. For forecasting purposes, the CNN-LSTM-GRU hybrid algorithm achieves lower MAE, RMSE, and MAPE values than LSTM alone. A hybrid algorithm yields more accuracy in silver value estimate than a single algorithm, according to test findings. It has also been shown that using the hybrid CNN-LSTM-GRU algorithm enhances closing silver price predicting outcomes and lowers error values when compared to other methods.

However, when the training and test results are compared, it is seen that the hybrid CNN-LSTM-GRU model has a clear advantage over the other models in both training and testing. The lowest RMSE, MAE, and MAPE values in both training and testing indicate that the model predicts the data accurately. The high R^2 value indicates that the model explains a large portion of the variance in the data set. Among the other models (CNN, LSTM, and GRU), the LSTM model performs better in both cases. The GRU model, on the other hand, performs the worst in both training and testing. These results show that the hybrid model is neither overfitting nor underfitting, so the model is overall efficient and balanced. In addition, comparing the training and testing results for the other models, the LSTM model performs well in both training and testing. The RMSE, MAE, and MAPE values are relatively low, and the R^2 value is high. The CNN model also performs reasonably well, while the GRU model has the highest error rates and the lowest R^2 values in both cases. This suggests that the GRU model is the least suitable for this dataset. Considering the performances of LSTM and CNN, it shows that although both of them perform lower than the hybrid model, they are compatible with the dataset and do not suffer from overfitting or underfitting problems.

As mentioned in the previous literature review, Sulistio et al. (2023) used six different deep learning algorithms for financial data prediction. When the results were analyzed, the CNN-LSTM-GRU hybrid algorithm performed better than the other methods. In addition, MSE decreased by 14%, MAE decreased by 13%, and R^2 increased by 3.9%. When these results are compared with the results of this study, the superior performance of the CNN-LSTM-GRU hybrid model is observed in both studies. On the other hand, in both studies, model performances were evaluated using metrics such as RMSE, MAE, and R^2. These metrics showed improvement in both studies. The parallelism of the results of the two studies strongly supports the potential of deep learning algorithms in this field.

However, comparing this study with the study by Goel et al. (2022) will provide different perspectives on the effectiveness of machine learning models in financial forecasting. Both studies used machine learning models to predict the prices of precious metals in particular, but the studies differ in terms of different models, data sets, and measurement metrics. Both studies focused on precious metals price forecasting, but for different markets (Indian market vs. Turkish market) and metals (gold and silver). In Goel et al.'s study, the RNN model showed high accuracy only for gold, while in this study, specific RNN structures were found to perform better for silver prediction.

Some enhancements and suggestions may be taken into consideration for next research on silver price forecasting, based on the findings of this study. First off, expanding access to a wider range of data sources might increase prediction accuracy. Furthermore, the prediction performance may be enhanced by the use of stronger deep learning algorithms. Investing may benefit from analyzing long-term silver price expectations. A more thorough investigation of the variables influencing silver pricing need to be conducted using factor analysis. In conclusion, it is worthwhile to contemplate the use of these projections in enhancing risk mitigation tactics. The study's relevance lies in the fact that industrial and investment users depend on precise silver price forecasts, and future research should focus on expanding knowledge and expertise in this field.

References

Alshaikhdeeb, A. J. & Cheah, Y. N. (2023). Utilizing Word Index Approach with LSTM Architecture for Extracting Adverse Drug Reaction from Medical Reviews. Journal of Advances in Information Technology, 14(3).

Ayzel, G., & Heistermann, M. (2021). The effect of calibration data length on the performance of a conceptual hydrological model versus LSTM and GRU: A case study for six basins from the CAMELS dataset. Computers & Geosciences, 149, 104708.

Brownlee, J. (2020), How to Grid Search Deep Learning Models for Time Series Forecasting, https://machinelearningmastery.com/how-to-grid-search-deep-learning-models-for-time-series-forecasting/ Access Date: 18.12.2023

Buslim, N., Rahmatullah, I. L., Setyawan, B. A., & Alamsyah, A. (2021, September). Comparing Bitcoin's Prediction Model Using GRU, RNN, and LSTM by Hyperparameter Optimization Grid Search and Random Search. In 2021 9th International Conference on Cyber and IT Service Management (CITSM) (pp. 1-6). IEEE.

Cansu, T., Kolemen, E., Karahasan, Ö., Bas, E., & Egrioglu, E. (2023). A new training algorithm for long short-term memory artificial neural network based on particle swarm optimization. Granular Computing, 1-14.

Chen, J. (2023). Analysis of bitcoin price prediction using machine learning. Journal of Risk and Financial Management, 16(1), 51.

Cho K., Van Merrienboer B., Gulcehre C.et al., (2014), Learning phrase representations using RNN encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078, 2014.

Doke, P., Shrivastava, D., Pan, C., Zhou, Q., & Zhang, Y. D. (2020). Using CNN with Bayesian optimization to identify cerebral micro-bleeds. Machine Vision and Applications, 31, 1-14.

Gao, Y., Wang, R. & Zhou, E. (2021). Stock Prediction Based on Optimized LSTM and GRU Models. Scientific Programming, 2021. https://doi.org/10.1155/2021/4055281

Goel, S., Saxena, M., Sarangi, P. K. & Rani, L. (2022). 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.

Hamayel, M. J. & Owda, A. Y. (2021). A novel cryptocurrency price prediction model using GRU, LSTM and bi-LSTM machine learning algorithms. AI, 2(4), 477-496.

Hsieh, C. H., Li, Y. S., Hwang, B. J. & Hsiao, C. H. (2020). Detection of atrial fibrillation using 1D convolutional neural network. Sensors, 20(7), 2136.

Investing.com Data: https://tr.investing.com/currencies/xagg-try-historical-data Access Date: 30.09.2023.

Kong, D., Liu, S. & Pan, L. (2021). Amazon spot instance price prediction with GRU network. In 2021 IEEE 24th international conference on computer supported cooperative work in design (CSCWD), (pp. 31-36). IEEE.

Kumar, G., Singh, U. P., & Jain, S. (2022). An adaptive particle swarm optimization-based hybrid long short-term memory model for stock price time series forecasting. *Soft Computing*, *26*(22), 12115-12135.

Lara-Benítez, P., Carranza-García, M., & Riquelme, J. C. (2021). An experimental review on deep learning architectures for time series forecasting. *International journal of neural systems*, *31*(03), 2130001.

Li, F., Zhou, H., Liu, M. & Ding, L. (2023). A Medium to Long-term Multi-influencing Factor Copper Price Prediction Method Based on CNN-LSTM. IEEE Access, (99), 1-1.

Lin, Y., Liao, Q., Lin, Z., Tan, B. & Yu, Y. (2022). A novel hybrid model integrating modified ensemble empirical mode decomposition and LSTM neural network for multi-step precious metal prices prediction. Resources Policy, 78, 102884.

Malik, A., Gupta, P. & Vijh, S. (2022). Towards a Stock Price Prediction on Time Series Data using Long-Short Term Memory Method. In 2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence), (pp. 525-529). IEEE.

Patel, N. P., Parekh, R., Thakkar, N., Gupta, R., Tanwar, S., Sharma, G., ... & Sharma, R. (2022). Fusion in cryptocurrency price prediction: A decade survey on recent advancements, architecture, and potential future directions. IEEE Access, 10, 34511-34538.

Pranolo, A., Mao, Y., Wibawa, A. P., Utama, A. B. P., & Dwiyanto, F. A. (2022). Robust LSTM With tuned-PSO and bifold-attention mechanism for analyzing multivariate time-series. *IEEE Access*, *10*, 78423-78434.

Rao, B. S., Bhattacharya, R., Tiwari, M. K., Kumari, K. A., Devmane, M. A. & Singh, K. (2023). Innovative Deep Learning Model-based Stock Price Prediction using a Hybrid Approach of CNN and Gradient Recurrent Unit. In 2023 8th International Conference on Communication and Electronics Systems (ICCES), (pp. 1304-1309). IEEE.

Sbrana, G., & Silvestrini, A. (2022). Random coefficient state-space model: Estimation and performance in M3–M4 competitions. International Journal of Forecasting, 38(1), 352-366.

Sulistio, B., Warnars, H. L. H. S., Gaol, F. L. & Soewito, B. (2023). Energy Sector Stock Price Prediction Using The CNN, GRU & LSTM Hybrid Algorithm. In 2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE), (pp. 178-182). IEEE.

Vidya, G. S. & Hari, V. S. (2020). Gold price prediction and modelling using deep learning techniques. In 2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS), (pp. 28-31). IEEE.

Vrigazova, B. (2021), "The Proportion for Splitting Data into Training and Test Set for the Bootstrap in Classification Problems", Business Systems Research, 12(1):228-242. DOI: https://doi.org/10.2478/bsrj-2021-0015

Wang, H., Dai, B., Li, X., Yu, N. & Wang, J. (2023). A Novel Hybrid Model of CNN-SA-NGU for Silver Closing Price Prediction. Processes, 11(3), 862.

Wibawa, A. P., Utama, A. B. P., Elmunsyah, H., Pujianto, U., Dwiyanto, F. A., & Hernandez, L. (2022). Time-series analysis with smoothed Convolutional Neural Network. *Journal of big Data*, *9*(1), 44.

Xu, Y., Hu, C., Wu, Q., Jian, S., Li, Z., Chen, Y., ... & Wang, S. (2022). Research on particle swarm optimization in LSTM neural networks for rainfall-runoff simulation. *Journal of hydrology*, *608*, 127553.

Yang, J., De Montigny, D. & Treleaven, P. (2022, May). ANN, LSTM, and SVR for gold price forecasting. In 2022 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr), (pp. 1-7). IEEE.

Zulfiqar, M., Gamage, K. A., Kamran, M., & Rasheed, M. B. (2022). Hyperparameter optimization of bayesian neural network using bayesian optimization and intelligent feature engineering for load forecasting. *Sensors*, *22*(12), 4446.