Acta Montanistica Slovaca
ISSN 1335-1788
actamont.tuke.sk

Optimizing Precious Metal Price Forecasting with Hybrid Deep Learning Models: An Operational Research Perspective

Harinarayanan KAYATHINGAL[1]**, *Marek VOCHOZKA*[2] and *Zuzana ROWLAND*[3]

Authors' affiliations and addresses:
[1] Institute of Technology and Business in České Budějovice, Okružní 517/10, 37001 České Budějovice, Czech Republic
e-mail: hari@mail.vstecb.cz

[2] Institute of Technology and Business in České Budějovice, Okružní 517/10, 37001 České Budějovice, Czech Republic
e-mail: vochozka@mail.vstecb.cz

[3] Institute of Technology and Business in České Budějovice, Okružní 517/10, 37001 České Budějovice, Czech Republic
e-mail: rowland@mail.vstecb.cz

***Correspondence:**
Harinarayanan Kayathingal, Institute of Technology and Business in České Budějovice, Okružní 517/10, 37001 České Budějovice, Czech Republic
tel.: +420 776 739 022
e-mail: hari@mail.vstecb.cz

Funding information:
Institute of Technology and Business in České Budějovice

How to cite this article:
Kayathingal, H., Vochozka, M. and Rowland, Z. (2025), Optimizing Precious Metal Price Forecasting with Hybrid Deep Learning Models: An Operational Research Perspective, *Acta Monstanistica Slovaca*, Volume 30 (1), 1-16

DOI:
https://doi.org/10.46544/AMS.v30i1.01

Abstract
In the current study, Wolfram Mathematica is employed to study the application of seven deep neural network architectures in predicting precious metal prices: LSTM, CNN, MLP, MLP-CNN, MLP-LSTM, LSTM-CNN, and LSTM-CNN-LSTM. We will utilize a daily price series over ten years for gold, silver, and platinum. Model performances were evaluated using RMSE, MAPE, and RMSE metrics. Results illustrate the strong performance of all neural network models as such techniques will capture long-term dependencies for better decision-making on the commodity markets. As compared to the individual neural networks, the hybrid neural network show superior performance. The LSTM-CNN-MLP model is a very dependable and robust solution for precious metals forecasting.This will clearly shows that NN plays a major role while predicting the time series data. These results contribute to the wealth of literature in operational research with improved predictive accuracy in financial systems and portfolio optimization strategies.

Keywords
Correlation, Neural Network, Prediction, LSTM, CNN, MLP, Gold, Silver, Palladium

Introduction

Precious metals, like gold, silver, and platinum (Mensi et al., 2021), have been of interest since antiquity and have always been critical assets for investors (Wang et al., 2019) and policymakers (Dhifaoui et al., 2022) because they provide hedges against inflation (Adekoya et al., 2021) and economic uncertainty (Mokni et al., 2021). Accurate forecasting of metal prices is the precondition for optimal and sustainable planning of mine production (Gligorić et al., 2020), a crucial role in the finance and industrial sectors (Lin et al., 2022). However, the nonlinear nature, large fluctuation, and aperiodic cycle of precious metals make their price forecasting a very difficult task (Foroutan & Lahmiri, 2024). Even in fluctuating situations, policymakers in commodity markets can take advantage of the superior performance of DL models in commodity price forecasting (Ben Ameur et al., 2023).

Some of the properties that are usually associated with precious metals include safe havens, hedges against inflation, and diversification benefits for risk-asset portfolios (Wei et al., 2023). Precious metals are used in a wide variety of industries due to their excellent corrosion resistance, electrical conductivity, and catalytic activity (Ding et al., 2019) . Precious metals are strategically important to a country's economic growth (Foroutan & Lahmiri, 2024). There exist statistically significant long-run and short-run relationships between economic growth and each of the precious metal types under consideration (Bildirici & Gokmenoglu, 2020). Kucher & McCoskey, (2017) found that Economic conditions strongly influence the long-term relationships between the prices of precious metals. These metals are more important to the stock of the precious rather than the industrial metals. Besides, this effect is also stronger for those firms that are domiciled in developed markets (Lazzarino et al., 2022).

This paper intends to implement and compare seven neural network architectures to forecast the prices of some precious metals using Wolfram Mathematica's advanced computational capabilities. In doing so, it aligns with the operational research aim of better decision-making by computational models and seeks to bridge the gap between theoretical advances and practical applications.

RQ1. How do the unique architectures and hybrid configurations of neural network models (MLP, LSTM, CNN, CNN-LSTM, CNN-MLP, MLP-LSTM, CNN-MLP-LSTM) affect the accuracy, robustness, and computational efficiency of precious metal price predictions?

Literature Review

It is challenging to develop accurate forecasting systems in practical applications (Santos Júnior et al., 2019). Researchers employed various predictive models, including statistical (Lauret et al., 2017; Ngadono & Ikatrinasari, 2018; S et al., 2023; Yıldıran & Fettahoğlu, 2017), machine learning (Derakhshani et al., 2024; Nabipour et al., 2020; Rostamian & O'Hara, 2022; Zhang et al., 2023), and econometric models, to ensure robust and accurate predictions.

The statistical prediction models hold a few significant predictors, implying their informative power is limited. The predictive statistical models also provide pseudo-correct regular statistical patterns, and while doing so, no previous knowledge of the causality of the data used is known (Grebovic et al., 2022). Traditional time series models, such as ARIMA and similar models, have exhibited shortcomings in capturing intricate and nonlinear patterns (Si et al., 2024**).** Adebiyi et al., (2014) examines ARIMA and neural networks for NYSE stock forecasting, confirming the superiority of neural networks and explaining prior conflicting findings. Neural networks also show the propensity for the identification of patterns present in the data, which an autoregressive and moving average model fails to detect (Hansen et al., 1999). While machine and deep learning models like variants of LSTM, SVR, and XGBoost efficiently predict metal futures, their results significantly vary concerning metal type, sample period, and length of input duration, pointing toward difficulties in constructing any unified theory of prediction (Varshini et al., 2024) .

Deep learning models represent a new learning paradigm in machine learning and AI (Emmert-Streib et al., 2020). It's interesting to note that neural network data analysis improves accuracy, processing speed, fault tolerance, latency, performance, volume, and scalability (Abiodun et al., 2018). In 90 years, since their official introduction in 1943, neural networks have made great progress. Due to the wide application and enormous research and development potential of this technology, more and more scientific and technological workers are joining in the study of neural networks (Pan, 2024).

Foroutan & Lahmiri, (2024) use deep learning systems for forecasting the prices of crude oil and precious metals. Sadorsky, (2021) study applies several machine learning tree-based classifiers, including random forests, stochastic gradient boosting, and bagging, to predict the price direction of gold and silver exchange-traded funds. Chen & Zhang, (2019) developed a combination forecasting model that uses a projection pursuit algorithm to select the influential factors and a BP neural network to learn and predict gold prices with better accuracy and efficiency. Oner & Oner, (2022) investigated which of the symmetric and asymmetric models of the ARCH family will be the best model for forecasting the volatility of the prices of precious metals during the COVID-19 pandemic.

In addition to technological forecasting approaches, a growing body of literature emphasizes the importance of intelligent decision-making systems and hybrid fuzzy models, particularly in contexts where uncertainty and multiple criteria are involved. For instance, Gavurova et al. (2023) developed a fuzzy decision support model for evaluating and selecting healthcare projects, demonstrating its potential in resolving complex selection problems with competing criteria. In a related study, Gavurova and Polishchuk (2025) proposed a fuzzy-based system for supporting travel planning for people with disabilities, which emphasizes the adaptability of fuzzy set theory in social and logistical decision-making. Similar techniques were employed in healthcare management, where Smolanka et al. (2024) investigated the integration of information technologies in medical institutions, and Gavurova et al. (2024) built an intellectual model for analyzing patient trust in healthcare providers – both highlighting the role of intelligent systems in operational assessment and trust analysis.

Furthermore, hybrid decision-making tools have also found application in the tourism and business sectors. Skare et al. (2023a) presented a decision support model for venture capital investment on crowdfunding platforms, while Skare et al. (2023b) introduced a large-scale model for evaluating regional tourism infrastructure funding, both emphasizing the scalability and predictive strength of integrated analytical systems. Complementing this, Moravec et al. (2025) explored algorithmic personalization and digital literacy, identifying gaps in public understanding and reinforcing the need for transparent and intelligent personalization frameworks in AI-supported environments. Recently, many researchers developed a Hybrid neural network model for better prediction (Abumohsen et al., 2024; Ehteram et al., 2023). Wang et al., (2023) proposed a new hybrid model, CNN-SA-NGU, to predict silver closing prices by integrating conventional neural networks, the self-attention mechanism, and the new gated unit. Li et al., (2023) We also propose a CNN-LSTM-based approach, utilizing the strengths of Convolutional Neural Networks (CNN) for feature extraction and Long-Short-Term Memory (LSTM) networks. Vidal & Kristjanpoller, (2020) We prove that the proposed CNN-LSTM model improves GARCH model forecasting by 21% and more than 10% in the classic LSTM model compared to its RMSE. These studies underscore the growing relevance of hybrid fuzzy models and intelligent decision support systems across various domains, providing methodological insights that align with current trends in AI-driven prediction, evaluation, and planning under uncertainty.

Material and Methods

For the analysis purpose, the historical data for the precious metal took from the investing.com website, and tha data is daily price for all precious metals ranging from 2014 January 1 to 2024 December 31.

In developing the preprocessing pipeline, there were seven hybrid deep learning models intended for the historical gold price dataset: CNN, LSTM, MLP, CNN-LSTM, LSTM-MLP, MLP-CNN, and LSTM-MLP-CNN. Each model would require a different representation based on its architecture, whereas every preprocessing step was intended to give the dataset consistency and quality. First, invalid or missing entries were cleaned from the raw dataset to ensure integrity. Normalization of data was done within the range of [0, 1] using the formula:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

Where:

x_{norm}	Normalized value.
x_{min}	Minimum value of metal price.
x_{max}	Maximum value of metal price.

Normalization was necessary to bring the data into a standard scale, so that models trained efficiently and were not biased by the magnitude of the original price values. After normalization, the sliding window approach was used to segment the data into sequences of 20 consecutive price values as input, with the 21st value serving as the target output. This captured temporal dependencies in the data and provided the context necessary for time-series forecasting.

In the case of CNN-based models, the input sequences were transformed into 20×20 grayscale images by tiling the 20-element sequences into square matrices. These images are used to leverage the capability of CNN in extracting spatial patterns.

For LSTM-based models, the input was prepared as 20×1 sequential arrays that maintained the temporal structure of the data. This format was optimized for the LSTM's capability to model long-term dependencies.

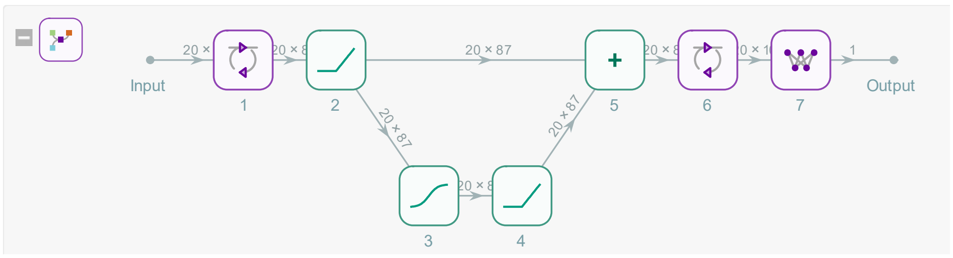
For MLP-based models, input was flattened into a 1D array of 20 values to allow the MLP to process feature-level relationships inherent in the data.

The hybrid models required a combination of these input formats. In the case of CNN-LSTM, it included both the 20×20 grayscale images for the CNN component and 20×1 sequential arrays for the LSTM component. For LSTM-MLP, the input was a combination of the 20×1 sequential arrays for the LSTM and 1D arrays for the MLP. In the case of MLP-CNN, this included 20×20 grayscale images for the CNN and 1D arrays for the MLP.

Lastly, the LSTM-MLP-CNN model required three forms of pre-processing: 20×1 arrays for LSTM, 20×20 images for CNN, and flattened 1D arrays for MLP.

This extensive preprocessing approach ensures that each model sees the input in the best format for its architecture and maximizes the effectiveness of the training process, thereby allowing strong performance across all seven models.

LSTM: An LSTM-based neural network was implemented in Mathematica using the NetGraph function. The architecture consisted of two LSTM layers interconnected by several nonlinear activation functions randomly picked from the set: {Tanh, Sin, Ramp, Logistic Sigmoid}. The structure was as follows:



Note: The architecture processes a 20×1 input sequence with two LSTM layers, sandwiched by three nonlinear activation layers. Asummation node aggregates the activation outputs from previous layers and a final linear layer predicts the normalized gold price correspondingto the input sequence.

The neural network consists of six main layers in addition to the input and output. The six layers between the input and output layers are hidden.

Input Layer: A matrix representing electricity consumption data is provided in the input layer. Different sizes of matrices will be investigated in the experiment (based on the input data). Here, the matrix of size is “20 x 1”.

1st Hidden Layer (LSTM Layer): The first hidden layer is an LSTM layer. It generates an output matrix of size m x n,' where 'm' is the time series lag and 'n' is chosen empirically. Here, the matrix at the input layer 23 x 1 will generate a matrix 20 x 87 at the output). The 'n' value grows by one.

2nd Hidden Layer (Elementwise Layer—Perceptron): This layer is built as an Elementwise Layer, a simple network that functions as a perceptron. The layers' activation functions will be chosen at random from a specified list of activation functions.

3rd Hidden Layer (Elementwise Layer): This layer is an Elementwise Layer with the same activation function selection criteria as the second hidden layer.

4th Hidden Layer (Plus Layer): The fourth hidden layer, known as the "Plus" layer, accomplishes summing. It receives input from the second and third hidden layers and transmits the output to the fifth hidden layer.

5th Hidden Layer (Elementwise Layer): This layer is an Elementwise Layer that uses the same activation function selection criteria as the second and third hidden layers.

6th Hidden Layer (Linear Layer): The sixth hidden layer is a Linear Layer, which acts on a data matrix at the input (a 20 x 87 matrix is shown in Figure 1). It generates a vector with one element as output.

Output Layer: The output layer predicts the metal commodity price.

Long-Short Term Memory Layer $[n]$ represents a network that receives an input matrix representing a sequence of vectors and outputs a sequence of the same length. Each input sequence element is a vector of size k, and each output sequence element is a vector of size n. The LSTM consists of four blocks: Input gate, output gate, forget gate, and memory gate.

$\{x1, x2, \ldots, xT\}$ is the input sequence, and the output of the LSTM is a series of states $\{s1, s2, \ldots, sT\}$. The cell state is defined as follows:

$$c_t = f_t * c_{t-1} + i_t * m_t \tag{2}$$

where

c_t	is a new variable state.
f_t	forget gate.
c_{t-1}	the initial state of the variable.
i_t	input gate.
m_i	memory gate.

The input gate is defined as follows:

$$i_t = \sigma[W_{ix}x_t + W_{is}S_{t-1} + b_i] \tag{3}$$

where

$$\sigma(z) = \frac{1}{(1 + exp(-z))} \tag{4}$$

σ	is Logistic Sigmoid.
W_{ix}	is an input weight in the input gate matrix $n\ x\ k$
x_t	is an input variable, matrix $n\ x\ k$.
W_{is}	Weight of the state in the input gate, matrix $n\ x\ n$.
S_{t-1}	The initial state.
b_i	Bias, vector size n.

The state is defined as follows:

$$s_t = o_t * Tanh[c_t] \tag{5}$$

where,

s_t	is a state of the variable.
o_t	output gate.
$Tanh$	Hyperbolic tangent.

Output gate is defined as follows:

$$o_t = \sigma[W_{ox}x_t + W_{os}S_{t-1} + b_o] \tag{6}$$

where,

W_{ox}	Defines the input weight in output gate, matrix $n\ x\ k$.
W_{os}	Weight of the state in output gate, matrix $n\ x\ n$.
b_o	Bias, vector size n.

When compared to, e.g., a Gated Recurrent Layer, the benefit of LSTM is in the forget gate:

$$f_t = \sigma[W_{fx}x_t + W_{fs}S_{t-1} + b_f] \tag{7}$$

where,

W_{fx}	is an input weight in forget gate, matrix $n\ x\ k$.
W_{fs}	is the weight of the state in forget gate, matrix $n\ x\ n$.
b_f	Bias, vector size n.

The main processes of LSTM include memory gate as follows:

$$= Tanh[W_{mx}x_t + W_{ms}S_{t-1} + b_m] \tag{8}$$

where,

W_{mx}	Defines the input weight in memory gate, matrix $n\ x\ k$.
W_{ms}	Weight of the state in memory gate, matrix $n\ x\ n$.
b_m	Bias, vector size n.

MLP

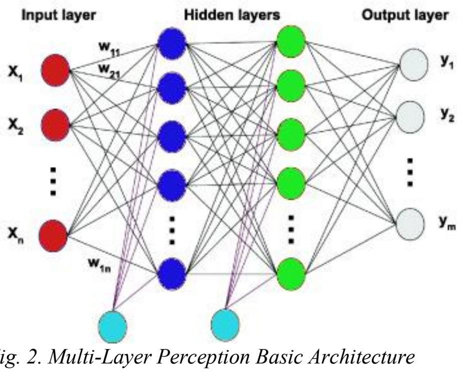


Fig. 2. Multi-Layer Perception Basic Architecture

An MLP consists of Three Layers:

Input layer: The input layer takes a preprocessed consumption data vector with 20 x 1 size, where '20' is representative of the number of time steps or hours utilized in order to predict the next value in consumption.Hidden layers: These layers perform computations and learn representations of the data.

First Hidden Layer: This includes a linear layer with a size of 100, which processes the input data and generates an output vector that helps in learning high-level and complex patterns in the data. An activation function applied at this layer is the Tanh function. This will introduce the non-linearity into the model.

Second Hidden Layer: Another linear layer of 50 units; the higher-level learned representations refine those from the previous layer. The same Tanh activation function is applied, enhancing the model in capturing intricate relationships in the data.

Output layer: The output layer contains one linear unit that generates, from the features learned by the previous layers, the forecast value of electricity consumption for the next hour.

MLPs are fully connected networks where each neuron in a layer is connected to every neuron in the subsequent layer. They are powerful because they can model complex relationships between inputs and outputs.

Architecture of MLP

An MLP with one hidden layer can be mathematically described by the following equations:

Input Layer

Let the input vector be $x = [x_1, x_2, \dots, x_n]$, where n n is the number of features.

Hidden Layer

Each hidden layer neuron computes a weighted sum of the input features, followed by the application of an activation function. For neuron j in the hidden layer:

$$z_j = \sum_{i=1}^{n} w_{ij} x_i + b_j \tag{9}$$

Where,

w_{ij}	is the weight between input neuron i and hidden neuron j.
x_i	is the bias term for the hidden neuron j.
b_j	is the linear combination of the input and weights for hidden neuron j.

The result z_j is then passed through an activation function, f, to introduce non-linearity:

$$h_j = f(z_j) = f(\sum_{i=1}^{n} w_{ij} x_i + b_j) \tag{10}$$

Where, f can be any non-linear activation function, such as:

Sigmoid:*f(z)* :

$$f(z) = \frac{1}{1 + e^{-z}} \tag{11}$$

ReLU (Rectified Linear Unit):

$$f(z) = \max(0, z) \tag{12}$$

Tanh:

$$f(z) = tanh(z) \tag{13}$$

Thus, the hidden layer outputs a vector $h = [h_1, h_{2,} \dots, h_m]$ where m m is the number of hidden neurons.

Output Layer

The output layer computes a weighted sum of the activations from the hidden layer. For output neuron k :

$$y_k = \sum_{j=1}^{m} v_{jk} x_j + c_k \tag{14}$$

Where,

v_{jk}	is the weight between hidden neuron j and output neuron k.
c_k	is the bias term for the output neuron k.
y_k	is the pre-activation value for the output neuron k .

The output may be passed through a final activation function depending on the task:

For regression, no activation (or a linear activation) is used, so y_k remains as is.

For binary classification, the sigmoid function is used:

$$\hat{y} = \sigma(y_k) = \frac{1}{1+e^{-y_k}} \tag{15}$$

For multi-class classification, the softmax function is applied to normalize outputs into probabilities:

$$\hat{y}_k = \frac{e^{y_k}}{\sum_{k^1} e^{y_{k^1}}} \tag{16}$$

Overall Equation of MLP:

For an MLP with a single hidden layer, the overall equation combining both the hidden and output layers can be written as:

$$\hat{y}_k = f_{output} = \left(\sum_{j=1}^{m} v_{jk} f \left(\sum_{i=1}^{n} w_{ij} x_i + b_j \right) + c_k \right) \tag{17}$$

Where,

f_{output}	is the activation function for the output layer.

f	f is the activation function for the hidden layer

CNN

The convolution operation is the core of CNNs. For an input image (or feature map) X , a filter (or kernel) K, and an output Y, the convolution operation is given by:

$$Y[i,j] = \sum_{M=0}^{M-1}\sum_{n=0}^{N-1} K[m,n].X[i+m,j+n] + b \tag{18}$$

Where,

$X[i,j]$	Pixel value at position (i , j) in the input image or feature map.
$K[m,n]$	Weight of the kernel at position (m , n).
M,N	Dimensions of the kernel.
$Y[i,j]$	The result of applying the kernel at location (i , j) in the input.
b	Bias term.

After the convolution operation, a non-linear activation function like ReLU (Rectified Linear Unit) is applied element-wise to introduce non-linearity:

$$Y[i,j] = ReLU\left(\sum_{M=0}^{M-1}\sum_{n=0}^{N-1} K[m,n].X[i+m,j+n] + b\right) \tag{19}$$

ReLU is defined as:

$$ReLU(z) = max\ (0,z)$$

After the convolution and activation steps, pooling layers (e.g., max pooling or average pooling) are applied to reduce spatial dimensions while preserving key features.

$$Y[i,j] = max\{p,q \in P\} X[i+p,j+q] \tag{20}$$

Where,

p	Pooling window size
$X[i+p,j+q]$	Values in the pooling window.

After several convolutional and pooling layers, the output feature map is flattened into a 1D vector and passed through fully connected layers

$$z = W.x + b \tag{21}$$

Where,

W	Weight matrix of the fully connected layer.
x	Flattened input vector.
b	Bias term.

To take advantage of the power of CNNs in time series forecasting, the data have been transformed into grayscale images. For every input sequence of 20 values, a 20×20 matrix has been obtained by repeating the sequence across rows, which provides a spatial representation of the data. These matrices were converted into grayscale images, whose pixel intensities were proportional to the normalized values from the input sequence. This transformation facilitates the use of CNNs because it can encode temporal dependencies as spatial features, thus feeding them to the convolutional layers.

Design of CNN Architectures

The twelve different CNN architectures, each using a different activation function to look for the respective impacts on the forecasting accuracies, were evaluated. Each of the CNN architectures was kept identical in design so that comparison would be feasible. The architecture of each CNN starts with a convolutional layer using a 2×2 kernel with a stride of 1, which extracts localized patterns from the input images. After that, an activation layer is applied with the usage of functions like ReLU, ELU, GELU, Swish, etc., to provide non-linearity. Then, the feature maps are passed through a flattening layer that reshapes them into a one-dimensional vector. Finally, it ends up with a fully connected linear layer that provides a single output prediction representing the next value in the sequence. This modular and consistent architecture allows for the systematic evaluation of activation functions.

Training and Optimization

The training process uses an aggressive setup to maximize model performance and robustness. Each CNN is trained using the ADAM optimization algorithm with a learning rate of 0.01. The batch size is set to 30, which is large enough for efficient processing but still varied enough to provide a good change in each batch to encourage generalization. Training is conducted over 70 epochs, giving ample iterations for convergence. The training dataset used in these networks consists of a grayscale representation of images paired with their respective normalized output values. These models try to minimize the error of prediction to learn strong relationships between sequential data and future values.

Model Evaluation and Selection

The architecture that best works is decided by correlation analysis for each of the CNN models after training. This is done by calculating the Pearson correlation coefficient between the predicted and actual values, which gives a measure of the linear relationship between them. The model with the highest correlation coefficient value is selected to be the best predictor. By doing this, it's ensured that the chosen model does the best job of representing the real dynamics of the data variation.

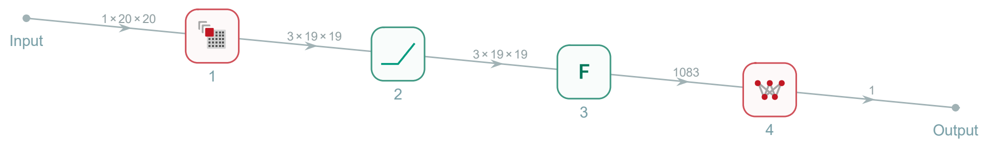


Fig 3. Architecture for CNN

CNN-LSTM

Data Transformation for CNN-LSTM

To reshape the preprocessed data for the hybrid CNN-LSTM architecture, the input sequences were transformed into 20×1 matrices, which encoded the temporal patterns as spatial data. The reshaping allows convolutional layers in the CNN part to capture local patterns in sequential data. The flattened output from the CNN feeds into the LSTM component, which then processes the temporal relationships in the sequence. This approach increases the capacity of the model by fusing spatial and temporal feature representations on mining complicated dependencies in the data.

Hybrid CNN-LSTM Architecture

The CNN-LSTM model will be developed in the light of deep learning based on the idea of hybrids, consisting of convolutional and recurrent layers. It combines two complementary elements:

Convolutional Layer: The CNN component applies convolutional filters to the reshaped input data to extract the localized patterns and features. The use of small 2×2 filters ensures the extraction of fine features that represent short-term tendencies in the data.

Flattening Layer: One can finally flatten the feature maps coming out of the CNNs into one-dimensional vectors for final integration with the LSTM components

LSTM Layer: The LSTM then processes sequential information embedded in the flattened features, learning long-term dependencies and temporal dynamics inside the price data.

Fully Connected Layer: The output of the LSTM is fed through a fully connected linear layer, the output of which is a single scalar prediction corresponding to the next price in the series.

Model Training and Optimization

The CNN-LSTM model is trained in a solid optimization framework. It will involve the ADAM optimizer for efficient gradient descent, the learning rate is set to 0.01 for stable convergence. Meanwhile, the batch size chosen for training is 30 over the span of 100 epochs, striking a perfect balance between computational efficiency and model performance. The loss function used here is MSE, which determines the difference between the values predicted and the actual value. It is used to drive the optimization process to minimize the prediction error of the model.

The inputs to the model are a normalized sequence of historical gold price data converted into 20×1 matrices, with the corresponding output being the next price value. Such a structure allows the model to iteratively refine parameters to come up with appropriate predictions.

Forecasting

The model, once trained and evaluated, is deployed for forecasting the gold prices. New sequences of normalized price data feed into the trained CNN-LSTM model, from which a prediction of the same, also at a normalized level, is obtained. It is then necessary to invert these predictions to obtain them on the original scale. The CNN-LSTM model will provide a strong tool for predicting price variations with high accuracy by capturing both localized and temporal features in the data.

CNN-MLP

Hybrid MLP-CNN Architecture

The hybrid MLP-CNN model is designed to bridge the gap between the effectiveness of MLPs in terms of sequential processing and CNNs regarding spatial feature extraction. Its MLP part takes the input as 1D sequential data and then processes the input through three fully connected layers. The first two layers contain 100 and 50 neurons, respectively, using the Tanh activation function for introducing non-linearity. The last layer outputs a single scalar value, capturing the relationships within the sequence. In parallel, the CNN component processes the 20×20 grayscale image input. First, it uses a convolutional layer with 2×2 kernels to extract localized spatial patterns. Then, the resulting feature maps are fed through a ReLU activation function and further downsampled with a flattening layer into a 1D vector. Further processing of these features is done by a fully connected layer comprising 50 neurons. The catenate layer then concatenates the outputs of the MLP and CNN parts into a combined feature representation, incorporating both sequential and spatial information. The final fully connected layer consists of one neuron and outputs the predicted next price in the sequence.

Training and Optimization

It provides the robust optimization framework used to train the hybrid model on the preprocessed dataset. The ADAM optimizer has been used with an iterable learning rate of 0.01 for model parameter tuning to assure efficiency in convergence. The Mean Squared Error loss function, which always minimizes the differences between squared actual and predicted values, was used, thus compelling the model to make precise predictions. Training is done with a batch size of 30 to balance computational efficiency and model generalization, while the maximum number of training rounds is set to 81 to allow for enough iterations in learning. The input consists of two parts: 1D sequential data and corresponding 20×20 grayscale images. During training, the model learns both temporal relationships from the MLP input and spatial features from the CNN input, optimizing both components together. The hybrid architecture ensures that the model effectively captures both localized and global patterns within the data.

Forecasting

Once trained and tested, the hybrid model is used for forecasting future gold prices. The prediction process involves the use of new sequences of normalized price data, both as 1D arrays for the MLP component and as 20×20 grayscale images for the CNN component. The model generates predictions in normalized form, which are then denormalized to their original scale for interpretability. The hybrid MLP-CNN model leverages both sequential and spatial representations of the input data to provide accurate and actionable forecasts, effectively capturing both temporal and spatial dependencies in the gold price dataset.

LSTM-MLP

Hybrid LSTM-MLP Architecture

It combines the strength of Long Short-Term Memory Networks with Multi-Layer Perceptrons to take advantage of temporal dependencies and nonlinear feature relationships in the data. The LSTM block processes the sequential 20×1 input to capture temporal patterns and long-term dependencies. It consists of two LSTM layers, the first with 87 units and the second with 100 units, interspersed with ElementwiseLayer operations that employ random activation functions such as Tanh, Sin, Ramp, and LogisticSigmoid. The intermediate outputs of the LSTM layers are combined using a summation operation, and the final output is passed through a fully connected layer with 50 neurons. While the MLP block, by contrast, takes care of the flattened 1D array of 20 price values and includes three fully connected layers containing 100, 50, and 1 neurons consecutively, with the tanh activation function after the first two layers to better represent non-linear relationships, are followed. Outputs coming out of the LSTM and the MLP blocks are further combined using CatenateLayer to form an integrated feature representation. This combined representation, the input to a final linear layer with a single neuron to produce the predicted price value, ensures that both temporal and feature-level dependencies are captured effectively and integrated.

Training and Optimization

The hybrid model was trained on the preprocessed dataset using a robust optimization strategy. In this regard, the ADAM optimizer was used to dynamically adjust the parameters of the model for efficient convergence. The Mean Squared Error loss function was used to quantify the error between the predicted and actual values, driving the model to minimize the prediction error. The training was done with a batch size of 30, balancing computational efficiency and generalization. The model had a maximum of 81 rounds of training to have enough learning iterations. The input data had two components: the sequential 20×1 array for the LSTM component and the flattened 20 20-dimensional array for the MLP component. During training, the LSTM block learned temporal dependencies from the sequential input, while the MLP block captured feature-level relationships. The concatenated features from both blocks were optimized together to produce accurate predictions. This integrated training process allowed the model to effectively leverage both LSTM and MLP capabilities.

Forecasting

After training and evaluation, the hybrid LSTM-MLP model was used to make the forecast of future gold prices. New input sequences were formatted as Both 20×1 arrays and flattened 1D arrays served as inputs for the model to generate the predictions. The predicted values, normalized initially, were transformed back into their original scale for better interpretability and practical use. This hybrid model combines long-term temporal modeling through its LSTM component with nonlinear feature extraction through its MLP component, yielding accurate and robust forecasts. It was good at capturing both short-term trends and long-term dependencies in the data.

LSTM-CNN-MLP

Hybrid LSTM-MLP-CNN Architecture

The proposed hybrid LSTM-MLP-CNN architecture brings together the complementary strengths of LSTM, MLP, and CNN components to comprehensively process sequential, feature, and spatial data. The LSTM block receives the 20×1 sequential array and processes it through two Long Short-Term Memory layers with 87 and 100 units, respectively. Elementwise layers with Tanh and Ramp activations are interleaved within the LSTM block to introduce non-linear transformations, and a summation operation combines intermediate outputs. The final output is fed into a fully connected layer of 50 neurons, capturing long-term temporal dependencies. The CNN block processes the 20×20 grayscale image, applying a convolutional layer with a 2×2 kernel and ReLU activation to extract spatial features. The resulting feature maps are flattened and passed through a fully connected layer with 50 neurons to consolidate spatial patterns. Simultaneously, the MLP block feeds the flattened 1D array of 20 price values through three fully connected layers with 100, 50, and 50 neurons, using Tanh activations to capture complex relationships between features. The outputs of the LSTM, CNN, and MLP blocks are then concatenated by a CatenateLayer into a unified feature representation. This combined representation is fed through a final linear layer with one neuron that generates the predicted price value. This hybrid design enables the model to harness the strengths of all three components, thereby effectively integrating temporal, spatial, and feature-level insights.

Training and Optimization

The hybrid LSTM-MLP-CNN model was trained with a dataset of paired inputs and target outputs using a robust training strategy. This utilized the ADAM optimizer in dynamically adjusting the model parameters with the assurance of efficient convergence. Mean Squared Error was the loss function used to quantify the error in prediction, hence forcing the model to minimize deviations between the predicted and actual values. The training process used a batch size of 30 to balance computational efficiency and generalization, and the model was trained for 81 rounds to achieve optimal performance. The tripartite input format consisted of 20×1 arrays for the LSTM component, The CNN inputs are grayscale images of size 20×20, and the inputs for the MLP are 1D flattened arrays. In this way, during training, each component has learned the pattern from its type of input, while concatenated features have been optimized altogether to ensure cohesive learning across the whole model. This allows the proposed hybrid architecture to understand complex relationships in the data effectively.

Forecasting

The trained model was then deployed to make a forecast for future gold prices. New input sequences were prepared in the same tripartite format as the training data and fed into the model to generate the predictions. These output predictions, having been normalized, were then denormalized back to their original scale for interpretability and practical application. This combination of long-term temporal modeling, localized spatial pattern extraction, and complex feature relationship analysis allowed the hybrid model to realize an accurate and robust forecast. Such an integrated approach makes the LSTM-MLP-CNN architecture particularly apt for catching the multifaceted dynamics of gold price trends.

Results

In this work, we compare the performances of deep learning models for precious metals price series forecasting (gold, silver). Evaluation of the models in four error metrics (Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and correlation coefficient.

Performance Metrics for Gold Forecasting

The results of performance metrics for gold prices are given below:

Table 1. *Performance Metrics for Gold Price Forecasting.*

	MAE	RMSE	MAPE	R
CNN	0.01902960	0.02527520	12.0137	0.993658
MLP	0.00682418	0.00936313	4.32170	0.999209
LSTM	0.00886243	0.01218730	4.99990	0.998588
MLP-CNN	0.00741245	0.01067920	3.86708	0.999223
MLP-LSTM	0.00632371	0.00881293	4.12012	0.999284
LSTM-CNN	0.00631120	0.00905016	3.65598	0.999274
LSTM-CNN-MLP	0.00601101	0.00855050	3.67573	0.999269

As one can see from the results, the LSTM-CNN-MLP hybrid model has the best performance concerning the lowest MAE (0.00601101), RMSE (0.0085505), 0r a similar MAPE (3.67573%), and correlation coefficient at(a very high heheheh level) 0.999269. Although both different hybrid models other than top-ranked (MLP-LSTM–and LSTM-CNN-hybrid) have a very skillful performance, the correlation coefficient is higher than 0.999, and their error metrics are very competitive with the top model. On the other hand, a single model such as CNN and LSTM shows relatively high error rates and lower correlation coefficients, which suggests that the hybrid frameworks can forgive the complex patterns when it comes to gold price data.

The results of performance metrics for silver prices are given below:

Table 1. *Performance Metrics for Silver Price Forecasting.*

	MAE	RMSE	MAPE	R
CNN	0.0349005	0.0487878	11.8175	0.971781
MLP	0.0137824	0.0190507	4.40592	0.996649
LSTM	0.0155308	0.0203800	5.72662	0.996244
MLP-CNN	0.0106659	0.0161304	3.49368	0.996723
MLP-LSTM	0.0106667	0.0145519	3.70762	0.997639
LSTM-CNN	0.0107948	0.0162004	3.41239	0.996878
LSTM-CNN-MLP	0.00606639	0.00808649	2.34891	0.999172

In line with silver price forecasting, the LSTM-CNN-MLP model showed best performance with the lowest MAE (0.00606639), RMSE (0.00808649), and MAPE (2.34891%), and highest R (correlation coefficient) of 0.999172 as shown. Moreover, the hybrid models such as MLP-CNN, MLP-LSTM and LSTM-CNN performed exceptionally well in contrast to standalone Architectures.

Table 1. *Performance Metrics for Palledium Price Forecasting.*

	MAE	RMSE	MAPE	R
CNN	0.0253315	0.04113625	12.248294	0.9860111
MLP	0.0102783	0.01521242	4.4007547	0.9983801
LSTM	0.0143736	0.01854710	9.7468207	0.9979762
MLP-CNN	0.0080040	0.00926510	3.7815979	0.9985415
MLP-LSTM	0.0069804	0.01049340	3.4363746	0.9989640
LSTM-CNN	0.0083587	0.01257198	4.3159546	0.9986511
LSTM-CNN-MLP	0.0048560	0.00678739	3.0187400	0.9995880

The best-performing model for palladium price prediction was LSTM-CNN-MLP, superior overall in all the measures with the least MAE (0.0048560), RMSE and MAPE (3.01874%), and highest correlation coefficient ("R"), indicating 0.999588. The hybrid models of MLP-LSTM and MLP-CNN did much better than single models, e.g., CNN or LSTM; Figure 4 below depicts the actual vs forecasted prices for the models that are being applied to precious metal forecasting consisting of gold, silver, and palladium. A panel for each model shows how well a model can keep pace with the price trends through time. The visualization comparison shows the alignment of forecasted and actual value

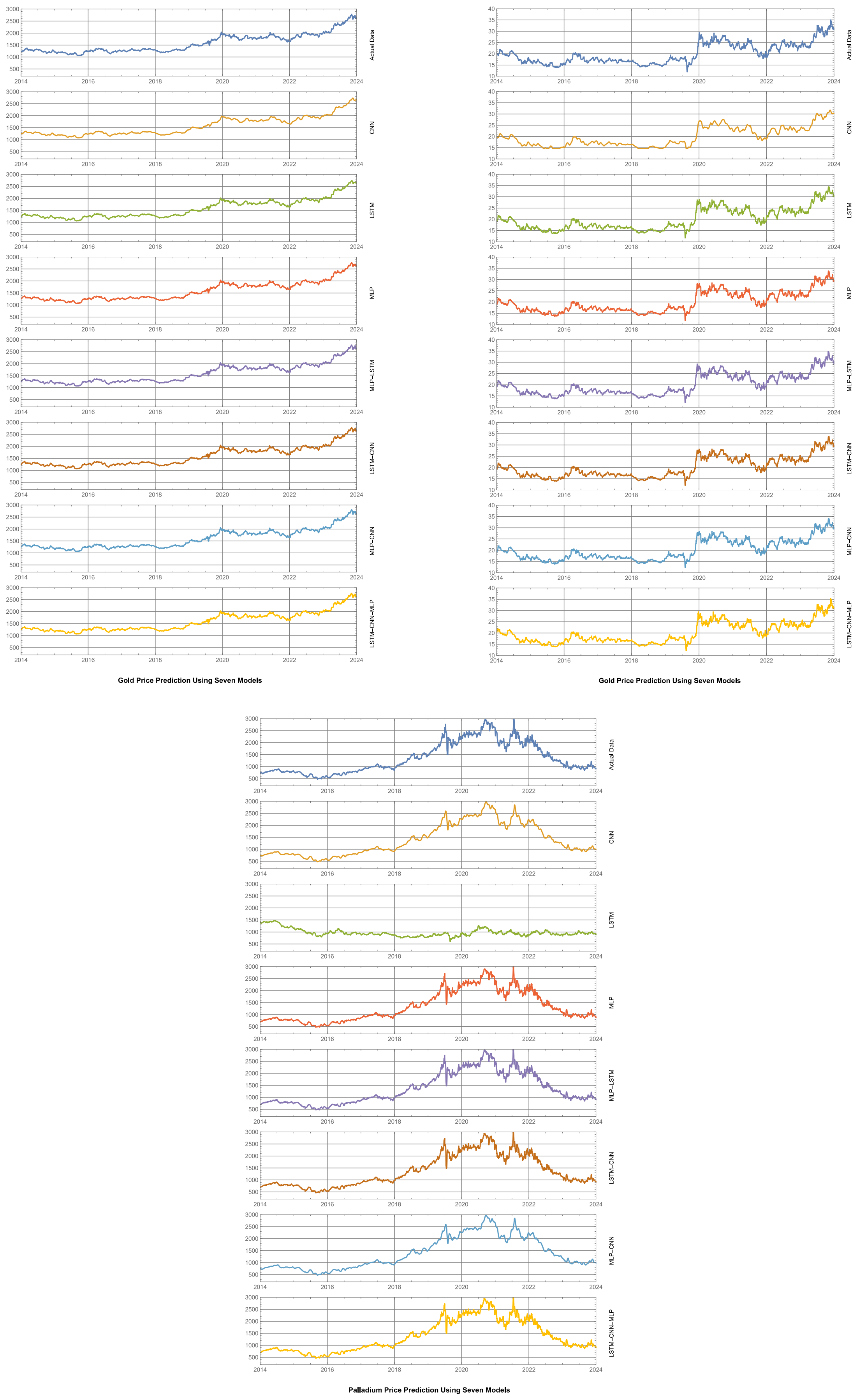


Figure 4. Performance Visualization of Forecasting Models for Precious Metals

Figures 5,6 and 7 represent the radial plot of error metrics for different models

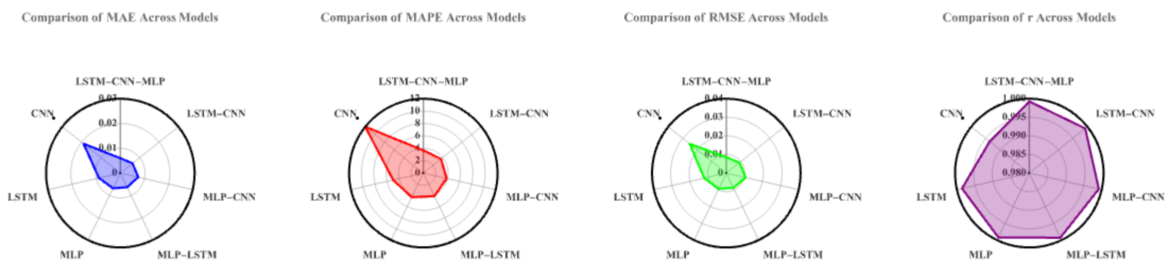


Figure 5. Radial Plot Comparison of Error Metrics for Gold Forecasting Model

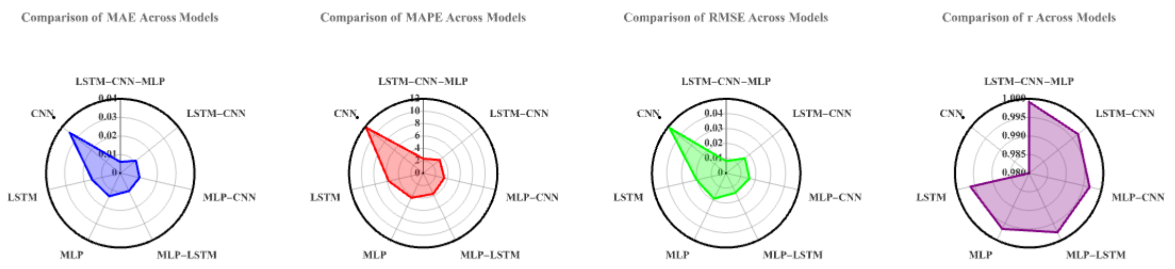


Figure 6.. Radial Plot Comparison of Error Metrics for Silver Forecasting Model

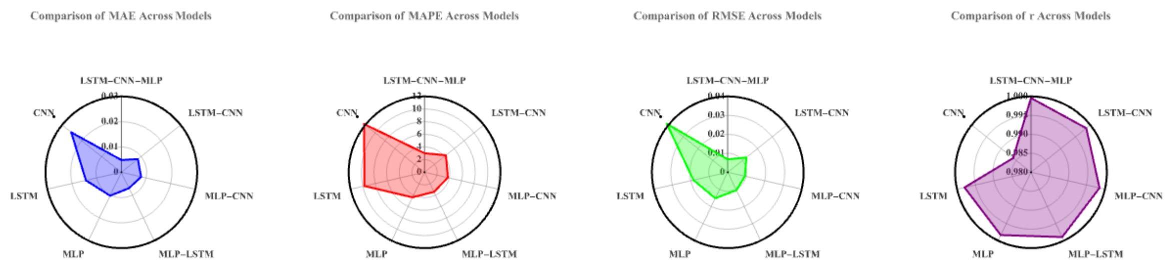


Figure 7.. Radial Plot Comparison of Error Metrics for Palladium Forecasting Model

Discussion

RQ1: How do the unique architectures and hybrid configurations of neural network models (MLP, LSTM, CNN, CNN-LSTM, CNN-MLP, MLP-LSTM, CNN-MLP-LSTM) affect the accuracy, robustness, and computational efficiency of precious metal price predictions?

The paper studies different neural topologies and hybrid schemes for precious metal price forecasting, by testing their performance with four different metrics: mean absolute error (MAE), root mean square error (RMSE), mean absolute percentage error (MAPE), and correlation coefficient(R). This research examines the price predictions of gold, silver, and palladium. The hybrid LSTM-CNN-MLP of gold gave the best followed by the lowest MAE (0.00601101), RMSE (0.0085505), and MAPE (3.67573%) with an excellent correlation coefficient R (= 0.999269). Thus the results speak of this hybrid technique being very good at complex pricing patterns. The second-best-performing models MLP-LSTM and LSTM-CNN hybrids showed low error rates with correlation coefficients over 0.999. Models like CNN and LSTM are more likely to give higher errors as well as worse R scores, which speaks to the fact that simple models like these cannot learn the complex process behind gold price data. hybrid models perform significantly better in gold pricing than traditional models FindingsLSTM-CNN-MLP is the best model in the case of MAE 0.00606639, RMSE 0.0080865, and MAPE 2.34891% as well, and Silver has the highest correlation (R). Hybrid (MLP-CNN, MLP-LSTM, and LSTM-CNN) model outperforms individual solution For example, MLP-CNN had a perfect all-around showing with MAE 0.0106659, RMSE 0.0161304 and MAPE 3.49368%.

Single models (CNN and LSTM) show very high rates in terms of error, and low correlation coefficients (r=0.971781 and 0.996244). The recurring theme indicates that hybrid frameworks serve the strengths of silver price forecasting exceptionally well. Similarly, for the prediction of palladium price, the LSTM-CNN-MLP model produced the best results across all dimensions. It was the LSTM (Mean Absolute Error:0.0048560 %, Root Mean Squared Error: 0.00678739, and MAPE:3.01874 %) and also the maximum correlation value (R =0.999588). Even the MLP-LSTM and MLP-CNN hybrids performed very well, surpassing standalone architectures as a whole. The error was lowered once more when models like MLP-LSTM performed an MAE of 0.0069804 and RMSE of 0.01049340. Meanwhile, independent models such as CNN and LSTM were found to suffer from higher error rates and lower correlation coefficient values, showing the benefits of hybrid models. The results highlight that hybrid neural network architectures are valuable in forecasting precious metal prices. In all models, the LSTM-CNN-MLP stacked ensemble has shown the best accuracy, stability, and precision concerning historical gold/silver/palladium. This helps to suggest that the hybrid frameworks are much better at encapsulating deep and non-linear correlations of precious metal pricing data than standalone models. The hybrid models MLP-LSTM and LSTM-CNN seemed to have quite good potential but the LSTM-CNN-MLP outperformed both. Individual models such as CNN and LSTM were not very successful in comparison to hybrid models. This paper provides important implications for designing efficient neural network topologies in financial time-series forecasting.

Conclusion

The results of this study illustrate that to forecast precious metal prices, hybrid deep learning models are powerful and significantly superior to any single-model architectures. Hybrid models outperformed single models on accuracy for all metals (gold, silver, palladium) measured in terms of benchmarks such as Mean Absolute Error, Root Mean Square Error, MAPE, and correlation coefficient The results show that the LSTM-CNN-MLP hybrid model provided the best performance with the smallest error and highest correlation for all three metals. The MLP-LSTM, MLP-CNN, and LSTM-CNN hybrid models performed very well as well, indicating that combining several architectures can indeed take advantage of the best features of both styles. In conclusion, this work demonstrates that hybrid deep learning remains superior in operational research problems, specifically focusing on financial forecasts. The LSTM-CNN-MLP model is a very dependable and robust solution for precious metals forecasting that practitioners should opt. These findings will facilitate further work and the practice of hybrid models across other domains in financial and operational analytics.

Reference

Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*, *4*(11), e00938. https://doi.org/10.1016/j.heliyon.2018.e00938

Abumohsen, M., Owda, A. Y., Owda, M., & Abumihsan, A. (2024). Hybrid machine learning model combining of CNN-LSTM-RF for time series forecasting of Solar Power Generation. *E-Prime - Advances in Electrical Engineering, Electronics and Energy*, *9*, 100636. https://doi.org/10.1016/j.prime.2024.100636

Adebiyi, A. A., Adewumi, A. O., & Ayo, C. K. (2014). Comparison of ARIMA and Artificial Neural Networks Models for Stock Price Prediction. *Journal of Applied Mathematics*, *2014*(1), 614342. https://doi.org/10.1155/2014/614342

Adekoya, O. B., Oliyide, J. A., & Tahir, H. (2021). What do we know about the inflation-hedging property of precious metals in Africa? The case of leading producers of the commodities. *Resources Policy*, *72*, 102120. https://doi.org/10.1016/j.resourpol.2021.102120

Ben Ameur, H., Boubaker, S., Ftiti, Z., Louhichi, W., & Tissaoui, K. (2023). Forecasting commodity prices: Empirical evidence using deep learning tools. *Annals of Operations Research*, *339*(1), Article 1. https://doi.org/10.1007/s10479-022-05076-6

Bildirici, M. E., & Gokmenoglu, S. M. (2020). Precious metal abundance and economic growth: Evidence from top precious metal producer countries. *Resources Policy*, *65*, 101572. https://doi.org/10.1016/j.resourpol.2019.101572

Chen, L., & Zhang, X. (2019). Gold Price Forecasting Based on Projection Pursuit and Neural Network. *Journal of Physics: Conference Series*, *1168*(6), 062009. https://doi.org/10.1088/1742-6596/1168/6/062009

de O. Santos Júnior, D. S., de Oliveira, J. F. L., & de Mattos Neto, P. S. G. (2019). An intelligent hybridization of ARIMA with machine learning models for time series forecasting. *Knowledge-Based Systems*, *175*, 72–86. https://doi.org/10.1016/j.knosys.2019.03.011

Derakhshani, R., GhasemiNejad, A., Amani Zarin, N., Amani Zarin, M. M., & Jalaee, M. sadat. (2024). Forecasting Copper Prices Using Deep Learning: Implications for Energy Sector Economies. *Mathematics*, *12*(15), Article 15. https://doi.org/10.3390/math12152316

Dhifaoui, Z., Khalfaoui, R., Abedin, M. Z., & Shi, B. (2022). Quantifying information transfer among clean energy, carbon, oil, and precious metals: A novel transfer entropy-based approach. *Finance Research Letters*, *49*, 103138. https://doi.org/10.1016/j.frl.2022.103138

Ding, Y., Zhang, S., Liu, B., Zheng, H., Chang, C., & Ekberg, C. (2019). Recovery of precious metals from electronic waste and spent catalysts: A review. *Resources, Conservation and Recycling*, *141*, 284–298. https://doi.org/10.1016/j.resconrec.2018.10.041

Ehteram, M., Najah Ahmed, A., Khozani, Z. S., & El-Shafie, A. (2023). Graph convolutional network – Long short term memory neural network- multi layer perceptron- Gaussian progress regression model: A new deep learning model for predicting ozone concertation. *Atmospheric Pollution Research*, *14*(6), 101766. https://doi.org/10.1016/j.apr.2023.101766

Emmert-Streib, F., Yang, Z., Feng, H., Tripathi, S., & Dehmer, M. (2020). *Frontiers | An Introductory Review of Deep Learning for Prediction Models With Big Data*. https://doi.org/10.3389/frai.2020.00004

Foroutan, P., & Lahmiri, S. (2024). Deep learning systems for forecasting the prices of crude oil and precious metals. *Financial Innovation*, *10*(1), 111. https://doi.org/10.1186/s40854-024-00637-z

Gavurova, B., Kelemen, M., Polishchuk, V., Mudarri, T., & Smolanka, V. (2023). A fuzzy decision support model for the evaluation and selection of healthcare projects in the framework of competition. *Frontiers in Public Health, 11*, 1222125. https://doi.org/10.3389/fpubh.2023.1222125

Gavurova, B., & Polishchuk, V. (2025). Decision-making support system for travel planning for people with disabilities based on fuzzy set theory. *Oeconomia Copernicana, 16*(1), 417–462. https://doi.org/10.24136/oc.3503

Gavurova, B., Smolanka, V., & Polishchuk, V. (2024). Intellectual model for analyzing and managing patient trust in medical staff of primary healthcare institutions. *Polish Journal of Management Studies, 30*(1), 98–115. https://doi.org/10.17512/pjms.2024.30.1.06

Gligorić, Z., Gligorić, M., Halilović, D., Beljić, Č., & Urošević, K. (2020). Hybrid Stochastic-Grey Model to Forecast the Behavior of Metal Price in the Mining Industry. *Sustainability*, *12*(16), Article 16. https://doi.org/10.3390/su12166533

Grebovic, M., Filipovic, L., Katnic, I., Vukotic, M., & Popovic, T. (2022). Overcoming Limitations of Statistical Methods with Artificial Neural Networks. *2022 International Arab Conference on Information Technology (ACIT)*, 1–6. https://doi.org/10.1109/ACIT57182.2022.9994218

Hansen, J. V., McDonald, J. B., & Nelson, R. D. (1999). Time Series Prediction With Genetic-Algorithm Designed Neural Networks: An Empirical Comparison With Modern Statistical Models. *Computational Intelligence*, *15*(3), 171–184. https://doi.org/10.1111/0824-7935.00090

Kucher, O., & McCoskey, S. (2017). The long-run relationship between precious metal prices and the business cycle. *The Quarterly Review of Economics and Finance*, *65*, 263–275. https://doi.org/10.1016/j.qref.2016.09.005

Lauret, P., David, M., & Pedro, H. T. C. (2017). Probabilistic Solar Forecasting Using Quantile Regression Models. *Energies*, *10*(10), Article 10. https://doi.org/10.3390/en10101591

Lazzarino, M., Berrill, J., & Šević, A. (2022). The importance of distinguishing between precious and industrial metals when investing in mining stocks. *Resources Policy*, *78*, 102802. https://doi.org/10.1016/j.resourpol.2022.102802

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*, *11*, 69458–69473. IEEE Access. https://doi.org/10.1109/ACCESS.2023.3288486

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. https://doi.org/10.1016/j.resourpol.2022.102884

Mensi, W., Xuan, V. V., & Kang, S. H. (2021). Time and frequency connectedness and network across the precious metal and stock markets: Evidence from top precious metal importers and exporters. *RESOURCES POLICY*, *72*, 102054. https://doi.org/10.1016/j.resourpol.2021.102054

Mokni, K., Al-Shboul, M., & Assaf, A. (2021). Economic policy uncertainty and dynamic spillover among precious metals under market conditions: Does COVID-19 have any effects? *Resources Policy*, *74*, 102238. https://doi.org/10.1016/j.resourpol.2021.102238

Moravec, V., Hynek, N., Skare, M., Gavurova, B., & Polishchuk, V. (2025). Algorithmic personalization: A study of knowledge gaps and digital media literacy. *Humanities and Social Sciences Communications, 12*, 341. https://doi.org/10.1057/s41599-025-04593-6

Nabipour, M., Nayyeri, P., Jabani, H., S., S., & Mosavi, A. (2020). Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; a Comparative Analysis. *IEEE Access*, *8*, 150199–150212. IEEE Access. https://doi.org/10.1109/ACCESS.2020.3015966

Ngadono, T. S., & Ikatrinasari, Z. F. (2018). Forecasting of PVB Film Using ARIMA. *IOP Conference Series: Materials Science and Engineering*, *453*(1), 012012. https://doi.org/10.1088/1757-899X/453/1/012012

Oner, H., & Oner, S. (2022). Precious Metals Price Forecast with ARCH Family Models in COVID-19 Pandemic Period. *The Economics and Finance Letters*, *9*(1), Article 1. https://doi.org/10.18488/29.v9i1.3023

Pan, Y. (2024). Different Types of Neural Networks and Applications: Evidence from Feedforward, Convolutional and Recurrent Neural Networks. *Highlights in Science, Engineering and Technology*, *85*, 247–255. https://doi.org/10.54097/6rn1wd81

Rostamian, A., & O'Hara, J. G. (2022). Event prediction within directional change framework using a CNN-LSTM model. *Neural Computing and Applications*, *34*(20), Article 20. https://doi.org/10.1007/s00521-022-07687-3

S, C., P, S., R, N. M., Mishra, P., & V, K. (2023). Probability analysis and rainfall forecasting using ARIMA model. *MAUSAM*, *74*(4), Article 4. https://doi.org/10.54302/mausam.v74i4.805

Sadorsky, P. (2021). Predicting Gold and Silver Price Direction Using Tree-Based Classifiers. *Journal of Risk and Financial Management*, *14*(5), Article 5. https://doi.org/10.3390/jrfm14050198

Si, Y., Nadarajah, S., Zhang, Z., & Xu, C. (2024). *Modeling opening price spread of Shanghai Composite Index based on ARIMA-GRU/LSTM hybrid model*. https://doi.org/10.1371/journal.pone.0299164

Skare, M., Gavurova, B., & Polishchuk, V. (2023a). A decision-making support model for financing start-up projects by venture capital funds on a crowdfunding platform. *Journal of Business Research, 158*, 113719. https://doi.org/10.1016/j.jbusres.2023.113719

Skare, M., Gavurova, B., & Polishchuk, V. (2023b). A large-scale decision-making model for the expediency of funding the development of tourism infrastructure in regions. *Expert Systems, 2023*, 13443. https://doi.org/10.1111/exsy.13443

Smolanka, V., Gavurova, B., & Polishchuk, V. (2024). Managing and evaluating the integration of information technologies in healthcare institutions. *Polish Journal of Management Studies, 30*(2), 297–313. https://doi.org/10.17512/pjms.2024.30.2.18

Varshini, A., Kayal, P., & Maiti, M. (2024). How good are different machine and deep learning models in forecasting the future price of metals? Full sample versus sub-sample. *Resources Policy*, *92*, 105040. https://doi.org/10.1016/j.resourpol.2024.105040

Vidal, A., & Kristjanpoller, W. (2020). Gold volatility prediction using a CNN-LSTM approach. *Expert Systems with Applications*, *157*, 113481. https://doi.org/10.1016/j.eswa.2020.113481

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), Article 3. https://doi.org/10.3390/pr11030862

Wang, X., Liu, H., Huang, S., & Lucey, B. (2019). Identifying the multiscale financial contagion in precious metal markets. *International Review of Financial Analysis*, *63*, 209–219. https://doi.org/10.1016/j.irfa.2019.04.003

Wei, Y., Wang, Y., Lucey, B. M., & Vigne, S. A. (2023). Cryptocurrency uncertainty and volatility forecasting of precious metal futures markets. *Journal of Commodity Markets*, *29*, 100305. https://doi.org/10.1016/j.jcomm.2022.100305

Yıldıran, C. U., & Fettahoğlu, A. (2017). Forecasting USDTRY rate by ARIMA method. *Cogent Economics & Finance*, *5*(1), 1335968. https://doi.org/10.1080/23322039.2017.1335968

Zhang, J., Ye, L., & Lai, Y. (2023). Stock Price Prediction Using CNN-BiLSTM-Attention Model. *Mathematics*, *11*(9), Article 9. https://doi.org/10.3390/math11091985