International Journal of Science and Engineering Applications
Volume 13-Issue 08, 13 – 20, 2024, ISSN:- 2319 - 7560
DOI: 10.7753/IJSEA1308.1003

Forecasting Gold Prices with MLP Neural Networks: A Machine Learning Approach

Arash Tashakkori
School of Business, Stevens Institute of Technology
Hoboken 07030,
New Jersey, USA.

Fatemeh Salboukh
Engineering and Applied Science, University of Massachusetts Dartmouth, USA.

Hossein Talebzadeh
Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

Mohammad Talebzadeh
Department of Civil and Environmental Engineering, Texas A&M University
College Station,
TX, 77840, USA

Lochan Deshmukh
College of Business and Technology, Sacred Heart University, 5151 Park Ave, Fairfield, CT 06825, USA.

Abstract: Predicting gold prices accurately is crucial for investors and policymakers alike, given gold's significance as a store of value and hedge against economic uncertainty. In this study, we propose a novel approach using Multilayer Perceptron (MLP) neural networks to forecast gold prices. Leveraging historical data on gold prices and relevant economic indicators, we trained an MLP neural network model. Our model achieved remarkable accuracy, with a prediction error for the test phase close to 0.001. This indicates the efficacy of MLP neural networks in capturing the complex relationships underlying gold price movements. Our research contributes to the growing body of literature on machine learning applications in financial forecasting and provides valuable insights for stakeholders in the gold market. Further exploration of this approach holds promise for enhancing gold price prediction models and informing investment decisions in the financial markets.
Keywords: Gold prediction; ANNs; MLP; Gold stock dataset.

1. INTRODUCTION

Navigating the complex landscape of financial markets, characterized by their volatility and multifaceted dynamics, presents an enduring challenge for investors and analysts alike. At the heart of this challenge lies the imperative to accurately forecast future stock prices, a pursuit essential for making informed investment decisions, mitigating risks, and optimizing returns within the ever-evolving market environment [1]. The significance of stock prediction reverberates throughout the realm of finance and investment. By providing insights into potential market trends and fluctuations, accurate forecasts empower investors to strategically navigate buying, selling, or holding securities, thereby maximizing profits and minimizing potential losses [2,3]. In a world where financial markets are influenced by a myriad of factors—ranging from economic indicators and geopolitical events to investor sentiment and market psychology—the ability to anticipate future price movements assumes paramount importance [4]. Furthermore, stock prediction plays a pivotal role in risk management strategies, enabling investors to hedge against adverse movements and safeguard their portfolios from unexpected downturns [5]. This proactive approach to risk mitigation is instrumental in preserving wealth and ensuring long-term financial stability. Beyond individual investors, the reliability of stock predictions bears broader implications for the overall health and stability of financial markets. Inaccurate forecasts can contribute to market volatility, speculative bubbles, and systemic risks, with profound consequences for economies and societies at large [6]. Conversely, dependable forecasts enhance market efficiency, liquidity, and investor confidence, fostering an environment conducive to sustainable economic growth and development [7].

Gold, as a timeless asset and a harbinger of economic stability, occupies a unique position within financial markets. The prediction of gold prices holds profound significance for investors seeking to diversify their portfolios and hedge against market volatility. Unlike stocks, gold serves as a reliable store of value, often sought after during periods of economic uncertainty or inflationary pressures. Therefore, accurate forecasting of gold prices not only informs individual investment decisions but also provides valuable insights into broader economic trends and geopolitical risks. Additionally, gold prices serve as barometers of market sentiment and risk appetite, reflecting shifts in investor perceptions and expectations. Given its global significance and intrinsic value, the ability to predict gold prices with precision is instrumental in navigating the intricacies of financial markets and fostering economic stability on both micro and macro levels.

Historically, stock prediction has primarily relied on traditional statistical models and fundamental analysis techniques. However, the emergence of machine learning, particularly the remarkable advancements in neural networks, has ushered in a new era of predictive analytics [8,9] These breakthroughs provide robust frameworks capable of processing massive datasets and discerning complex patterns that were once challenging to identify. At the forefront of this technological revolution are Multilayer Perceptron (MLP) networks, which have become indispensable tools in the realm of stock prediction. Multilayer Perceptron networks, a type of artificial neural network, have garnered widespread attention and adoption due to their ability to model nonlinear relationships and capture intricate dependencies within data [10,11]. By employing multiple layers of interconnected neurons, MLPs excel at learning complex representations of input-output mappings, making them well-suited for tasks like financial forecasting. The iterative training process, typically facilitated by algorithms like backpropagation, enables MLPs to continually refine their predictions based on observed data, iteratively adjusting the network's parameters to minimize prediction errors. The application of machine learning, including MLP neural networks, in gold price prediction has opened up new frontiers in financial analysis and decision-making. Unlike traditional econometric models, which may struggle to capture the nonlinear and dynamic nature of gold markets, machine learning techniques offer unparalleled flexibility and adaptability. By ingesting vast amounts of historical price data, along with relevant economic indicators and market variables, MLP neural networks can discern subtle patterns and relationships that may elude human analysts. This ability to uncover hidden insights enables investors and policymakers to make more informed decisions about gold investments, hedging strategies, and macroeconomic policies. Moreover, machine learning techniques like MLP neural networks have broader applications beyond gold price prediction, extending to portfolio optimization, risk management, algorithmic trading, and market sentiment analysis. As financial markets continue to evolve and become increasingly interconnected, the role of machine learning in driving insights and innovation will only grow in importance. By harnessing the power of MLP neural networks and other advanced machine learning methods, analysts and investors can gain a competitive edge in navigating the complexities of modern financial markets and capitalizing on emerging opportunities.

In this paper, we undertake a comprehensive investigation into the utilization of MLP neural networks for predicting gold prices, examining their constraints, and evaluating their potential impact on investment decision-making. Furthermore, we strive to bridge the divide between theoretical advancements and practical applications by scrutinizing the performance of MLPs within the distinctive domain of the gold stock market index, as provided in the Kaggle. We construct an MLP model using the gold dataset, encompassing real-time gold prices (in USD) spanning from 2012 to 2022. Through this endeavor, our objective is to enhance our comprehension of the effectiveness of MLPs across diverse financial landscapes and lay the groundwork for more informed and efficacious investment strategies.

2. RELATED WORKS

In this section, we present related work on the application of various methods for predicting gold prices, categorized based on their approaches. These approaches include methods applying established predictive models on new datasets without proposing new enhancements, methods introducing new techniques to improve existing models, and hybrid approaches closely related to our proposed methodology. Commodity prices, including gold, oil, silver, platinum, and others, exert significant influence on economic and financial sectors. In previous studies exploring general methods for predicting commodity prices, researchers employed established techniques such as Auto Regression of Integrated Moving Average (ARIMA), Artificial Neural Networks (ANN), K-Nearest Neighbors (K-NN), and Support Vector Machines (SVM). While these studies did not introduce innovative additions, their application of these methods on diverse datasets offers valuable research insights [12-15].

For instance, Uche-Ikonne Okezie [16] investigated gold price prediction using the ARIMA statistical method for the Indian local gold market, demonstrating the model's reliability in forecasting future gold prices based on 25 years of data. Livieris [17] conducted a comparative study of three machine learning methods—K-NN, SVM, and Naïve Bayes—for predicting gold prices obtained from Yahoo Finance archives, with K-NN exhibiting superior performance. Jabeur [18] proposed a linear regression model to predict gold prices based on gold price time series, crude oil prices, and historical US dollar data, achieving an accuracy of 85%. Kristjanpoller & Minutolo [19] explored the effectiveness of Linear Regression in gold price prediction, with their study suggesting high accuracy compared to other models such as ANN, ARIMA, and ANFIS [20].

Other studies delved into the efficacy of deep learning architectures for predicting gold prices. For instance, ul Sami & Junejo [21] compared classical ANN against ARIMA, with ANN outperforming ARIMA in both the training and testing phases. Sadorsky [22] investigated the reliability of Convolutional Neural Networks (CNN) for gold price prediction, suggesting that CNNs are among the best models suited for nonlinear time series. Hajek & Novotny [23] conducted a comparative study of three different neural network models—MLP, RNN, and LSTM—for predicting global iron prices, with LSTM emerging as the most reliable model.

3. METHODS AND MATERIALS

In this section, we outline the methodologies utilized in our research to predict gold stock prices using MLP networks. We begin by preprocessing the dataset, ensuring uniformity, and eliminating any discrepancies or anomalies that could impact model accuracy negatively [26]. Following this, we deploy a Multi-layer Perceptron neural network architecture to forecast future price trends [27].

3-1- Data

The dataset comprises real-time gold prices in USD spanning from 2012 to 2022. It includes several key features such as the date on which the price was noted, the closing price of gold in USD, the volume representing the sum of buys and sells of the gold commodity, as well as the open, high, and low prices of gold for each respective day. This comprehensive dataset provides a rich source of information for analyzing historical trends and patterns in gold prices, facilitating the development and evaluation of predictive models for forecasting future price movements.

3-2- Multi-layer Perceptron Network (MLP)

The FNN architecture, inspired by the organization of neurons in the human brain, comprises interconnected layers where information flows from input to output without any loops or cycles. Each layer consists of nodes, also known as neurons, which receive input from the previous layer, perform computations, and pass the result to the next layer. The input layer receives raw data, the hidden layers process this data through various transformations, and the output layer produces the final result. The connections between neurons are weighted, representing the strength of the connection.

An MLP is a type of FNN with multiple layers, including at least one hidden layer between the input and output layers. The presence of hidden layers enables MLPs to learn complex relationships and patterns within the data. Each neuron in the MLP computes a weighted sum of its inputs, adds a bias term, and passes the result through an activation function to produce the output. This process is repeated for each layer until the final output is generated. MLPs are trained using supervised learning algorithms, such as backpropagation, to adjust the weights and biases iteratively based on the error between the predicted and actual outputs. An MLP network consists of three layers. The input layer consists of neurons that receive the raw features of the data. Each neuron represents a feature, and the values of these neurons are the input to the network. Hidden layers perform computations on the input data to extract relevant features and patterns. These layers contain multiple neurons, each connected to every neuron in the previous layer. The number of hidden layers and neurons in each layer is a design choice that affects the model's capacity to learn complex relationships. Finally, the output layer produces the final result of the network. The number of neurons in this layer depends on the nature of the task—binary classification, multi-class classification, or regression. Each neuron in the output layer represents a class or a numerical value.

Moreover, activation functions introduce non-linearity to the model, enabling MLPs to learn complex mappings between inputs and outputs. Common activation functions include Sigmoid, Tanh, ReLU (Rectified Linear Unit), and Leaky ReLU. While Sigmoid and Tanh functions are bounded between 0 and 1 or -1 and 1, respectively, ReLU and Leaky ReLU functions introduce sparsity and address the vanishing gradient problem. For the model training, MLPs are trained using optimization algorithms, such as gradient descent, to minimize a loss function that quantifies the difference between predicted and actual outputs [28-31]. During training, the weights and biases of the MLP are adjusted iteratively to minimize the loss. Regularization techniques, such as L1 and L2 regularization, are often employed to prevent overfitting and improve generalization performance.

The parameters of an MLP include the input features, weights (w), and biases (b). The output of the MLP is computed through three main steps:

1- Initialization: Each input in the MLP network is assigned a weighted sum score. The weighted sum is calculated using the equation:

$$\text{weighted sum} = \sum_{i=1}^{n} w_{ij} X_i + b_j$$

(1)

Here, n represents the number of inputs in the MLP, wij is the weight vector linked to input i in hidden neuron j,X_i is the input number i, and b_j is the bias of hidden neuron$j$.

2- Activation function: The output of the weighted sum is processed using an activation function. While the Sigmoid activation function is commonly used, we opt for the Leaky ReLU function (Tsantekidis et al., 2017) for predicting gold prices. The Leaky ReLU function is defined as:

3-

$$\text{LeakyReLU} = \max(\alpha x, x)$$

(2)

his function ensures outputs within the range of -1 to +1, making it more suitable for our framework.

4- Final output computation: The output of the last layer is computed using the equation:

5-

$$\text{output} = \sum_{j=1}^{n} w_{jk} + b_k$$

(3)

Here, w_{jk} represents the weight linked from hidden neuron j to output neuron k, and b_k is the bias of output neuron k. Optimizing the values of weight and bias vectors is crucial for improving the performance of the MLP model, leading to better classification accuracy.

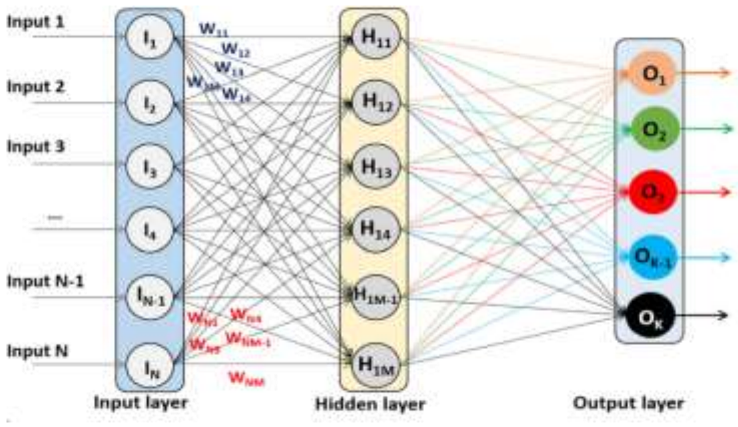

Figure 1Network structure of MLPs with only single one hidden layer

International Journal of Science and Engineering Applications
Volume 13-Issue 08, 13 – 20, 2024, ISSN:- 2319 - 7560
DOI: 10.7753/IJSEA1308.1003

4. EXPERIMENTS

4.1 Data visualization

In this section, we introduce several figures representing the data under analysis. First, we load the dataset containing historical gold price data. The dataset comprises columns such as 'Date', 'Close/Last', 'Volume', 'Open', 'High', and 'Low', each representing essential aspects of gold prices on different dates. Upon loading the data, we display the first five rows of the dataframe to provide a glimpse into the dataset's structure and contents. The displayed rows showcase information including the date, closing/last price, trading volume, opening price, highest price, and lowest price for gold on specific dates, facilitating further analysis and visualization of gold price trends over time.

Table 1. The first five rows of the dataset

	Date	Close/Last	Volume	Open	High	Low
0	2022-10-28	1648.3	186519.0	1667.2	1670.9	1640.7
1	2022-10-27	1668.8	180599.0	1668.8	1674.8	1658.5
2	2022-10-26	1669.2	183453.0	1657.7	1679.4	1653.8
3	2022-10-25	1658.0	178706.0	1654.5	1666.8	1641.2
4	2022-10-24	1654.1	167448.0	1662.9	1675.5	1648.0

To gain insights into the trend of gold prices over time, we focus on visualizing the closing prices. We selected the 'Close/Last' column. Subsequently, we generate a line plot of the closing prices. The plot is displayed allowing for clear visualization of the fluctuations in gold prices over the specified period.

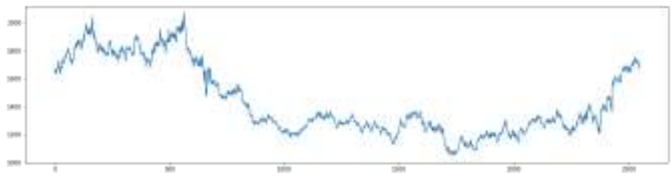

Figure 2 Closing prices over time

This visualization serves as a preliminary exploration of the data, offering a visual representation of the temporal evolution of gold prices and providing a foundation for further analysis and modeling.

The dataset contains a total of 2,547 entries, each representing a specific date. It comprises six columns: 'Date', 'Close/Last', 'Volume', 'Open', 'High', and 'Low'. The 'Date' column is of datetime64 data type, facilitating the handling of dates. The 'Close/Last' column, representing the closing or last price of gold on each date, is of float64 data type. The 'Volume' column contains the trading volume, with 2,508 non-null entries, indicating that some values are missing. The 'Open', 'High', and 'Low' columns represent the opening, highest, and lowest prices of gold on each date, respectively, all of which are of float64 data type. A descriptive summary of the dataset reveals insights into the distribution and statistics of the numerical columns. The 'Close/Last' column exhibits a mean closing price of approximately $1,437.56, with a standard deviation of $255.90, indicating considerable variability in closing prices. The trading volume ranges from a minimum of 1 unit to a maximum of 787,217 units, with an average volume of approximately 182,067.67 units. The 'Open', 'High', and 'Low' columns similarly showcase statistical measures such as mean, standard deviation, minimum, 25th percentile, median, 75th percentile, and maximum values, providing a comprehensive overview of gold price trends over the specified time period.

Table 2. Dataset properties

	Close/Last	Volume	Open	High	Low
count	2547.000000	2508.000000	2547.000000	2547.000000	2547.000000
mean	1437.557008	182067.668660	1437.743031	1447.083235	1427.891891
std	255.898467	97589.342619	256.239938	257.924158	253.641116
min	1049.600000	1.000000	1051.500000	1062.700000	1045.400000
25%	1243.450000	120901.000000	1243.000000	1251.000000	1235.250000
50%	1318.500000	168425.500000	1319.000000	1326.300000	1310.900000
75%	1698.100000	231754.000000	1701.450000	1715.300000	1684.000000
max	2069.400000	787217.000000	2076.400000	2082.100000	2049.000000

In this section, we utilize Seaborn's `pairplot` function to visualize pairwise relationships between different numerical columns in the dataset. The pairplot provides a grid of scatterplots for each pair of variables, allowing us to observe correlations, distributions, and potential patterns within the data. Each scatterplot within the grid represents the relationship between two variables, with points indicating individual data points. Additionally, diagonal axes display histograms for each variable, illustrating their distributions. This comprehensive visualization aids in identifying potential correlations or trends between different features of the dataset, facilitating further exploration and analysis of the data's underlying structure.

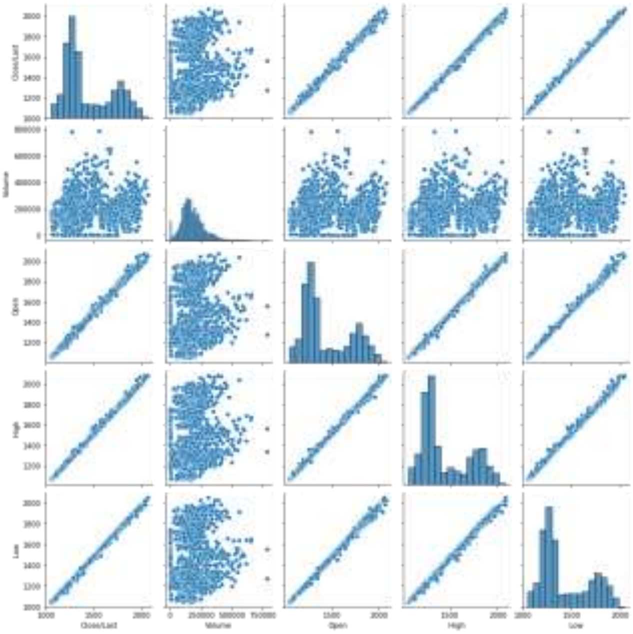

Figure 3 pairwise relationships between different numerical columns in the dataset

We generate a heatmap using Seaborn's heatmap function to visualize the correlation matrix of the numerical columns in the dataset. The correlation matrix quantifies the relationship between pairs of variables, with values ranging from -1 to 1. A value of 1 indicates a perfect positive correlation, -1 indicates a perfect negative correlation, and 0 indicates no correlation. The heatmap visualizes these correlation values using color gradients, with warmer colors representing stronger positive correlations and cooler colors representing stronger negative correlations. This visualization allows for a quick assessment of the relationships between different features in the dataset, aiding in the identification of potential patterns or dependencies among variables.

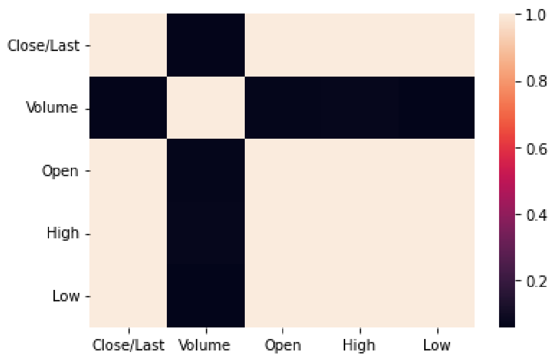

Figure 4 a heatmap to visualize the correlation matrix of the numerical columns in the dataset

4.2 Model training

In this section, we partition the dataset into training and testing sets. The dataset is divided into features (X) and target variables (y). We specify a test size of 20%, indicating that 20% of the data will be reserved for testing, while the remaining 80% will be used for training. Additionally, we set `shuffle` to False to maintain the order of the data. Upon splitting, the training set consists of 2006 samples, while the testing set comprises 502 samples. Both the training and testing sets for features (X) have a shape of (number of samples, number of features), with 3 features each. Similarly, the target variables (y) have shapes of (number of samples, 1), indicating a single target variable for each sample in both the training and testing sets. This partitioning scheme ensures that we have distinct subsets of data for training and evaluating our machine learning model's performance.

We visualize the target variables (y) for both the training and testing sets. We create two separate plots, each displaying the target variable values over time. The first plot represents the target variable values in the training set, while the second plot illustrates the target variable values in the testing set. These visualizations provide insights into the distribution and patterns of the target variable values in both the training and testing sets, aiding in the assessment of the data's characteristics and informing subsequent modeling and analysis.

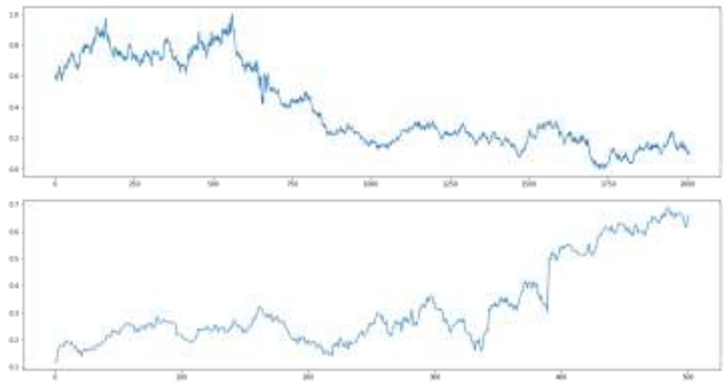

Figure 5 target variable values over time. Up: training data, Down: test data

```
Model: "sequential"
_________________________________________________________________
Layer (type)                Output Shape              Param #
=================================================================
dense (Dense)               (None, 40)                160
_________________________________________________________________
dense_1 (Dense)             (None, 40)                1640
_________________________________________________________________
dense_2 (Dense)             (None, 40)                1640
_________________________________________________________________
dense_3 (Dense)             (None, 1)                 41
=================================================================
Total params: 3,481
Trainable params: 3,481
Non-trainable params: 0
_________________________________________________________________
```

Figure 6 Model's summary

We construct an ANN model with three hidden layers, each containing 40 neuron nodes. The Rectified Linear Unit

(ReLU) activation function is employed for all hidden layers, facilitating non-linearity and enabling the model to capture complex relationships within the data. The output layer comprises a single neuron node, utilizing a linear activation function to yield continuous output values. To compile the ANN model, we utilize the Adam optimizer function and the mean squared error loss function. The Adam optimizer is a popular choice for training neural networks due to its adaptive learning rate and efficient optimization capabilities. Mean squared error (MSE) is a suitable loss function for regression tasks, measuring the average squared difference between predicted and actual values. Following compilation, we proceed to train the model using the fit method. We specify 100 epochs for training, with a batch size of 50 rows. The training process includes 20% cross-validation on the training set, facilitating model evaluation and preventing overfitting.

4.3 Results

We conducted experiments on a gold dataset to predict gold prices using Multilayer Perceptron (MLP). Our objective was to utilize the MLP architecture for forecasting future gold stock prices, leveraging historical data. Our analysis included training models on a subset of the dataset and assessing their performance using metrics like root mean squared error (RMSE) on both training and testing data. Employing the MLP model enabled us to capture complex patterns within the gold stock market data, enhancing the accuracy of our price forecasts for the future.

By plotting these metrics against the number of epochs, we visualized the model's learning progress. The x-axis represents the number of epochs, while the y-axis denotes the loss rate. The plotted curves illustrate how the loss rates change over the training process. The training loss curve typically decreases over epochs as the model learns from the training data, while the validation loss curve provides insight into the model's generalization performance on unseen data. By observing the convergence or divergence of these curves, we gain valuable insights into the model's training dynamics and its ability to generalize to new data.

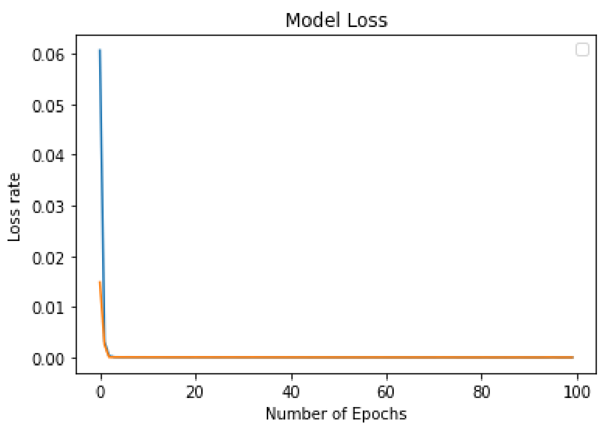

Figure 7 training and test error

The prediction results of a model applied to a test dataset is shown in Figure 8. Upon making predictions using the trained model on the test dataset, a plot is generated to compare these predicted values against the actual values. The figure showcases two lines: one representing the predicted values labeled as "Predicted value" and another depicting the actual values labeled as "Actual value.", while the x-axis labeled "Day" indicates the time progression or sequence of observations. The legend distinguishes between the predicted and actual values, aiding in interpretation.

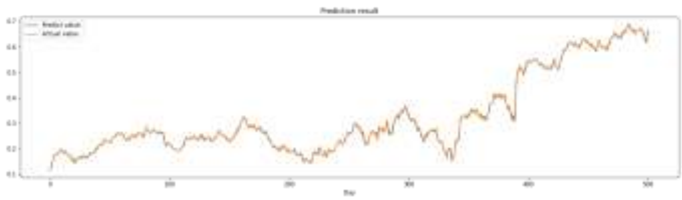

Figure 8 Prediction results by MLP

This visualization enables a direct comparison between the model's predictions and the ground truth, facilitating an assessment of the model's performance and its ability to accurately forecast future outcomes. Additionally, it's worth noting that the test error, measured at 0.001, provides a quantitative assessment of the model's performance. This metric indicates the average discrepancy between the model's predictions and the actual values in the test dataset. A test error of 0.001 suggests that, on average, the model's predictions deviate from the actual values by a very small margin. Such a low test error indicates that the model is performing well and has effectively learned patterns from the training data, allowing it to make accurate predictions on unseen data. This validation of the model's performance adds further confidence in its reliability for future predictions and applications.

5. CONCLUSION

In conclusion, our exploration into predicting gold prices using Multilayer Perceptron (MLP) models has yielded promising results. By leveraging historical data and employing the MLP architecture, we were able to develop a model capable of forecasting future gold stock prices with a high degree of accuracy. The visualizations of prediction results, juxtaposing predicted values against actual values, provide clear insight into the model's performance. Notably, the test error of 0.001 further validates the effectiveness of our model, indicating minimal deviation between predictions and actual outcomes. These findings underscore the potential of data-driven approaches, particularly MLP models and graph recurrent network [32], in capturing complex patterns within datasets. Moving forward, our study lays a foundation for further research and application of machine learning techniques in financial forecasting, with implications for investment strategies and risk management in the gold market and beyond. In future research, utilizing content analysis techniques [33] to analyze qualitative data sources alongside quantitative data [34] could provide a more comprehensive understanding of factors influencing gold prices and further improve forecasting models.

6. REFERENCES

[1] Bams, D., Blanchard, G., Honarvar, I., & Lehnert, T. (2017). Does oil and gold price uncertainty matter for the stock market?. *Journal of Empirical Finance*, *44*, 270-285.

[2] Khorsandi, H., & Bayat, M. (2022). Prioritizing operational strategies of saman bank. International Journal of Health Sciences, 6(S7), 1442–1453. https://doi.org/10.53730/ijhs.v6nS7.11548

[3] Sanaei, F. (2024). How customers' satisfaction change with the use of AR shopping application: A conceptuall model. *arXiv preprint arXiv:2401.10953*. https://doi.org/10.48550/arXiv.2401.10953

[4] Karabulut, E., Gholizadeh, F., & Akhavan-Tabatabaei, R. (2022). The value of adaptive menu sizes in peer-to-peer platforms. *Transportation Research Part C: Emerging Technologies*, *145*, 103948.

[5] Zhang, F., & Liao, Z. (2014). Gold price forecasting based on RBF neural network and hybrid fuzzy clustering algorithm. In Proceedings of the Seventh International Conference on Management Science and Engineering Management: Focused on Electrical and Information Technology Volume I (pp. 73-84). Springer Berlin Heidelberg.

[6] Hashemian, F., Maleki, N., & Zeinali, Y. (2024). From User Behavior to Subscription Sales: An Insight Into E-Book Platform Leveraging Customer Segmentation and A/B Testing. *Services Marketing Quarterly*, 1-29.

[7] Wen, F., Yang, X., Gong, X., & Lai, K. K. (2017). Multi-scale volatility feature analysis and prediction of gold price. *International Journal of Information Technology & Decision Making*, *16*(01), 205-223.

[8] Hashemi, A., Jang, J., & Beheshti, J. (2023). A Machine Learning-Based Surrogate Finite Element Model for Estimating Dynamic Response of Mechanical Systems. IEEE Access. 11, 54509-54525, 2023, doi: 10.1109/ACCESS.2023.3282453.

[9] Mirshekari, S., Moradi, M., Jafari, H., Jafari, M., & Ensaf, M. (2024). Enhancing Predictive Accuracy in Pharmaceutical Sales Through an Ensemble Kernel Gaussian Process Regression Approach. *International Journal of Computer and Information Engineering*, *18*(5), 255-260.

[10] Vahdatpour, M. S., & Zhang, Y. (2024). Latency-Based Motion Detection in Spiking Neural Networks. *International Journal of Cognitive and Language Sciences*, *18*(3), 150-155.

[11] Farhang, M., & Safi-Esfahani, F. (2020). Recognizing mapreduce straggler tasks in big data infrastructures using artificial neural networks. *Journal of Grid Computing*, *18*(4), 879-901.

[12] Dooley, G., & Lenihan, H. (2005). An assessment of time series methods in metal price forecasting. *Resources Policy*, *30*(3), 208-217.

[13] Aye, G., Gupta, R., Hammoudeh, S., & Kim, W. J. (2015). Forecasting the price of gold using dynamic model averaging. *International Review of Financial Analysis*, *41*, 257-266.

[14] Li, Y., Wang, S., Wei, Y., & Zhu, Q. (2021). A new hybrid VMD-ICSS-BiGRU approach for gold futures price forecasting and algorithmic trading. *IEEE Transactions on Computational Social Systems*, *8*(6), 1357-1368.

[15] Kiaghadi, M., Sheikholeslami, M., Alinia, A. M., & Boora, F. M. (2024). Predicting the performance of a photovoltaic unit via machine learning methods in the existence of finned thermal storage unit. *Journal of Energy Storage*, *90*, 111766.

[16] Uche-Ikonne Okezie, O., Offorha Bright, C., & Ukomah Henry, I. (2020). Forecasting Monthly Prices of Gold Using Artificial Neural Network. *Journal of Statistical and Econometric Methods*, *9*(3), 19-28.

[17] Livieris, I. E., Pintelas, E., & Pintelas, P. (2020). A CNN–LSTM model for gold price time-series forecasting. *Neural computing and applications*, *32*, 17351-17360.

[18] Jabeur, S. B., Mefteh-Wali, S., & Viviani, J. L. (2024). Forecasting gold price with the XGBoost algorithm and SHAP interaction values. *Annals of Operations Research*, *334*(1), 679-699.

[19] Kristjanpoller, W., & Minutolo, M. C. (2015). Gold price volatility: A forecasting approach using the Artificial Neural Network–GARCH model. *Expert systems with applications*, *42*(20), 7245-7251.

[20] He, Z., Zhou, J., Dai, H. N., & Wang, H. Gold price forecast based on LSTMCNN model. In2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech) 2019 Aug 5 (pp. 1046–1053).

[21] ul Sami, I., & Junejo, K. N. (2017). Predicting future gold rates using machine learning approach. *International Journal of Advanced Computer Science and Applications*, *8*(12).

[22] Sadorsky, P. (2021). Predicting gold and silver price direction using tree-based classifiers. *Journal of risk and financial management*, *14*(5), 198.

[23] Hajek, P., & Novotny, J. (2022). Fuzzy rule-based prediction of gold prices using news affect. *Expert Systems with Applications*, *193*, 116487.

[24] Liu, J., & Huang, X. (2021). Forecasting crude oil price using event extraction. *IEEE Access*, *9*, 149067-149076.

[25] Wu, J. Q., Wu, J. Q., & Chen, X. B. (2012, July). Import iron ore price forecasting based on PSO-SVMs model. In *2012 7th International Conference on Computer Science & Education (ICCSE)* (pp. 32-35). IEEE.

[26] Asadollahi, A., Latifi, H., Zeynali, S., Pramanik, M., & Qazvini, H. (2022). Accuracy of peak-power compensation in fiber-guided and free-space acoustic-resolution photoacoustic microscopy. *Biomedical Optics Express*, *13*(3), 1774-1783.

[27] https://www.kaggle.com/datasets/saikumartamminana/gold-price-prediction/data

[28] Manshour N, He F, Wang D, Xu D. Integrating Protein Structure Prediction and Bayesian Optimization for Peptide Design. InNeurIPS 2023 Generative AI and Biology (GenBio) Workshop 2023. https://openreview.net/forum?id=CsjGuWD7hk

[29] Larijani, A., & Dehghani, F. (2023). A Computationally Efficient Method for Increasing Confidentiality in Smart Electricity Networks. *Electronics*, *13*(1), 1-15. https://doi.org/10.3390/electronics13010170

[30] Ghafariasl, P., Mahmoudan, A., Mohammadi, M., Nazarparvar, A., Hoseinzadeh, S., Fathali, M., ... & Garcia, D. A. (2024). Neural network-based surrogate modeling and optimization of a multigeneration system. *Applied Energy*, *364*, 123130. https://doi.org/10.1016/j.apenergy.2024.123130

[31] Larijani, A., & Dehghani, F. (2023). An Efficient Optimization Approach for Designing Machine Models Based on Combined Algorithm. *FinTech*, *3*(1), 40-54.

[32] Mehrabian, A., Bahrami, S., & Wong, V. W. (2023, May). A dynamic Bernstein graph recurrent network for wireless cellular traffic prediction. In *ICC 2023-IEEE International Conference on Communications* (pp. 3842-3847). IEEE. https://doi.org/10.1109/ICC45041.2023.10279102

[33] Darvishinia, N. (2023). AI in Education: Cracking the Code Through Challenges: A Content Analysis of one of the recent Issues of Educational Technology and Society (ET&S) Journal. Partners Universal International Innovation Journal, 1(4), 61-71.

[34] Dokhanian, S., Sodagartojgi, A., Tehranian, K., Ahmadirad, Z., Moghaddam, P. K., & Mohsenibeigzadeh, M. (2024). Exploring the impact of supply chain integration and agility on commodity supply chain performance. *World Journal of Advanced Research and Reviews*, *22*(1), 441-450.