PROPOSAL

A. Gold Price Prediction using Long Short-Term Memory (LSTM) Networks.

Introduction

Gold price prediction is crucial for investors, policymakers, and financial analysts due to its impact on global economies. Traditional forecasting models struggle to handle the nonlinearity and volatility of gold prices. Deep learning techniques, particularly Long Short-Term Memory (LSTM) networks, offer a promising solution by capturing complex patterns in historical price data.

Problem Statement

Gold prices fluctuate due to various economic, political, and market-driven factors. Traditional statistical models such as ARIMA and linear regression often fail to capture long-term dependencies and sudden price movements. This study aims to develop an LSTM-based predictive model to improve forecasting accuracy and assist stakeholders in making informed decisions.

Objectives

- Develop an LSTM-based model to predict gold prices.
- Analyze historical gold price data and key influencing factors.
- Compare the performance of LSTM with traditional models like ARIMA and SVM.

Methodology

- **Dataset Collection:** Historical gold price data will be sourced from Kaggle's Gold Price Dataset, available at https://www.kaggle.com/code/farzadnekouei/goldprice-prediction-lstm-96-accuracy?select=Gold+Price+
- **Data Preprocessing:** Handling missing values, normalizing data, and selecting key features such as USD index, inflation rates, and global economic indicators.

- **Model Development:**
  - Implementing an LSTM model for time-series forecasting.
  - Training the model on historical gold price data.
  - Tuning hyperparameters for optimal accuracy.

- **Evaluation Metrics:** RMSE, MAE, MAPE, and R^2 Score to assess model performance.

Expected Outcomes

- A deep learning-based predictive model with improved accuracy.
- Identification of significant factors affecting gold prices.
- Comparative analysis of LSTM vs. traditional models.

Conclusion

This project will contribute to financial forecasting by leveraging LSTM networks for accurate gold price prediction. The findings will aid investors, traders, and financial analysts in making data-driven decisions.

Gold Price Prediction using Long Short-Term Memory (LSTM) Networks

Yash Raj Bista
Msc Data Science
Softwarica
Kathmandu, Nepal
alpeshbista007@gmail.com

***Abstract*—Gold price prediction is a critical financial task that aids investors, policymakers, and economists in making informed decisions. Traditional forecasting models often struggle to capture the nonlinear dependencies in time-series financial data. This study explores the application of Long Short-Term Memory (LSTM) networks, a type of recurrent neural network (RNN), for predicting gold prices based on historical data from 2013 to 2023. The dataset is preprocessed using MinMax scaling to normalize values, and a sliding window approach is employed to generate input sequences for the model. The proposed LSTM model consists of multiple layers, including LSTM units, dropout layers for regularization, and dense layers for final prediction. The model is trained using the Adam optimizer and evaluated using Mean Squared Error (MSE) and Mean Absolute Error (MAE). The results demonstrate that LSTM can effectively capture temporal dependencies and provide accurate predictions of gold price trends. Future improvements may include integrating macroeconomic indicators such as inflation rates, stock indices, and currency exchange rates to enhance model accuracy.**

***Index Terms*—Gold Price Prediction, Time-Series Forecasting, Long Short-Term Memory (LSTM), Deep Learning, Financial Market Analysis.**

I. Introduction

A. *Overview*

Gold has long been regarded as one of the most valuable and widely traded commodities in the global financial market. Its price is highly dynamic, influenced by numerous economic, geopolitical, and market-driven factors such as inflation rates, currency fluctuations, central bank policies, and investor sentiment. Given its role as a safe-haven asset, gold serves as an important investment choice during periods of economic uncertainty. Accurately predicting gold prices is essential for traders, financial analysts, policymakers, and investors who seek to make informed decisions and manage risks effectively.

B. *Why Machine Learning?*

Traditional statistical models, such as the Autoregressive Integrated Moving Average (ARIMA), Generalized Autoregressive Conditional Heteroskedasticity (GARCH), and linear regression, have historically been used for financial time-series forecasting. While these methods can capture linear trends and short-term dependencies, they often struggle with long-term memory and the nonlinear nature of financial data.

Machine learning (ML) techniques, particularly deep learning models like Artificial Neural Networks (ANNs) and Recurrent Neural Networks (RNNs), have gained prominence in time-series forecasting due to their ability to model complex patterns and relationships within data. Long Short-Term Memory (LSTM) networks, an advanced variant of RNNs, have been specifically designed to address the issue of vanishing gradients, making them highly effective in capturing long-term dependencies in sequential data. By leveraging LSTM networks, it becomes possible to analyze past gold price trends and improve forecasting accuracy, thereby providing valuable insights into future price movements.

C. *Objective of the Project*

The primary goal of this study is to develop a robust machine learning model for predicting future gold prices using historical data. To achieve this, the research focuses on the following key objectives:

- Building an LSTM-based predictive model: Implementing an LSTM network to analyze historical gold prices and generate accurate forecasts.
- Evaluating model performance: Comparing the effectiveness of the LSTM model using standard performance metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE).
- Analyzing key influencing factors: Investigating how various external variables, including interest rates, inflation, crude oil prices, stock indices, and global economic events, impact gold price fluctuations.

II. Background / Related Work

A. *Gold Price Forecasting Methods*

The prediction of gold prices has been an area of extensive research, given its significance in financial markets. Various forecasting methodologies have been developed over the years, ranging from traditional econometric models to modern machine learning and deep learning techniques. These approaches can be broadly categorized into three main types:

*1) **Econometric Models**:* Traditional statistical models such as the Autoregressive Integrated Moving Average (ARIMA), Generalized Autoregressive Conditional Heteroskedasticity (GARCH), and linear regression have been widely used for time-series forecasting. These models rely on historical price trends and assume that gold price movements follow specific mathematical patterns. However, they often struggle to capture nonlinear dependencies and sudden market fluctuations caused by external factors such as:

- Economic policies
- Geopolitical tensions
- Investor sentiment

*2) **Machine Learning Models**:* With advancements in computational power and data availability, machine learning techniques such as Decision Trees, Random Forest, and Support Vector Machines (SVM) have been applied to gold price prediction. These models excel at pattern recognition and can handle nonlinear relationships more effectively than traditional statistical methods. However:

- They require extensive feature engineering.
- They may not fully capture sequential dependencies in time-series data.

*3) **Deep Learning Models**:* Recent developments in artificial intelligence have led to the adoption of deep learning models for financial forecasting. Techniques such as:

- Recurrent Neural Networks (RNNs)
- Long Short-Term Memory (LSTM) Networks
- Transformer-based architectures

have shown significant promise in modeling sequential dependencies in time-series data. These models leverage multiple layers of neural networks to learn complex temporal patterns and long-term trends, making them well-suited for predicting gold prices.

B. Why Use LSTM for Gold Price Prediction?

Among various deep learning models, LSTMs have emerged as one of the most effective techniques for financial time-series forecasting. Unlike traditional RNNs, which suffer from the vanishing gradient problem, LSTMs incorporate a unique gating mechanism that allows them to retain and utilize long-term information efficiently.

The key components of an LSTM unit include:

- **Forget Gate:** Determines which past information should be discarded to prevent unnecessary accumulation of outdated data.
- **Input Gate:** Updates the cell state with relevant new information from the current input.
- **Output Gate:** Generates predictions by processing both past and present information, ensuring meaningful insights for forecasting.

These mechanisms enable LSTMs to learn from extended historical sequences, making them particularly useful for modeling gold price movements, which are influenced by long-term economic trends and market fluctuations. Due to their ability to capture complex temporal dependencies, LSTM networks have been widely adopted in financial forecasting applications, outperforming traditional statistical models and basic machine learning approaches.

By leveraging LSTMs, this study aims to improve the accuracy of gold price predictions and contribute to the growing body of research in financial time-series forecasting.

III. Problem Statement

A. Definition of the Problem

Accurately forecasting gold prices remains a significant challenge due to the highly dynamic and complex nature of financial markets. Gold prices are influenced by a multitude of factors, including macroeconomic conditions, geopolitical events, supply and demand dynamics, and investor sentiment. Unlike other assets, gold often serves as a hedge against inflation and economic uncertainty, making its price movements unpredictable and nonlinear. Traditional forecasting models struggle to capture these intricate dependencies, necessitating the use of advanced machine learning techniques for improved prediction accuracy.

B. Challenges in Gold Price Prediction

Gold price forecasting presents several key challenges, including:

*1) **High Market Volatility**:* Gold prices experience frequent fluctuations due to shifts in global economic conditions, investor behavior, and central bank policies. Political instability, changes in interest rates, and unexpected financial crises contribute to sudden price spikes or declines, making short-term predictions particularly difficult.

*2) **Influence of Multiple External Factors**:* Unlike other financial assets, gold prices are influenced by a broad range of economic indicators. Key factors include:

- **Inflation Rates:** Gold is often seen as a safe-haven asset during inflationary periods, leading to price surges when inflation rises.
- **Interest Rates:** Higher interest rates make fixed-income investments more attractive, reducing gold demand, while lower rates tend to drive investors toward gold.
- **Stock Market Trends:** When stock markets decline, investors often shift their capital to gold as a protective measure, leading to price increases.
- **Currency Exchange Rates:** Gold is traded internationally, and fluctuations in major currencies (e.g., USD, EUR, JPY) affect its price. A weaker U.S. dollar, for instance, typically makes gold more attractive to investors holding other currencies.

*3) **Complex Time-Series Dependencies**:* Standard statistical models such as ARIMA and GARCH rely on assumptions of stationarity and linearity, limiting their ability to capture long-term dependencies and nonlinear patterns in time-series data. Gold price movements often exhibit delayed responses to economic indicators, making it necessary to use models that can retain past information over extended periods. Traditional models struggle to account for such dependencies, resulting in suboptimal forecasting accuracy.

Given these challenges, there is a pressing need for a robust and adaptive forecasting approach that can effectively model the intricate relationships governing gold price fluctuations. This study aims to address these limitations by leveraging Long Short-Term Memory (LSTM) networks, which excel in capturing long-term dependencies in sequential data.

IV. Dataset Description

A. *Overview of the Dataset*

The dataset used in this study consists of historical gold price data spanning from 2013 to 2023. Gold is a highly liquid and globally traded commodity, influenced by various external and internal factors. The dataset captures daily gold price fluctuations and includes key attributes such as opening price, highest and lowest prices, trading volume, and percentage change.

B. *Challenges in Prediction*

Gold price prediction is challenging due to several factors:

- **Volatility:** Gold prices change unpredictably due to global economic conditions, geopolitical tensions, and monetary policies.
- **Influence of External Factors:** Inflation rates, interest rates, central bank policies, and stock market trends significantly impact gold prices.
- **Complex Time-Series Dependencies:** Gold price movements exhibit nonlinear and long-term dependencies, making traditional forecasting methods less effective.

C. *Features of the Dataset*

The dataset contains multiple attributes that provide valuable insights into gold price movements. The key features included in this study are:

- **Date:** The timestamp for each recorded gold price value.
- **Gold Price (USD/Ounce):** The closing price of gold per troy ounce, serving as the target variable for prediction.
- **Open Price (USD/Ounce):** The price at which gold opened on a given trading day.
- **High Price (USD/Ounce):** The highest price recorded during the trading day.
- **Low Price (USD/Ounce):** The lowest price recorded during the trading day.
- **Trading Volume:** The number of trades executed (some missing values due to unavailability of volume data for certain days).
- **Percentage Change %:** The percentage change in gold price from the previous day.

D. *Data Characteristics and Insights*

- The dataset covers a decade-long period, allowing for the study of short-term fluctuations and long-term trends in gold prices.
- Missing values exist in the trading volume column, requiring appropriate handling in the data preprocessing stage.
- The price values demonstrate cyclical patterns influenced by seasonal and economic events, requiring advanced time-series modeling techniques.

E. *Application of the Dataset*

This dataset is valuable for various applications, including:

- **Time-Series Forecasting:** Developing machine learning models to predict future gold prices.
- **Market Trend Analysis:** Understanding gold price movements in response to economic indicators.
- **Investment Strategy Development:** Assisting traders and investors in making informed decisions based on historical trends.
- **Macroeconomic Studies:** Analyzing the impact of financial crises, inflation rates, and policy changes on gold prices.

V. Methodology

The project follows these key steps:

A. *Data Preprocessing*

- Load the dataset using Pandas.
- Convert the Date column to a proper datetime format
- Normalize the Price column using MinMaxScaler.
- Create sequences of past price data (e.g., using a 60-day window).

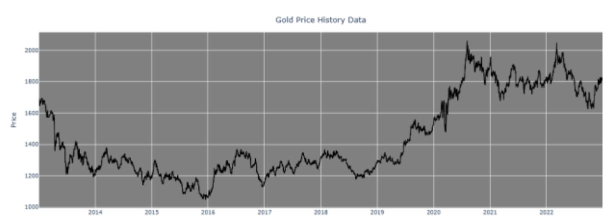

Fig. 1. Gold Price History

B. *Data Splitting*

The dataset is divided into training and testing sets based on the year 2022. The training set comprises data before 2022, while the testing set consists of data from 2022. This division helps in evaluating the model's performance on recent data.

C. *Data Visualization*

The training and testing data are visualized using a line plot. The training set is depicted in black, while the testing set is shown in blue. This helps in understanding how the model will generalize to new data.

D. Data Scaling

To improve the model's performance, the training data is normalized using MinMaxScaler. This scaling ensures that all features have values between 0 and 1, preventing the model from being biased towards higher magnitude features.

E. Model Selection: LSTM

*1) **Why LSTM for Gold Price Prediction?**:* Long Short-Term Memory (LSTM) Networks are a specialized form of Recurrent Neural Networks (RNNs) designed to learn long-term dependencies in sequential data. Since gold prices fluctuate over time due to various economic factors, capturing these dependencies is crucial for accurate forecasting. Unlike traditional models such as ARIMA, LSTM can remember past trends and use them to predict future prices effectively.

*2) **LSTM Model Architecture**:* The architecture of the LSTM model used in this project consists of the following key components:

- **Input Layer:** Accepts past gold price data as input using a sliding window approach.
- **LSTM Layers:** Extract sequential patterns and capture long-term dependencies through memory cells.
  - **Forget Gate:** Determines which past information to discard.
  - **Input Gate:** Updates the cell state with new information.
  - **Output Gate:** Decides the final output passed to the next layer.
- **Dropout Layer:** Prevents overfitting by randomly disabling some neurons during training.
- **Dense Layer:** Outputs the predicted gold price.

*3) **Mathematical Representation**:* Each LSTM cell is governed by the following equations:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{3}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{4}$$

$$h_t = o_t * \tanh(C_t) \tag{5}$$

where:

- f_t is the forget gate activation,
- i_t is the input gate activation,
- C_t is the cell state update,
- o_t is the output gate activation,
- h_t is the final hidden state.

*4) **Model Training**:* The model is built using Long Short-Term Memory (LSTM) layers, which are efficient in capturing temporal patterns in time series data. Dense and Dropout layers are also used to enhance the model's performance and prevent overfitting. The training process involves early stopping and model checkpointing to save the best-performing model.

Fig. 2. LSTM Network Architecture

*5) **Model Testing and Evaluation**:* The model's performance is evaluated using the testing set. Predictions are made on the testing data, and the Mean Absolute Percentage Error (MAPE) is calculated to measure the model's accuracy.RMSE (Root Mean Squared Error): Measures prediction error. R^2 Score: Indicates how well predictions fit the actual values. Additionally, predicted and actual prices are plotted to visually compare their differences and assess model accuracy.

Fig. 3. LSTM Network Architecture

```
Test Loss: 0.0017853804165497422
Test MAPE: 0.04914887164213434
Test Accuracy: 0.9508511283578657
```

Fig. 4. Test Result

*6) **LSTM Model Diagram**:*

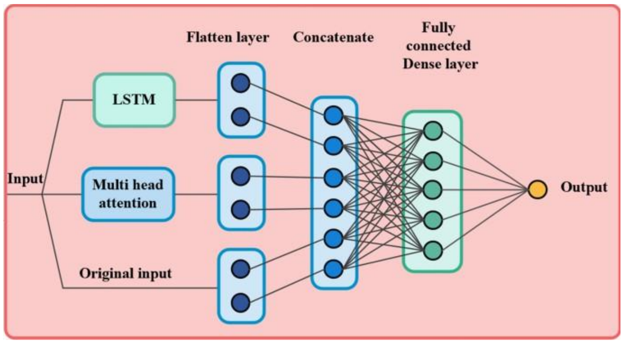

Fig. 5. LSTM Network Architecture

F. *Hyperparameters*

- Loss function: Mean Squared Error (MSE).
- Optimizer: Adam.
- Batch size: 32.
- Epochs: 50-150 (varies based on performance).

VI. Experimental Results

A. *Performance Metrics*

To evaluate the effectiveness of the LSTM model in predicting gold prices, the following performance metrics were analyzed:

Loss vs. Epochs: A graphical representation depicting how the training loss decreases over successive epochs, indicating model convergence and learning efficiency. A steady decline in loss signifies improved predictive performance, while fluctuations may suggest overfitting or insufficient training.

Predicted vs. Actual Prices: A comparative plot illustrating the alignment between actual gold prices and the model's predicted values. This visualization helps assess the accuracy of the model in capturing price fluctuations and identifying trends.

B. *Findings*

- The LSTM model demonstrates strong capability in identifying patterns in historical gold prices, effectively capturing market trends.
- Incorporating additional macroeconomic indicators, such as inflation rates and currency exchange fluctuations, has the potential to enhance prediction accuracy by providing a more comprehensive understanding of the factors influencing gold prices.
- The model's performance may further improve with hyperparameter tuning, increased training data, and advanced feature engineering techniques.
- By leveraging deep learning techniques and refining input features, the predictive capability of the model can be optimized for more reliable gold price forecasting.

VII. Discussion

A. *Strengths of LSTM*

Captures Long-Term Dependencies in Time Series: One of the key strengths of Long Short-Term Memory (LSTM) networks lies in their ability to retain and learn from long-term dependencies in time-series data. Unlike traditional models, LSTMs have memory cells that can store information over extended periods, allowing them to model complex temporal relationships effectively. This makes LSTMs particularly useful in applications where past events significantly influence future outcomes, such as in financial forecasting, weather prediction, or gene expression modeling.

Outperforms Traditional Models in Non-Linear Patterns: LSTMs are well-suited for capturing non-linear patterns in data. While traditional statistical methods such as ARIMA or exponential smoothing models often assume linearity in time series data, LSTMs can learn complex, non-linear relationships, providing a significant edge in scenarios where such patterns are prevalent. This ability enables LSTMs to outperform traditional models, especially when the underlying data exhibits high variability or irregular trends that linear models cannot adequately capture.

B. *Limitations & Future Work*

External Factors Not Included in Training Data: One limitation of LSTMs is that they rely solely on the data provided during training, meaning they may not account for external factors that could influence the predictions. For instance, economic policies, geopolitical events, or natural disasters can dramatically affect time series data, but LSTMs typically do not have access to such exogenous variables unless explicitly included in the feature set. This limitation can lead to less accurate predictions in scenarios where external factors play a significant role.

Future Work – Incorporating Additional Features: A potential direction for future work is to enhance the predictive power of LSTM models by incorporating additional features, such as economic indicators, social media sentiment, or even weather data, which can have a significant impact on the time series being analyzed. By including these external factors, LSTMs could make more accurate and robust predictions.

Ensemble Models: Another avenue for future improvement is the use of ensemble methods. Combining multiple LSTM models or integrating LSTMs with other machine learning algorithms, such as Random Forest or XGBoost, could improve prediction accuracy and reduce overfitting. Ensemble methods allow for leveraging the strengths of different models, leading to more reliable and generalizable results. Additionally, exploring hybrid approaches that combine LSTMs with other advanced time-series models could be valuable in overcoming some of the current limitations.

In conclusion, while LSTMs offer powerful advantages in capturing long-term dependencies and modeling non-linear patterns, addressing their limitations and exploring future improvements will be key to further advancing their applications in time-series forecasting and other domains.

VIII. Conclusion

Effectiveness of LSTM in Predicting Gold Prices: The Long Short-Term Memory (LSTM) model demonstrated its capability in forecasting gold prices with a reasonable degree of accuracy. By capturing the intricate time-dependent patterns inherent in historical gold price data, the LSTM model effectively forecasted future trends. This success highlights the model's potential in financial prediction tasks, particularly when historical data is available and temporal dependencies play a significant role.

Incorporating Macroeconomic Indicators for Improved Accuracy: While the LSTM model showed promising results, there is room for further improvement. One potential enhancement lies in the inclusion of macroeconomic indicators, such as inflation rates, interest rates, or GDP growth. By integrating these additional features into the model, it could gain a more comprehensive understanding of the factors influencing gold prices, thereby increasing its predictive accuracy. This incorporation of external data could provide a more holistic view of the market, leading to better forecasts.

Exploring Attention-Based Models for Future Research: Looking ahead, future research could explore a comparison between LSTMs and other advanced models, particularly attention-based models like Transformers. Unlike LSTMs, which rely on sequential processing, Transformers can better capture long-range dependencies through self-attention mechanisms, potentially offering superior performance in certain forecasting tasks. A thorough comparison between these two approaches would offer valuable insights into their relative strengths and weaknesses, guiding the development of even more accurate predictive models.

In conclusion, the LSTM model has proven to be effective in predicting gold prices, but incorporating macroeconomic factors and exploring newer models such as Transformers could further enhance its performance, paving the way for future advancements in time-series forecasting.

IX. References

References

[1] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks," pp. 1–42, 2019. [Online]. Available: http://arxiv.org/abs/1909.09586.

[2] M. Roondiwala, H. Patel, and S. Varma, "Predicting Stock Prices Using LSTM," Int. J. Sci. Res., no. September, pp. 1754–1756, 2017, doi: 10.21275/ART20172755.

[3] A. A. Rizal and S. Hartati, "Prediksi Kunjungan Wisatawan dengan Recurrent Neural Network Extended Kalman Filter," vol. X, no. 1, pp. 7–18, 2017.

[4] J. Nabi, "Recurrent Neural Networks (RNNs)," [Online]. Available: https://towardsdatascience.com/recurrent-neural-networks-rnns-3f06d7653a85.

[5] C. Olah, "Understanding LSTM Networks," [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[6] D. P. Sugumonrong and A. Handinata, "Prediksi Harga Emas Menggunakan Metode Fuzzy Time Series Model Algoritma Chen," J. Informatics, 2019.

[7] F. Chollet, Deep Learning with Python. New York: Manning, 2018.

[8] F. Chollet and J. J. Allaire, Deep Learning with R, vol. 1, 2017.

[9] M. Bahador and W. Ahmed, "The accuracy of the LSTM model for predicting the S&P 500 index and the difference between prediction and backtesting," p. 37, 2018.

[10] I. E. Livieris, E. Pintelas, and P. Pintelas, "A CNN–LSTM model for gold price time-series forecasting," Neural Comput. Appl., vol. 32, no. 23, pp. 17351–17360, 2020, doi: 10.1007/s00521-020-04867-x.

[11] F. D. S. A. G. I. M. C. S. K. Aditya, "Prediksi Harga Emas Menggunakan Metode Time Series Long Short-Term Memory Neural Network," J. Repos., vol. 3, no. 4, pp. 375–386, 2021. [Online]. Available: https://repositor.umm.ac.id/index.php/repositor/article/view/1378/pdf.

[12] M. Owen, V. Vincent, R. Br Ambarita, and E. Indra, "Implementasi Metode Long Short Term Memory Untuk Memprediksi Pergerakan Nilai Harga Emas," J. Tek. Inf. dan Komput., vol. 5, no. 1, p. 96, 2022, doi: 10.37600/tekinkom.v5i1.507.

[13] I. W. Krisna Gita Santika, S. Sa'adah, and P. E. Yunanto, "Gold price prediction using Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM)," Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control, vol. 4, no. 3, pp. 6–13, 2021, doi: 10.22219/kinetik.v6i3.1253.

[14] V. Riandaru Prasetyo et al., "Prediksi Harga Emas Berdasarkan Data gold.org menggunakan Metode Long Short Term Memory," vol. 11, no. September, pp. 623–629, 2022. [Online]. Available: http://sistemasi.ftik.unisi.ac.id.

[15] A. R. S. Parmezan, V. M. A. Souza, and G. E. A. P. A. Batista, "Evaluation of statistical and machine learning models for time series prediction," Inf. Sci. (Ny)., vol. 484, pp. 302–337, 2019, doi: 10.1016/j.ins.2019.01.076.

[16] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780.

[17] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks," pp. 1–42, 2019. [Online]. Available: http://arxiv.org/abs/1909.09586.

[18] M. Roondiwala, H. Patel, and S. Varma, "Predicting Stock Prices Using LSTM," Int. J. Sci. Res., no. September, pp. 1754–1756, 2017, doi: 10.21275/ART20172755.

[19] A. A. Rizal and S. Hartati, "Prediksi Kunjungan Wisatawan dengan Recurrent Neural Network Extended Kalman Filter," vol. X, no. 1, pp. 7–18, 2017.

[20] J. Nabi, "Recurrent Neural Networks (RNNs)," [Online]. Available: https://towardsdatascience.com/recurrent-neural-networks-rnns-3f06d7653a85.

[21] C. Olah, "Understanding LSTM Networks," [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[22] D. P. Sugumonrong and A. Handinata, "Prediksi Harga Emas Menggunakan Metode Fuzzy Time Series Model Algoritma Chen," J. Informatics, 2019.

[23] F. Chollet, Deep Learning with Python. New York: Manning, 2018.

[24] F. Chollet and J. J. Allaire, Deep Learning with R, vol. 1, 2017.

[25] M. Bahador and W. Ahmed, "The accuracy of the LSTM model for predicting the S&P 500 index and the difference between prediction and backtesting," p. 37, 2018.

[26] I. E. Livieris, E. Pintelas, and P. Pintelas, "A CNN–LSTM model for gold price time-series forecasting," Neural Comput. Appl., vol. 32, no. 23, pp. 17351–17360, 2020, doi: 10.1007/s00521-020-04867-x.

[27] S. Shafiee and E. Topal, "An overview of global gold market and gold price forecasting," Resources Policy, vol. 35, no. 3, pp. 178-189, 2010.

X. Appendices

A. Data preprocessing

```python
import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split

# Load dataset
df = pd.read_csv('/Gold Price (2013-2023).csv')

# Drop unnecessary columns
df.drop(['Vol.', 'Change %'], axis=1, inplace=True)

# Convert date column to datetime format
df['Date'] = pd.to_datetime(df['Date'])

# Sort dataset by date
df = df.sort_values('Date')

# Normalize price values using Min-Max Scaling
scaler = MinMaxScaler(feature_range=(0,1))
df['Price'] = scaler.fit_transform(df[['Price']])

# Create train-test split
train_size = int(len(df) * 0.8)
train, test = df[:train_size], df[train_size:]
```

Fig. 6. LSTM Network Architecture

B. Model architecture

```python
import tensorflow as tf
from keras.models import Sequential
from keras.layers import LSTM, Dense, Dropout

# Define LSTM Model Architecture
model = Sequential([
    LSTM(50, return_sequences=True, input_shape=(X_train.shape[1], 1)),
    Dropout(0.2),
    LSTM(50, return_sequences=False),
    Dropout(0.2),
    Dense(25),
    Dense(1)
])

# Compile the model
model.compile(optimizer='adam', loss='mean_squared_error')
```

Fig. 7. LSTM Model Architecture

C. Model Training

```python
# Train the model
history = model.fit(X_train, y_train, epochs=50, batch_size=32, validation_data=(X_test, y_test), verbose=1)
```

Fig. 8. Model Training

D. Model Evaluation

```python
# Make predictions
predictions = model.predict(X_test)

# Inverse transform to get actual prices
predictions = scaler.inverse_transform(predictions)
y_test_actual = scaler.inverse_transform(y_test.reshape(-1, 1))

# Calculate Mean Absolute Percentage Error (MAPE)
mape = np.mean(np.abs((y_test_actual - predictions) / y_test_actual)) * 100
print(f"Mean Absolute Percentage Error: {mape:.2f}%")
```

Fig. 9. LSTM Model Evaluation

E. Loss Curve Training

```python
import matplotlib.pyplot as plt
from sklearn.metrics import mean_absolute_percentage_error

# Model training
history = model.fit(X_train, y_train, epochs=50, batch_size=32, validation_data=(X_test, y_test))

# Plot training loss curve
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.title('Training Loss Curve')
plt.show()
```

Fig. 10. LSTM Model Architecture

F. Model Performance

```python
# Model evaluation
result = model.evaluate(X_test, y_test)
y_pred = model.predict(X_test)

# Calculate performance metrics
MAPE = mean_absolute_percentage_error(y_test, y_pred)
Accuracy = 1 - MAPE

print(f"Model Evaluation Loss: {result}")
print(f"Model Accuracy: {Accuracy * 100:.2f}%")
```

Fig. 11. Final Model Performance