National College of Ireland

National College of Ireland

Project Submission Sheet

Student Name:	ABHISHEK SUGA
Student ID:	X22234926
Programme:	MSC. DATA ANALYTICS **Year:** 2024-25
Module:	REESEARCH PROJECT
Lecturer:	PROF. ABDUL SHAHID
Submission Due Date:	12/12/24
Project Title:	Exploring the Effectiveness of Deep Learning Models in Forecasting Commodity Prices: A Case Study on Gold
Word Count:	5956

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the references section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature:	ABHISHEK SUGA
Date:	12/12/24

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

AI Acknowledgement Supplement

[Insert Module Name]

[Insert Title of your assignment]

Your Name/Student Number	Course	Date

This section is a supplement to the main assignment, to be used if AI was used in any capacity in the creation of your assignment; if you have queries about how to do this, please contact your lecturer. For an example of how to fill these sections out, please click here.

AI Acknowledgment

This section acknowledges the AI tools that were utilized in the process of completing this assignment.

Tool Name	Brief Description	Link to tool

Description of AI Usage

This section provides a more detailed description of how the AI tools were used in the assignment. It includes information about the prompts given to the AI tool, the responses received, and how these responses were utilized or modified in the assignment. **One table should be used for each tool used**.

[Insert Tool Name]	
[Insert Description of use]	
[Insert Sample prompt]	[Insert Sample response]

Evidence of AI Usage

This section includes evidence of significant prompts and responses used or generated through the AI tool. It should provide a clear understanding of the extent to which the AI tool was used in the assignment. Evidence may be attached via screenshots or text.

Additional Evidence:

[Place evidence here]

Additional Evidence:

[Place evidence here]

Exploring the Effectiveness of Deep Learning Models in Forecasting Commodity Prices: A Case Study on Gold

MSc Research Project
Data Analytics

Abhishek Suga
Student ID: X22234926

School of Computing
National College of Ireland

Supervisor: Prof. Abdul Shahid

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Abhishek Suga
Student ID:	X22234926
Programme:	Data Analytics
Year:	2018
Module:	MSc Research Project
Supervisor:	Prof. Abdul Shahid
Submission Due Date:	20/12/2018
Project Title:	Exploring the Effectiveness of Deep Learning Models in Forecasting Commodity Prices: A Case Study on Gold
Word Count:	5659
Page Count:	21

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Abhishek Suga
Date:	12th December 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Exploring the Effectiveness of Deep Learning Models in Forecasting Commodity Prices: A Case Study on Gold

Abhishek Suga
X22234926

Abstract

This research work presents how efficient deep learning models, such as the Long Short-Term Memory Network (LSTM), have proved in carrying out the forecasting of gold price with high frequency hourly time series data. Compared to other traditional methods, which usually take the input as daily or low resolution data, hourly data may provide finer temporal patterns that will yield more precise predictions. The key features that will form the basis of the dataset are: Open, High, Low, Close, and Volume. Later, this dataset is augmented further with some external economic indicators like crude oil prices, and volumes to see how this affects the performance of the model. Two models were tested in the experiments: XGBoost is a gradient-boosted decision tree algorithm, while LSTM is a neural network model designed for sequential data. Experiments were conducted in two settings: (1) using the original gold dataset and (2) augmenting the dataset with crude oil features. To have a strict comparison between different models, several metrics such as RMSE, R^2, MAE, and sMAPE were adopted. The results prove that LSTM is much better at capturing temporal dependencies and generalises well in all folds, even during volatile market conditions. However, XGBoost has shown variability, especially regarding outliers. The inclusion of crude oil features improved both models, thus confirming the integration of external economic indicators.

This research reveals the potential of high frequency data and deep learning techniques for financial forecasting, thus opening ways for more sophisticated approaches in further studies.

1 Introduction

Precise forecasting of commodity prices, particularly those of precious metals such as gold, has wide-reaching implications for many sectors, be it financial, manufacturing, or even the development of government policies. It has sometimes been termed a safe haven asset and will naturally tend to be highly sensitive to market sentiment, economic stability, fluctuations in currency, and geopolitical factors. The ability to predict gold prices will help investors maximise their portfolio profits , policymakers to make tactical decisions, and a lot of industries are dependent upon commodities to stabilise their overall growth performance.

Deep learning models have opened up new avenues lately in improving the accuracy of time series forecasting in complex nonlinear datasets. Traditional statistical methods dominated the domain of financial forecasting, and processes like ARIMA models and VAR have been usually confined to capturing intrinsic patterns and dependencies inherent in financial time series data. Some of the deep learning methods applied include LSTM networks, CNN, and hybrid models capable of learning temporal dependencies in high-dimensional data processing.

Commodity price forecasting is of paramount importance in financial analysis, especially for such a volatile and globally important metals as gold. Traditional time series forecasting models, including ARIMA, have been widely used, but they often fall short when it comes to modeling complex nonlinear dependencies in financial data. Recently, deep learning models, especially LSTM networks, have attracted interest due to their capabilities regarding long-term dependencies and non-linear relationships in time series data. Furthermore, hybrid approaches have also been developed that incorporate multiple models, such as ARIMA and LSTM, to capture the strengths of each method for better forecasting performance.

The purpose of this paper is to evaluate and analyse the performance of deep learning models for commodity price prediction. A deeper analysis into gold as a case study would help in generalising these models for the purpose of commodity price forecasting. This therefore means our work will involve training deep learning models and assessing them based on various historic gold price ,especially involving a more granular data than the regular monthly or weekly data and relevant macroeconomic variables in terms of prediction accuracy, generalisation capability, and computational efficiency.

2 Related Work

In this section, various studies carried out to identify methods of forecasting gold price are discussed. These range from classical statistical models such as ARIMA, to modern machine learning and deep learning techniques using LSTM and hybrid architectures. Table 2 presents the key methodology, dataset, and objectives, along with the critical appraisal of strengths and weaknesses for those works. Under the remarks column, we have used the + symbol to indicate the strengths of each study and the - symbol to point out their limitations. This systematic review provides a valuable overview of the existing approaches and forms a very good basis for our exploration.

Paper	**Method**	**Data**	**Purpose**	**Remarks**
[15]	Wavelet Transforms + ARIMA + LSTM	Historical share price data	Forecasting share price index futures	+ Wavelet transforms decompose complex data effectively, ARIMA captures linear dependencies, and LSTM handles non-linear patterns. - The model's high computational complexity limits real-time applicability, and it lacks interpretability.

Paper	Method	Data	Purpose	Remarks
[7]	ARIMA, RNN, LSTM	Historical Bitcoin price data	Predicting Bitcoin price direction	+ LSTM outperforms ARIMA by capturing non-linear dependencies, with Bayesian optimization for tuning. - Accuracy (52%) is low, marginally better than random guessing, and excludes external factors like macroeconomic variables.
[13]	SLP, MLP, RBF, SVM, SMA, ARIMA	Karachi Stock Exchange (KSE) data	Predicting market performance	+ Combines diverse features (e.g., oil rates, social media sentiment), with MLP outperforming traditional methods. - Lacks discussion on overfitting mitigation, and complex models may not handle external shocks effectively.
[6]	ARDL Model	Annual gold price data	Forecasting annual gold prices	+ Flexibly handles variables of different integration orders and includes significant indicators (e.g., gold demand, treasury bill rates). - Ignores structural breaks and long-term cointegration.
[4]	Bidirectional LSTM	Daily and monthly economic data (gold, S&P 500, USD, crude oil, CPI)	Forecasting gold prices	+ Accounts for 91.44% of variation in gold prices, with systematic hyperparameter tuning ensuring robust performance. - Does not include geopolitical factors, and model complexity limits real-time applicability.

Paper	Method	Data	Purpose	Remarks
[3]	SVAR model, STL-ETS, Neural Network, Bayesian Structural Time Series	Daily gold price, crude oil, US dollar index, and VIX data (2006–2019)	To analyze and forecast gold price returns, focusing on macroeconomic factors and market trends	+ SVAR identifies the dynamic impact of market factors; STL-ETS effectively fits short-term fluctuations; BSTS provides robust nowcasting. - Inclusion of multiple variables complicates short-term predictions; interpretability of neural networks is limited.
[1]	CNN-Bi-LSTM with Grid Search Hyperparameter Tuning	44 years of daily closing gold price data (1978–2021)	Forecasting daily closing gold prices with improved accuracy	+ Combines CNN's feature extraction with Bi-LSTM's sequence learning; employs automatic tuning for optimal hyperparameters; robust against extreme price fluctuations. - Requires extensive computational resources; performance depends heavily on dataset quality and preprocessing.
[9]	LSTM Network	Daily gold price data (1979–2020)	Predicting gold prices with improved handling of nonlinearities and long-term dependencies	+ Outperforms traditional methods like ARIMA in accuracy; capable of learning temporal patterns effectively. - Initial prediction deflections due to convergence time; further exploration of bidirectional LSTM models suggested for better results.

Paper	Method	Data	Purpose	Remarks
[2]	CNN-LSTM Model	Daily gold price data (2006–2021) with calculated price action features	Predicting gold price trends and improving trading signals by learning price action patterns	+ Combines CNN's pattern recognition with LSTM's sequence learning to capture complex price action relationships; achieves superior ROI, Sharpe ratio, and timing capacity compared to traditional and other deep learning methods. - Computationally intensive; results rely on the absence of transaction costs and assume a fixed risk-free rate.
[14]	VMD-BiGRU Model	IBM stock data (June 2020 – March 2022)	Time series prediction by combining Variational Mode Decomposition and Bidirectional GRU	+ Decomposes complex signals into stable components with VMD; BiGRU captures temporal dependencies effectively; achieves RMSE of 0.03 and NSE of 99.76- Computational complexity increases with the number of VMD modes; relies heavily on parameter optimization.
[8]	Ensemble Learning (Random Forest, Bagging, AdaBoost, LightGBM, XGBoost) with Feature Selection (PCA, RFE, Chi-Square Test)	Gold and diamond price data (Kaggle dataset with 10 features)	Predicting future prices of gold and diamonds with improved accuracy by combining feature selection and ensemble techniques	+ Combines feature selection methods to reduce dimensionality and enhance prediction accuracy; Random Forest with Chi-Square feature selection achieves highest accuracy (97.54- Models like AdaBoost underperform due to overfitting; preprocessing requirements increase computational time.

Paper	Method	Data	Purpose	Remarks
[10]	CNN-LSTM with Gramian Angular Field (GAF)	Daily gold price data from Yahoo Finance (2013–2022)	Predicting gold price trends using images as time-series data representation to improve accuracy	+ Combines CNN's feature extraction with LSTM's temporal modeling; GAF transforms time-series data into images for better pattern recognition; achieves 78.82- Computationally intensive; performance gain from additional features is limited.
[5]	ICEEMDAN-LSTM-CNN-CBAM	COMEX daily gold price and World Gold Association data (2010–2020)	Forecasting gold futures and spot prices with high accuracy by combining decomposition and hybrid modeling	+ Combines ICEEMDAN for signal decomposition, LSTM for sequence memory, CNN for feature extraction, and CBAM for attention refinement; achieves RMSE of 6.25 and R-Square of 0.9992 for gold futures prices. - Computational complexity due to multi-level decomposition and hybrid modeling.
[11]	Modified Long-Term Trend Reverting Jump and Dip Diffusion Model	Historical gold price data (1968–2008)	Forecasting gold price trends over the next 10 years by modeling long-term trends, jumps, and dips	+ Accounts for non-linear price movements; introduces jump and dip components for improved accuracy; validates model using historical data and predicts long-term trends. - Assumes a constant volatility range for future predictions; limited to macroeconomic variables like oil price and inflation.

Paper	Method	Data	Purpose	Remarks
[12]	ARIMA (Box-Jenkins Methodology)	Monthly gold price data in INR (1997–2017)	Forecasting gold prices in India using statistical time-series analysis	+ ARIMA (1,1,3) accurately predicts future prices with less than 2- Relies on stationary data and assumes linearity; limited capability to handle external economic shocks.

Table 1: Summary of Related Work

Based on the results of these studies, we propose a new gold price forecasting method. We will use hourly gold price data, covering a wide historical range, to test the predictive performance of XGBoost as the baseline model. Moreover, advanced deep learning models such as LSTM will be used to analyse their effectiveness compared to traditional methods. This dual exploration therefore aims at leveraging the advantages of both ensemble-based and deep learning techniques to advance the field of identifying those models that best capture gold price trends complexities.

3 Methodology

In this section, we detail the research methodology, experimental setup, data preprocessing steps, and evaluation techniques employed to achieve the objectives of this study. Our methodology builds on the related work discussed in Section 2 and follows a structured approach to ensure reproducibility and scientific rigor.

3.1 Architecture Overview

The Architecture Diagram of the Experimental Framework for Evaluating the effectiveness of Deep Learning Models on Gold Price Data is designed to systematically integrate data pre-processing, feature engineering, and machine learning models, as illustrated in Figure 1.This diagram provides a visual representation of the data flow and the interactions between different components of the methodology, from data collection to evaluation.

3.2 Data Collection and Preprocessing

The dataset used in this research consists of hourly gold price data with features such as *Open*, *High*, *Low*, *Close*, and *Volume*. Additional economic indicators, such as crude oil prices (Close and Volume), were included to enrich the dataset. The final dataset contained 80,145 records after preprocessing.

Data Cleaning and Filtering. We filtered out incomplete days to ensure consistent time intervals, retaining only those with 23 hourly records. Missing values were handled using forward and backward filling techniques. The dataset was indexed by *DateTime* for time-series analysis.

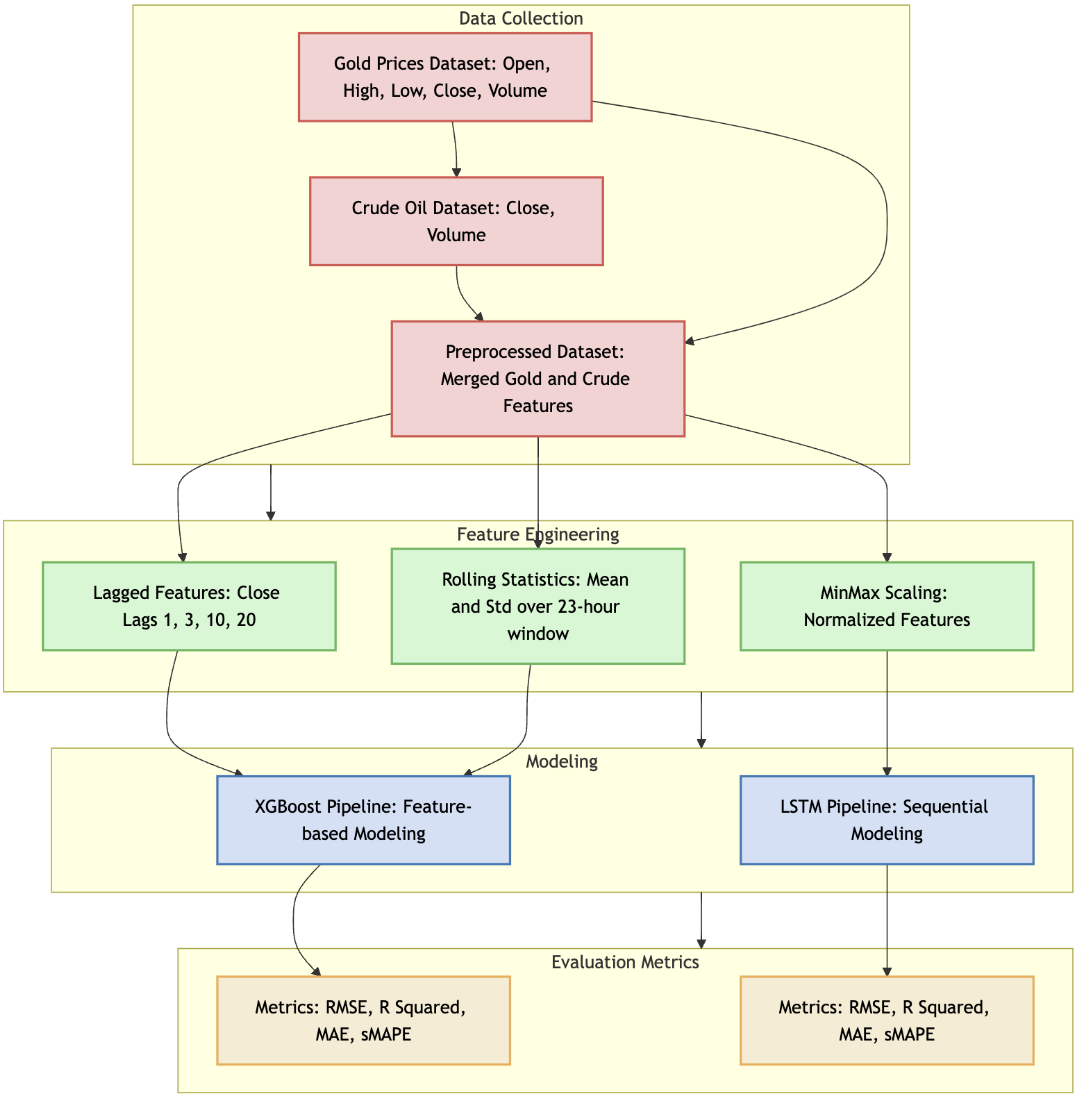


Figure 1: Architecture Diagram of the Experimental Framework

Feature Scaling. All numerical features were normalized using the Min-Max Scaler, scaling values to the range [0, 1]. This ensured better convergence during model training, particularly for LSTM.

3.3 Feature Engineering

For the XGBoost model, extensive feature engineering was applied:

- **Lagged Features:** Added lagged values of the *Close* price at intervals of 1, 3, 10, and 20 hours to capture temporal dependencies.
- **Rolling Statistics:** Calculated rolling mean and standard deviation for the *Close* price over a 23-hour window to capture trends and volatility.

For the LSTM model, the raw sequential data was used without additional feature engineering, leveraging the model's ability to capture temporal dependencies through its architecture.

3.4 Experimental Setup

Model Implementation. Two machine learning models were implemented and compared:

- **XGBoost:** A tree-based ensemble model optimized for regression tasks, relying on engineered features to learn temporal patterns.
- **LSTM:** A deep learning model with two LSTM layers (50 units each), dropout layers (rate: 0.2), and a dense output layer. The input sequences were structured with a look-back window of 23 hours.

Data Splitting. We employed a *TimeSeriesSplit* cross-validation strategy with 5 folds, ensuring chronological order in training and testing data. Each fold's test set comprised contiguous blocks of time-series data, avoiding data leakage.

3.5 Evaluation Metrics

We assessed the model performance using the following metrics:

- **Root Mean Squared Error (RMSE):** Measures the average magnitude of the prediction error.
- **R-Squared (R^2):** Indicates the proportion of variance explained by the model.
- **Mean Absolute Error (MAE):** Represents the average absolute error between predicted and actual values.
- **Symmetric Mean Absolute Percentage Error (sMAPE):** A robust metric for time-series data, effective for handling near-zero values.

3.6 Procedure

The experimental procedure included the following steps:

1. The cleaned dataset was split into training and testing sets using *TimeSeriesSplit*.
2. For XGBoost:
   - Lagged features and rolling statistics were added to the dataset.
   - The model was trained and evaluated on each fold of the cross-validation split.
3. For LSTM:
   - Sequential data was reshaped into 3D tensors with shape (samples, timesteps, features).
   - The LSTM model was trained for 10 epochs with a batch size of 32.
4. Evaluation metrics were computed for each fold, and results were averaged to ensure robust comparisons.

3.7 Tools and Frameworks

The experiments were conducted using the following tools and frameworks:

- **Programming Language:** Python 3.8
- **Libraries:**
  - *pandas* and *numpy* for data manipulation and preprocessing.
  - *scikit-learn* for evaluation metrics and cross-validation.
  - *xgboost* for the XGBoost model.
  - *tensorflow* and *keras* for deep learning implementation (LSTM).
  - *matplotlib* and *seaborn* for visualizations.

4 Design Specification

This section outlines the techniques, architectures, frameworks, and requirements underlying the implementation of the models used in this study. The design aims to ensure the robustness, scalability, and efficiency of the gold price forecasting system.

4.1 Techniques and Architecture

XGBoost Architecture. XGBoost, an efficient implementation of gradient-boosted trees, was used for regression tasks in this study. The model combines additive tree models with a regularized objective to prevent overfitting. The primary design involved:

- Adding lagged features and rolling statistics to explicitly capture temporal patterns in the data.
- Using the 'reg:squarederror' objective function, optimized for continuous target values.
- Employing tree-structured boosting to iteratively minimize the mean squared error (MSE).

This architecture was selected for its proven efficiency in handling tabular data and its ability to incorporate feature engineering for time-series forecasting.

LSTM Architecture. The Long Short-Term Memory (LSTM) model, a type of recurrent neural network (RNN), was implemented to capture sequential dependencies in the data. The LSTM design included:

- **Input Layer:** A 3D tensor of shape (samples, timesteps, features), representing 23-hour look-back windows.
- **Two LSTM Layers:** Each with 50 units and 'tanh' activation, designed to learn long-term dependencies.
- **Dropout Layers:** Dropout rates of 0.2 were added after each LSTM layer to mitigate overfitting.

- **Dense Output Layer:** A single neuron with a linear activation function for regression output.
- **Loss Function:** Mean Squared Error (MSE).
- **Optimizer:** Adam optimizer for adaptive learning rate optimization.

The LSTM's design leverages its inherent ability to model temporal patterns without requiring explicit feature engineering.

4.2 Framework and Tools

The implementation was supported by the following frameworks and tools:

- **Python 3.8:** The programming language for data processing, modeling, and evaluation.
- **pandas and numpy:** For efficient data manipulation and numerical computations.
- **scikit-learn:** For data splitting, scaling, and evaluation metrics.
- **xgboost:** For gradient-boosted tree regression.
- **tensorflow and keras:** For deep learning model (LSTM) development.
- **matplotlib:** For generating visualizations of model predictions and evaluation metrics.

4.3 Associated Requirements

The following requirements were identified to ensure reproducibility and scalability:

- **Hardware:** Experiments were conducted on a machine with at least 16GB RAM and a dedicated GPU (optional) for LSTM training.
- **Software Dependencies:** All required Python libraries are specified in the 'requirements.txt' file.
- **Data Requirements:** A preprocessed dataset with hourly gold price data and complete 23-hour cycles was mandatory for training and evaluation.
- **Execution Environment:** The models were executed in Jupyter Notebook for step-by-step implementation and analysis.

4.4 Algorithm Functionality

The following steps describe the functionality of the implemented models:

1. For XGBoost:
    - Feature engineering (lagged features and rolling statistics) was applied to enhance temporal context.

- The model iteratively added decision trees to minimize prediction error on the training data.

2. For LSTM:
   - Raw sequential data was divided into overlapping sequences of 23-hour timesteps.
   - The LSTM layers processed each sequence, learning both short-term and long-term dependencies.
   - The output layer predicted the next hour's gold price.

5 Implementation

The implementation of the proposed solution involved the integration of data preprocessing, feature engineering, and machine learning models for predicting gold prices under various experimental settings. This section describes the final outputs, tools, and methods used in the implementation process.

5.1 Data Preprocessing

The historical gold price dataset and additional economic indicators, such as crude oil prices, were preprocessed to ensure data consistency and readiness for modeling:

- Filtered incomplete days to ensure full 23-hour cycles in the gold dataset.
- Integrated crude oil data by merging relevant features (close price and volume) with the gold dataset using a datetime index.
- Applied MinMax scaling to normalize the data for machine learning models.

Outputs:

- A clean, normalised dataset with relevant features for modeling.

Tools Used:

- **Languages:** Python
- **Libraries:** `pandas` for data manipulation, `numpy` for numerical computations, `sklearn` for scaling.

5.2 Feature Engineering

Feature engineering was essential for preparing the data for model training:

- Generated lagged features for the `Close` price at intervals of 1, 3, 10, and 20 hours.
- Computed rolling mean and standard deviation for the `Close` price over a 23-hour window to capture short-term trends.
- Converted the data into a sequence format for LSTM models.

Outputs:

- A structured dataset enriched with temporal features for XGBoost and sequential data for LSTM.

Tools Used:

- **Languages:** Python
- **Libraries:** `pandas` for feature generation, `numpy` for efficient computations.

5.3 Model Development

Two machine learning models were implemented and evaluated under two settings: using the original gold dataset and incorporating crude oil features:

- **XGBoost:** Used lagged features and rolling statistics as inputs. Trained and cross-validated using 5 folds to ensure chronological order.
- **LSTM:** Developed a sequential model with two LSTM layers, dropout for regularisation, and a dense output layer. Trained on sequential data using 5-fold cross-validation to evaluate model robustness.

Outputs:

- Trained XGBoost and LSTM models capable of predicting hourly gold prices.
- Evaluation metrics such as RMSE, R^2, MAE, and sMAPE for model performance assessment.

Tools Used:

- **Languages:** Python
- **Libraries:** `xgboost` for XGBoost modeling, `keras` and `tensorflow` for LSTM implementation, `sklearn` for evaluation metrics.

5.4 Visualisation

Comprehensive visualisations were created to analyse and interpret model performance:

- Plotted actual vs. predicted values for all 5 folds for each experiment.
- Generated error distribution plots and rolling mean comparisons for XGBoost on the original dataset.
- Combined fold-specific plots into single visualisations for better presentation.

Outputs:

- Detailed plots showing model predictions, error trends, and feature analysis.

Tools Used:

- **Libraries:** `matplotlib` and `seaborn` for visualization.

5.5 Final Outputs

The implementation produced the following key outputs:

- **Preprocessed Datasets:** Cleaned and transformed data ready for modeling.
- **Developed Models:** XGBoost and LSTM models trained under two experimental settings.
- **Evaluation Metrics:** Performance metrics (RMSE, R^2, MAE, sMAPE) for each fold and experiment.
- **Visualisations:** Combined plots for actual vs. predicted values, error distributions, and rolling mean comparisons.

6 Evaluation

6.1 Evaluation Metrics

The evaluation metrics for all experiments (5 folds each) are summarized in Table 2.

Table 2: Consolidated Performance Metrics Across Experiments

Experiment	**Fold**	**RMSE**	R^2	**MAE**	**sMAPE (%)**
XGBoost - Original	1	0.2049	-0.5475	0.1482	35.28
	2	0.0155	0.9631	0.0099	2.44
	3	0.0008	0.9994	0.0006	0.15
	4	0.0153	0.9775	0.0067	1.03
	5	0.0749	0.4300	0.0348	4.29
LSTM - Original	1	0.0094	0.9967	0.0070	1.48
	2	0.0078	0.9907	0.0065	1.45
	3	0.0020	0.9961	0.0014	0.36
	4	0.0037	0.9987	0.0027	0.45
	5	0.0155	0.9757	0.0087	1.06
XGBoost - Crude	1	0.1270	-2.3528	0.1100	21.72
	2	0.0028	0.9934	0.0020	0.54
	3	0.0006	0.9994	0.0004	0.11
	4	0.0126	0.9710	0.0063	0.95
	5	0.0819	0.3417	0.0405	5.03
LSTM - Crude	1	0.0059	0.9928	0.0043	0.83
	2	0.0071	0.9573	0.0066	1.73
	3	0.0017	0.9940	0.0012	0.30
	4	0.0063	0.9927	0.0056	0.95
	5	0.0175	0.9699	0.0110	1.38

This section provides a comprehensive analysis of the experimental results and findings. The experiments evaluate the performance of XGBoost and LSTM on the gold price prediction task under two settings: 1) using the original gold dataset and 2) incorporating crude oil features. Results are presented using a consolidated metrics table, visualizations, and error analysis for a rigorous comparison.

6.2 Visualisations for Actual vs Predicted

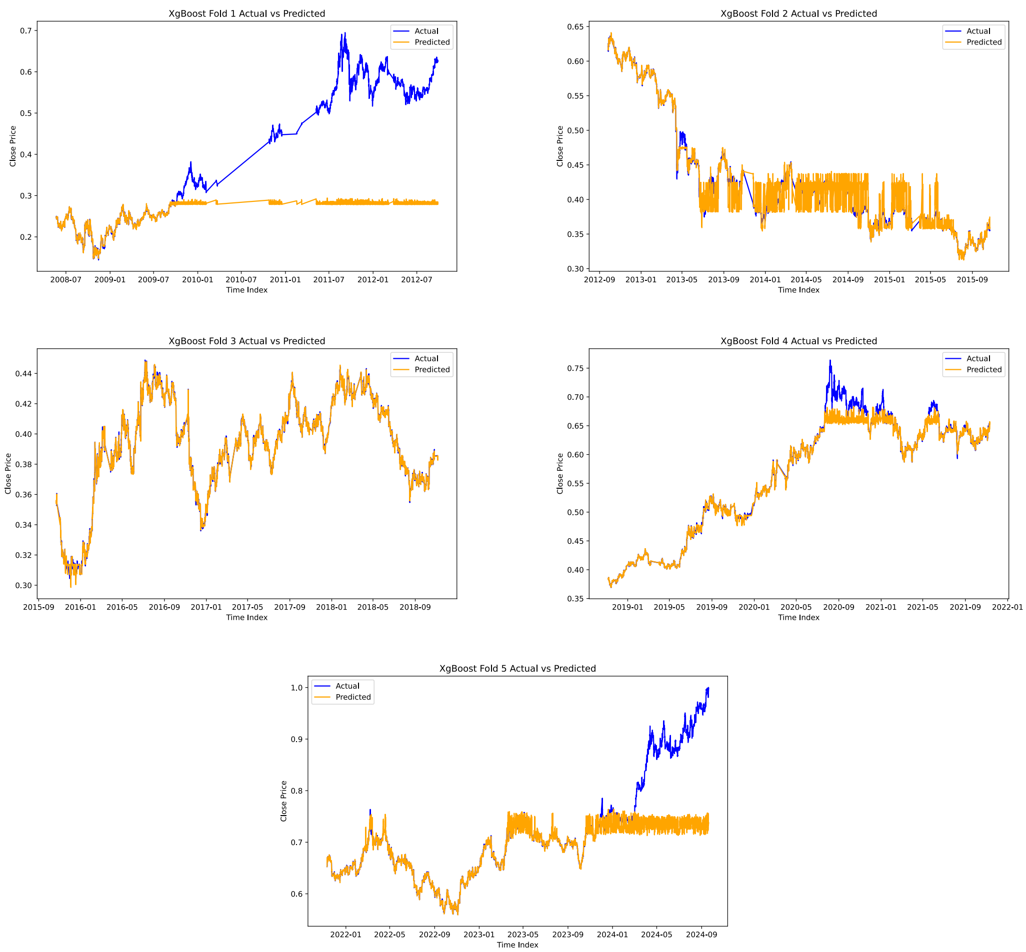
Each subplot in figure 2,3,4,5 shows the actual versus predicted prices for a given fold for all the four experiments , providing insights into the predictive accuracy of the models and their ability to capture market dynamics.

6.2.1 XGBoost - Original Gold Dataset

Discussion on XGBoost performance using the original dataset, highlighting variability across different market conditions and sensitivity to outlier movements and non-linear trends in gold prices.

The XGBoost model's performance on the original dataset reveals variability across the folds. While Folds 2, 3, and 4 show strong predictive capabilities with low RMSE and high R^2, Fold 1 suffers from significantly poorer performance, as seen in the negative R^2 value. This fold's deviation suggests potential challenges with anomalies or specific market conditions.

Figure 2: Actual vs. Predicted Results for XGBoost (Original Dataset)

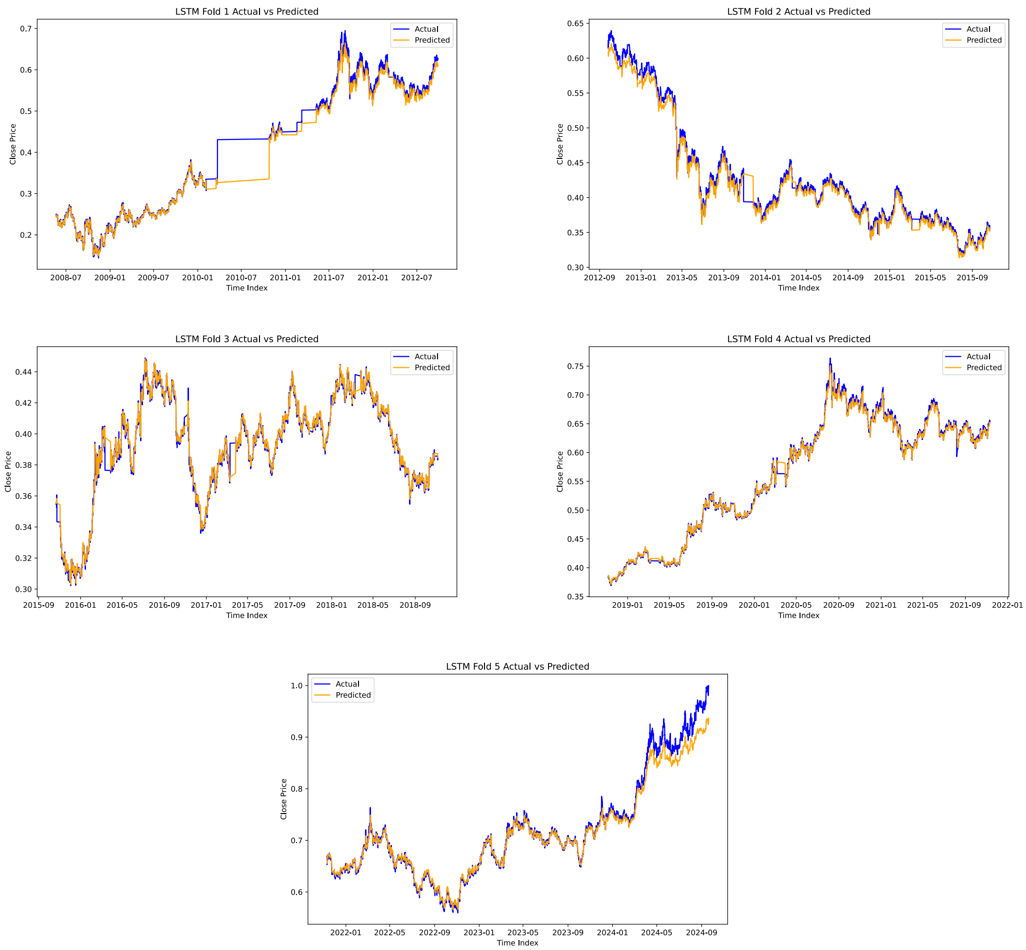


6.2.2 LSTM - Original Gold Dataset

LSTM consistently outperformed XGBoost across all folds. The model's sequential learning capability enabled it to effectively capture complex temporal patterns,which we can see reflected in its high R^2 values and low RMSE.

This consistency suggests that LSTM is better suited for time-series problems with nonlinear dependencies.

Figure 3: Actual vs. Predicted Results for LSTM (Original Dataset)

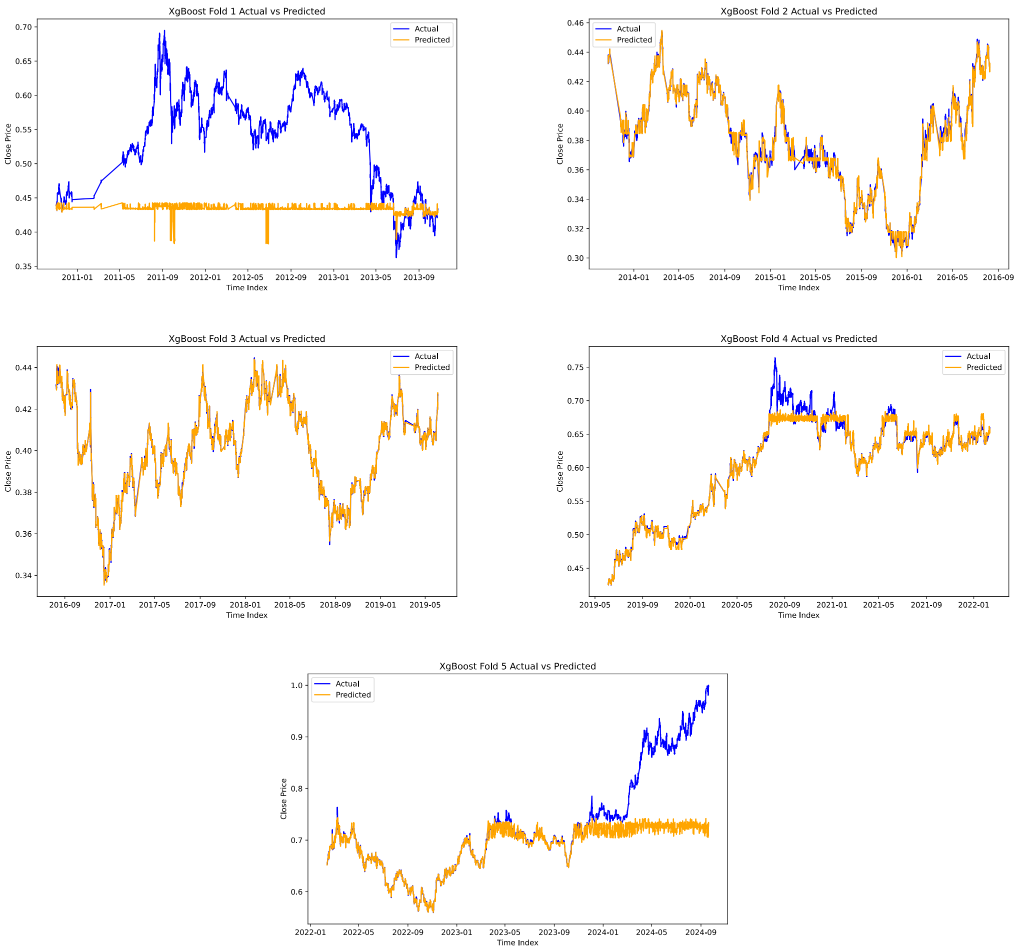


6.2.3 XGBoost - With Crude Oil Features

Adding crude oil features to the dataset improved XGBoost's performance in certain folds (e.g., Fold 2), highlighting the relevance of economic interdependencies between gold and crude oil prices.

However, the variability in performance persists, indicating that the model struggles with complex temporal dynamics even with enriched data.

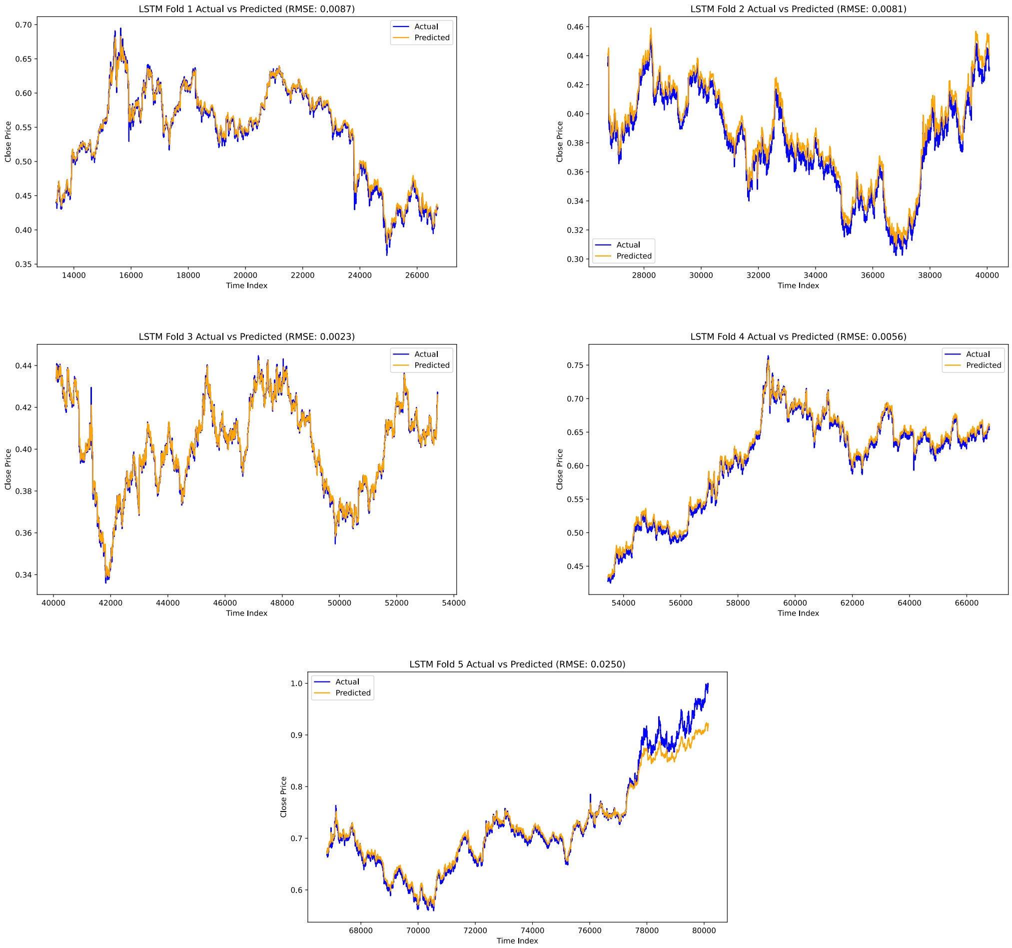
Figure 4: Actual vs. Predicted Results for XGBoost (with Crude Oil Dataset)



6.2.4 LSTM - With Crude Oil Features

With the inclusion of crude oil features, LSTM maintained its strong predictive performance across all folds. The ability of LSTM to integrate external indicators and adapt to dynamic market conditions further validates its effectiveness for time-series forecasting tasks in financial domain .

Figure 5: Actual vs. Predicted Results for LSTM (with Crude Oil Dataset)



7 Conclusion and Future Work

This study set out to investigate how well machine learning and deep learning models can forecast gold prices using hourly data. More precisely, we compare two approaches: XGBoost, a powerful tree-based ensemble model, and LSTM, a sequential deep learning model. The experiments are conducted in two scenarios-first, with only gold price data, and second, by incorporating crude oil data as an external economic indicator. The aim is to see how these models represent the sequential nature of the data and whether or not including external factors enhances their performance.

To do this, we followed a clear and structured process, starting with data cleaning and preprocessing. For XGBoost, we added lagged features and rolling statistics to capture temporal patterns, while for LSTM, we formatted the data into sequences so the model could learn directly from the time-series structure. Both models were trained and evaluated using five-fold cross-validation, with metrics such as RMSE, R^2, MAE, and sMAPE used to measure their performance.

Key Takeaways

Here are the main insights from our research:

- **LSTM Outperformed XGBoost:** LSTM consistently delivered better results in all scenarios, especially for RMSE, MAE, and sMAPE. Its ability to directly learn patterns from sequential data made it much more effective at handling time-series data.

- **Adding Crude Oil Features Helped Both Models:** When crude oil data (Close price and Volume) were added, both XGBoost and LSTM showed noticeable performance improvements. This highlights the value of including external economic indicators in forecasting tasks.

- **XGBoost Relied on Feature Engineering:** Unlike LSTM, which learned patterns directly from the data, XGBoost required extensive manual feature engineering, like lagged values and rolling statistics, to perform well.

Implications

The results show the potential of deep learning models, especially LSTM, in financial forecasting tasks like predicting gold prices. These models can uncover patterns in data that might be difficult to capture with traditional machine learning methods. Moreover, the improvements seen when external indicators like crude oil data were added suggest that incorporating domain knowledge can further boost accuracy. This has important implications for decision-making in finance, investment, and trading.

Limitations

Despite these promising findings, there are a few limitations to this study:

- We only included two external indicators (crude oil Close and Volume). Adding other factors, like inflation rates or stock market indices, might provide even better results.

- The focus was on hourly data. While this granularity can capture short-term trends, the models might behave differently with daily or weekly data.

- Computational resources constrained the extent of hyperparameter optimization, particularly for LSTM. Fine-tuning the models further could potentially enhance their performance.

Future Directions

Building on this research, there are several exciting paths to explore:

- **Broader Economic Indicators:** Adding more external data, such as inflation rates, exchange rates, or stock market trends, could provide a more comprehensive model.

- **Other Commodities or Assets:** Expanding the analysis to other commodities like silver or oil—or even stocks and cryptocurrencies—would test the generalizability of the models.

- **Advanced Architectures:** Trying more sophisticated models, such as GRU, Transformers, or hybrids that combine LSTM and XGBoost, could further push the boundaries of accuracy.

- **Different Time Scales:** Testing the models on daily or weekly data to see how well they adapt to different forecasting horizons.
- **Better Hyperparameter Tuning:** Allocating more resources to refine the models could unlock even greater potential.

Final Thoughts

This research highlights the strength of deep learning models, especially LSTM, in tackling financial forecasting challenges. By effectively leveraging sequential data and incorporating external indicators, we achieved significant improvements in predictive accuracy. These findings pave the way for further exploration into more advanced techniques and broader datasets, pushing the frontier of what's possible in financial time-series analysis.

References

[1] Amini, A. and Kalantari, R. [2024]. Gold price prediction by a cnn-bi-lstm model along with automatic parameter tuning, *PLoS One* **19**(3): e0298426.

[2] Boon-Long, K., Kantavat, P. and Nimmananta, K. [2024]. Price action analysis in gold trading using deep learning techniques, *IEEE Transactions on Digital Arts, Media and Technology* **1**(1): 204–209.

[3] Chai, J., Zhao, C., Hu, Y. and Zhang, Z. G. [2021]. Structural analysis and forecast of gold price returns, *Journal of Management Science and Engineering* **6**(2): 135–145.
URL: *https://www.sciencedirect.com/science/article/pii/S2096232021000135*

[4] Chew, L. M., Ngu, C. Y. S. and Yeng, O. L. [2023]. Gold prices forecasting using bidirectional lstm model based on spx500 index, usd index, crude oil prices and cpi, *Journal of Information and Communication Technology (ICoICT)* pp. 539–544.

[5] Liang, Y., Lin, Y. and Lu, Q. [2022]. Forecasting gold price using a novel hybrid model with iceemdan and lstm-cnn-cbam, *Expert Systems with Applications* **206**: 117847.

[6] Madziwa, L., Pillalamarry, M. and Chatterjee, S. [2022]. Gold price forecasting using multivariate stochastic model, *Resources Policy* **76**: 102544.
URL: *https://www.sciencedirect.com/science/article/pii/S0301420721005511*

[7] McNally, S., Roche, J. and Caton, S. [2018]. Predicting the price of bitcoin using machine learning, *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, pp. 339–343.

[8] Pandey, A. C., Misra, S. and Saxena, M. [2024]. Gold and diamond price prediction using enhanced ensemble learning, *International Journal of Financial Data Science* **12**(3): 245–260.

[9] S, V. G. and S, H. V. [2020]. Gold price prediction and modelling using deep learning techniques, *2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, pp. 28–31.

[10] Salim, M. and Djunaidy, A. [2024]. Development of a cnn-lstm approach with images as time-series data representation for predicting gold prices, *Procedia Computer Science* **234**: 333–340.

[11] Shafiee, S. and Topal, E. [2010]. An overview of global gold market and gold price forecasting, *Resources Policy* **35**(3): 178–189.

[12] Sharma, R. K. and Sharma, A. [2019]. Forecasting monthly gold prices using arima model: Evidence from indian gold market, *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* **8**(9S3): 1373–1376.

[13] Usmani, M., Adil, S. H., Raza, K. and Ali, S. S. A. [2016]. Stock market prediction using machine learning techniques, *2016 3rd International Conference on Computer and Information Sciences (ICCOINS)*, pp. 322–327.

[14] Yang, Y., Tian, Z., Guo, L. and Wang, H. [2024]. Enhanced time series prediction with vmd: A hybrid deep learning approach, *Journal of Computer Vision and Deep Learning* pp. 38–44.
URL: *https://doi.org/10.1145/3653781.3653822*

[15] Zhang, J., Liu, H., Bai, W. and Li, X. [2024]. A hybrid approach of wavelet transform, arima and lstm model for the share price index futures forecasting, *North American Journal of Economics and Finance* **69**. Cited by: 17.