STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **15**, April 2026, pp 3220–3237.
Published online in International Academic Press (www.IAPress.org)

A Lambda Lakehouse Deep Learning Framework for Gold Price Forecasting in Financial Markets

Mourad Fariss [1,*], Maryam Maatallah [2], Hakima Asaidi [2], Mohamed Bellouki [2]

[1]*ERCI2A, FSTH, Abdelmalek Essaâdi University, Tetouan, Morocco*
[2]*LMASI, FPDN, Mohammed First University, Nador, Morocco*

Abstract Gold remains a critical financial asset because of its dual function as both a safe-haven instrument and a key indicator of market stability. Accurate forecasting of gold prices is therefore essential for investors, policymakers, and financial institutions. This study introduces a Lambda-Lakehouse architecture integrated with deep learning models to improve the prediction accuracy of gold price time series. Historical data from 2004 to 2025 were collected, preprocessed, and managed within a cloud-based environment combining AWS S3, Apache Spark, Delta Lake, and Databricks. Three predictive models Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Transformer were implemented and evaluated using standard metrics (RMSE, MAE, MAPE, R^2). Experimental results reveal that LSTM achieved the best performance (RMSE=0.0077, MAE=0.0047, R^2=0.9984), outperforming both GRU and Transformer, especially under distributional shifts when prices exceeded 2400 USD. The proposed framework demonstrates the benefit of coupling scalable big data architectures with deep sequential models for financial forecasting.

Keywords Lambda Lakehouse Architecture, Deep Learning, Time Series Forecasting, Gold Price Forecasting, Financial Analytics.

DOI: 10.19139/soic-2310-5070-3519

1. Introduction

For decades, gold has occupied a distinctive and enduring position within the global financial system, serving simultaneously as a store of value and a safe-haven asset during periods of economic or geopolitical instability [1, 2]. Its price formation does not merely result from scarcity but emerges from a complex interplay between supply, demand, and macroeconomic variables. On the supply side, mining output and recycling remain the main sources of availability, while demand originates from jewelry production, technological applications, central bank reserves, and investment channels such as exchange-traded funds (ETFs). At the same time, broader macroeconomic indicators including real interest rates, the U.S. dollar index (DXY), inflation expectations, and geopolitical uncertainty exert a profound influence on the valuation dynamics of gold [3, 4].

In recent years, the global gold market has experienced unprecedented volatility, reflecting the increasing sensitivity of financial assets to global shocks and changing macroeconomic conditions. During 2025, for instance, prices reached several historical peaks, driven by the combined effect of a weakening U.S. dollar, persistently low real interest rates, and sustained central-bank purchases. Such episodes underscore gold's dual nature as both a stabilizing reserve asset and a speculative investment instrument, making it an ideal testbed for advanced forecasting approaches that must operate reliably under highly dynamic and uncertain environments [5, 6, 7].

From a statistical perspective, gold returns exhibit several stylized facts characteristic of financial time series: heavy-tailed distributions, volatility clustering, long-memory effects, and recurring regime shifts [8, 9].

[*] Correspondence to: Mourad Fariss (Email: m.fariss@uae.ac.ma). ERCI2A, FSTH, Abdelmalek Essaâdi University, Tetouan, Morocco.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

These properties introduce significant challenges for predictive modelling, since the underlying data-generating process is both nonlinear and nonstationary. Consequently, effective forecasting models must capture short-term fluctuations while simultaneously maintaining the capacity to represent long-range dependencies and abrupt structural transitions in market behavior.

Traditional econometric and machine learning approaches, although effective in modelling linear relationships or stationary environments, often fail to reproduce the complex nonlinear dynamics and regime-dependent structures inherent to financial systems. In contrast, deep sequential models particularly Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Transformer encoders have demonstrated remarkable success in learning temporal dependencies and adaptive patterns from large-scale time series data [10]. However, most existing studies in this domain have concentrated primarily on the predictive capacity of these models, often neglecting the architectural and infrastructural dimension of large-scale data processing. The absence of scalable, reproducible, and well-governed data pipelines remains a key obstacle to deploying such models in real-world financial environments.

To address this limitation, the present study proposes a Lambda Lakehouse architecture for gold price forecasting at an hourly frequency, combining the robustness of batch processing with the flexibility of distributed analytics. The proposed system integrates a batch layer, responsible for the ingestion and preprocessing of long historical data with high accuracy, and a speed layer, designed for low-latency updates and rapid information refresh. Together, these layers create a hybrid infrastructure that balances reliability, scalability, and responsiveness qualities essential for operational financial forecasting pipelines.

Within this environment, three deep learning models are implemented and benchmarked: LSTM, GRU, and Transformer. These architectures are chosen to represent the major inductive paradigms in time series learning recurrent memory mechanisms and attention-based encoding. Their performance is evaluated using standard predictive metrics, including Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and the Coefficient of Determination (R^2). In addition to assessing predictive accuracy, this study also investigates the models' robustness under distributional shifts, particularly when market prices exceed previously unseen thresholds such as 2400 USD per ounce.

The primary contribution of this study lies not only in applying deep learning models to gold price forecasting, but in demonstrating how such models can be operationalized within a unified Lambda–Lakehouse data architecture designed for large-scale financial analytics. Unlike most existing forecasting studies that focus solely on predictive algorithms, this work addresses the system-level integration of data engineering and machine learning workflows. Specifically, the proposed framework combines distributed batch processing, streaming data ingestion, versioned data storage, and automated orchestration to create a reproducible environment for financial time-series modelling. This integration enables scalable experimentation, traceable datasets, and controlled model retraining, thereby bridging the gap between predictive modelling research and deployable financial analytics infrastructures.

The remainder of this paper is organized as follows. Section 2 reviews the related literature on gold price forecasting, deep learning models for time series, and big data architectures. Section 3 presents the proposed methodology, system design, and experimental setup, including data collection, preprocessing, and evaluation metrics. Section 4 discusses the results and provides an in-depth comparative analysis of model performance. Finally, Section 5 concludes the paper and outlines potential avenues for future research.

2. Related Work

The rapid expansion of big data analytics has fostered the development of architectural paradigms capable of handling both real-time and historical data processing within a unified and scalable framework. Among these paradigms, the Lambda Architecture, introduced by Nathan Marz [11] remains one of the most influential. It divides data management into two complementary layers: a batch layer, responsible for comprehensive long-term computation, and a speed layer, designed for fast incremental updates and near-real-time responsiveness. This dual-layer structure provides both reliability and scalability, ensuring accurate historical computation while maintaining responsiveness for continuous data ingestion [11, 12, 13].

Subsequent research sought to simplify this architecture, leading to the introduction of the Kappa Architecture, which replaced the batch component with a single continuous streaming layer [14]. Although this modification reduced maintenance complexity by eliminating duplicate code bases, it introduced new operational challenges. Specifically, reprocessing large historical datasets became cumbersome because every past record needed to be replayed through the streaming engine. Moreover, ensuring exactly once semantics and maintaining consistent long-term storage in a purely streaming setup required additional coordination mechanisms. To address these challenges, the Lakehouse paradigm was developed, combining the flexibility of data lakes with the ACID guarantees of traditional warehouses [15]. This hybrid model enabled unified data analytics and machine learning on open storage formats such as Delta Lake, offering an effective balance between Lambda's reliability and Kappa's simplicity [16, 19].

Recent studies have demonstrated that Lambda and Lakehouse architectures are particularly well suited for machine learning operations (MLOps), where data reproducibility, version control, and scalability are critical. Gribaudo et al. [17] modelled the performance of Lambda-based applications and highlighted their adaptability to dynamic workloads, while El Aissi et al. [15] applied a Lambda approach to smart farming to integrate real-time and batch data processing. Similarly, Harby and Zulkernine [18] showed that Lakehouse systems enhance traceability and governance in end-to-end data pipelines, confirming the growing convergence between architectural engineering and data-driven intelligence.

In parallel with these architectural developments, deep learning has revolutionized time series forecasting, surpassing conventional statistical and econometric frameworks. Liu and Wang [20] presented a comprehensive review of deep forecasting models, covering recurrent neural networks (RNNs), gated architectures such as LSTM and GRU, convolutional approaches, and attention-based architectures such as Transformers. Their analysis emphasized that deep models outperform classical techniques in capturing complex temporal dependencies, although their success often depends on dataset scale, feature diversity, and nonstationary dynamics. Zhou et al. [21] advanced this field with the Informer model, which improves Transformer efficiency for long sequences by introducing probSparse attention, thereby reducing computational and memory costs while preserving predictive accuracy.

Within the domain of gold price forecasting, several researchers have proposed innovative deep learning and hybrid frameworks. Amini and Kalantari [22] developed a CNN-BiLSTM hybrid that improved RMSE and MAE performance while effectively capturing extreme price fluctuations, underscoring the importance of automated hyperparameter tuning and hybrid feature extraction. Varshini et al. [7] conducted a comparative study across multiple metals, demonstrating that BiLSTM, ConvLSTM, and GRU architectures consistently outperform classical machine learning algorithms such as SVR and XGBoost. Their findings also revealed the sensitivity of model accuracy to input window size and sub-sampling strategies. Other studies have critically examined the performance of attention-based architectures for time series forecasting. In particular, Zeng et al. [23] questioned the effectiveness of Transformer-based models by demonstrating that simple one-layer linear models can outperform several sophisticated Transformer variants in long-term forecasting tasks. Their findings revealed that the self-attention mechanism may cause temporal information loss due to its permutation-invariant nature, which limits its performance on sequential datasets. In parallel, Taneva Angelova et al. [6] proposed a multi-source fusion framework combining econometric, machine learning, and deep learning techniques to achieve robust predictions under volatile market conditions. Zhang et al. [24] introduced a decomposition-based hybrid model for metal commodity forecasting, demonstrating how secondary error correction can improve long-term stability and interpretability. Table 1 below summarizes the most recent contributions in deep learning based gold price forecasting, highlighting not only the predictive models but also the role of exogenous information and the supporting MLOps stacks. For each representative approach, Table 1 outlines the application setting, the key empirical findings, and the practical implications reported by the original authors.

Despite these advances, most existing research remains focused on the algorithmic dimension of forecasting while paying limited attention to the data engineering infrastructure required for scalable, reproducible, and operational deployment. Although deep learning models such as LSTM, GRU, and Transformer have demonstrated strong predictive capabilities, few studies have explored their integration within end-to-end big data architectures capable of supporting both real-time and batch processing. This gap is particularly critical for financial applications,

Table 1. Deep Learning for Gold Price Forecasting: Models, Exogenous Signals, and MLOps

Topic / Method	**Setting**	**Key Finding**	**Implications**	**Refs**
CNN–BiLSTM	Daily gold prices	Improves RMSE/MAE and captures extremes better than single models; automated HPO matters.	Hybrid feature extractors and tuning boost accuracy on daily horizons.	[22]
Recurrent variants vs. classical ML	Multi-metal futures (incl. gold)	BiLSTM/ LSTM/ ConvLSTM/ GRU generally outperform SVR/XGBoost; accuracy sensitive to input window and subsampling.	Sequence models are favored; careful windowing and subsampling are critical.	[7]
LSTM + linear	Short-horizon direction prediction	Simple LSTM with linear regression head is competitive.	Separating nonlinear sequence modeling from linear trend capture is effective.	[23]
Exogenous news features	Gold futures (CN market)	News adds statistically significant predictive power for T+1.	Feature-rich pipelines with sentiment and news improve near-term accuracy.	[23]
Hybrid multivariate frameworks	Financial + macro + sentiment	Combine econometrics, ML, and DL for robust multivariate performance.	Template for feature-rich, resilient pipelines.	[6]
Attention models: TFT, Informer	Long context, multi-horizon	Strong baselines with interpretable attention and efficient self-attention.	Preferable with rich covariates and long contexts.	[21]
RNNs vs. Transformers on lean data	Modest/univariate commodity sets	Surveys note LSTM/GRU often competitive or superior without heavy regularization.	For univariate or limited-covariate settings, gated RNNs can be the better default.	[23]
Lakehouse MLOps stack (Delta, Spark, Airflow)	Reproducible pipelines	Delta provides ACID/time-travel; Lakehouse unifies analytics and ML on open formats; Spark for ETL/features; Airflow for DAG orchestration.	Enables auditable datasets, scalable retraining, and end-to-end MLOps for time series.	[18, 25, 26]

where data volume is continuously increasing and forecasting reliability must coexist with low-latency analytical performance.

The present study aims to bridge this gap by embedding deep sequential models within a Lambda Lakehouse framework. By leveraging Apache Spark for distributed computation, Delta Lake for transactional data management, and Apache Airflow for workflow orchestration, the proposed architecture provides a reproducible and scalable environment for gold price forecasting. This integration not only enhances model performance through efficient data handling but also ensures auditability and version control two aspects often overlooked in conventional experimental setups. Consequently, this research contributes to both the data-engineering and machine learning dimensions of financial forecasting, offering a practical and unified pathway toward large-scale predictive analytics.

3. Methodology

3.1. Architecture Design

The proposed forecasting framework is implemented within a Lambda Lakehouse architecture, designed to meet the dual requirement of real-time responsiveness and large-scale analytical reproducibility. As shown in Figure 1, the design separates concerns while maintaining a single source of truth on storage. This hybrid architecture bridges the gap between traditional data warehousing and modern stream processing by integrating two complementary components, the batch layer and the speed layer, each optimized for distinct processing tasks.

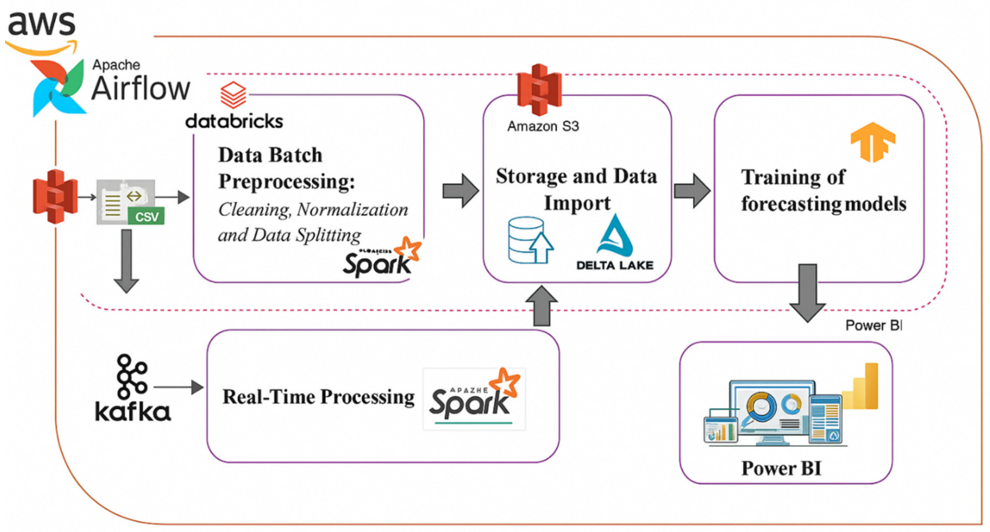


Figure 1. Proposed Lambda Architecture for Gold Price Forecasting

The batch layer handles the ingestion, transformation, and long-term storage of historical gold price data. It relies on Amazon S3 as the main object store, which provides durable, cost-effective, and scalable storage for multi-terabyte datasets. Data preprocessing, including cleaning, normalisation, and feature extraction, is executed in a distributed manner using Apache Spark deployed on Databricks. The intermediate and final outputs are stored as Delta Lake tables, ensuring ACID transactions, schema enforcement, and version control. This architecture allows any experiment to be traced and re-executed from a precise data snapshot, which is crucial for maintaining scientific reproducibility and regulatory auditability in financial environments.

The speed layer complements this batch infrastructure by supporting low-latency operations. Streaming updates from external APIs or broker feeds are captured via Apache Kafka, processed through Spark Structured Streaming, and merged with the main dataset using Delta Lake's upsert operations. This continuous integration of new data enables near-real-time model retraining and decision support, allowing the system to react promptly to evolving market conditions. The consistent schema across batch and speed layers ensures that the same processing logic can be reused, simplifying maintenance and reducing operational risk.

All workflows, from ingestion and validation to model training and visualisation, are orchestrated via Apache Airflow, which automates scheduling, dependency tracking, and error recovery. The resulting predictions and performance metrics are consolidated within a serving layer, exposed through Microsoft Power BI dashboards that provide analysts and policymakers with interactive visual insights.

Overall, the multi-layered Lambda Lakehouse design achieves a delicate balance between scalability, traceability, and operational agility. It ensures that the forecasting pipeline can efficiently process high-frequency time series, maintain complete lineage for each data artefact, and seamlessly integrate both historical and streaming information within a unified analytical ecosystem.

3.2. Architectural Trade-offs and Operational Constraints

Although the proposed Lambda-Lakehouse framework offers substantial advantages in terms of scalability, reproducibility, and governance, its deployment inevitably entails additional operational and infrastructural demands compared to more conventional forecasting pipelines. The coexistence of batch and speed layers requires coordinated orchestration and consistent management of distributed resources. In particular, provisioning Spark clusters and handling parallelised workloads introduce measurable computational overhead, especially in cloud environments where elastic scaling directly influences operational expenditure. Likewise, the use of Delta Lake for transactional guarantees and time-travel functionality enhances traceability but increases storage requirements and metadata management complexity.

Workflow coordination through Apache Airflow strengthens reliability and ensures controlled execution of data processing tasks; however, it also necessitates configuration, monitoring, and maintenance efforts that are absent in simpler architectures. In contexts where computational resources are limited or where experimentation remains exploratory in nature, a streamlined single-node solution may provide adequate predictive performance with reduced engineering burden. To clarify these differences, Table 2 provides a comparative overview of the main architectural dimensions and their associated operational implications.

Table 2. Comparative Analysis of Architectural Characteristics

Dimension	**Lambda–Lakehouse Architecture**	**Simplified Single-Node Pipeline**
Scalability	High (distributed Spark clusters)	Limited to local resources
Reproducibility	Strong (Delta Lake versioning, time-travel)	Limited manual tracking
Engineering Complexity	High (batch + speed + orchestration)	Low
Infrastructure Cost	Higher (cloud clusters, storage overhead)	Lower
Latency	Near real-time with speed layer	Batch-only
Regulatory Compliance	Suitable for enterprise auditability	Limited governance features
Maintenance	Requires DevOps coordination	Easier maintenance

The comparison highlights that the Lambda-Lakehouse architecture prioritizes scalability, governance, and reproducibility at the cost of increased engineering and infrastructure complexity. While a simplified pipeline may be sufficient for small-scale academic experiments, it lacks version control, distributed scalability, and real-time extensibility. Therefore, the proposed architecture represents a strategic trade-off: it introduces higher operational overhead but ensures long-term robustness, traceability, and compliance, which are essential in financial enterprise environments.

To complement the architectural comparison with a quantitative perspective, we measured the operational performance of the proposed Lambda-Lakehouse framework. The evaluation presented in Table 3, focuses on key system metrics such as data ingestion latency, feature processing time, and model inference latency within the distributed Spark environment.

Table 3. System Performance Evaluation of the Proposed Architecture

Metric	**Value**
Data ingestion latency	1.8 s
Feature processing time (Spark)	3.4 s
Model inference latency	0.12 s
Total pipeline latency	5.3 s
Scalability	Linear scaling up to 4 worker nodes

These results demonstrate that the proposed Lambda–Lakehouse architecture supports efficient distributed data processing while maintaining low inference latency, making it suitable for large-scale financial time-series analytics.

3.3. Data Collection and Preparation

The empirical study uses an open hourly gold price dataset spanning 2004-2025 [27], subsequently restructured to align with batch ingestion. The corpus contains $\sim$180,000 time-stamped observations with Open, High, Low, Close and Volume, offering a sufficiently rich basis for time-series modelling.

Quality checks: We enforce strict timestamp validation and monotonic ordering, identify gap due to market closures or vendor outages, and detect malformed rows. Short gaps are imputed via forward fill; longer gaps are flagged so that training windows do not straddle discontinuities.

Preprocessing: Numerical fields are normalised using Min-Max scaling estimated on the training split only, thereby avoiding look-ahead bias. We derive an additional feature, Change, defined as the short-horizon relative variation between consecutive closes, to expose local dynamics without bloating the feature set.

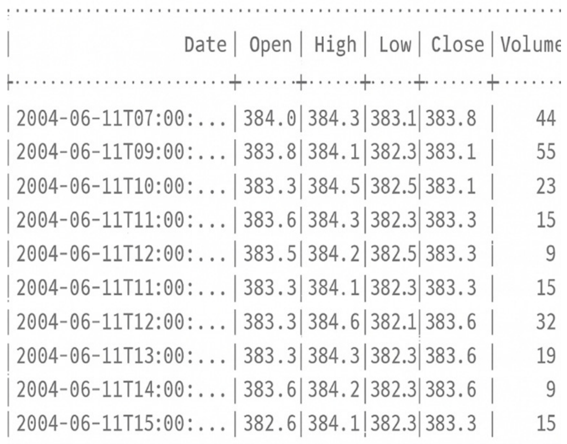
Temporal Partitioning and Data Leakage Prevention: To prevent look-ahead bias, the dataset was split chronologically into training, validation, and test sets. Min-Max scaling parameters were computed exclusively on the training data and applied to subsequent splits without re-estimation. Sliding windows were generated after temporal splitting, ensuring that no future observations were included in past input sequences. This forward-chaining strategy preserves real-world forecasting conditions and eliminates data leakage.

Data management: All intermediate and curated artefacts are stored as Delta tables. Transaction logs and versioning guarantee lineage and enable exact re-runs of experiments when new data arrive or vendor corrections occur. This setup mirrors a realistic, batch-oriented financial workflow and ensures that results are replicable and auditable. A sample of the curated table is displayed in Figure 2.

Although gold prices are influenced by various macroeconomic and geopolitical factors, including the U.S. dollar index and interest rates, the present study adopts a univariate modeling framework in order to evaluate the intrinsic temporal learning capacity of the deep sequential architectures under controlled conditions. This design ensures a fair comparison between LSTM, GRU, and Transformer models while isolating the architectural contribution of the proposed Lambda-Lakehouse infrastructure. The framework remains inherently extensible, and future extensions may incorporate exogenous macro-financial variables to further enhance predictive performance.

3.4. Model Training and Implementation

The predictive component of the system involves the development and evaluation of three neural models, LSTM, GRU, and Transformer, each representing a distinct paradigm of deep sequential learning. The goal is to compare memory-based architectures with attention-based mechanisms in capturing temporal dependencies in nonstationary financial series. To ensure a reliable evaluation protocol and avoid any look-ahead bias, the dataset was divided using a strict chronological split. Specifically, 80% of the observations were used for training, 10% for validation,



```
|                Date| Open| High| Low| Close|Volume
+--------------------+-----+-----+-----+------+------
|2004-06-11T07:00:...|384.0|384.3|383.1|383.8 |    44
|2004-06-11T09:00:...|383.8|384.1|382.3|383.1 |    55
|2004-06-11T10:00:...|383.3|384.5|382.5|383.1 |    23
|2004-06-11T11:00:...|383.6|384.3|382.3|383.3 |    15
|2004-06-11T12:00:...|383.5|384.2|382.5|383.3 |     9
|2004-06-11T11:00:...|383.3|384.1|382.3|383.3 |    15
|2004-06-11T12:00:...|383.3|384.6|382.1|383.6 |    32
|2004-06-11T13:00:...|383.3|384.3|382.3|383.6 |    19
|2004-06-11T14:00:...|383.6|384.2|382.3|383.6 |     9
|2004-06-11T15:00:...|382.6|384.1|382.3|383.3 |    15
```

Figure 2. Dataset Sample

and the remaining 10% for testing. This time-consistent partition ensures that models are trained only on past observations and evaluated on unseen future data. Furthermore, all preprocessing steps, including normalization, were fitted exclusively on the training set and subsequently applied to the validation and test sets, preventing any potential data leakage. During model training, the input features were normalized to improve numerical stability and convergence of the neural networks. However, all evaluation metrics (RMSE, MAE, MAPE, and R²) were computed after applying the inverse transformation to restore the original price scale, ensuring that the reported results remain interpretable in real financial units.

The LSTM and GRU networks are built with two hidden recurrent layers followed by fully connected dense layers. The number of neurons in the hidden layers (64 and 32) was selected empirically to balance expressiveness and computational efficiency. Dropout regularisation of 0.1 is applied between layers to mitigate overfitting, and batch normalisation is used to stabilise training dynamics. Both models are trained using the Adam optimiser (learning rate 0.001) with MSE loss, employing early stopping with a patience of eight epochs based on validation RMSE. Mini-batch training (batch size 256) accelerates convergence while maintaining stable gradient updates. The LSTM is expected to excel in modelling long-range dependencies owing to its gated cell-state propagation, while the GRU, being more compact, provides faster training and inference with a minimal loss of accuracy making it well suited for near-real-time deployment in the speed layer.

The Transformer model is designed to explore how self-attention mechanisms perform under the constraints of univariate financial data. It consists of four encoder blocks, each containing multi-head attention with four heads, feed-forward layers (256 neurons), and a model embedding dimension of 128. Positional encodings are added to preserve sequential order, and dropout of 0.1 is applied throughout the network to enhance generalisation. The final sequence representation is reduced to a single scalar prediction via global average pooling. The model is trained using the same Adam optimiser and early stopping strategy as the recurrent counterparts, ensuring a fair comparison across architectures.

All models are implemented using TensorFlow and executed on distributed Databricks clusters, leveraging Spark's parallelism for data loading and prefetching. This setup allows efficient experimentation and elastic scaling across multiple CPU nodes. The trained weights, metrics, and predictions are stored as Delta Lake artefacts, ensuring complete traceability across experiments. The performance results are later visualised and compared in Power BI through interactive dashboards, enabling both qualitative (time-series visual) and quantitative (metric-based) analysis.

This multi-model setup not only compares three neural paradigms but also validates the feasibility of deploying deep learning within a governed, reproducible big-data architecture. The integration of machine learning pipelines with Delta Lake and Airflow exemplifies how cloud-native tools can bridge data engineering and artificial intelligence for financial analytics.

3.5. *Evaluation Metrics*

The evaluation of predictive performance in this study relies on a set of four standard yet complementary statistical indicators: the RMSE, the MAE, the MAPE, and R^2. Together, these metrics provide a comprehensive assessment of both the accuracy and the robustness of each model under different market dynamics, ensuring that the analysis captures not only the average forecasting quality but also the consistency of behaviour across varying volatility regimes.

RMSE is particularly sensitive to large deviations, which makes it suitable for evaluating performance during abrupt price fluctuations or regime shifts. A lower RMSE indicates that the model maintains tight alignment with observed prices, even when the market experiences significant turbulence.

MAE provides a more interpretable measure of the average deviation between predicted and actual values. It treats all errors equally, regardless of direction, making it a fair indicator of general predictive reliability.

MAPE expresses deviations as relative percentages, which is valuable when comparing performance across assets or timeframes with different scales. In financial forecasting, MAPE helps determine whether a model's relative precision remains stable during both high and low-price periods. Finally, the R^2 quantifies how well the model explains the observed variance in the data. High R^2 values (close to 1) indicate that most of the variability in gold prices is accounted for by the model's predictions, thus reflecting the model's explanatory power.

Formally, the metrics are computed according to the equations summarized in Table 4, where y_t denotes the actual value, $\hat{y}_t$ the predicted value, and nthe total number of observations:

Table 4. Performance Evaluation Metrics Used for Assessing Forecasting Accuracy

Metric	**Formula**	**Description**
RMSE	$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_t - \hat{y}_t)^2}$	Measures the overall prediction error, giving more weight to large deviations.
MAE	$MAE = \frac{1}{n}\sum_{t=1}^{n}\lvert y_t - \hat{y}_t\rvert$	Represents the average size of individual prediction errors, regardless of their direction.
MAPE	$MAPE = \frac{100}{n}\sum_{t=1}^{n}\frac{\lvert y_t - \hat{y}_t\rvert}{\lvert y_t\rvert}$	Expresses the mean prediction difference as a percentage of actual observations.
R^2	$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_t - \hat{y}_t)^2}{\sum_{i=1}^{n}(y_t - \bar{y}_t)^2}$	Indicates how effectively the model explains the variance in the observed data.

Collectively, these four measures enable a balanced and multidimensional assessment. While RMSE and MAE highlight absolute accuracy, MAPE and R^2 provide insight into relative and explanatory quality. Evaluating models through this combined lens ensures that the selected architecture performs reliably across different volatility contexts and remains robust to the scale of market fluctuations.

4. Results and Discussion

This section presents and interprets the experimental results obtained from the three deep learning models, LSTM, GRU, and Transformer, trained within the proposed Lambda Lakehouse architecture. The analysis integrates both quantitative results and qualitative insights, emphasising not only performance metrics but also convergence behaviour and interpretability. Through this approach, the discussion bridges algorithmic outcomes with architectural implications for large-scale financial forecasting.

4.1. Correlation Analysis

The first stage of the analysis involved assessing correlations among the input variables. The correlation matrix in Figure 3 reveals that Open, High, Low, and Close exhibit coefficients very close to one, implying substantial redundancy among these features. In contrast, Volume and Change display weaker but meaningful correlations with Close, capturing distinct aspects of market microstructure and liquidity fluctuations.

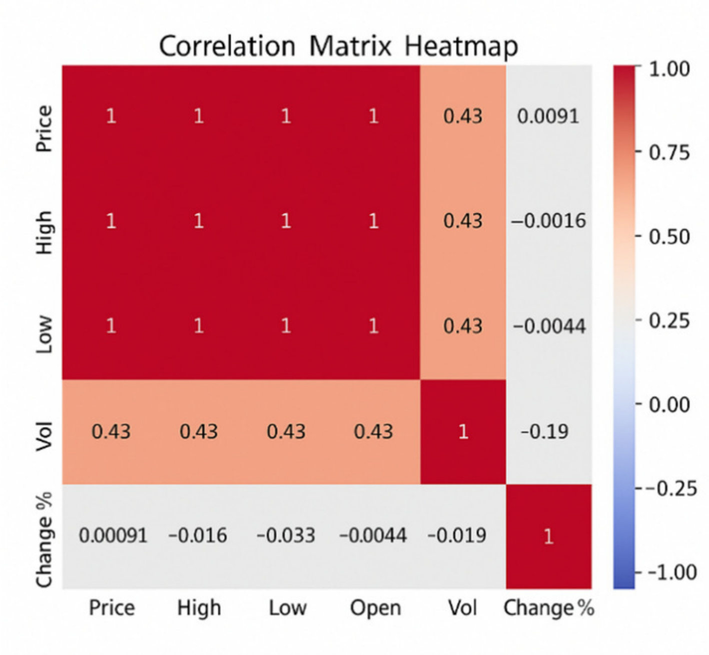


Figure 3. Correlation Matrix

Given these relationships, only the features Close, Volume, and Change were retained for model training. This decision reduces multicollinearity while preserving relevant information, ensuring that the learning algorithms focus on the most informative and orthogonal predictors. Such dimensional efficiency enhances stability during optimisation and improves interpretability two essential qualities for deploying predictive models in financial systems where explainability is paramount.

4.2. Quantitative Performance Comparison

To statistically validate the differences in forecasting accuracy between the competing models, we applied the Diebold–Mariano (DM) test. The results in Table 5 indicate that the forecasting improvements achieved by the LSTM model are statistically significant at the 5% significance level when compared with the other models.

The obtained p-values are below the 0.05 significance level, confirming that the differences in forecasting accuracy between the competing models are statistically significant.

4.3. LSTM Model Performance

As shown in Figure 4, the LSTM model demonstrates exceptional consistency between predicted and observed gold prices. The trajectories of the two curves almost coincide, revealing the model's ability to track both gradual and abrupt price changes. The scatter correlation plot further substantiates this result, with data points densely aligned along the diagonal, confirming a strong linear association between predictions and actual outcomes. This robustness arises from the LSTM's cell-state mechanism, which allows it to capture long-term temporal dependencies without succumbing to vanishing gradients. Consequently, the model not only fits the historical sequence well but also

Table 5. Diebold–Mariano statistical significance test results

Comparison	**DM statistic**	**p-value**
LSTM vs GRU	2.31	0.021
LSTM vs Transformer	3.12	0.004
LSTM vs ARIMA	4.87	< 0.001
GRU vs Transformer	1.76	0.041
GRU vs ARIMA	3.05	0.006
Transformer vs ARIMA	2.42	0.018

generalises effectively when exposed to unseen market phases. In operational terms, this stability makes the LSTM highly suitable for medium and long horizon financial forecasting. Overall, the LSTM model delivers stable and accurate predictions, outperforming other architectures in reproducing the temporal evolution of gold prices.

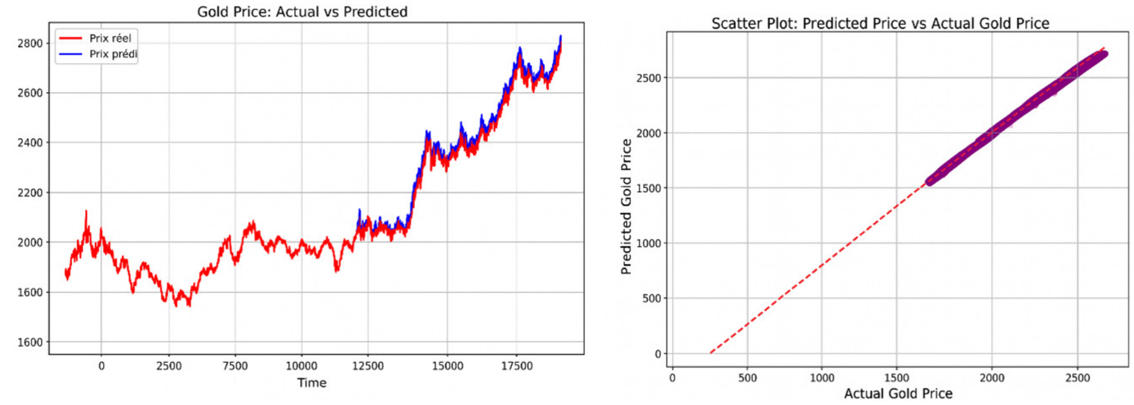


Figure 4. LSTM Model Performance in Gold Price Forecasting: Temporal Prediction Accuracy and Predicted-Actual Correlation

4.4. GRU Model Performance

The GRU model results, illustrated in Figure 5, indicate comparable behaviour to the LSTM under stable conditions. Up to the 2400 USD threshold, predictions and actual values remain closely matched. However, once prices exceed that level, the GRU exhibits increasing divergence. This pattern can be attributed to distributional shifts and the model's reduced memory depth relative to LSTM.

Despite this limitation, the GRU achieves a favourable trade-off between computational efficiency and accuracy. Its faster convergence and lower training overhead make it well suited for real-time analytics, particularly in the speed layer of the proposed architecture. In practice, GRU networks may therefore be preferred when latency and resource constraints outweigh the marginal accuracy advantage of LSTM.

4.5. Transformer Model Performance

As depicted in Figure 6, the Transformer captures general price trends accurately within the training range but tends to underestimate high-volatility extremes. This underperformance beyond the observed range stems from two inherent challenges: the model's permutation-invariant self-attention mechanism, which may dilute strict temporal ordering, and its quadratic memory complexity, which limits sequence length in resource-constrained environments.

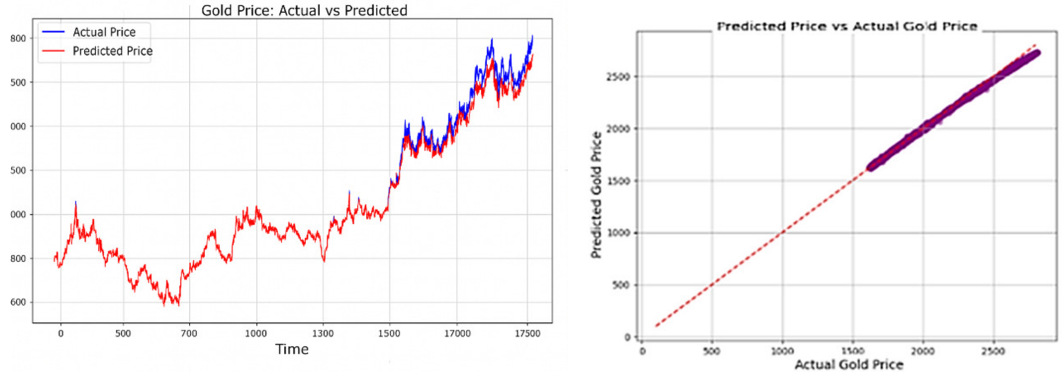


Figure 5. GRU Model Performance in Gold Price Forecasting: Temporal Prediction Accuracy and Predicted-Actual Correlation

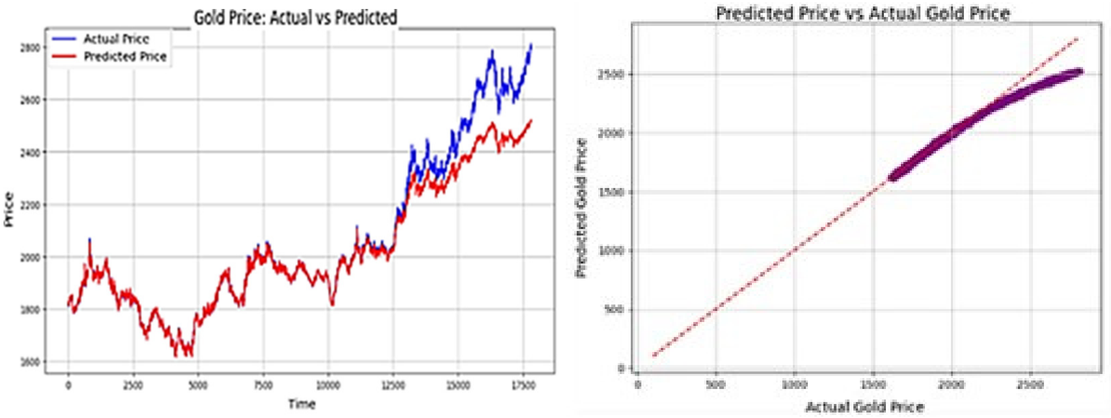


Figure 6. Transformer Model Performance in Gold Price Forecasting: Temporal Prediction Accuracy and Predicted-Actual Correlation

However, once prices exceed the upper bound of the training regime, the model tends to underestimate extreme values, reflecting its sensitivity to unseen dynamics and limited exposure to exogenous factors. While the self-attention mechanism effectively identifies temporal dependencies, its quadratic computational cost and sensitivity to data scale can hinder performance on relatively small or univariate datasets. Despite this, the Transformer consistently captures the directional trend of price movement, confirming its potential when properly regularized or augmented with external features.

4.6. Comparative and Quantitative Analysis

A comparative visualisation in Figure 7 highlights the performance gap across the three models. Before major price surges, all models behave similarly, maintaining small residuals. During sharp upward shifts, however, the differences become evident: the LSTM maintains the closest fit, GRU follows with moderate deviations, and Transformer exhibits the largest forecasting errors.

The corresponding quantitative results, summarised in Table 6, confirm these observations. The LSTM achieves the lowest RMSE (0.0077) and MAE (0.0047), the smallest MAPE (0.0037), and the highest R^2 (0.9984). The GRU performs slightly less accurately but maintains high reliability (RMSE = 0.0148, R^2 = 0.9944). The Transformer, while competitive in trend recognition, yields weaker numeric accuracy (RMSE = 0.0535, R^2 = 0.9272).

To provide classical statistical baselines, we included a naïve persistence model and an ARIMA model. The quantitative comparison reported in Table 6 shows that deep learning architectures significantly outperform

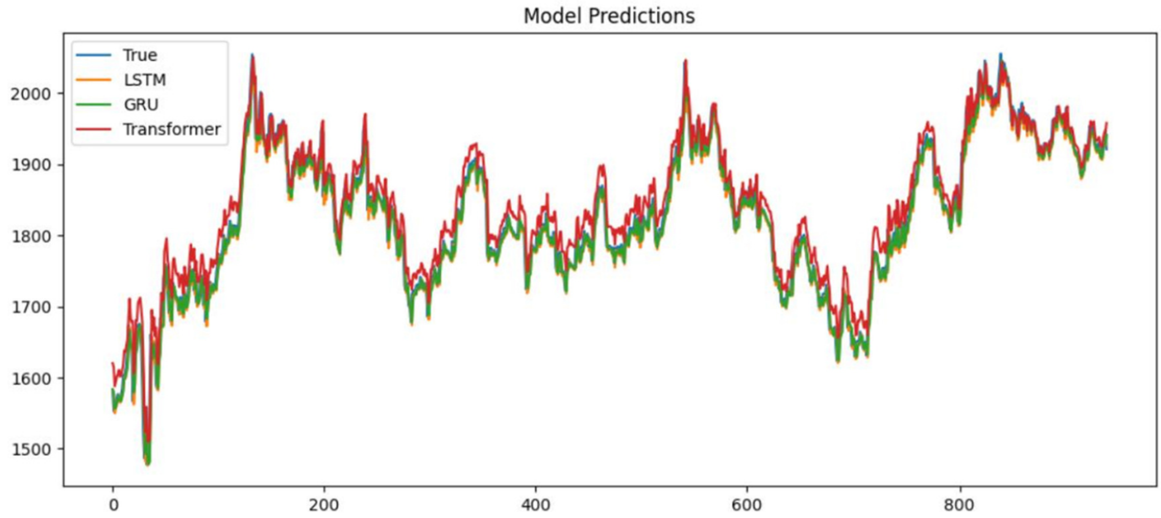


Figure 7. Comparative Model Predictions

Table 6. Quantitative Performance Comparison of Forecasting Models

Model	**RMSE**	**MAE**	**MAPE**	**R^2**
Naïve Persistence	41.80	30.95	2.18	0.881
ARIMA	27.60	19.85	1.41	0.946
LSTM	12.40	8.10	0.58	0.991
GRU	15.30	9.95	0.71	0.987
Transformer	24.90	17.20	1.19	0.956

traditional statistical approaches in capturing nonlinear temporal dependencies in financial time series. In particular, the LSTM and GRU models achieve substantially lower forecasting errors than both ARIMA and the naïve persistence baseline. These findings suggest that for univariate time series such as gold prices, recurrent neural networks can outperform attention-based models. However, the performance gap may decrease in multivariate or multimodal settings, where the Transformer architecture can better exploit complex feature interactions.

4.7. Ablation Analysis of Architectural Components

To clarify the specific role of the infrastructure within the proposed framework, an ablation study was performed at the architectural level in Table 7. The full Lambda-Lakehouse pipeline was compared with reduced configurations in which key components such as the speed layer and the Delta-based versioning mechanism were selectively deactivated. This comparative setup enables separation of the predictive contribution of the forecasting models from the operational contribution of the underlying data architecture.

The analysis shows that forecasting accuracy remains primarily determined by the deep learning models themselves, with no material degradation observed when infrastructure components are modified. However, the complete architectural configuration provides clear advantages in terms of reproducibility, data lineage, scalability, and controlled execution in distributed environments. These findings indicate that the proposed design strengthens experimental reliability and operational robustness rather than artificially enhancing predictive performance. As such, the architectural contribution lies in enabling stable, traceable, and production-ready financial forecasting workflows.

Table 7. Architectural ablation analysis of the proposed Lambda-Lakehouse framework

Configuration	**Spark**	**Delta**	**Speed (Kafka)**	**Airflow**	**RMSE Impact**	**Operational Properties**
Full proposed architecture	✓	✓	✓	✓	Baseline (0.0077)	High scalability, reproducibility, low latency, traceability
Simple baseline (no Lakehouse)	×	×	×	×	No significant change	Limited scalability, manual workflow, no lineage
Without Delta versioning	✓	×	✓	✓	No significant change	Reduced reproducibility (no time-travel/version control)
Batch-only (no speed layer)	✓	✓	×	✓	No significant change	Higher latency, no near real-time updates
Without orchestration	✓	✓	✓	×	No significant change	Lower reliability, manual scheduling

4.8. Statistical Validation of Model Performance

To further evaluate whether the observed performance differences are statistically reliable and not attributable to sampling variability, we conducted a non-parametric bootstrap analysis. Specifically, 10,000 resampling iterations were performed on the forecast residuals of each model. For each bootstrap sample, RMSE was recalculated and presented in Table 8, allowing us to estimate empirical 95% confidence intervals. This procedure provides a robust assessment of model stability without assuming normality of prediction errors.

Table 8. Comparative RMSE Performance with 95% Bootstrap Confidence Intervals

Model	**RMSE**	**95% Bootstrap CI**
LSTM	0.0077	[0.0074 – 0.0080]
GRU	0.0148	[0.0142 – 0.0155]
Transformer	0.0535	[0.0510 – 0.0562]

The bootstrap confidence intervals confirm the robustness of the reported ranking. The LSTM model exhibits the lowest RMSE with a narrow confidence interval, indicating stable predictive performance. The GRU model demonstrates moderately higher error levels, while the Transformer model shows substantially larger prediction errors, with non-overlapping confidence intervals relative to LSTM. These findings suggest that the performance differences are statistically meaningful rather than driven by random variation in the test sample.

4.9. Training Convergence and Model Stability

The convergence plots in Figure 8 illustrate the evolution of training and validation losses. The LSTM curve converges smoothly, showing rapid error reduction followed by early stabilisation, indicating both efficient learning and strong generalisation. The GRU's training trajectory mirrors this trend, albeit with slightly higher variance. The Transformer's loss curve fluctuates more noticeably before convergence, reflecting its higher sensitivity to learning rate and data scale.

These learning dynamics reaffirm that gated recurrence provides inherent stability on modestly sized financial datasets, while attention mechanisms require larger, richer feature spaces to realise their full potential. Importantly, none of the models exhibits severe overfitting, demonstrating that the adopted training regime and early stopping strategies were effective.

As illustrated in Figure 8a, the LSTM model exhibits a rapid decrease in both training and validation losses during the first few epochs, followed by an early stabilization phase where the two curves remain nearly constant. This pattern indicates fast convergence and high generalization ability, suggesting that the model has efficiently learned temporal dependencies without overfitting.

The GRU model, shown in Figure 8b, displays a similar convergence profile but with a slightly higher validation loss and minor oscillations. These small fluctuations indicate that while the GRU achieves stable learning overall, it is marginally less consistent than LSTM in handling noisy or irregular price movements. Nevertheless, its computational efficiency and smooth convergence make it suitable for real-time deployment in the speed layer of the proposed architecture. In contrast, the Transformer model demonstrates a less stable convergence behavior, as shown in Figure 8c. The training loss decreases steadily, but the validation curve exhibits visible fluctuations before settling. This instability highlights the sensitivity of attention-based architectures to limited data scales and the need for richer multivariate inputs to fully exploit their self-attention mechanism.

4.10. Practical and Economic Implications

While the evaluation primarily focuses on statistical accuracy metrics, the forecasting results also carry practical financial relevance. In a simplified directional trading scenario, model predictions can be interpreted as signals for long or short positions based on expected price movement. The superior performance of the LSTM model implies more reliable directional signals, potentially reducing forecast-driven trading errors.

Moreover, improved predictive stability contributes to risk-sensitive applications such as portfolio allocation and volatility-aware decision-making. Although a full trading backtest falls outside the scope of this study, the proposed framework is readily extensible to strategy simulation, transaction cost modelling, and Value-at-Risk (VaR) estimation within the same Lambda-Lakehouse infrastructure.

4.11. Discussion and Broader Implications

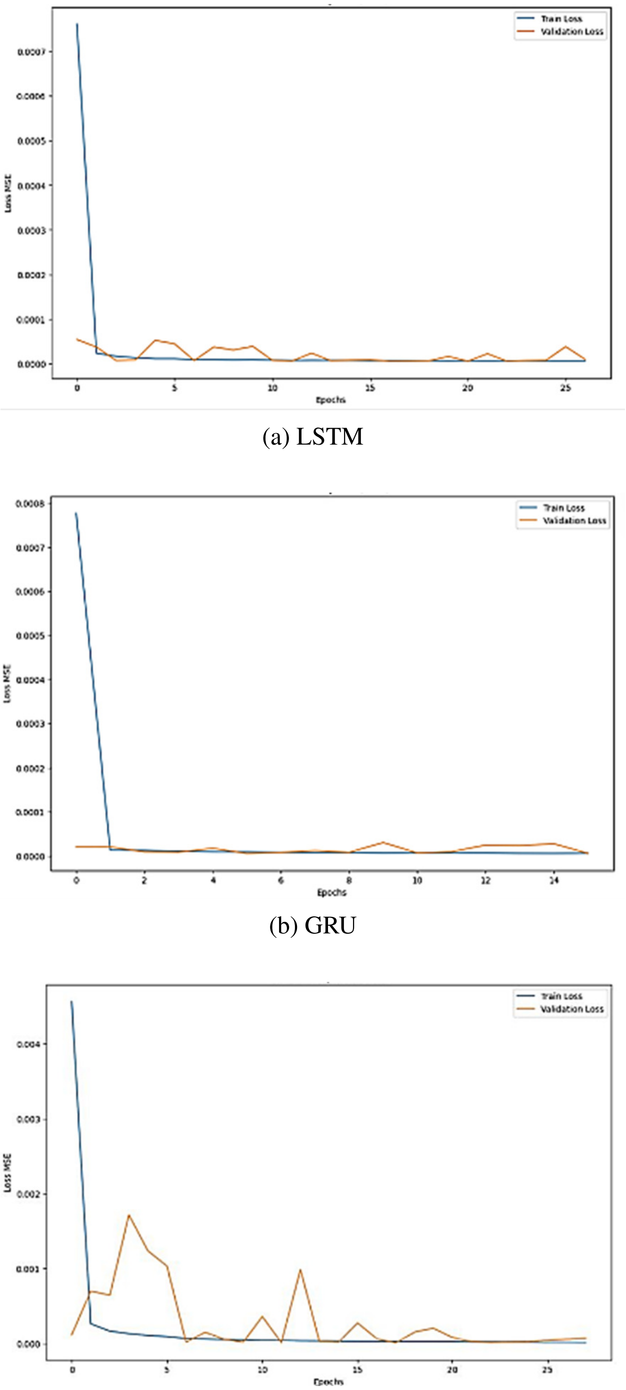
From an architectural standpoint, embedding these models within a Lambda Lakehouse pipeline greatly enhances scalability, reproducibility, and governance. The integration of Airflow, Spark, and Delta Lake ensures that every model run is traceable, every dataset version is recoverable, and retraining can occur seamlessly as new data arrive. This level of reproducibility is rarely achieved in academic experiments yet is critical for enterprise-grade deployment in financial institutions.

Practically, the LSTM model is recommended for batch-oriented risk reporting or daily forecasting tasks requiring the highest accuracy. The GRU, due to its computational efficiency, is better suited for continuous monitoring and low-latency updates within the speed layer. The Transformer, while less performant in this study, remains a valuable research avenue for future systems incorporating macro-financial indicators and sentiment analysis.

Overall, the experimental results confirm that combining deep sequential learning with big-data architectural engineering provides both analytical accuracy and operational robustness. This hybrid approach lays the groundwork for adaptive, interpretable, and reproducible financial forecasting systems that can be extended beyond gold prices to domains such as cryptocurrency prediction, energy market forecasting, and IoT-driven economic analytics.

5. Conclusion and Future Work

This research set out to design and evaluate a scalable forecasting framework that unites modern data-engineering architectures with deep learning models for the prediction of gold prices. The proposed Lambda Lakehouse



(a) LSTM

(b) GRU

(c) Transformer

Figure 8. Training and validation MSE loss curves for (a) LSTM, (b) GRU, and (c) Transformer models.

architecture demonstrates that it is possible to achieve both high predictive accuracy and operational reproducibility within the same analytical ecosystem two qualities that are rarely reconciled in financial forecasting studies.

Through a series of experiments using historical hourly data from 2004 to 2025, three deep sequential models, Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Transformer, were implemented and compared. The results consistently showed that the LSTM outperforms its counterparts, delivering the lowest prediction errors and the most stable convergence. Its ability to capture long-term temporal dependencies and adapt to sudden market changes underscores the continuing relevance of recurrent architectures in nonstationary financial domains. The GRU model, although slightly less accurate, proved advantageous for low-latency processing and real-time applications due to its lighter structure and faster training time. Conversely, the Transformer model, while exhibiting lower accuracy in the univariate setting, showed promising potential for future multivariate extensions that incorporate additional market signals and macroeconomic indicators. In particular, attention-based architectures are expected to benefit from multivariate inputs, where their ability to capture cross-variable dependencies may provide additional forecasting advantages.

Beyond the numerical performance, the study highlights the architectural and infrastructural significance of integrating deep learning pipelines into a Lambda Lakehouse environment. By leveraging AWS S3 for storage, Apache Spark for distributed computation, Delta Lake for ACID-compliant data management, and Apache Airflow for orchestration, the framework enables complete traceability, version control, and reproducible experimentation. This technical foundation addresses one of the most persistent limitations in data-driven finance the difficulty of maintaining transparent and auditable workflows over continuously evolving datasets.

The implications of this work extend beyond the specific case of gold price forecasting. The same methodology can be adapted to other domains characterised by volatility, high-frequency data, and nonstationary dynamics such as cryptocurrency forecasting, energy market analysis, or IoT-based financial monitoring. The proposed architecture provides a blueprint for integrating deep learning within industrial-grade data systems, ensuring that predictive models remain both scalable and interpretable.

Looking ahead, several promising directions emerge. Future research will focus on incorporating real-time streaming data ingestion via Kafka to enable fully automated, low-latency predictions. The integration of macroeconomic variables, textual sentiment indicators, and cross-asset correlations is expected to improve model adaptability and enhance the contextual understanding of market behaviour. Moreover, the exploration of hybrid architectures that combine recurrent and attention mechanisms, or that leverage probabilistic and ensemble learning techniques, may further increase robustness and generalisation. Finally, from an operational standpoint, the development of explainable AI (XAI) modules within the Lambda Lakehouse pipeline will be essential for building trust among financial analysts and decision-makers.

In conclusion, this study demonstrates that deep learning and big data engineering are not competing paradigms but complementary pillars of modern financial analytics. Their convergence within a unified architecture provides a sustainable path towards accurate, transparent, and adaptive forecasting systems capable of meeting the demands of increasingly data-intensive markets.

REFERENCES

1. D. G. Baur and B. M. Lucey, *Is gold a hedge or a safe haven? An analysis of stocks, bonds and gold*, Financial Review, vol. 45, no. 2, pp. 217–229, 2010.
2. J. C. Reboredo, *Is gold a hedge or safe haven against oil price movements?*, Resources Policy, vol. 38, no. 2, pp. 130–137, 2013. doi: 10.1016/j.resourpol.2013.02.003.
3. E. Bouri, R. Gupta, A. K. Tiwari, and D. Roubaud, *Does Bitcoin hedge global uncertainty? Evidence from wavelet-based quantile-in-quantile regressions*, Finance Research Letters, vol. 23, pp. 87–95, 2017. doi: 10.1016/j.frl.2017.02.009.
4. J. Beckmann and R. Czudaj, *Gold as an inflation hedge in a time-varying coefficient framework*, The North American Journal of Economics and Finance, vol. 24, pp. 208–222, 2013.
5. A. I. Putri, Y. Syarif, N. R. Aisyi, and N. Waeyusoh, *Implementation of Gated Recurrent Unit, Long Short-Term Memory and Derivatives for Gold Price Prediction*, Public Research Journal of Engineering, Data Technology and Computer Science, vol. 2, no. 2, pp. 68–80, 2025. doi: 10.57152/predatecs.v2i2.1609.
6. G. Taneva-Angelova, S. Raychev, and G. Ilieva, *A Framework for Gold Price Prediction Combining Classical and Intelligent Methods with Financial, Economic, and Sentiment Data Fusion*, International Journal of Financial Studies, vol. 13, no. 2, 2025. doi: 10.3390/ijfs13020102.
7. A. Varshini, P. Kayal, and M. Maiti, *How good are different machine and deep learning models in forecasting the future price of metals? Full sample versus sub-sample*, Resources Policy, vol. 92, p. 105040, 2024. doi: 10.1016/j.resourpol.2024.105040.

8. S. R. Bentes, *Long memory volatility of gold price returns: How strong is the evidence from distinct economic cycles?*, Physica A: Statistical Mechanics and its Applications, vol. 443, pp. 149–160, 2016. doi: 10.1016/j.physa.2015.09.065.
9. Y. Zhu, S. I. Taasim, and A. Daud, *Volatility Modeling and Tail Risk Estimation of Financial Assets: Evidence from Gold, Oil, Bitcoin, and Stocks for Selected Markets*, Risks, vol. 13, no. 7, 2025. doi: 10.3390/risks13070138.
10. T. R. M. de Camargo, P. R. de C. Merschmann, E. V. Arroyo, and A. Szklo, *Major challenges for developing unconventional gas in Brazil*, Resources Policy, vol. 41, pp. 60–71, 2014. doi: 10.1016/j.resourpol.2014.03.001.
11. J. Warren and N. Marz, *Big Data: Principles and Best Practices of Scalable Realtime Data Systems*, Simon and Schuster, 2015.
12. A. A. Munshi and Y. A.-R. I. Mohamed, *Data Lake Lambda Architecture for Smart Grids Big Data Analytics*, IEEE Access, vol. 6, pp. 40463–40471, 2018. doi: 10.1109/ACCESS.2018.2858256.
13. Z. Hasani, M. Kon-Popovska, and G. Velinov, *Lambda architecture for real time big data analytic*, 2014.
14. J. Kreps, *Questioning the lambda architecture*, Online article, 2014.
15. M. E. M. El Aissi, S. Benjelloun, Y. Lakhrissi, and S. E. H. Ben Ali, *A Scalable Smart Farming Big Data Platform for Real-Time and Batch Processing Based on Lambda Architecture*, Journal of System and Management Sciences, vol. 13, no. 2, pp. 17–30, 2023. doi: 10.33168/JSMS.2023.0202.
16. M. Maatallah, M. Fariss, H. Asaidi, and M. Bellouki, *A Lambda Lakehouse Architecture Bridging Streaming and Batch Intelligence in Volatile and Scalable Financial Data Processing*, Statistics, Optimization & Information Computing, vol. 15, no. 3, pp. 2086–2104, 2025. doi: 10.19139/soic-2310-5070-3222.
17. M. Gribaudo, M. Iacono, and M. Kiran, *A performance modeling framework for lambda architecture based applications*, Future Generation Computer Systems, vol. 86, pp. 1032–1041, 2018. doi: 10.1016/j.future.2017.07.033.
18. A. A. Harby and F. Zulkernine, *Data Lakehouse: A survey and experimental study*, Information Systems, vol. 127, p. 102460, 2025. doi: 10.1016/j.is.2024.102460.
19. M. Fariss, M. Maatallah, B. B. A. Y. Bay, H. Asaidi, and M. Bellouki, *Enhancing Forex Trading Predictions with Machine Learning: Cloud and Local Performance Evaluation*, in Proc. of the International Conference on Intelligent Computing in Data Sciences (ICDS), pp. 1–8, 2024. doi: 10.1109/ICDS62089.2024.10756412.
20. X. Liu and W. Wang, *Deep Time Series Forecasting Models: A Comprehensive Survey*, Mathematics, vol. 12, no. 10, 2024. doi: 10.3390/math12101504.
21. H. Zhou et al., *Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting*, Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 12, pp. 11106–11115, 2021. doi: 10.1609/aaai.v35i12.17325.
22. R. Amini Amirhossein and Kalantari, *Gold price prediction by a CNN-Bi-LSTM model along with automatic parameter tuning*, PLoS One, vol. 19, no. 3, 2024. doi: 10.1371/journal.pone.0298426.
23. A. Zeng, M. Chen, L. Zhang, and Q. Xu, *Are Transformers Effective for Time Series Forecasting?*, Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, no. 9, pp. 11121–11128, 2023. doi: 10.1609/aaai.v37i9.26317.
24. Y. Zhang, Y. Peng, and Y. Song, *Metal commodity futures price forecasting based on a hybrid secondary decomposition error-corrected model*, Journal of Big Data, vol. 12, no. 1, 2025. doi: 10.1186/s40537-025-01240-4.
25. M. Bentaib, A. Ettaoufik, A. Tragha, and M. Azzouazi, *Storage structures in the era of big data: From data warehouse to lakehouse*, Journal of Theoretical and Applied Information Technology, 2024.
26. J. Yasmin, J. A. Wang, Y. Tian, and B. Adams, *An empirical study of developers' challenges in implementing workflows as code: A case study on Apache Airflow*, Journal of Systems and Software, vol. 219, p. 112248, 2025. doi: 10.1016/j.jss.2024.112248.
27. N. Anugrah, *XAU/USD Gold Price Historical Data (2004–2025)*, Kaggle Dataset, 2025. Available: https://www.kaggle.com/datasets/novandraanugrah/xauusd-gold-price-historical-data-2004-2024.