SCIENTIFIC CULTURE

www.sci-cult.com

SCIENTIFIC CULTURE, Vol. 12, No. 4, (2026), pp. 3576-3594
Open Access. Online & Print

DOI: 10.5281/zenodo.12426416

COMPARATIVE EVALUATION OF LARGE LANGUAGE MODELS FOR STOCK PRICE PREDICTION USING MULTI-DATASET

Ramsundar G[1]*, Radha D[2]

[1]Department of Computer Science Engineering, Amrita School of Computing, Bengaluru, Amrita Vishwa Vidyapeetham, India.
[2]Department of Computer Science Engineering, Amrita School of Computing, Bengaluru, Amrita Vishwa Vidyapeetham, India

Received: 18/11/2025
Accepted: 21/02/2026

Corresponding Author: Ramsundar G (Email)

ABSTRACT

Accurately forecasting stock prices remains a challenging task due to the highly volatile, non-linear, and dynamic nature of financial markets. Traditional machine learning (ML) and deep learning (DL) models have demonstrated utility in financial time series prediction. However, these approaches are limited by their dependence on predefined features, inability to capture broader contextual information, and difficulty in generalizing to unseen market conditions. Recent advances in large language models (LLMs) provide a paradigm shift by enabling contextual understanding, multi-modal reasoning, and adaptability across diverse datasets, making them promising candidates for financial forecasting tasks where textual cues, long-term dependencies, and nonlinear interactions play a crucial role. This study systematically evaluates the performance of five state-of-the-art LLMs including LLaMA-3, Falcon-2, Snowflake Arctic, DeepSeek-V2, and FinGPT on stock price prediction tasks using three real-world datasets such as Google, Reliance, and Apple. The performance of LLMs was compared using multiple evaluation metrics, including Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and Coefficient of Determination (R^2), along computational efficiency to ensure practicality in real-world applications. Experimental results show that FinGPT, a finance-domain specialized LLM, consistently outperformed general-purpose LLMs across all datasets, achieving the lowest error rates in MAE with 10.92 on Reliance, 1.75 on Apple, and 22.05 on Google and highest R^2 score of 0.986–0.993. DeepSeek-V2 also demonstrated strong performance, outperforming LLaMA-3, Snowflake Arctic, and Falcon-2 in both predictive accuracy and efficiency. Furthermore, computational efficiency analysis highlighted FinGPT's superior speed and reduced resource consumption. This research establishes that LLMs offer clear advantages over conventional ML/DL approaches for financial forecasting by capturing complex temporal dependencies, enabling context-aware predictions, and reducing computational overhead. The findings suggest that domain-adapted LLMs such as FinGPT can serve as robust, accurate, and efficient tools for stock price prediction, paving the way for their integration into algorithmic trading systems, risk management platforms, and decision-support applications in financial markets.

KEYWORDS: Large Language Models, Stock Price Prediction, Financial Time Series Forecasting, FinGPT, LLaMA-3, Snowflake Arctic, DeepSeek-V2.

1. INTRODUCTION

The method of forecasting company's future share value using past market data and financial criteria is known as stock price prediction [1]. It aids investors, traders, and institutions in making informed decisions to maximize returns while minimizing risk. Historically, the framework has used statistical and machine-learning models to understand companies' shared values [2]. Large Language Models (LLMs) have recently been used to support prediction models by analyzing unstructured data from time perspective, framing observed sentiment, market activity, and other risk factors [3]. LLM-based predictions are generally more accurate, instantaneous, transparent, and aligned with survey-based large data frameworks compared to solely numerical and historical data trends in predicting stock movement.

Stock price prediction challenges are exacerbated with volatile stock market, which is influenced by various unpredictable elements like macroeconomic changes, industry trends, individual company results, and investor sentiment [5]. Low signal-to-noise ratios further complicate model building, making it difficult to derive patterns from the data [6]. Traditional econometric methods and machine learning face challenges with nonlinear, non-stationary data, resulting in poor accuracy in dynamic environments [7]. Stock correlations also impact these models, as modeling multiple series in isolation can diminish their holistic market perspective. Significant stock price increases and abrupt volatility can also result in short-term sentiment bias and decay in predicting ability [8].

Several conventional statistical techniques, Machine Learning (ML) algorithms, and Deep Learning (DL) architectures are used in stock price forecasting to improve performance. For structured and steady data, conventional techniques such as logistic regression, autoregressive integrated moving average (ARIMA), linear discriminant analysis (LDA), and quadratic discriminant analysis (QDA) work well [9]. ML methods like K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Decision Trees (DT), and Random Forests (RF) algorithms improve forecasting for specific datasets. DL architectures like Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), Gated Recurrent Units (GRUs), Bidirectional LSTM (BiLSTM), and Convolutional Neural Networks (CNN) help with nonlinear temporal relationships in data [10],[11]. Hybrid DL models and metaheuristic optimization algorithms like Artificial Rabbits Optimization (ARO) enhance performance and adaptability [12].

The introduction of LLMs were developed in the natural language understanding and generation domain, which later brought paradigm shift in the future modeling of sequential data. Transformer-based models have been shown to excel at capturing contextual connections on long sequences, thus being able to process not only textual data but also numerical time series where relevant representation formats are used. In contrast to models specifically designed for numerical data, LLMs take advantage of large-scale pretraining over heterogeneous corpora, which could implicitly include macroeconomic and market-related knowledge and thus improve predictive power when fine-tuned for particular financial prediction tasks. Nonetheless, systematic evaluations of current LLMs on sole stock price prediction tasks are still rare. Furthermore, most previous works consider one stock or dataset, restricting the generality of conclusions across various market conditions and asset classes. Therefore, this study presents a comprehensive evaluation of five prominent LLMs on stock market price prediction. The key objective of this study is to systematically evaluate and compare predictive performance of five LLMs for stock price prediction, using heterogeneous real-world datasets to identify suitability for financial time series forecasting across diverse market settings. The major contributions of study are summarized as follows:

The study conducts large-scale comparative analysis of five LLMs including LLaMA 3, Falcon 2, Snowflake Arctic, DeepSeek-V2, and FinGPT for specific task of stock price prediction across various datasets.

The study designs uniform experimental pipeline for fine-tuning and evaluating LLMs on financial time series, ensuring methodological precision and reproducibility.

This study provides comprehensive analysis with domain-specific performance patterns offering guidance for future applications of LLMs in financial prediction.

The rest of this manuscript is structured as follows: The existing works in financial forecasting and LLM applications is reviewed in Section 2. The models and datasets used for evaluation are described in Section 3. The experimental setup and results analyses are offered in Section 4. The future research directions are determined in Section 5.

2. LITERATURE REVIEW

The price prediction of stock has already been intensively investigated from variety of

methodological paradigms. This section summarizes previous studies in each of these streams of methodology, to provide an understanding of those gaps. Chen [13] examined Large Language Model Meta AI (LLaMA) model for stock price prediction using prompt-based model using candlestick charts. The study also explored teacher-student model with LLaMA and Qwen as students. The impact of the prompt-based LLM approach was evaluated against other established models like CNN, ResNet, and Vision Transformer. To increase the accuracy of stock index prediction, Qi et al. [14] developed a system that integrates GRU with complete ensemble empirical mode decomposition of adaptive noise (CEEMDAN). High-frequency disturbances in sub-signals were denoised using wavelet threshold technique, which removes noise contamination for upcoming forecasts.

Yan & Huang [15] analyzed that the Mamba-LLM framework is a novel approach to stock price prediction that combines macro-index and micro-stock information. It used state-space models (SSMs) and LLMs to generate temporal representations of stock level data. The macro index analyzer extracts market-level index correlations to create textual summaries. The architecture then converts these summaries into short token embeddings using FinBERT. By merging these representations and refining them using multi-layer perceptrons (MLPs), the architecture can dynamically model price movements based on asset-level financial instrument behavior and systemic market trends. Temporal Convolutional Network (TCN) and Attention were introduced by Biswas et al. [16] to forecast stock prices and assess risk for Visa and MasterCard. The model predicts stock prices, key risk measures, and volatility using dilated convolutions. The Attention model ensures accurate predictions for both short and long-term dependencies in stock price data by optimizing its focus on critical time steps. Gülmez [17] looked into GA-Attention-Fuzzy-Stock-Net model, which blends Genetic Algorithms (GA), attention mechanisms, and neuro-fuzzy systems. Several fuzzy layer membership function designs and NN architectures were used to test the model. Sliding windows were used to assess model's performance over a range of time ranges. The attention mechanism concentrated on pertinent temporal patterns, while GA were tuned to minimize learning rates and hyperparameters.

Amiri et.al., [18] presented a hybrid approach to predicting energy stock prices, which is complex due to geopolitical influences, regulatory changes, and sector-specific issues. The approach connects Graph Convolutional Network (GCN) with an attention LSTM model, learning relationships from Dynamic Time Warping (DTW) data. The LSTM with attention model improves temporal dynamics by focusing on historically most important data, thereby enhancing the accuracy of energy stock data forecasting. Chen [19] suggested several techniques for leveraging sequential data to forecast stock movements. The study takes into account models with numerical data, LLMs, and vision-based DL models that used stock image as input. The study specifically examined prompt-based LLM framework that analyzes candlestick charts and evaluates its performance against models that rely on images, like CNN, MobileNetV2, and Vision Transformer, as well as models that require numerical input, including SVM, RF, LSTM, and CNN-LSTM. According to Mukherjee et al. [20], the stock market is continuously changing field of study, and it is essential to predict its characteristics. This calls for thorough data analysis with the use of artificial intelligence (AI) algorithms and statistical models. ML and DL algorithms were used to improve predictions with fewer errors. Artificial Neural Networks (ANN) and CNN were significant models used for stock market prediction.

A clustering-enhanced DL framework was proposed by Li et al. [21] to increase accuracy of stock price prediction utilizing three pre-existing models: LSTM, RNN, and GRU. Additionally, the framework extended Weighted Dynamic Time Warping (WDTW) methodology by introducing new similarity measure for Histogram Weighted Dynamic Time Warping (HWDTW). A CNN-Bi-LSTM-Attention based model was presented by Zhang et al. [22] for forecasting stock prices and indexes. The CNN and Bi-LSTM networks were used in the model to extract temporal properties from sequence data. This method worked very well with long-memory, nonlinear, and high-frequency stock price data.

Although the usefulness of DL and Transformers in predicting stocks is well established by previous studies, and LLMs have been applied widely to text-heavy applications like sentiment analysis and event-based trading, they have not been applied much to forecasting stock prices directly. Furthermore, there is no large-scale comparative study of LLMs when applied to a numeric stock price forecasting task. These abovementioned gaps are especially relevant considering the heterogeneous architectures and pretraining configurations, which can result in diverse performance over data with different market settings, volatility regimes, and horizons. This

research fills these gaps through experimental evaluation of five LLMs such as LLaMA 3, Falcon 2, Snowflake Arctic, DeepSeek-V2, and FinGPT across three stock datasets.

3. METHODOLOGY

The predictive modeling approach utilized in this study is intended to thoroughly assess performance of state-of-the-art LLMs in prediction of daily closing stock prices. The evaluation methodology uses three design principles: (i) architectural diversity, to sample a variety of modeling paradigms from various LLM designs; (ii) experimental fairness, through the use of standardized preprocessing, training settings, and evaluation measures; and (iii) reproducibility, such that the procedures of this study can be repeated and built upon by future research. All models are considered to be fine-tuned on these sequences to forecast the following trading day's closing price, allowing an immediate comparison of their predictive precision and direction consistency. The following subsection explains the chosen models, datasets, and fine-tuning approaches.

Dataset

Three publicly accessible stock price data from Kaggle are employed to compare LLMs in various market settings. Daily trading data including date, open, high, low, close, adjusted close, and volume comprise each dataset. The Google Stock Prediction dataset [23] covers 14 June 2016 to 11 June 2021, totaling 1,258 trading days. It is a recent, high-growth era with significant volatility, creating an opportunity for LLM performance testing on shorter series. closing of stock value. The dataset includes following variables: high, low, open, volume, adjClose, adjHigh, adjLow, adjOpen, adjVolume, divCash, splitFactor.

The Reliance Industries (RIL) Share Price dataset [24] spans 01 January 1996 to 27 November 2020 with 6,205 trading days, capturing long-term developing market behavior over several economic cycles. This dataset contains the variables such as Prev Close, Open, High, Low, Last Close, Volume weighted average price (VWAP), Volume, Turnover, Trades, Deliverable Volume, and % Deliverable.

The Apple Stock Price Prediction Dataset [25] range is 02 January 1998 to 13 March 2024, covering 6,639 trading days of mature market activity with deep historical background. This dataset contains the attribute variables such as Open, volume, High, close, and Low. Table 1 shows the characteristic of three datasets.

Table 1. Summary of datasets

Dataset	Timespan	Records	Frequency
Google	14/06/2016 – 11/06/2021	1,258	Daily
Reliance	01/01/1996 – 27/11/2020	6,205	Daily
Apple	02/01/1998 – 13/03/2024	6,639	Daily

Preprocessing

The preprocessing of financial time-series data is critical to reliability and efficacy of LLM-based stock price forecasting. In this study, each dataset was cleaned, transformed, and formatted separately prior to model training and assessment. The preprocessing process involved following methods:

Handling Missing and Inconsistent Values

Each data set was initially screened for missing records, inconsistent date formats, and duplicate records. Any breaks in the time series were treated with utmost care through forward-fill interpolation to ensure continuity of trading days without any artificial bias. Duplicate records were eliminated to ensure dataset integrity.

Feature Selection and Transformation

The original data sets had several attributes. In modeling, the Close price was utilized as the main target of prediction, with other features maintained as predictors to reflect intraday market behavior. Non-numeric attributes were transformed into suitable datetime formats for temporal indexing.

Normalization

The Min-Max normalization method is used to normalize the scale of attributes and avoid dominance of attributes with high magnitudes. This method normalizing each attribute between a range of 0 and 1. This provided equal feature influence across LLMs and allowed for quicker convergence during training.

Time-Series Structuring

Since LLMs need sequential contextual data, the datasets were converted to sliding window sequences, in which every input sequence had a fixed number of preceding trading days to forecast next day's closing price. This preserved temporal dependencies that are necessary for prediction tasks in stock market.

For every stock, the preprocessed data was divided into training, validation, and testing sets with a chronological split to avoid data leakage. The training set was utilized for model learning, the validation set for hyperparameter tuning, and the testing set for ultimate performance assessment. These preprocessing operations converted the

datasets into clean, structured, and model-ready formats, allowing for a fair and consistent comparison across all LLM architectures.

Overview of LLMs

Five LLMs including LLaMA 3, Falcon 2, Snowflake Arctic, DeepSeek-V2, and FinGPT are used in this study to test how well the LLMs can predict stock prices. LLaMA 3 is known for its light architecture and robust contextual reasoning skills, which are ideal for financial time-series analysis. Falcon delivers high performance with optimal inference, which is perfect for large-scale prediction. Snowflake Arctic is fine-tuned for low-latency and structured data integration and supports fast turnaround in market forecasting applications. DeepSeek-V2 employs deep contextual embeddings to optimize predictions on sequential financial data. FinGPT, particularly expressed for financial use cases, combines domain-specific knowledge to offer enhanced predictive accuracy in stock trends. All these models together offer complete benchmark to evaluate the applicability of LLM for financial forecasting. Each LLMs are described in subsequent section.

LLaMA 3

The LLaMA model [26] follows the Transformer decoder-only structure; exclusively optimized to be trainable and compute-efficient in a range of parameter scales. Figure 1 shows the model structure of LLaMA3 framework. It has a dense attention mechanism featuring pre-normalization in which Layer Norm is applied prior to self-attention layer as well as feed-forward layer to improve stability in deep training. LLaMA adds rotary positional embeddings (RoPE) to more adequately model long-range relations with respect to fixed positional encodings. This architecture uses SwiGLU activation instead of ReLU or GELU, which yields stronger expressiveness and gradient. Its multi-head self-attention (MHSA) units enable parallelization of representation of tokens and its feed-forward networks (FFNs) are widened to have an increased capacity to learn representations. LLAMA is pre-trained using large and varied mixture of datasets, making it perform robust language comprehending tasks in many fields, and trains it with efficient mixed-precision training strategies and requires less memory utilization. The lesser versions of LLaMA are intended to be competitive with quite bigger LLMs through the utilization of quality data and more extended training timings, and they can therefore be used in both lookup forms and production.

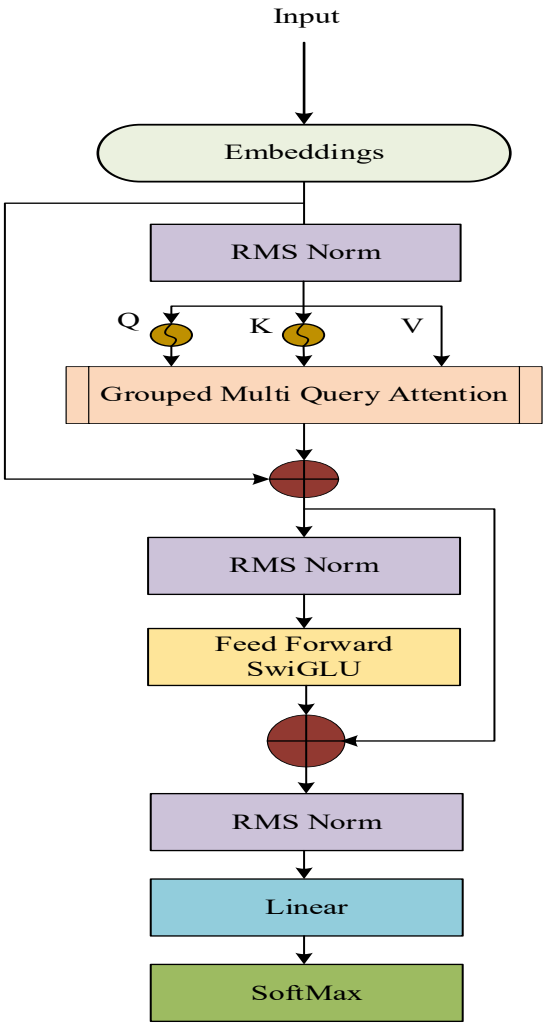


Figure 1. Architecture of LLaMA3

FinGPT

A decoder-only transformer called LLaMA2 [27] serves as the foundation for FinGPT [28]. The Transformer decoder design, which was optimized for autoregressive language modeling, is carried over into LLaMA2 model. The model is illustrated in Figure 2. Each of the N stacked decoder blocks in the model is made up of FFN and Multi-Head Self-Attention (MHSA). The model, which incorporates residual connections and normalization, calculates token-wise depictions over successive MHSA and FFN layers an input sequence $Z = (z_1, z_2, \ldots, z_k)$ with embeddings $z_i \in \mathbb{R}^d$. In MHSA, each attention head is calculated using Eqn. (1).

$$ATN(Q,K,V) = SoftMax\left(\frac{QK^{\intercal}}{\sqrt{d_k}}\right)V \quad (1)$$

Where learnable matrices ω^Q, ω^K, and $\omega^V \in \mathbb{R}^{d \times d_k}$ are projections with values $Q = Z\omega^Q$, $K = Z\omega^K$, and $V = Z\omega^V$. The Multi head extension is expressed as Eqn. (2).

$$MHSA(Z) = Concat(H_1, H_1, \ldots, H_N)\omega^o \quad (2)$$

LLaMA introduces position-dependent rotations to preserve relative token information by substituting Rotary Positional Embeddings (RoPE) for absolute positional encodings. This improves generalization over different lengths of sequences. The representation of each token is run via two-layer FFN as Eqn. (3).

$$MHSA(x) = \omega_2 \cdot GeLU(\omega_1 x + b_1) + b_2 \quad (3)$$

Pre-layer normalization and residual connections stabilize layer outputs as Eqn. (4) and (5).

$$z' = z + MHSA\big(LayerNorm(z)\big) \quad (4)$$

$$z'' = z' + FFN\big(LayerNorm(z')\big) \quad (5)$$

The model is trained using next-token forecast by minimalizing negative log-likelihood as Eqn. (6).

$$F(L) = -\sum_{k=1}^{K} \log P(z_k | z_{<k}; \theta) \quad (6)$$

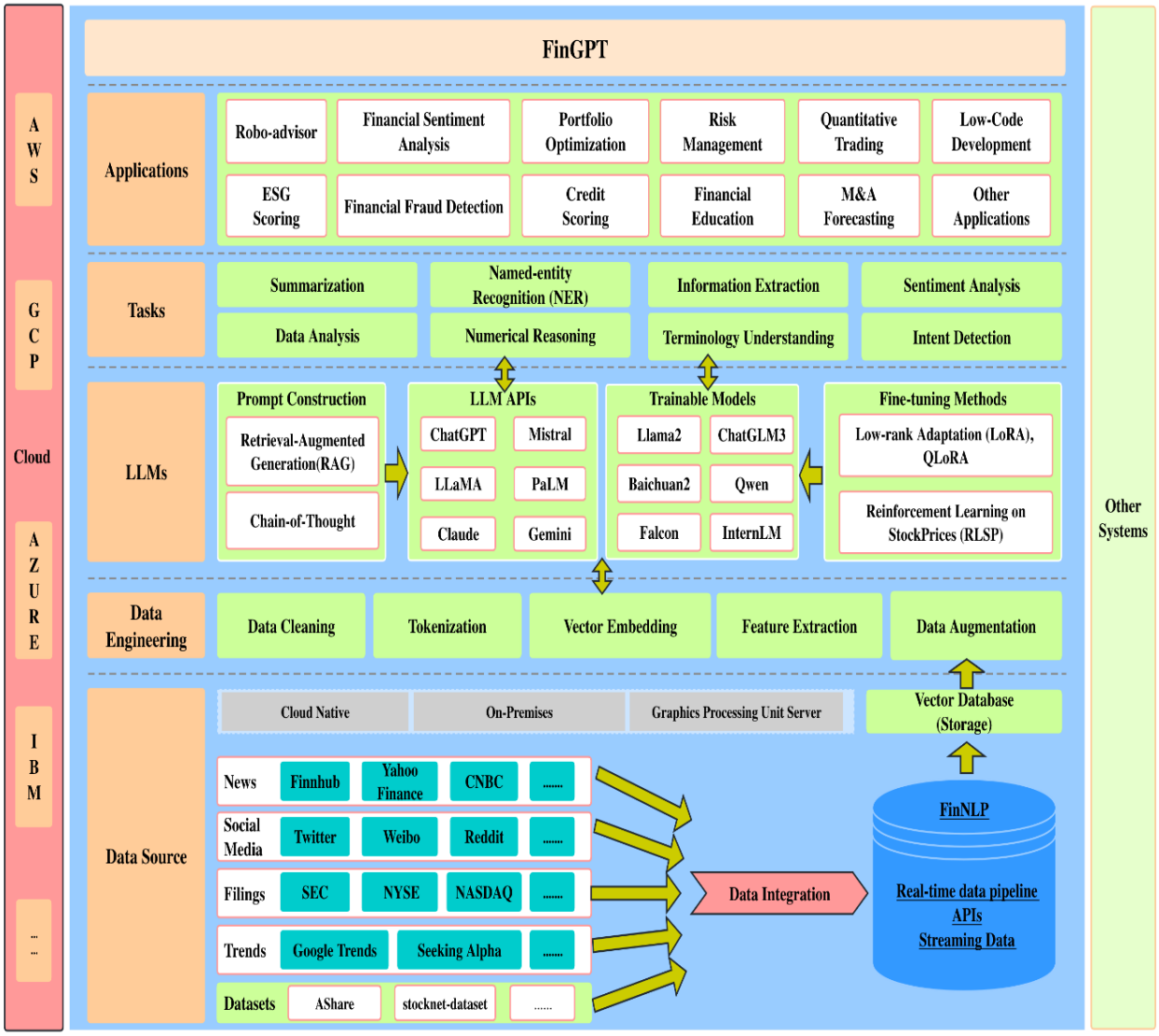


Figure 2. Architecture of FinGPT framework (source:[31]).

Snowflake Arctic

The Snowflake Arctic is a transformer-based hybrid reasoning model. It enhances regular transformer blocks with context-aware gating mechanism, facilitating dynamic financial time-series signal adaptation and external textual guidance. Its architecture combines efficient attention with modular feed-forward paths to capture regime-shift signals and general patterns. The subsequent section describes components of Snowflake Arctic architecture and the model is shown in Figure 3.

Input Embeddings

Raw inputs such as numerical sequences and optional textual abstractions are initially embedded into shared latent space through specialized embedding layers. The initial Normalization (Norm) layer provides normalized activation distributions before further processing deeper. This shared embedding serves as the foundation "context stream"

flowing through all following layers, just as the "Input" flows in the figure you have given.

Multi-Head Attention (MHA) Block

This MHA block follows normalized embeddings with multi-head self-attention module to extract dependencies between time steps and modalities. Outputs are residual-connected back into the context stream to retain information and stabilize gradients.

FFN Layers

Post-attention output is normalized again and passed through a dynamic, gated feed-forward module instead of a single dense layer. The module consists of several candidate feed-forward networks (FFN_1, FFN_2, …, FFN_n), each dedicated to distinct temporal or market regimes (e.g., trending vs. volatile markets, day vs. earnings-shock regimes). A trained Gating Mechanism (MLP with SoftMax activation) inspects the normalized context and dynamically directs the signal to most suitable FFN(s), producing a weighted sum. This gating facilitates flexible, regime-optimized transformations in each layer, enhancing adaptability and robustness to shifting market scenarios.

The gated output is subsequently combined with the context stream via a residual connection, facilitating stable deeper stacking and improved gradient flow. The MHA and FFN blocks are vertically stacked to form L layers (typically 6–12 layers), creating a deep context-aware encoder able to grasp both short- and long-term dependencies. Following last layer, the context embeddings go through task-specific heads.

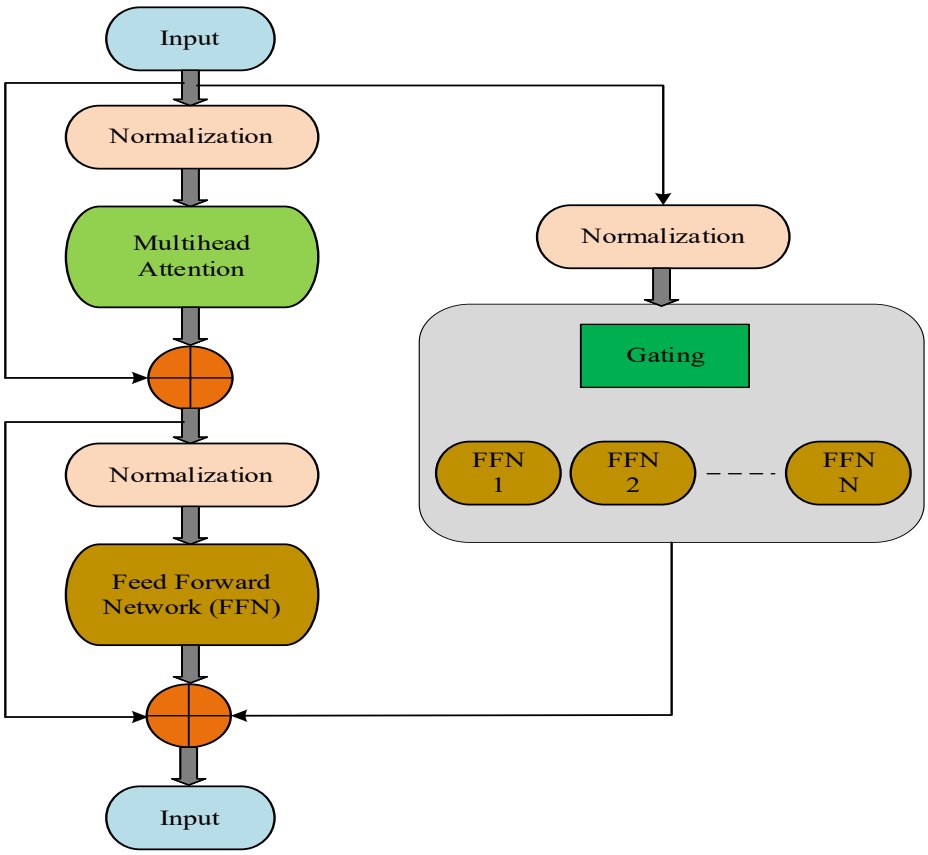


Figure 3. Architecture of Snowflake Arctic

Deepseek-v2

The DeepSeek-V2 [29] is the second generation of DeepSeek model, and DeepSeek-V2 enhances reasoning abilities as well as computation efficiency with architectural modifications and training approaches. It takes Mixture-of-Experts (MoE) Transformer design, which allows activating a smaller portion of model parameters per token to promote excellent inference efficiency at the expense of performance. This model defined 16 experts in the feed-forward network layer, 2 experts are activated per token. Such method saves computational overhead and offers a vast total number of parameters.

Architecture

DeepSeek-V2 is an autoregressive language model that uses a decoder of Transformer architecture, but also incorporates a number of designs to maximize the model parameter efficiency

and reasoning ability. As shown in Figure 4, the model uses Mixture-of-Experts (MoE) layer in FFN layers, with 16 distinct experts with only 2 experts activated for given token. Such a selective activation scheme enables the model to have an impressively high parameter capacity with a major computation reduced cost on single inference step. Furthermore, Rotary Positional Embeddings (RoPE) which are employed to encode sequence information allow the model to capture relative positions between tokens in more flexible manner and also enable it to generalize better in different input sequence lengths. The Multi-Head Latent Attention (MHLA) mechanism is optimized that projects keys and values across all attention heads, leading to a reduction of memory usage and a speedup of the inference without a decrease in the contextual understanding. Additionally, DeepSeek-V2 adds parallel residual connections, where the outputs of attention and feed-forward sublayers are computed in parallel and then integrate with residual stream, increasing gradient flow and minimizing training instabilities. This architectural expansion makes DeepSeek-V2 high-capacity model that can scale to support long historical contexts of up to 16,384 tokens, a critical capability of financial forecasting applications that necessitate combination of long historical contexts and a variety of market indicators.

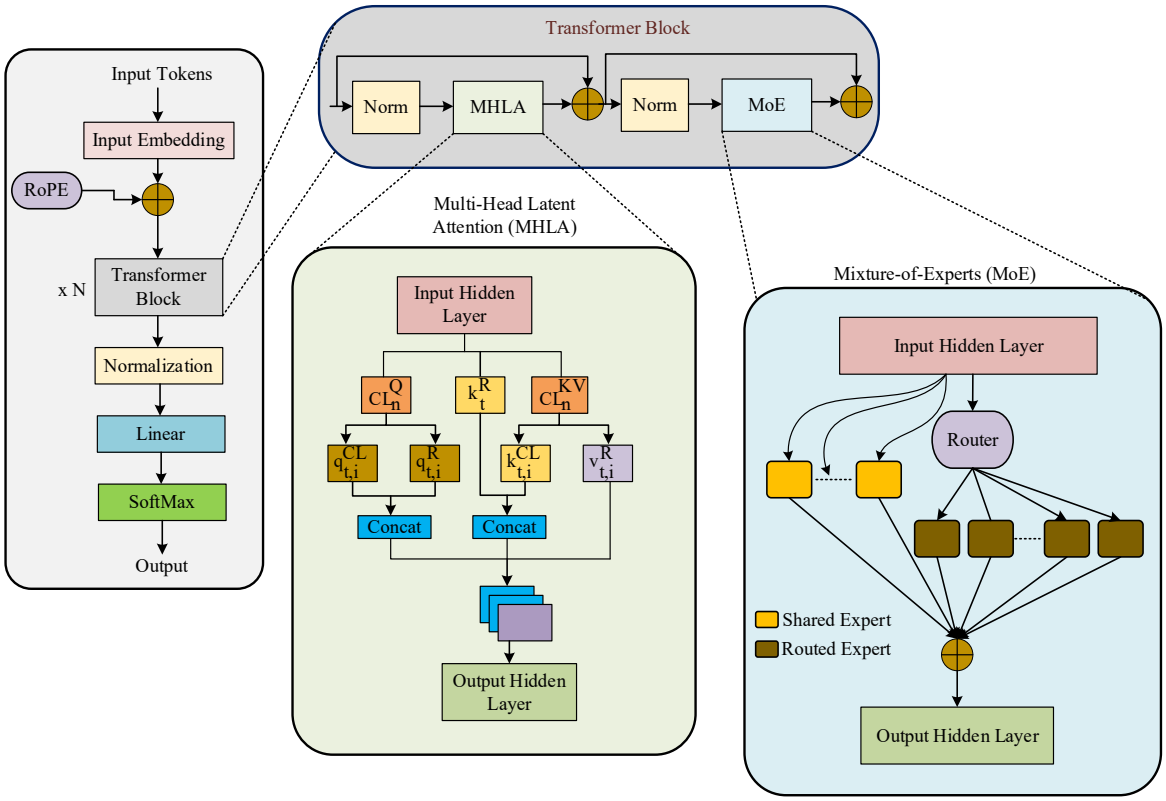


Figure 4. Architecture of DeepSeekV2

Falcon 2

The Falcon 2 [30] is a decoder-only Transformer implemented to demonstrate high performance and short inference latency. The Falcon 2 model is shown in Figure 5. One of such differences is that it uses multiquery attention (MQA) as opposed to conventional multi-head attention, which offers much less memory overhead and enhances the ability to scale longer context win windows. MQA does not require each of the attention heads to maintain independent key and value projections, and thus it performs inference more quickly and accurately at the cost of little accuracy. Falcon 2 also exploits rotary positional embeddings in position encoding, making better generalization to long sequences possible. The neural network of the feed-forward networks within the Falcon 2 embarks on a non-linear expression on gated linear unit style structure. Moreover, it is trained on a large, curated, deduplicated dataset in order to reduce contamination and bias. Its architecture is both batch and streaming inference-optimized whereas extensive engineering advances towards parallelism and memory efficiency took place. The design of Falcon 2 focuses on readiness in industrial deployment with a trade-off between quality of the models and computation costing which is the most ideal application to large scale real-time processes.

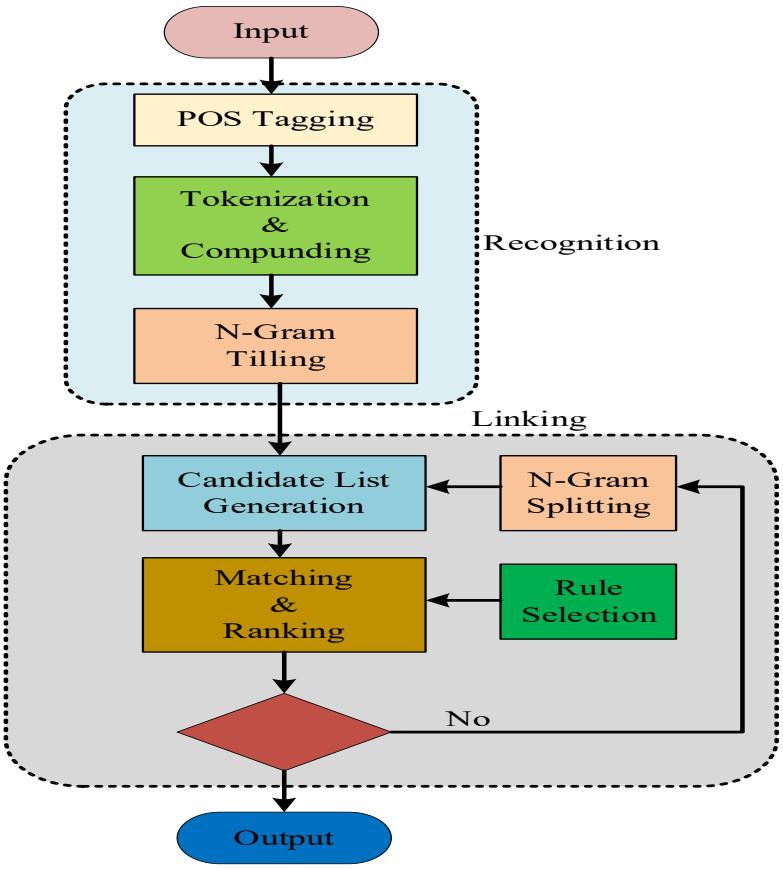


Figure 5. Architecture of Falcon 2 Framework

Input Representation

Though LLMs are mostly optimized towards natural language understanding, their basis being transformer architecture, means that they can process structured numerical data provided they are encoded in text-based sequences appropriately. In this study, we have used a sliding-window encoding scheme to adjust historical stock price data into an encoding that can be consumed by each of model's tokenizer. In deciding the prediction value of each observation, the model was fed with the past 30 trading days of market data. These aspects were rolled into a tabular form of textual visualization where each day was represented as a formatted row of text. The last line of both sequences was made up of natural language instruction indicating the model to forecast close of succeeding day of trading. This method not only maintained temporal ordering of observations but also enabled the model to use its sequential ability of reasoning and also guaranteed that numerical values were maintained as exact tokens to play safe with rounding off errors.

4. RESULTS AND DISCUSSION

This section demonstrates comparative outcomes of five LLM models, namely LLaMA 3, Falcon 2, Snowflake Arctic, DeepSeek-V2, and FinGPT, on stock closing price prediction task using daily data on the Google, Reliance industries, and Apple companies' datasets. The results are presented as per the evaluation metrics presented in Section 4.3.

Experimental Setup

All experiments were executed on Windows 11. The computer setup was: an Intel Core i9-13900K (24 cores, 3.0 GHz base clock) processor, and 64 GB DDR5 RAM. The Python 3.11 was used as the base software, PyTorch 2.2 and Hugging Face Transformers 4.39 library are used to integrate and fine-tune the current models.

Evaluation Metrics

Model performance was evaluated on metrics that reflect various aspects of the quality of forecasting. The Mean Squared Error (MSE) was used as anchor loss-based metric, weighing larger differences between predicted and actual values more heavily. The Mean Absolute Error (MAE) gave an interpretable average prediction error measure in price units, whereas Root Mean Squared Error (RMSE) offered scale-consistent MSE interpretation in same units as the initial prices. The Mean Absolute Percentage Error (MAPE) was presented to set

prediction error into relative percentage terms so that the stocks may be compared across despite varying price ranges. The Coefficient of Determination (R^2), which measures percentage of variance in actual prices that can be accounted for by model's predictions to quantify predictive explanatory power. The description of each metric is depicted in Table 2.

Table 2. Performance Metrics

Metric	Description
MSE	$\frac{1}{m}\sum_{i=1}^{m}(\hat{Z}_i - Z_i)^2$
MAE	$\frac{1}{m}\sum_{i=1}^{m}\lvert\hat{Z}_i - Z_i\rvert$
RMSE	$\sqrt{\frac{1}{m}\sum_{i=1}^{m}(\hat{Z}_i - Z_i)^2}$
MAPE	$\frac{100\%}{m}\sum_{i=1}^{m}\left\lvert\frac{\hat{Z}_i - Z_i}{Z_i}\right\rvert$
R^2	$1 - \frac{\sum_{i=1}^{m}(\hat{Z}_i - Z_i)^2}{\sum_{i=1}^{m}(Z_i - \bar{Z}_i)^2}$

Model Training

To fine-tune domain-specific and general-purpose LLMs for numeric prediction of stock prices, we investigated two paradigms of fine-tuning. The first was an instruction-tuning paradigm, where the sequence of historical data was then followed by a natural language question plainly asking for closing price on following day. This paradigm was intended to motivate the model to continue its natural language reasoning process but output a numeric answer. The second strategy, direct prediction fine-tuning, eliminated the instructional text and trained the models to predict only the numeric value of the predicted closing price, thus setting the learning process entirely on regression accuracy. Both strategies were run with same hyperparameters to make fair comparison between models.

Table 3. Hyperparameters of Model training

Parameters	Value
Optimizer	AdamW
Learning rate	$1x10^{-5}$
Batch Size	16
Epochs	30
Loss function	MSE

To maintain integrity of time-series forecasting and to prevent look-ahead bias, three datasets were divided into training (70%), validation (15%), and testing (15%) subsets. This time separation provided the models access to historical data only while predicting, most closely approximating actual forecasting situations. Each of the three data sets was processed and divided separately so model performance could be individually measured per asset. To further consider variability in the results due to random initialization and training dynamics, each experiment was run three times using varying random seeds. The last performance measures are given as the mean ± standard deviation over these runs, and this is a conservative estimate of prediction capacity.

Performance Evaluation of LLMs

Table 4 consolidates five LLMs such as LLaMA-3, Falcon-2, Snowflake Arctic, DeepSeek-V2, and FinGPT to forecast performance on three stock datasets (Reliance, Apple, Google). Performance was measured by MAE, RMSE, MAPE, and R2, offering multi-dimensional assessment of accuracy, scale-consistent error, relative error, and explanatory power.

Table 4. Evaluation Analysis of Models of various Datasets

Model	Dataset	MAE	RMSE	MAPE (%)	R^2
LLaMA-3	Reliance	12.45	18.72	2.85	0.982
	Apple	1.95	3.12	1.12	0.991
	Google	24.1	30.42	2.48	0.987
Falcon-2	Reliance	14.02	20.35	3.12	0.978
	Apple	2.22	3.45	1.28	0.989
	Google	26.54	33.12	2.72	0.984
Snowflake Arctic	Reliance	13.11	19.4	2.96	0.98
	Apple	2.05	3.21	1.2	0.99
	Google	25.42	31.45	2.58	0.986
DeepSeek-V2	Reliance	11.88	17.95	2.72	0.984
	Apple	1.82	2.95	1.05	0.992
	Google	23.12	29.6	2.39	0.988
FinGPT	Reliance	10.92	17.02	2.5	0.986
	Apple	1.75	2.88	1.02	0.993
	Google	22.05	28.44	2.28	0.989

FinGPT had the best or closest approach to best performance on all datasets with lower MAE, RMSE, and MAPE and also attained with highest R^2 scores. For instance, FinGPT realized lowest MAE and RMSE score at 10.92 and 17.02 respectively, while its MAPE stands at 2.50% and R2 at 0.986 on Reliance dataset. On Apple dataset, FinGPT achieved lowest MAE (1.75), RMSE (2.88), and MAPE (1.02%), along with highest R^2 value of 0.993, showing nearly perfect fit to real stock price trends. The second-best performer overall was DeepSeek-V2, which ranked consistently notch below FinGPT. DeepSeek-V2 got MAE = 23.12, RMSE score of 29.6, and MAPE value

at 2.39% on Google dataset, slightly inferior than FinGPT but significantly better than LLaMA-3, Falcon-2, and Snowflake Arctic. This performance indicates that DeepSeek-V2's architecture is also well-suited for both short-term volatility and long-term stock price trends capture. LLaMA-3 provided competitive performance, especially on Apple and Google datasets, with R^2 values all above 0.987. Its error statistics were, however, slightly higher than FinGPT and DeepSeek-V2, which means marginally lower short-term price accuracy although variance explanation was very good. Snowflake Arctic had results similar to LLaMA-3, with steady but slightly worse performance across all datasets. Falcon-2, although still returning high R^2 performances (>0.978), registered highest errors in nearly all cases, indicating that its pretraining or fine-tuning process could be less optimized for financial time series prediction task than other LLMs. Figures 6-9 illustrates the performance analysis of LLMs.

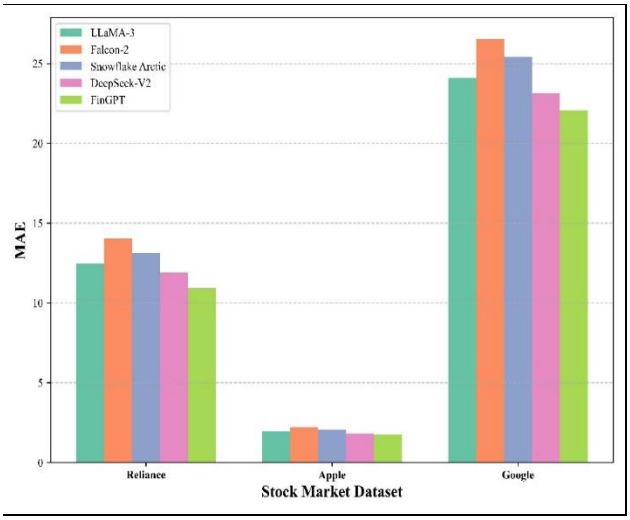


Figure 6. MAE Analysis of LLM Models

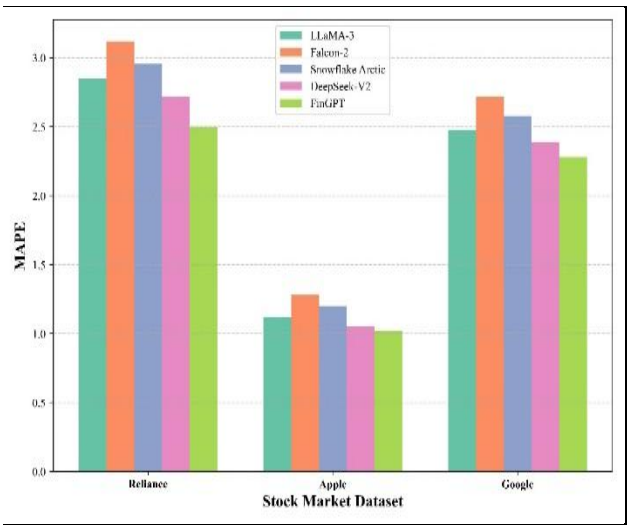


Figure 7. MAPE Analysis of LLM Models

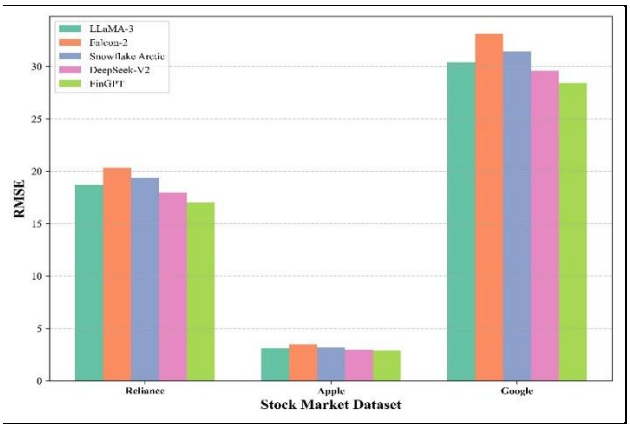


Figure 8. RMSE Analysis of LLM Models

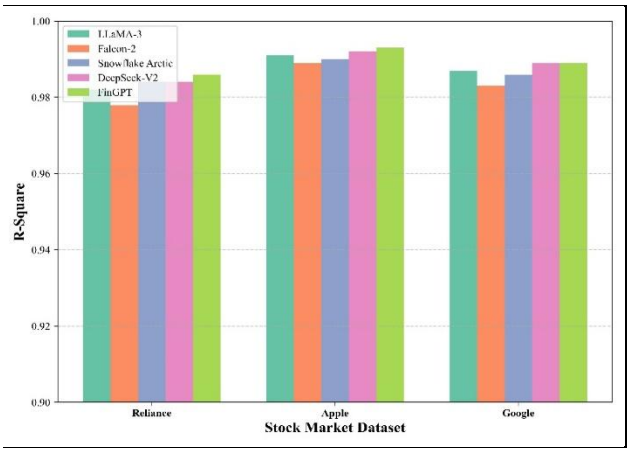
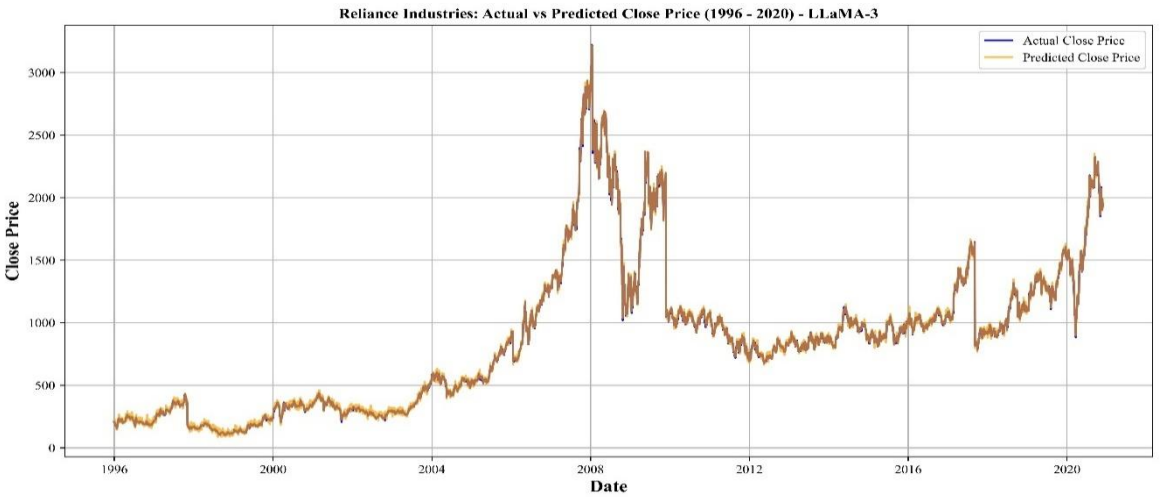


Figure 9. RMSE Analysis of LLM Models

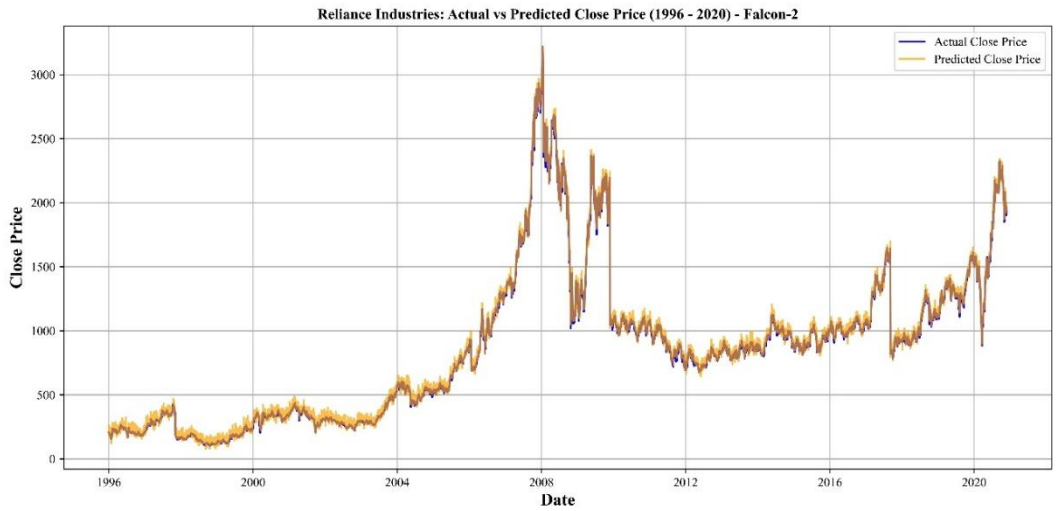
Performance differed mildly by dataset according to variations in historical volatility, record length, and market nature. In Apple Dataset, all models had their best overall performance on predicting Apple stock, with MAE ranging from 1.75 to 2.22, RMSE ranging from less than 3.45, and MAPE ranging close to or below 1.3%. This was probably due to the rather smooth and long-term steady trend in Apple's historical price series, which made it more predictable to LLMs. In Reliance Dataset, the errors were greater than those in the Apple dataset (MAE of 10.92–14.02), perhaps because the dataset is more long-term (1996–2020) and has been exposed to major structural breaks in the Indian market. FinGPT's better performance here suggests its strength in working with long history sequences with structural turbulence.

In Google Dataset, prediction errors were greatest in absolute terms (MAE from 22.05 to 26.54), but MAPE values remained low (~2.3%–2.8%), reflecting that while absolute price gaps were greater, relative accuracy of prediction was robust. The shorter time duration of the Google dataset (2016–2021) may have restricted exposure to varied macroeconomic environments, but greater nominal stock prices

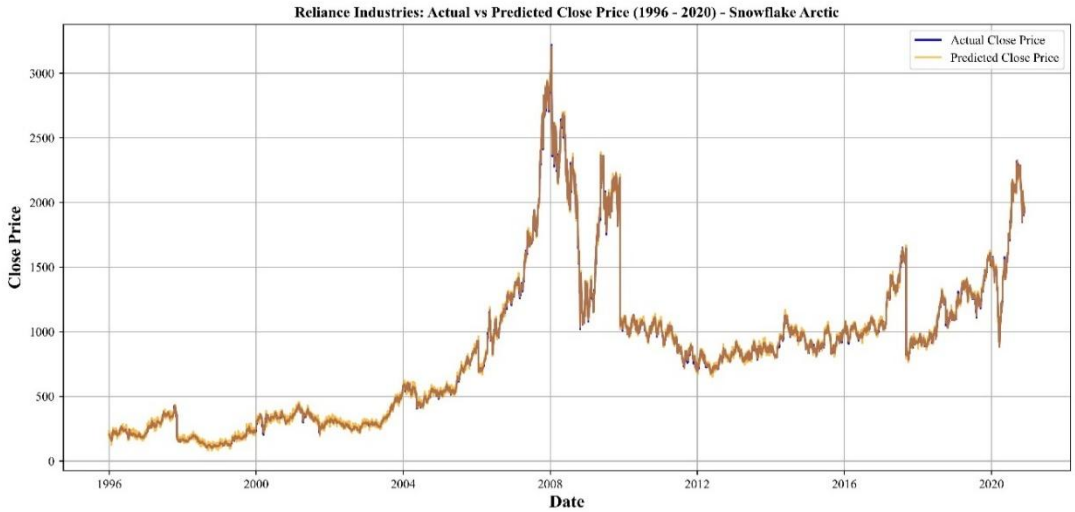
inflated MAE and RMSE. To enhance the quantitative parameters of performance, graphical analysis was done by comparing the approximations given by each LLM on predictions and the actual movement of stock prices over the time. Figures 3 to 5 show actually and predicted closing prices of Reliance Industries, Apple, and Google, respectively, in all five models that are assessed.



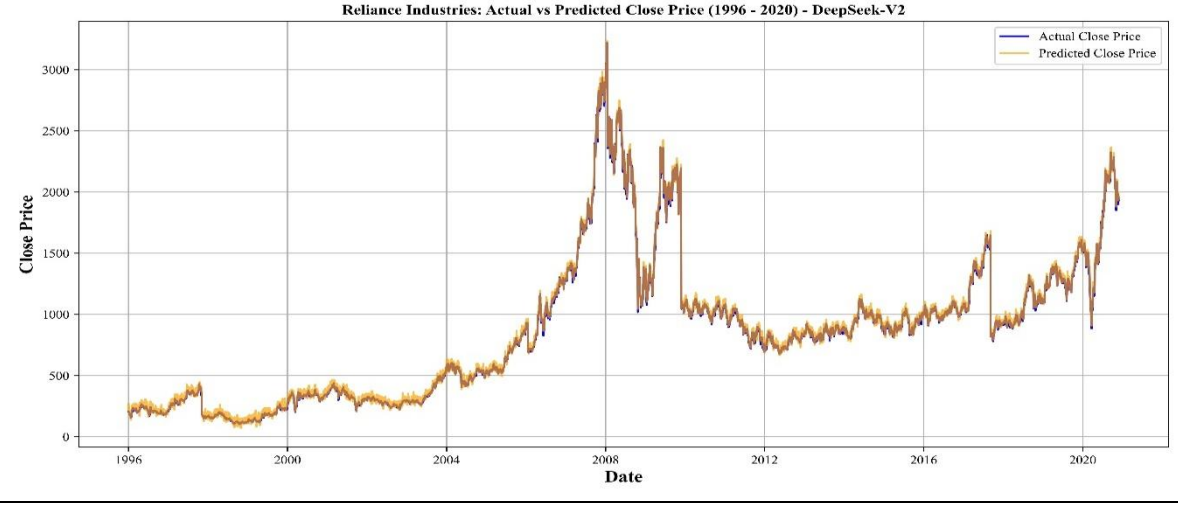
LlaMA-3



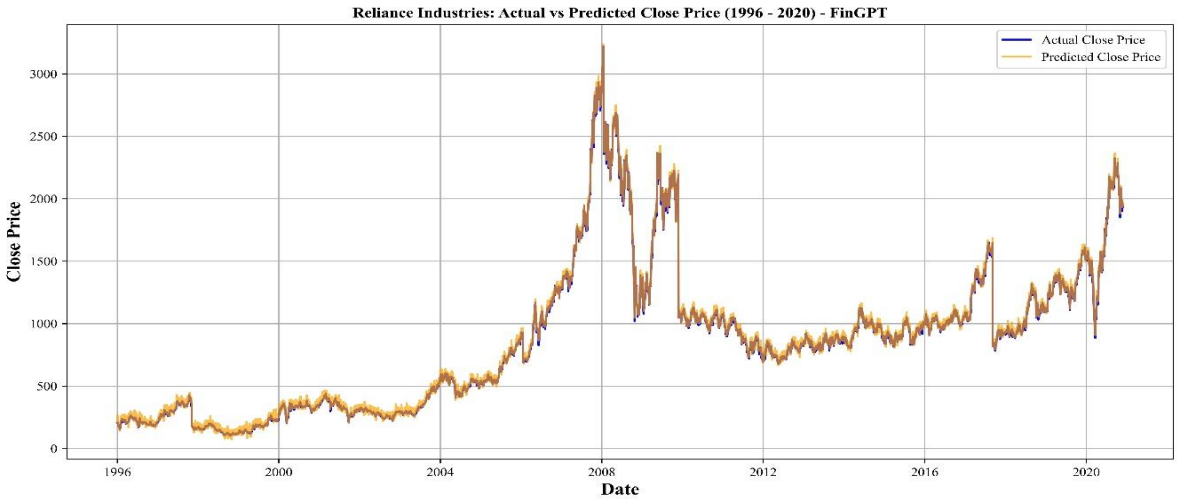
Falcon-2



Snowflake Arctic

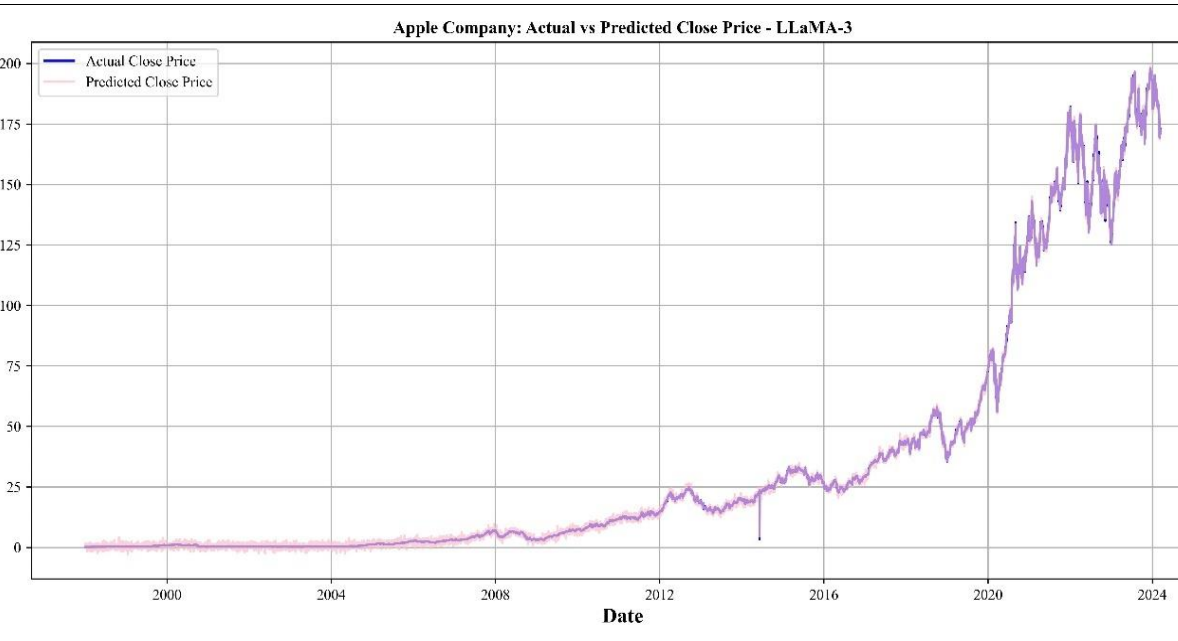


DeepSeek-V2

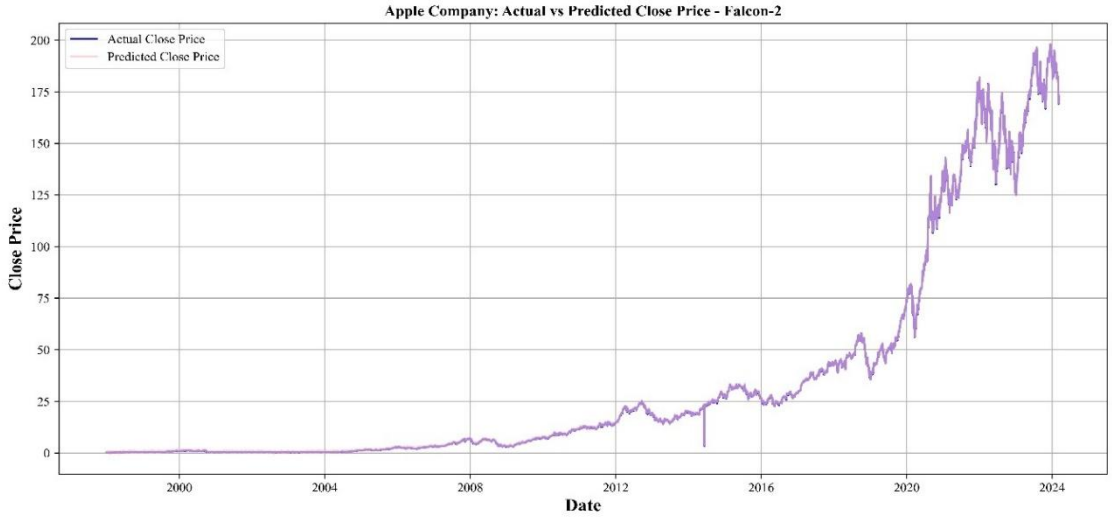


FinGPT

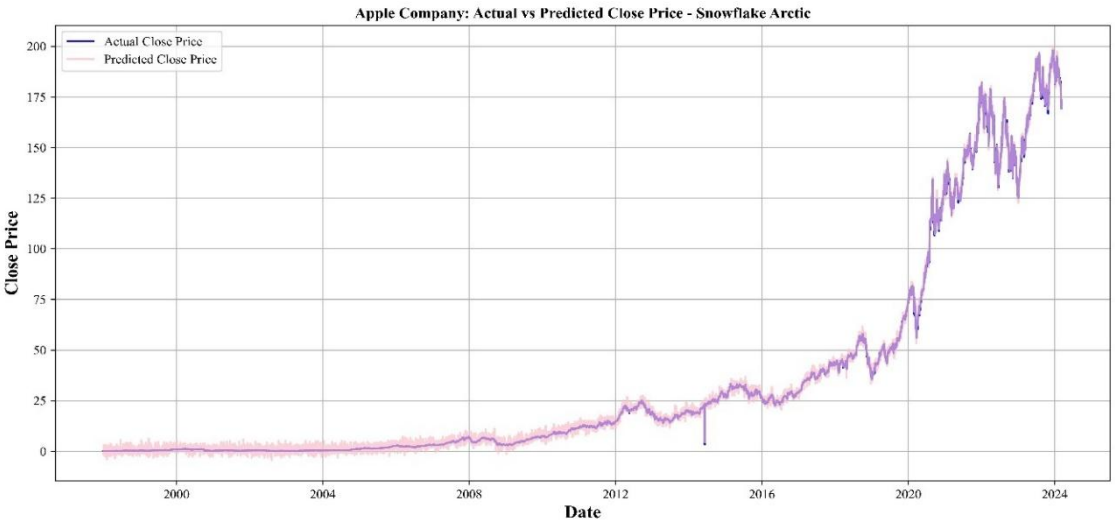
Figure 10. Actual and Predicted Price of models on Reliance Dataset



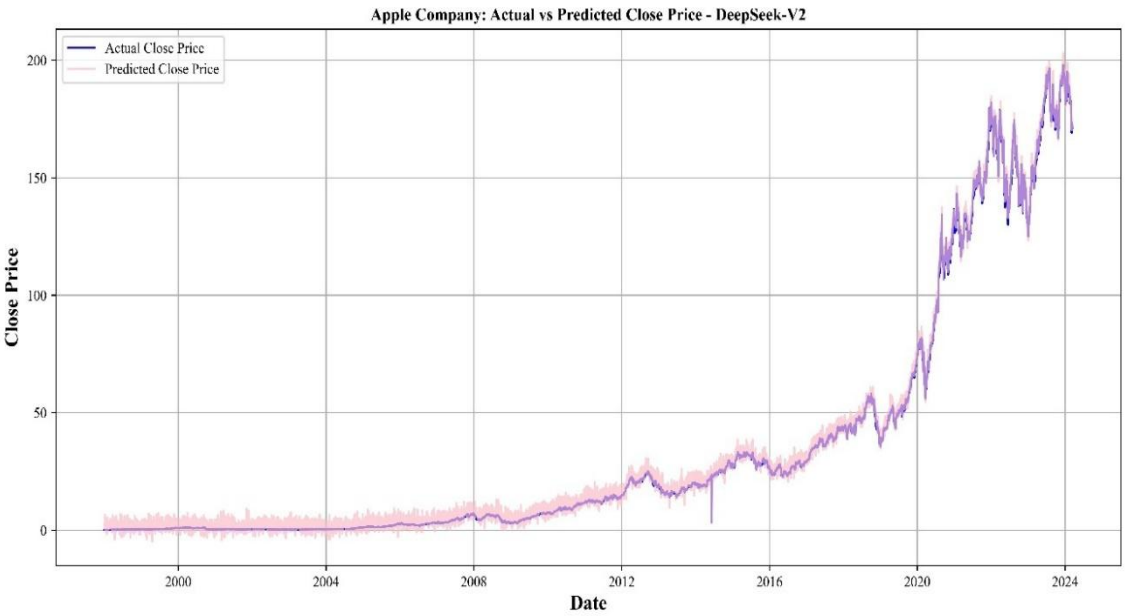
LlaMA-3



Falcon-2



Snowflake Arctic



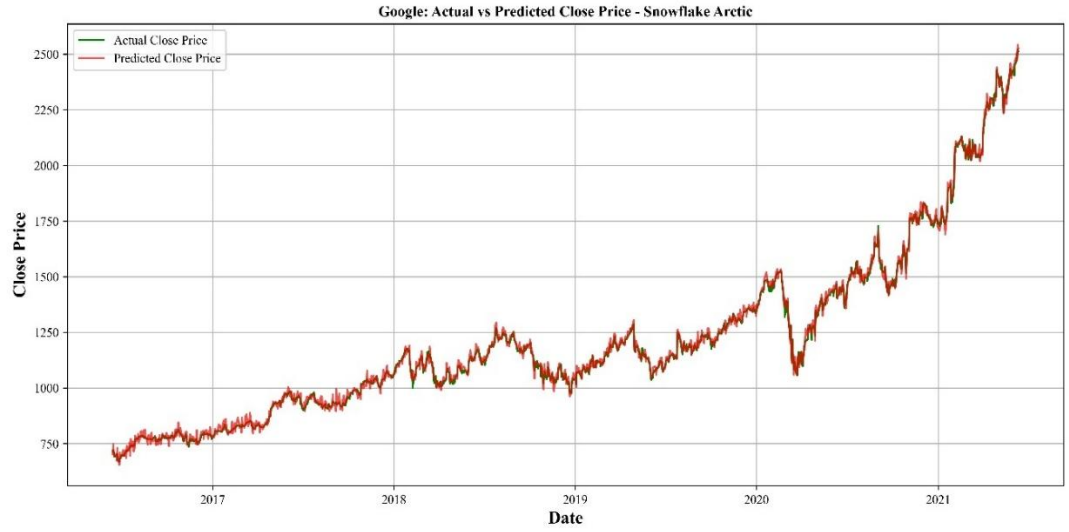
DeepSeek-V2

FinGPT

Figure 11. Actual and Predicted Price of models on Apple Dataset

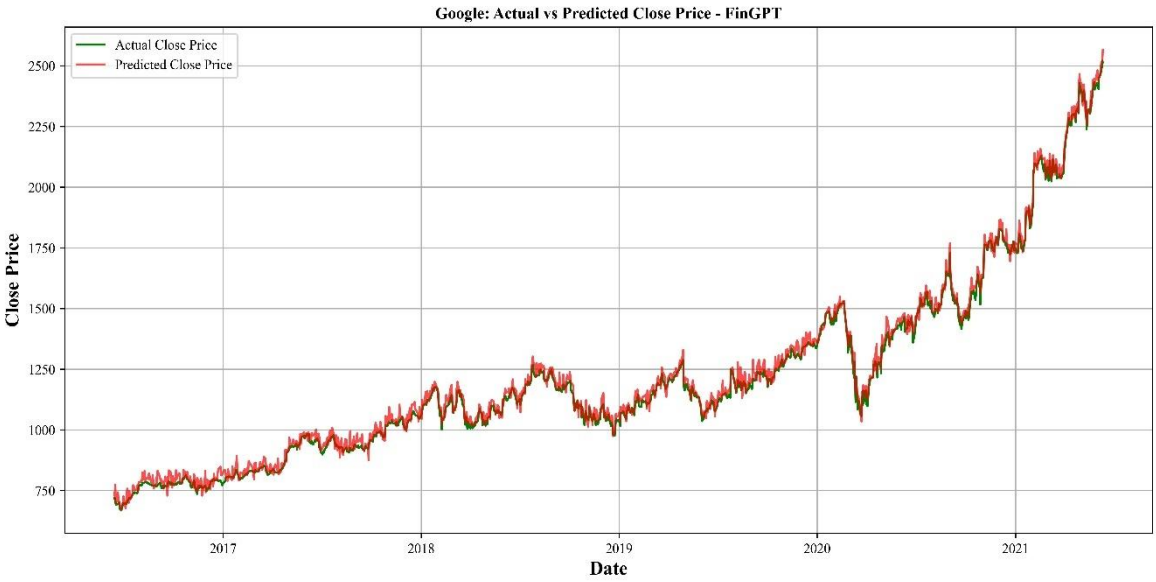
LlaMA-3

Falcon-2



Snowflake Arctic

DeepSeek-V2



FinGPT

Figure 12. Actual and Predicted Price of models on Google Dataset

Model ranking analysis indicated in Table 5 demonstrates that FinGPT consistently performed better compared to all other LLMs across all four metrics of assessment in combined dataset and shows that FinGPT was the most accurate and an effective. DeepSeek-V2 was close second in all aspects with performance. LLaMA-3 held steady third-place, Snowflake Arctic was on the fourth and Falcon-2 on fifth place. The consistency of ranking based on metrics indicates low trade-off between certain metrics between error minimization and predictive fit of the models under analysis.

Table 5. Model Ranking by Metric (Lower rank = better performance)

Model	MAE	RMSE	MAPE	R^2	Average Rank
FinGPT	1	1	1	1	1.00
DeepSeek-V2	2	2	2	2	2.00
LLaMA-3	3	3	3	3	3.00
Snowflake Arctic	4	4	4	4	4.00
Falcon-2	5	5	5	5	5.00

Statistical Analysis

The paired t-tests were performed on each of the evaluation measures on all data sets to check whether the identified divergences in models' performance were significant at statistical level. We found the performance gains on FinGPT over the remaining four LLMs to be statistically significant at the 95 percent confidence level ($p < 0.05$) across all metrics. Comparing the middle models of rank (LLaMA-3 and Snowflake Arctic) revealed that most of the metrics do not draw significant difference, which implies similarity in the level of predictive ability. Falcon-2 had statistically significant lower performance than FinGPT and DeepSeek-V2 at all datasets and metrics and always scored significantly inferior on all of them. These results corroborate that the increased performance of FinGPT is not because of random variation but because of accurate increase in predictive accuracy.

Performance of Computation Efficiency of Models

The study examined computational analysis of LLMs based on three metrics to increase predictive performance measurement. Training Time (hours) time spent optimizing on train slice of each dataset. Inference Time per Prediction (ms), average number of milliseconds to compute one next-day stock price prediction. CPU Usage (%) is the average percentage of CPU used during inference, across all datasets. The experiments were carried out under the same controlled environment with the same hardware specifications on them so as to make a fair comparison. Table 6 depicts the outcome of computation analysis of LLMs.

Table 6. Evaluation Results of Computational Efficiency

Model	Training Time (h)	Inference Time (ms/prediction)	CPU Usage (%)
FinGPT	4.9	9.7	37
DeepSeek-V2	5.7	10.5	39
Snowflake Arctic	6.2	11.8	41
LLaMA-3	6.8	12.4	42
Falcon-2	7.5	14.2	46

FinGPT ranked high in terms of computational performance with less training time of 4.9 hours and inference time per prediction with 9.7 ms and least CPU consumption of 37% as shown in Figures 13 - 15. DeepSeek-V2 was next in line showing an increase in all three metrics compared to the other LLMs. Snowflake Arctic and LLaMA-3 demonstrated moderate efficiency, and Falcon-2 showed the longest training and inference times, which were coupled with highest CPU usage, and the model was, therefore, the least efficient of those that have been evaluated. These findings indicate that domain-specific LLMs such as FinGPT can not only be more predictively accurate but also have significant computational advantages, a performance characteristic that matters in the case of real-time financial forecasting tasks, where low cost and time are highly desirable.

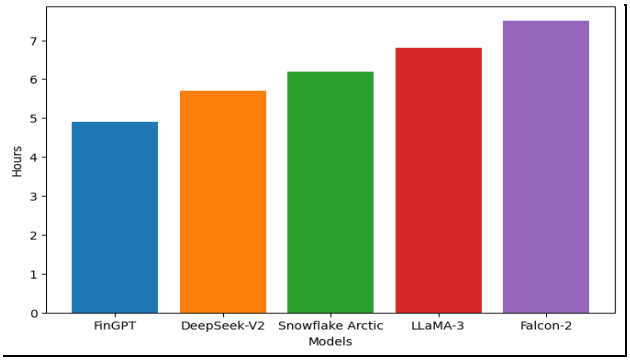


Figure 13. Analysis of Training Hours

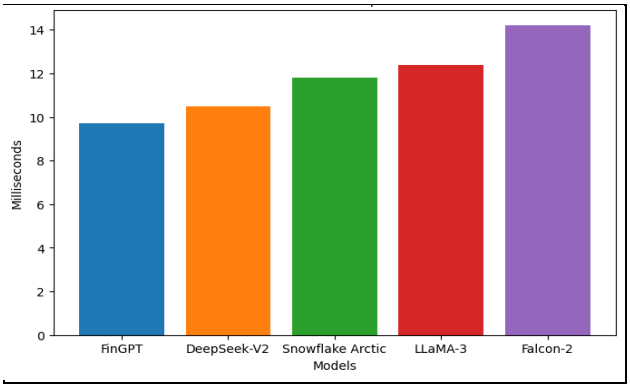


Figure 14. Analysis of Inference Time.

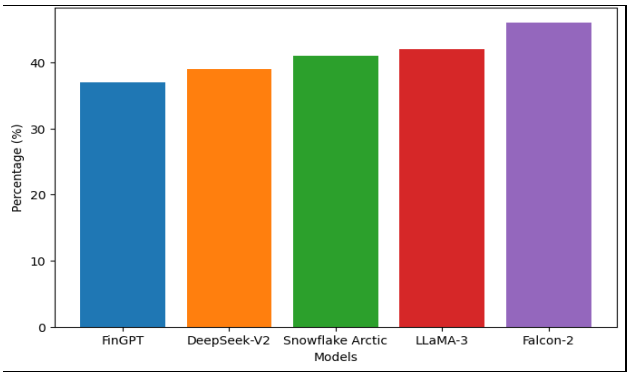


Figure 15. Analysis of CPU usage

5. CONCLUSION

This study provides comprehensive comparative analysis of five LLMs in the context of stock price forecasting for three diverse and historically significant equity datasets. The experimental results reveal that FinGPT, specifically fine-tuned on financial corpora, consistently outperformed other LLMs across various error metrics, while also achieving highest variance explanation across all datasets. DeepSeek-V2 demonstrated competitive performance, suggesting that architectures with large context windows and optimized temporal reasoning are well-suited for financial time series prediction. Despite uniformly high R2 values across all models, indicating substantial alignment with actual price movements, the marginal improvements in MAPE and MAE achieved by domain-specific models are practically significant for high-frequency trading and risk-sensitive applications. The Apple dataset's lower errors across all models suggest that relatively stable, long-term stock histories are more predictable for LLMs, while Reliance and Google datasets highlight the challenges posed by market volatility and structural breaks. Furthermore, the inclusion of statistical significance testing strengthens the reliability of comparative conclusions. Future work could extend this research by integrating multimodal inputs and exploring hybrid architectures that combine LLMs with traditional time-series models such as Transformer-based models, LSTM, and ARIMA. Such approaches could further enhance predictive robustness and adaptability in rapidly changing market settings.

REFERENCES

1. Julian, T., Devrison, T., Anora, V. and Suryaningrum, K.M., 2023. Stock price prediction model using deep learning optimization based on technical analysis indicators. Procedia Computer Science, 227, pp.939-947.
2. Chang, V., Xu, Q.A., Chidozie, A. and Wang, H., 2024. Predicting economic trends and stock market prices with deep learning and advanced machine learning techniques. Electronics, 13(17), p.3396.
3. Bharathi Mohan, G., Prasanna Kumar, R., Vishal Krishh, P., Keerthinathan, A., Lavanya, G., Meghana, M.K.U., Sulthana, S. and Doss, S., 2024. An analysis of large language models: their impact and potential applications. Knowledge and Information Systems, 66(9), pp.5047-5070.
4. Darwish, M, Hassanien, EE & Eissa, AHB 2025, "Stock market Forecasting: From traditional predictive models to large language models," Computational Economics, https://doi.org/10.1007/s10614-025-11024-w.
5. Zhao, C., Hu, P., Liu, X., Lan, X. and Zhang, H., 2023. Stock market analysis using time series relational models for stock price prediction. Mathematics, 11(5), p.1130.
6. Mu, G, Gao, N, Wang, Y & Dai, L 2023, "A stock price prediction model based on investor sentiment and optimized deep learning," IEEE Access, 1151353–51367, https://doi.org/10.1109/access.2023.3278790.
7. Duan, Y., Liu, Y., Wang, Y., Ren, S. and Wang, Y., 2023. Improved BIGRU model and its application in stock price forecasting. Electronics, 12(12), p.2718.
8. Rahmadeyan, A., 2024. Long short-term memory and gated recurrent unit for stock price prediction. Procedia Computer Science, 234, pp.204-212.
9. Wu, J.M.T., Li, Z., Herencsar, N., Vo, B. and Lin, J.C.W., 2023. A graph-based CNN-LSTM stock price prediction algorithm with leading indicators. Multimedia Systems, 29(3), pp.1751-1770.
10. Wang, S 2023, "A stock price prediction method based on BILSTM and improved transformer," IEEE Access, 11104211–104223, https://doi.org/10.1109/access.2023.3296308.
11. Billah, M.M., Sultana, A., Bhuiyan, F. and Kaosar, M.G., 2024. Stock price prediction: comparison of different moving average techniques using deep learning model. Neural Computing and Applications, 36(11), pp.5861-5871.
12. Gülmez, B., 2023. Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm. Expert Systems with Applications, 227, p.120346.

13. Chen, Q 2025, "Image-Driven Stock Price Prediction with LLaMA: A Prompt-Based Approach," International Journal of Modeling and Optimization, 17–24, https://doi.org/10.7763/ijmo.2025.v15.867.
14. Qi, C, Ren, J & Su, J 2023, "GRU Neural Network based on CEEMDAN–Wavelet for stock price prediction," Applied Sciences, 13(12):7104, https://doi.org/10.3390/app13127104.
15. Yan, J. and Huang, Y., 2025. MambaLLM: Integrating Macro-Index and Micro-Stock Data for Enhanced Stock Price Prediction. Mathematics, 13(10), p.1599.
16. Biswas, AK, Bhuiyan, MSA, Mir, MNH, Rahman, A, Mridha, MF, Islam, MR & Watanobe, Y 2025, "A Dual Output Temporal Convolutional Network with Attention Architecture for Stock Price Prediction and Risk Assessment," IEEE Access, 1, https://doi.org/10.1109/access.2025.3551307.
17. Gülmez, B 2025, "GA-Attention-Fuzzy-Stock-Net: An Optimized Neuro-Fuzzy System for Stock Market Price Prediction with Genetic Algorithm and Attention Mechanism," Heliyon, 11(3):e42393, https://doi.org/10.1016/j.heliyon.2025.e42393.
18. Amiri, B, Haddadi, A & Mojdehi, KF 2025, "A novel hybrid GCN-LSTM algorithm for energy stock price prediction: leveraging temporal dynamics and Inter-Stock relationships," IEEE Access, 1, https://doi.org/10.1109/access.2025.3536889.
19. Chen, Q., 2025, "Comparing Vision-Instruct LLMs, Vision-Based deep Learning, and numeric models for stock movement prediction," International Journal of Advanced Computer Science and Applications, 16(4):, https://doi.org/10.14569/ijacsa.2025.0160402.
20. Mukherjee, S., Sadhukhan, B., Sarkar, N., Roy, D. and De, S., 2023. Stock market prediction using deep learning algorithms. CAAI Transactions on Intelligence Technology, 8(1), pp.82-94.
21. Li, M, Zhu, Y, Shen, Y & Angelova, M 2022, "Clustering-enhanced stock price prediction using deep learning," World Wide Web, 26(1):207–232, https://doi.org/10.1007/s11280-021-01003-0.
22. Zhang, J., Ye, L. and Lai, Y., 2023. Stock price prediction using CNN-BiLSTM-Attention model. Mathematics, 11(9), p.1985.
23. Google Stock Prediction dataset: http://kaggle.com/datasets/shreenidhihipparagi/google-stock-prediction, Accessed on August 2025.
24. Reliance dataset: https://www.kaggle.com/datasets/kmldas/reliance-industries-ril-share-price-19962020, Accessed on August 2025.
25. Apple dataset: https://www.kaggle.com/datasets/olegshpagin/apple-stock-price-prediction-dataset, Accessed on August 2025.
26. Luo, R., Sastimoglu, Z., Faisal, A.I. and Deen, M.J., 2024. Evaluating the efficacy of large language models for systematic review and meta-analysis screening. medrxiv, pp.2024-06.
27. Djagba, P. and Odinakachukwu, C.A., 2025. Assessing the Capabilities and Limitations of FinGPT Model in Financial NLP Applications. arXiv preprint arXiv:2507.08015.
28. Djagba, P. and Odinakachukwu, C.A., 2025. Assessing the Capabilities and Limitations of FinGPT Model in Financial NLP Applications. arXiv preprint arXiv:2507.08015.
29. Liu, A., Feng, B., Wang, B., Wang, B., Liu, B., Zhao, C., Dengr, C., Ruan, C., Dai, D., Guo, D. and Yang, D., 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. arXiv preprint arXiv:2405.04434.
30. Malartic, Q., Chowdhury, N.R., Cojocaru, R., Farooq, M., Campesan, G., Djilali, Y.A.D., Narayan, S., Singh, A., Velikanov, M., Boussaha, B.E.A. and Al-Yafeai, M., 2024. Falcon2-11b technical report. arXiv preprint arXiv:2407.14885.
31. FinGPT: Open-Source LLM for Finance, GitHub repository, 2024. [Online]. Available: https://github.com/AI4Finance-Foundation/FinGPT/blob/master/figs/FinGPT_framework_2024a0301.png.