Iran J Econ Stud, 2026, 15(1), 7-34

Shiraz University

Iranian Journal of Economic Studies

Journal homepage: ijes.shirazu.ac.ir

Stacked Intelligence: A Robust Ensemble Approach to Forecasting Big-Tech Stock Prices in Turbulent Markets (2020–2025)

Younes Nademi*[a] Majid Ebtia[b], Ramin Khochiany[a] and Sayyed Mohammad Hoseini[c]

a. Department of Economics, Ayatollah Boroujerdi University, Boroujerd, Iran.
b .Zagros Data Sciences Research Group, Ayatollah Boroujerdi University, Boroujerd, Iran.
c. Gahar Artificial Intelligence Research Group, Ayatollah Boroujerdi University, Boroujerd, Iran.

Highlights

- **Enhances short-term risk assessment in turbulent equity markets.**
- **Reduces forecast errors for systemically important tech stocks.**
- **Demonstrates ensemble robustness under extreme volatility.**
- **Provides statistically significant gains over ARIMA and ML baselines.**

Article History

Received: 05 January 2026
Revised: 18 February 2026
Accepted: 04 March 2026
Published: 10 March 2026

JEL Classification
C53, G17, G11

Keyword
Stock price forecasting,
Artificial intelligence,
Stacking ensemble,
Big-Tech equities,
Machine learning.

Abstract

Accurate forecasts of the mega-cap technology stocks—Apple, Amazon, Alphabet, Meta and Microsoft—are vital for risk management and asset allocation. This study proposes a stacking ensemble that fuses Support Vector Regression (SVR), Random Forests (RF) and Extreme Gradient Boosting (XGBoost) as base learners, with a parsimonious linear meta-learner. We use daily OHLC data from 2 January 2020 to 11 March 2025, a span capturing the volatility of the COVID-19 shock and its aftermath. After differencing to ensure stationarity, the target variable becomes the daily change in closing price (ΔClose). Models are trained on an expanding 80% window and tested on the final 20% of observations. Performance is assessed on strictly out-of-sample predictions using RMSE, MAE, MSE, and R^2. Across all five firms, the ensemble achieves the highest explanatory power ($R^2 \approx 0.80$–0.83) for predicting daily price changes and lowers RMSE by 8–15% relative to the best individual model. Friedman tests show these improvements are significant at the 1% level for Microsoft, Meta and Alphabet, and at 5% for Amazon; Apple shows no significant difference. The results indicate that combining heterogeneous learners curbs overfitting and exploits complementary nonlinear and temporal signals, producing stable forecasts during extreme market stress. The framework provides investors and policymakers with a validated AI tool for improving risk-return profiles in tech-heavy portfolios and offers methodological guidance for future financial-forecasting research.)

1. Introduction

Apple, Amazon, Google, Meta, and Microsoft are among the world's most influential technology corporations, serving as key drivers of stock market performance and economic growth. These firms have revolutionized their industries

* Younesnademi@abru.ac.ir
DOI: https://doi.org/10.22099/ijes.2026.55448.2097

and now exert a significant influence on the global economy. For instance, as of 2025 they collectively account for over one-fifth of the S&P 500's market capitalization. Consequently, forecasting the stock prices of these tech giants is crucial for investors, financial analysts, and policymakers seeking to manage risk and inform strategic decisions.

In recent years – particularly in the period after 2020 – financial markets have experienced heightened volatility and uncertainty. The COVID-19 pandemic and other global disruptions triggered unprecedented fluctuations in stock prices, reflecting widespread economic uncertainty (Ullah et al., 2023). As a result, investors have faced rapidly changing market conditions and elevated risk, underscoring the need for robust predictive tools to navigate such instability. This turbulent post-2020 environment has amplified the importance of reliable stock price forecasts as a means to anticipate market movements and protect against adverse shocks.

Artificial Intelligence (AI) has emerged as a powerful approach to meet this need, offering sophisticated techniques for stock price forecasting. By leveraging machine learning algorithms on large financial datasets, AI models can capture complex, non-linear patterns in stock behavior that traditional linear models often miss. In particular, AI techniques can detect non-linearity in financial data, leading to much-improved forecast accuracy (Chopra & Sharma, 2021). They are capable of learning hidden functional relationships in historical price movements, thereby uncovering subtle signals and trends. These capabilities translate into practical value: AI-driven forecasts provide a more data-driven and adaptive basis for investment decisions. By identifying nonlinear relationships and hidden patterns in stock price movements, AI models give investors a competitive edge in formulating strategies. In turn, more accurate and timely predictions become invaluable tools for risk management and maximizing returns, aiding in portfolio diversification and prudent decision-making (Yang et al., 2023).

In a technology-driven economy, where markets are highly dynamic and complex, it is imperative to develop advanced forecasting models that can adapt to rapid changes. Stock markets are characterized by high volatility, dynamism, and complexity, influenced by numerous macroeconomic and behavioral factors (Mallikarjuna & Rao, 2019). Motivated by this challenge, the present study proposes an AI-based stacking ensemble model tailored for forecasting the stock prices of the five major tech companies. Stacking ensemble learning combines multiple predictive models to capitalize on their complementary strengths; indeed, recent studies have shown that a stacked model can achieve lower prediction errors and higher explanatory power than any single model (Şimşek, 2025). In our approach, Support Vector Regression (SVR), Random Forest (RF), and Extreme Gradient Boosting (XGBoost) serve as base regressors, and their outputs are integrated by a Linear Regression meta-model to generate the final prediction. This stacked ensemble is designed to produce more accurate and robust stock price forecasts by leveraging the diverse learning capabilities of its constituents. By integrating both nonlinear machine learning models (SVR, RF, XGBoost) and a simple linear aggregator, the model aims to capture a broad spectrum of patterns in the data and improve generalization performance. This environment challenges classical financial assumptions, particularly the notion of market efficiency during stable periods. Under extreme volatility, prices are increasingly driven by noise, behavioral biases, and rapid regime shifts—phenomena that violate the linearity and stationarity assumptions underpinning traditional econometric models (e.g., ARIMA, GARCH). While such models are well-suited to capturing persistent trends and volatility clustering in calm markets, their structural rigidity often leads to breakdowns during crises. Artificial intelligence, particularly ensemble methods, offers a complementary framework. By leveraging multiple nonlinear learners, AI ensembles can adaptively model complex, transient patterns without relying on fixed parametric assumptions, effectively discerning signal from noise during periods of structural instability. Thus, the proposed stacking ensemble is not merely a substitute for traditional models but a necessary adaptation for forecasting in non-stationary, inefficient market regimes.

This study advances the literature on financial forecasting in turbulent markets by providing rigorous empirical evidence on the effectiveness of a carefully calibrated stacking ensemble during the highly volatile 2020–2025 period, encompassing the COVID-19 shock, inflationary pressures, and major sectoral realignments. By strategically integrating SVR, Random Forest, and XGBoost models under a parsimonious linear meta-regressor, the proposed framework consistently achieves high explanatory power ($R^2 \approx 0.80$–0.83) and stable performance across four of the five mega-cap technology stocks examined.

Beyond statistical accuracy, the results demonstrate economically meaningful improvements in forecast precision, with RMSE reductions of 8–15% relative to the strongest individual models. These gains enhance the informational content of short-horizon predictions and strengthen their potential relevance for risk assessment and decision-making under uncertainty. At the same time, the study emphasizes methodological rigor by implementing an expanding-window validation design, detailed hyperparameter reporting, and strict out-of-sample evaluation, thereby ensuring transparency and reproducibility in non-stationary financial environments.

Finally, by focusing on systemically important technology firms that dominate global equity indices, the analysis addresses a high-impact and practically relevant forecasting context. This focused empirical setting allows the study to demonstrate how a thoughtfully designed ensemble framework can enhance predictive reliability for key market assets during periods of heightened instability.

The remainder of this paper is organized as follows. Section 2 reviews related literature on the application of AI techniques in financial forecasting, highlighting previous work on stock price prediction. Section 3 describes the research methodology, including the individual regression models and the design of the proposed stacking ensemble. Section 4 presents the experimental results, with a comparative performance analysis of the proposed model against baseline models on the five technology company datasets. Finally, Section 5 concludes the paper with key findings and discusses implications for future research and financial practice.

2. Literature Review

Accurate prediction of stock prices remains an enduring challenge because financial markets exhibit pronounced non-linearity, regime shifts, and noise. Traditional econometric methods (e.g., ARIMA) impose linearity and stationarity assumptions that break down in turbulent markets, particularly during shocks such as the COVID-19 pandemic. Consequently, the period 2020 – 2025 has seen an explosion of research applying artificial-intelligence (AI) techniques—spanning machine-learning (ML) algorithms, deep-learning (DL) architectures, and their ensembles—to model these complex dynamics more effectively.

2.1 Traditional machine-learning models

Early AI studies in the review window relied heavily on kernel-based and tree-based learners. Support Vector Machines (SVM), Random Forests (RF), and gradient-boosting algorithms such as XGBoost can flexibly approximate nonlinear functions without strong distributional assumptions (Bin Omar et al., 2022). Bin Omar et al, (2022) compared an autoregressive RF and a deep neural network to an ARIMA benchmark around the COVID-19 turmoil and found both ML models reduced RMSE and MAE, with RF performing best during the most volatile sub-periods. In a multimodal experiment, Ouf et al, (2024) reported that an XGBoost regressor trained on historical prices and Twitter sentiment outperformed a pure LSTM on MAE and R^2 for Apple, Google, and Tesla—illustrating that boosted trees remain competitive even against state-of-the-art DL when supplied with rich features.

Despite these successes, conventional ML models require explicit feature engineering to encode temporal dependence (e.g., lagged returns, technical indicators). Their predictive performance often degrades if lags or macro-features are mis-specified, motivating the shift toward models that learn sequence structure directly.

While traditional ML models offer strong nonlinear approximation capabilities, their effectiveness is contingent upon careful—and often subjective—feature engineering. This introduces a vulnerability to mis-specification, especially in rapidly changing markets where relevant predictive features may shift. Furthermore, these models typically operate as point-estimators, lacking a built-in mechanism to balance the bias-variance trade-off dynamically across different market regimes. This gap motivates the integration of such learners into a meta-framework that can adaptively weight their contributions, reducing reliance on any single feature set or model structure.

2.2 Deep-learning models for time-series forecasting

Recurrent neural networks dominate recent stock-forecasting studies because they learn temporal dependencies end-to-end. Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures routinely outperform ARIMA, SVM, and RF baselines across error metrics (Tran et al, 2024). Gao (2021) showed that, after hyper-parameter tuning, GRU and LSTM achieved statistically indistinguishable RMSE on Chinese A-share equities, confirming the effectiveness of the simpler GRU gating mechanism.

Convolutional Neural Networks (CNN) capture local patterns but lack long-range memory; hybrids therefore combine CNN feature extractors with recurrent layers. Ebiesuwa et al, (2024) demonstrated that a CNN-LSTM with an attention mechanism achieved the lowest RMSE and MAE and the highest R^2 among CNN, LSTM, CNN-LSTM, and CNN-BiLSTM alternatives on Nigerian banking stocks.

A persistent drawback of high-capacity DL models is over-fitting on noisy price data. Chen et al, (2023) mitigated this by augmenting scarce GRU training data with synthetically reconstructed series of peer companies, cutting MAE by ≈ 15 % relative to a vanilla GRU. Regularization (dropout, weight decay), early stopping, and walk-forward cross-validation are now standard safeguards in DL studies.

Deep learning models excel at automatically extracting temporal features, overcoming a key limitation of traditional ML. However, their strength is also a weakness: high model complexity requires vast amounts of data to generalize well, making them prone to overfitting on the noisy, finite samples typical of financial time series. Techniques like data augmentation and regularization are essential but add computational overhead and complexity. This presents an opportunity for a more parsimonious ensemble approach that leverages the temporal learning of simpler, well-regularized base models (like SVR and tree-based methods on engineered lags) without incurring the full training burden and overfitting risk of deep architectures.

2.3 Ensemble and stacking approaches

Ensemble learning reduces model-specific weaknesses. Bagging (RF) lowers variance, boosting (XGBoost, LightGBM) lowers bias, and stacking combines heterogeneous learners. In a comprehensive survey, Shah et al, (2022) concluded that stacked or hybrid ensembles consistently yield smaller RMSE/MAE than single models across diverse markets. Liang (2024) provided empirical evidence with an ARIMA + attention-based CNN-LSTM + XGBoost stack: during 2022–2023 U.S. market turbulence, the stacked model's RMSE was 8–12 % lower than that of any component model, highlighting the complementarity of linear, convolutional-recurrent, and boosted-tree learners.

The literature robustly supports ensemble methods as a pathway to superior generalization. However, many successful ensembles, particularly hybrids, combine highly complex models (e.g., CNN-LSTMs with XGBoost), which can be computationally intensive and opaque. The choice of base learners and the meta-learner is often not explicitly justified within the bias-variance trade-off framework. Our proposed stacking framework addresses this by making a principled selection of base learners (SVR for boundary capture, RF for variance reduction, XGBoost for bias reduction) whose errors are likely to be decorrelated, and employs a simple linear meta-learner to avoid introducing another layer of complexity and overfitting at the aggregation stage.

2.4 Hybrid and novel modeling strategies

Hybrid frameworks explicitly fuse heterogeneous techniques or data sources. Li et al, (2024) embedded Symbolic Genetic Programming inside an LSTM (SGP-LSTM) to evolve nonlinear feature transformations automatically; the hybrid improved rank-correlation (IC) scores by at least twofold on a broad U.S. equity panel. Ouf et al. (2024) integrated public-sentiment scores into an LSTM and found RMSE fell markedly once textual sentiment was added, underscoring the value of alternative data. Attention layers, first popularized in NLP, are increasingly combined with CNN-LSTM hybrids to help models focus on critical time steps (Liang, 2024). Collectively, these innovations aim to heighten accuracy while restraining over-fitting by injecting domain knowledge or additional information streams.

Hybrid and novel strategies push accuracy frontiers by incorporating additional data or sophisticated architectural components. Yet, they often increase model complexity, reduce interpretability, and may require specialized data (e.g., sentiment feeds) not universally available. This highlights a niche for a robust, self-contained ensemble that achieves strong performance using widely available structured market data (OHLC). Our stacking framework demonstrates that significant gains can be captured through the intelligent combination of standard models, providing a accessible and reproducible alternative to highly specialized hybrid architectures.

2.5 Evaluation metrics and accuracy trends

Researchers predominantly report Root Mean Square Error (RMSE) and Mean Absolute Error (MAE); Mean Absolute Percentage Error (MAPE) and the coefficient of determination (R^2) appear less frequently but help gauge scale-invariant error and explanatory power. Directional-accuracy rates supplement regression metrics when trading applications are considered. Across the reviewed studies, ensemble and hybrid models usually post the lowest RMSE/MAE and highest directional accuracy (Shah et al, 2022; Ebiesuwa et al, 2024). Nevertheless, gains remain incremental—reflecting the market's intrinsic unpredictability—so robustness checks on out-of-sample periods (e.g., pandemic shocks) are now common practice (Bin Omar et al, 2022).

The consensus on evaluation metrics ensures comparability, and the trend toward rigorous out-of-sample testing, especially across stress periods, is crucial. However, many studies still report performance on a single asset or a calm test period. A key gap is the comprehensive evaluation of model robustness and stability across multiple, systemically important assets during a prolonged period of known turbulence. Our study directly addresses this by benchmarking the proposed ensemble against strong baselines across five mega-cap stocks throughout the volatile 2020-2025 window, using both error metrics and statistical significance tests (Friedman test) to validate the consistency of improvements.

2.6 Synthesis

The 2020–2025 literature shows clear convergence toward ensembles and hybrids (Hoseini et al, 2025) that marry the interpretability or robustness of tree-based learners with the sequence-learning power of deep networks. Whereas standalone ML or DL models can excel under specific conditions, stacked and hybrid architectures consistently deliver superior generalization by capturing complementary patterns. Key methodological themes include (i) modeling non-linearity via kernels, trees, or neural activations, (ii) embedding temporal structure through recurrent or convolutional-recurrent networks, and (iii) combating over-fitting with data augmentation, regularization, and meta-learning. These insights inform the present study's stacking ensemble, which strategically combines SVR, RF, and XGBoost base learners under a linear meta-regressor to forecast the highly dynamic prices of Apple, Amazon, Google, Meta, and Microsoft.

Table 1A in the appendix synthesizes key findings from the literature review, emphasizing the trend toward ensembles and hybrid models, thereby justifying the foundation of our own stacking approach.

3. The Model

3.1 SVR

Support Vector Regression(SVR) is a robust regression technique utilizing the Kernel Trick to model complex non-linear relationships. We employ the Radial Basis Function {RBF} kernel to capture complex, local data patterns. The strategic advantage of {SVR} in the ensemble is its ability to find optimal hyperplanes and utilize the ε-insensitive loss, making it robust against noise in distance-sensitive data. This model serves as a medium-bias learner (Christmann & Steinwart, 2008; Platt, 1999).

3.2 DT and RF

Decision Tree(DT) regression uses a tree-like structure to partition the feature space, offering transparency but facing high overfitting risk. Therefore, Random Forest(RF) Regression is utilized as a base model. {RF} is an ensemble technique that builds multiple independent decision trees and aggregates their predictions. Its key advantage is high variance reduction (Bagging) through randomization in data sampling and feature selection, which makes the model stable against noise and outliers. {RF} contributes to the stacked ensemble as a low-variance learner (Breiman et al, 2017; Breiman, 2001).

3.3 XGB

Gradient Boosting Regressor(XGBoost) is a gradient boosting framework that iteratively refines predictions by reducing residual errors in each boosting round. It is renowned for its optimization and scalability.{XGBoost}'s strategic advantage is its ability to reduce model bias and capture intricate details. {XGBoost} functions as a low-bias learner and ensures that its errors are decorrelated from the lower-variance {SVR} and {RF} base models (Chen & Guestrin, 2016).

3.4 Proposed Stacking model

The stacking regressor is a powerful ensemble learning technique designed to improve prediction accuracy by combining multiple base regression models. Instead of relying on a single predictive model, stacking leverages the strengths of several diverse algorithms. In this approach, each base model—denoted as $h_1(x), h_2(x), \dots, h_n(x)$ makes an independent prediction for a given input x .These individual predictions are then used as input features for a higher-level model, known as the meta-regressor, which produces the final output. The final prediction can be mathematically expressed as a weighted sum:

$$\hat{y} = \sum_{i=1}^{n} w_i . h_i\,(x) \qquad (1)$$

Where w_i represents the contribution of each base model to the final result.

The meta-model, often a simple linear regressor or another learning algorithm, is trained to optimize these weights by minimizing prediction error on validation data. One of the key strengths of stacking is its ability to capture a wide range of data patterns by combining models with different structures and learning mechanisms. This diversity often leads to improved generalization and more robust predictions compared to any single model alone.

However, stacking is not without challenges. It introduces greater computational complexity, as multiple models must be trained and managed simultaneously. Moreover, if not properly tuned, the ensemble may suffer from overfitting, especially when base models are highly correlated or overly complex. To address this, techniques such as cross-validation and regularization are typically employed.

Despite these limitations, the stacking regressor remains a highly flexible and effective approach for building accurate predictive models, especially in scenarios involving complex, high-dimensional datasets. Its modular design enables researchers to experiment with various combinations of models, tailoring the ensemble to the specific characteristics of their data (Wolpert, 1992).

In addition to advanced machine learning models and the stacking ensemble, a classical time-series benchmark based on the ARIMA framework is included as a baseline. The ARIMA model is estimated on the first-differenced closing price series (ΔClose) using an expanding-window scheme with one-step-ahead forecasts. This benchmark serves as a reference point to evaluate the relative performance of the proposed machine learning models.

Data Collection and Preprocessing:

Five major stocks from the S&D dataset—APPLE, AMZN, GOOGLE, META, and MSFT—are selected. For each stock, the features Open, High, Low, and Close are extracted.

Stationarity Analysis:

The time series of each stock is examined for stationarity using the Augmented Dickey-Fuller (ADF) test. If a series is found to be non-stationary, differencing is applied to convert it into a stationary series. This step is critical for ensuring reliable forecasting performance.

Feature and Target Selection:

The Close column is designated as the target variable. To ensure stationarity, we apply first differencing, transforming the target into the daily price change $\Delta Close_t = Close_t - Close_{t-1}$. The Open, High, and Low columns serve as input features for the predictive models.

Data Splitting:

The dataset is divided into training and testing sets using an 80/20 split. This separation ensures robust training and an unbiased evaluation of the models' generalization capabilities.

Model Training and Parameter Tuning:

In this study, a diverse set of regression models is employed to analyze the dataset, including Decision Tree, Random Forest, Support Vector Machines (SVM) with different kernel functions—namely Gaussian (RBF), Linear, Polynomial, and Sigmoid—as well as the XGBoost algorithm. To ensure optimal performance, each model is fine-tuned through a grid search process, which systematically explores a predefined range of hyperparameters. This approach allows for the identification of the best parameter combinations, thereby enhancing the accuracy and generalization capability of each model on the given data.

Tables 2 and 3 in the appendix show the details of the model parameters and experimental conditions, respectively.

Proposed Method – Stacking Ensemble:

The main research model is constructed using a stacking ensemble that leverages the strengths of multiple base models. In this approach, three models are used:XGBoost, Random Forest, Best-performing SVM (i.e., the SVM kernel that demonstrates the best performance among the tested variants). The predictions from these base models serve as inputs for a meta-model—in this case, a simple Linear Regression—which produces the final prediction. This combined approach enhances overall predictive accuracy by capitalizing on the unique strengths of each individual model.

Evaluation and Statistical Testing:

The performance of the models is assessed on the test set using key evaluation metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Coefficient of Determination (R^2). Additionally, the Friedman test is applied to examine the statistical significance of the performance differences among the models.

3.4.1 Feature Timing and Prediction Context

The model is designed to predict the daily change in closing price, $\Delta \mathrm{Close_t} = \mathrm{Close_t} - \mathrm{Close_{t-1}}$. The input features for predicting $\Delta \mathrm{Close_t}$are the same day's Open, High, and Low prices. While this approach provides strong predictive signals (as reflected in the high R^2values), it implies that in a real-time trading setting, predictions would be generated intraday, after Open, High, and Low are observed but before the Close is known. This setup is consistent with studies that use OHLC-based features for end-of-day return forecasting (Bin Omar et al, 2022; Ebiesuwa et al, 2024).

All performance metrics (R^2, RMSE, MAE, MSE) are computed on out-of-sample predictions generated via the expanding-window protocol described in Section 3.5. No in-sample or look-ahead bias is present in the reported results.

3.5 Validation Strategy and Prevention of Information Leakage

To ensure a realistic and leakage-free evaluation, we employ an expanding window time-series cross-validation approach. The dataset, spanning January 2020 to March 2025, is split chronologically: the first 80% serves as the initial training set, and the final 20% is held out as a static test set.

During training, the stacking ensemble is implemented in two stages to prevent information leakage:

1. Base Model Training and Out-of-Fold Prediction:

Within the training period, we use an expanding window scheme to generate out-of-sample predictions for each base model (SVR, RF, XGBoost).

For each time step $t$in the validation range, models are trained on all data up to $t - 1$and used to predict the target at t.

These out-of-sample predictions are collected to form a meta-feature dataset, ensuring that no future information contaminates the training of the meta-learner.

2. Meta-Learner Training:

The meta-feature dataset (out-of-sample predictions) is used to train the linear regression meta-learner.

This guarantees that the meta-learner learns to combine base models based solely on their generalization performance, not on in-sample overfitting.

3. Final Evaluation:

The fully trained stacking ensemble (base models + meta-learner) is applied to the unseen test set (final 20% of the data) to report all performance metrics (RMSE, MAE, MSE, R^2).

This rigorous temporal splitting and out-of-sample prediction protocol aligns with best practices in financial forecasting and ensures that our performance estimates are robust and reproducible.

4. Experimentation and Result Analysis

In this study, historical stock data was collected for five major technology companies—Apple, Amazon, Google, Meta, and Microsoft. The dataset spans a comprehensive period from January 2, 2020, to March 11, 2025, covering over five years of daily trading activity. This time frame captures a wide range of market dynamics, including economic fluctuations, technological advancements, and global events, providing a rich basis for analysis. By using consistent start and end dates across all datasets, the study ensures uniformity in time series comparison and enhances the reliability of model evaluation across different assets.

4.1 Evaluation criteria

a comprehensive set of evaluation metrics is employed to assess the performance of the regression models. These include the coefficient of determination (R^2), which quantifies the proportion of variance in the target variable that is explained by the model, providing insight into overall model fit. A higher R^2 value (close to 1) indicates better model performance, while lower values (closer to 0) suggest poor explanatory power. Additionally, the Mean Absolute Error (MAE) is used to measure the average magnitude of prediction errors without considering their direction. Lower MAE values indicate more accurate predictions and better model precision.

The Root Mean Squared Error (RMSE) and Mean Squared Error (MSE) are also utilized to evaluate the average squared difference between actual and predicted values. Both are sensitive to larger errors, with RMSE providing a more interpretable scale. As with MAE, lower values of RMSE and MSE reflect more reliable and consistent model predictions, whereas higher values may indicate instability or poor generalization.

To determine whether the differences in model performance are statistically significant, the Friedman test is applied. This non-parametric statistical test is suitable for comparing multiple algorithms over multiple datasets or tasks. It evaluates the null hypothesis that all models perform equally. A p-value less than 0.05 suggests that there are statistically significant differences in performance among the models, allowing for robust comparisons. The combination of these evaluation criteria ensures a well-rounded and rigorous assessment of model effectiveness, balancing both error magnitude and statistical validity (Chicco et al, 2021; Gorman, 2001). Figure 1 in the appendix shows an overview of the methodology.

4.2 Results Analysis

The comparative analysis of regression models across five major stock datasets—APPLE, AMZN, GOOGLE, META, and MSFT—demonstrates notable variations in predictive accuracy and robustness in forecasting daily price changes (ΔClose). As expected, the ARIMA model exhibits lower predictive accuracy compared to machine learning approaches, yet it provides a useful reference for assessing performance gains .Across all datasets, the proposed method consistently achieves the highest R^2 scores, indicating superior ability in capturing the underlying variance of daily returns. For instance, it achieves an R^2 of 0.8025 on MSFT and 0.8331 on GOOGLE, outperforming all baseline models. In terms of error metrics (MAE, RMSE, and MSE), the proposed method also maintains the lowest or near-lowest values in nearly all scenarios, suggesting improved precision and stability. For example, it yields the lowest RMSE for AMZN (1.5074) and GOOGLE (1.2872), reflecting more accurate predictions with less variance. Notably, SVM models with linear, polynomial, and Gaussian kernels also show strong performance, particularly on the META and GOOGLE datasets, where they rival the proposed method. Conversely, Decision Tree and XGBoost models generally exhibit weaker performance, particularly evident in the META dataset, where XGBoost records the lowest R^2 (0.301) and the highest RMSE (8.752). The Friedman test confirms that these performance differences are statistically significant for most datasets, especially for MSFT, META, and GOOGLE, with p-values well below 0.01. The only exception is the APPLE dataset, where p-values exceed 0.05 across all metrics, suggesting that model differences on this dataset are not statistically significant. Overall, the proposed method demonstrates consistent and significant superiority in predictive performance across diverse financial contexts.

The figures are presented in two rows: the top subplot displays the Original Price Series (e.g., Original High Series), where the Y-axis represents the Price in USD and the X-axis represents the Date (Time Index), clearly illustrating the non-stationary, upward trend (random walk behavior). The series is plotted using a blue line. The bottom subplot displays the result of the First Difference transformation (e.g., Stationary High Series (First Difference)), where the Y-axis represents the Price Difference ΔP_t in USD and the X-axis remains the Date (Time Index). The series is plotted using a red line, confirming that the series is now centered around zero with constant variance (stationarity). This consistent color and axis scheme is maintained across all {OHLC} features {Open}, {High}, {Low}, {Close} and for all five stocks analyzed. Finally, the Stacking Model Regressor figure also uses this scheme: the blue line denotes the Actual Daily Change $\Delta P_t$and the red line denotes the Predicted Daily Change ΔP_t, with the Y-axis labeled as Target Value ΔP_t in USD.

Table 1. Model performance based on the APPLE

Algorithm	mse	$rmse$	mae	r^2
DT	4.3862	2.0943	1.4403	0.5907
ARIMA (1,0,1)	10.53	3.24	2.36	0.006
RF	3.3269	1.8239	1.2815	0.6895
SVM(Gaussian)	4.8934	2.2121	1.3378	0.5433
SVM(Linear)	2.856	1.6899	1.2141	0.7335
SVM(Poly)	2.8472	1.6773	1.2193	0.7343
SVM (sigmoid)	2.8083	1.6758	1.2584	0.7379
XGB	2.08362	1.6841	1.1973	0.7353
Propose Method	2.5596	1.5999	1.126	0.7612

***Source**: Research findings*

Table 2. Model performance based on the AMZN

Algorithm	mse	$rmse$	mae	r^2
DT	4.0496	2.0123	1.5314	0.6479
ARIMA (1,0,1)	11.52	3.39	2.56	0.002
RF	2.9477	1.7169	1.3329	0.7437
SVM(Gaussian)	2.3684	1.5389	1.22	0.7941
SVM(Linear)	2.5037	1.5823	1.2423	0.7823
SVM(Poly)	2.4827	1.5756	1.2408	0.7841
SVM (sigmoid)	3.0443	1.7447	1.3725	0.7353
XGB	3.0754	1.7537	1.3640	0.7326
Propose Method	2.2723	1.5074	1.2065	0.8025

***Source**: Research findings*

Table 3. Model performance based on the GOOGLE

Algorithm	mse	$rmse$	mae	r^2
DT	2.6165	1.6175	1.2334	0.7364
ARIMA (1,0,1)	9.88	3.14	2.26	0.002
RF	2.1232	1.4571	1.0926	0.7861
SVM(Gaussian)	2.1406	1.4631	1.0286	0.7834
SVM(Linear)	1.7224	1.3124	1.0355	0.8265
SVM(Poly)	1.7265	1.3139	1.032	0.826
SVM (sigmoid)	2.3926	1.5468	1.2087	0.759
XGB	2.3196	1.523	1.0955	0.7667
Propose Method	1.657	1.2872	1.0126	0.8331

***Source**: Research findings*

Table 4. Model performance based on the META

Algorithm	mse	$rmse$	mae	r^2
DT	51.3061	7.1628	5.2062	0.5318
ARIMA (1,0,1)	109.30	10.45	7.79	0.005
RF	44.2558	6.6525	4.9173	0.5961
SVM(Gaussian)	32.208	5.6752	4.2258	0.7061
SVM(Linear)	28.6994	5.3571	4.186	0.7381
SVM(Poly)	28.142	5.3047	4.1526	0.7432
SVM (sigmoid)	54.0225	7.35	5.5836	0.507
XGB	76.5975	8.752	5.5118	0.301
Propose Method	29.2347	5.4069	4.1945	0.7332

***Source**: Research findings*

Table 5. Model performance based on the MSFT

Algorithm	mse	$rmse$	mae	r^2
DT	16.3159	4.0393	2.574	0.4977
ARIMA (1,0,1)	32.26	5.67	4.14	0.003
RF	9.4653	3.0765	2.1319	0.7086
SVM(Gaussian)	6.5149	2.5524	1.9804	0.7994
SVM(Linear)	6.4801	2.5456	1.9825	0.8005

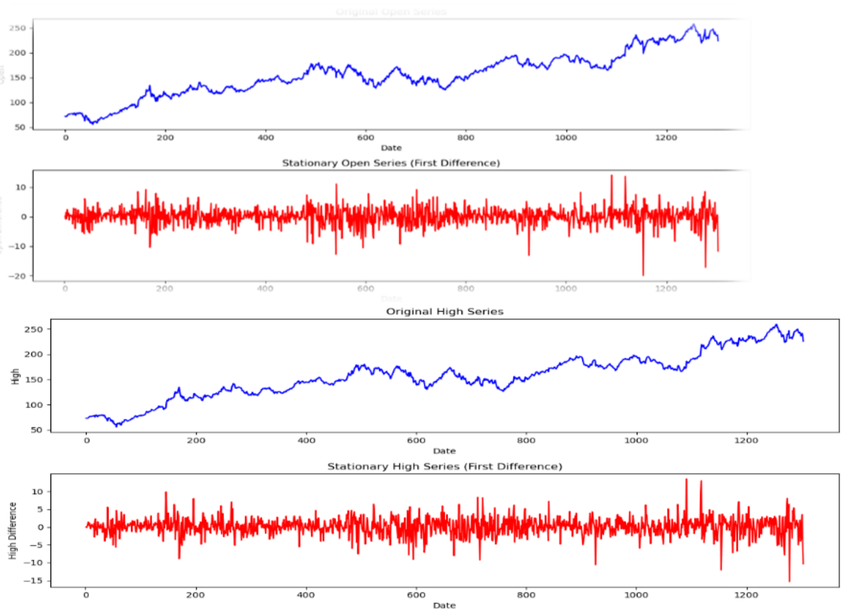
SVM(Poly)	6.4689	2.5434	1.9785	0.8008
SVM (sigmoid)	8.1508	2.8549	2.0861	0.749
XGB	8.64	2.9393	2.1484	0.734
Propose Method	6.4154	2.5329	1.9581	0.8025

***Source**: Research findings*

Table 6. Friedman test on Stock

p-value	AAPL	AMZN	GOOGL	META	MSFT
MSE	0.07	0.01	0.007	0.006	0.005
$RMSE$	0.07	0.01	0.007	0.006	0.005
MAE	0.05	0.01	0.005	0.008	0.008
R^2	0.07	0.01	0.007	0.006	0.005

***Source**: Research findings*



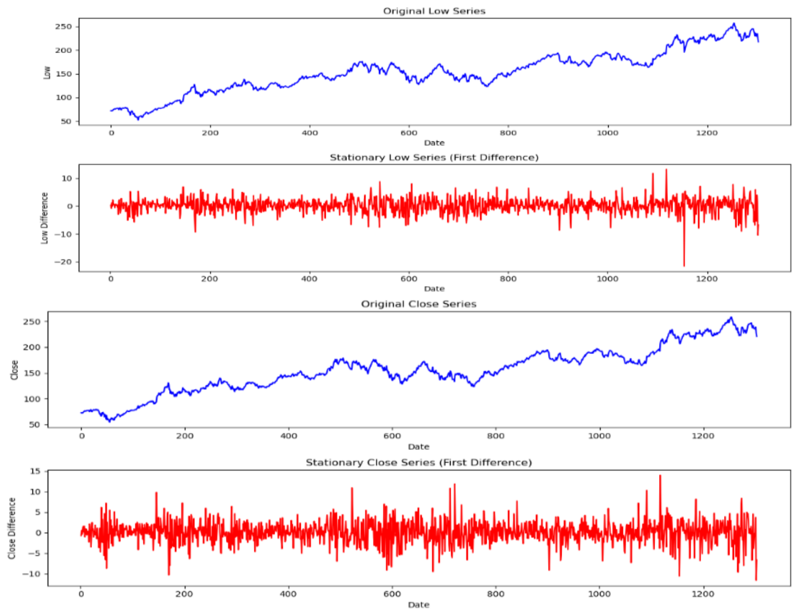
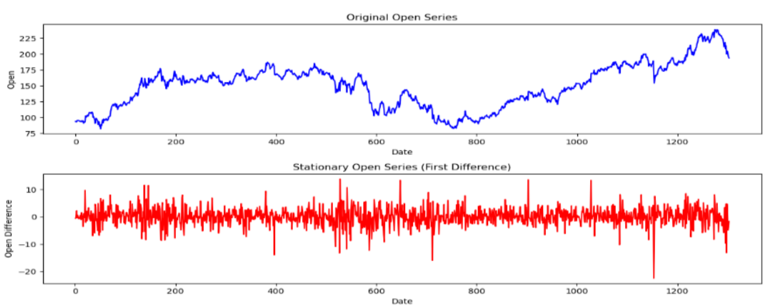
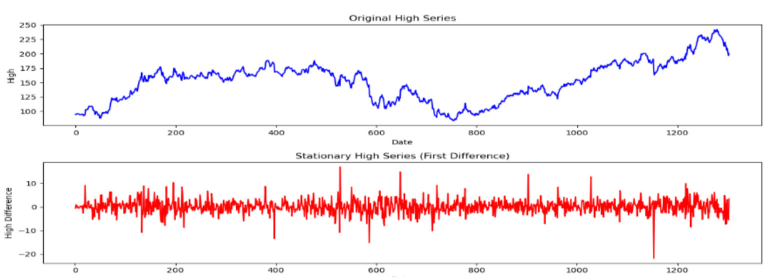


Figure 1. Comparison of Original and Stationarized OHLC APPLE.

Note: *Bottom panels show the first-differenced series (ΔPt), which is the forecasting target*

Source*: Research findings*





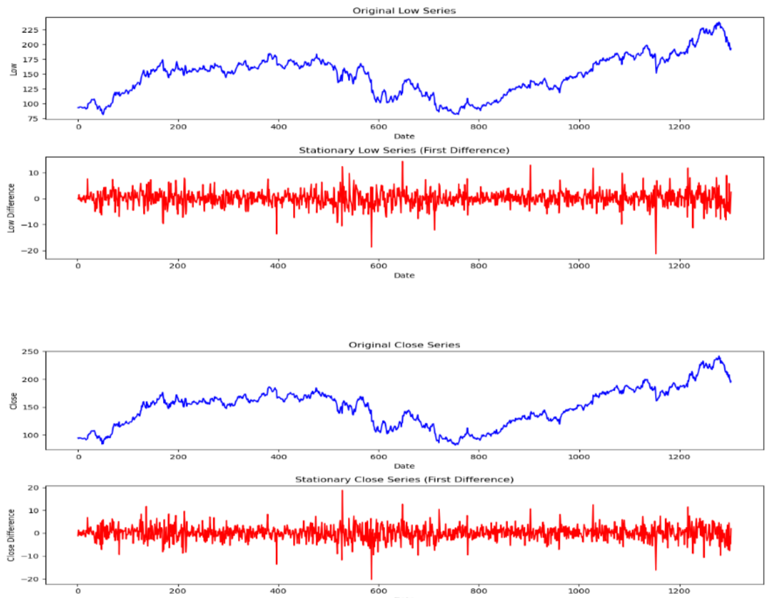
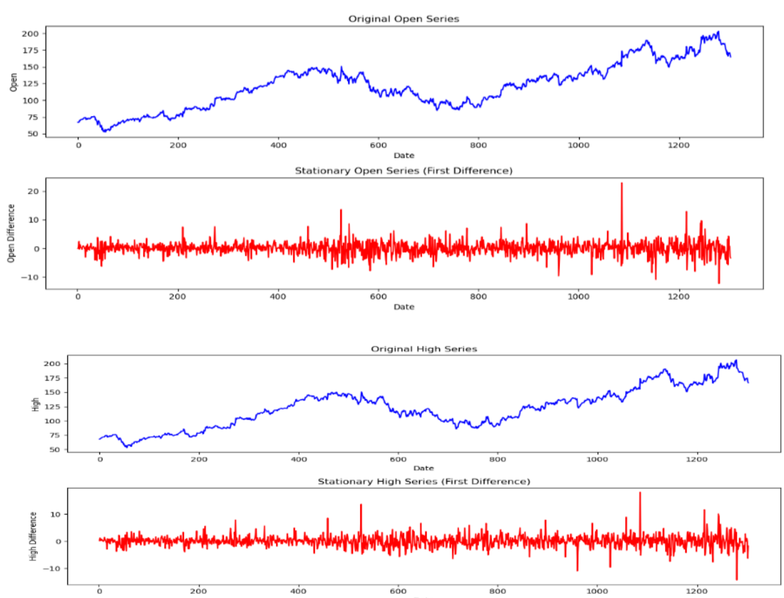


Figure 2. Comparison of Original and Stationarized OHLC AMZN

Note: *Bottom panels show the first-differenced series (ΔPt), which is the forecasting target*

Source*: Research findings*



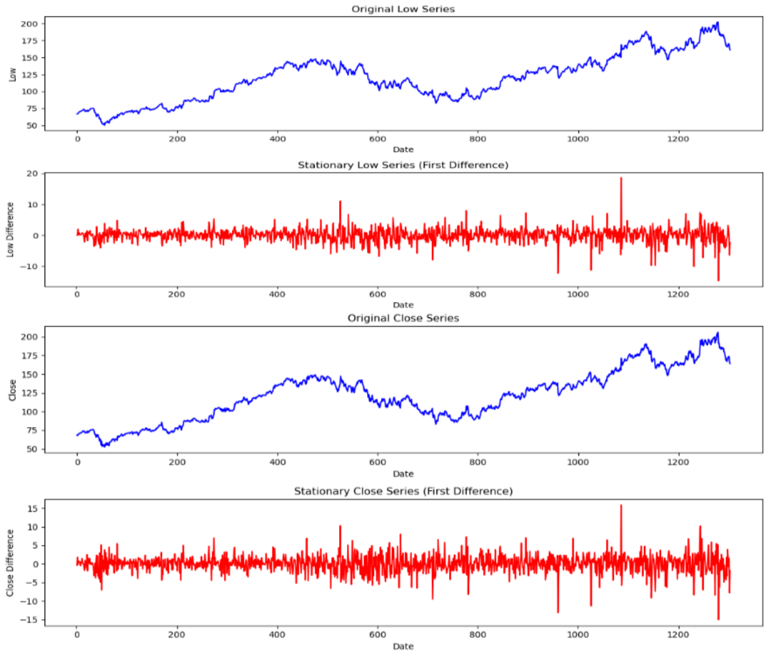
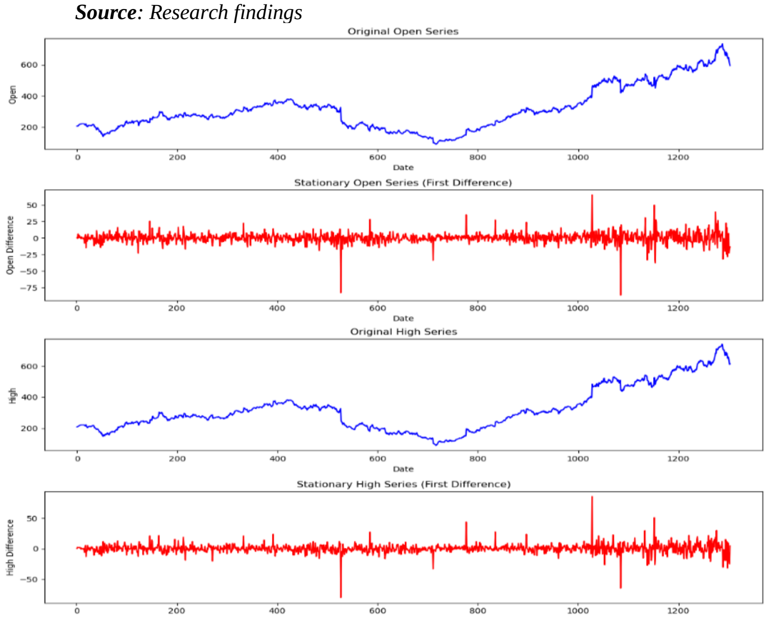


Figure 3. Comparison of Original and Stationarized OHLC GOOGLE

Note: *Bottom panels show the first-differenced series (ΔPt), which is the forecasting target*

Source*: Research findings*



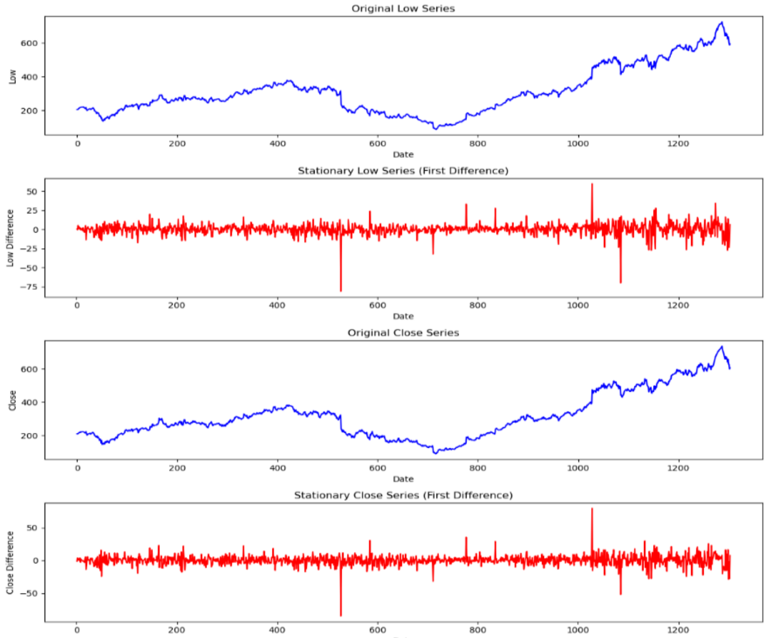
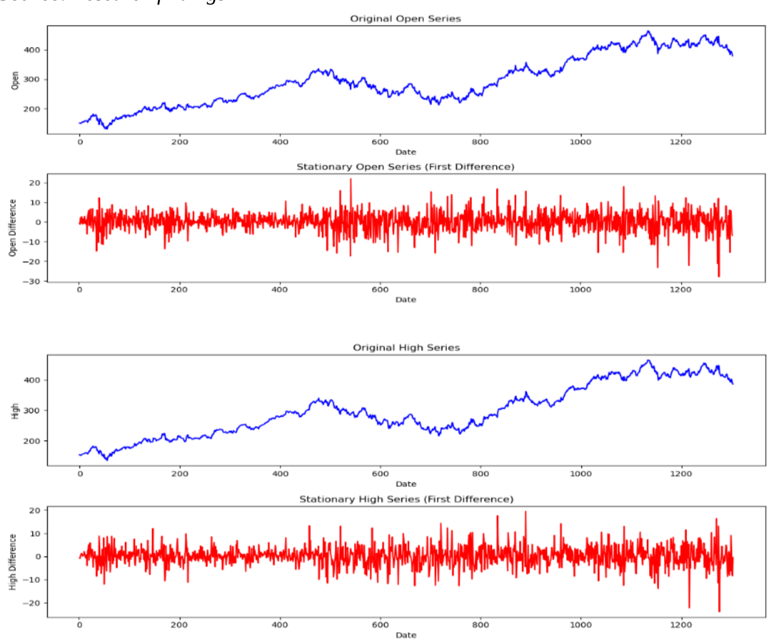


Figure 4. Comparison of Original and Stationarized OHLC META

Note: *Bottom panels show the first-differenced series (ΔPt), which is the forecasting target*

Source*: Research findings*



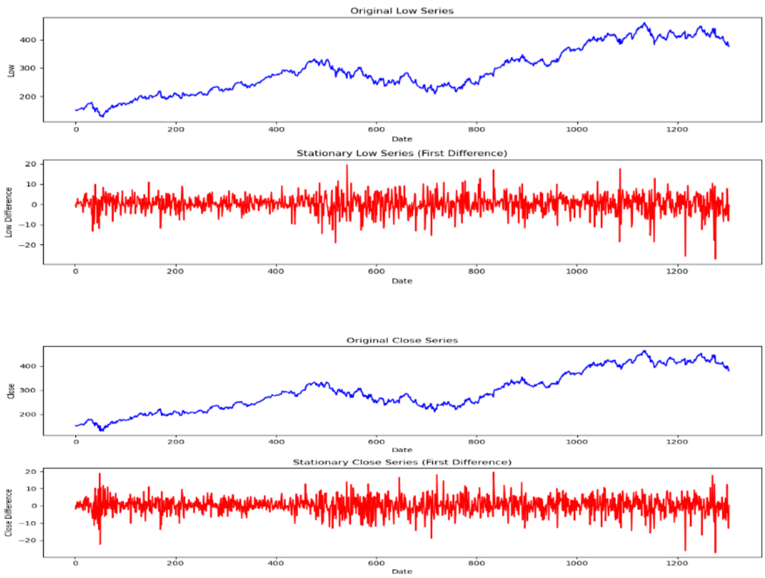


Figure 5. Comparison of Original and Stationarized OHLC MSFT

Note: *Bottom panels show the first-differenced series (ΔPt), which is the forecasting target*

Source*: Research findings*

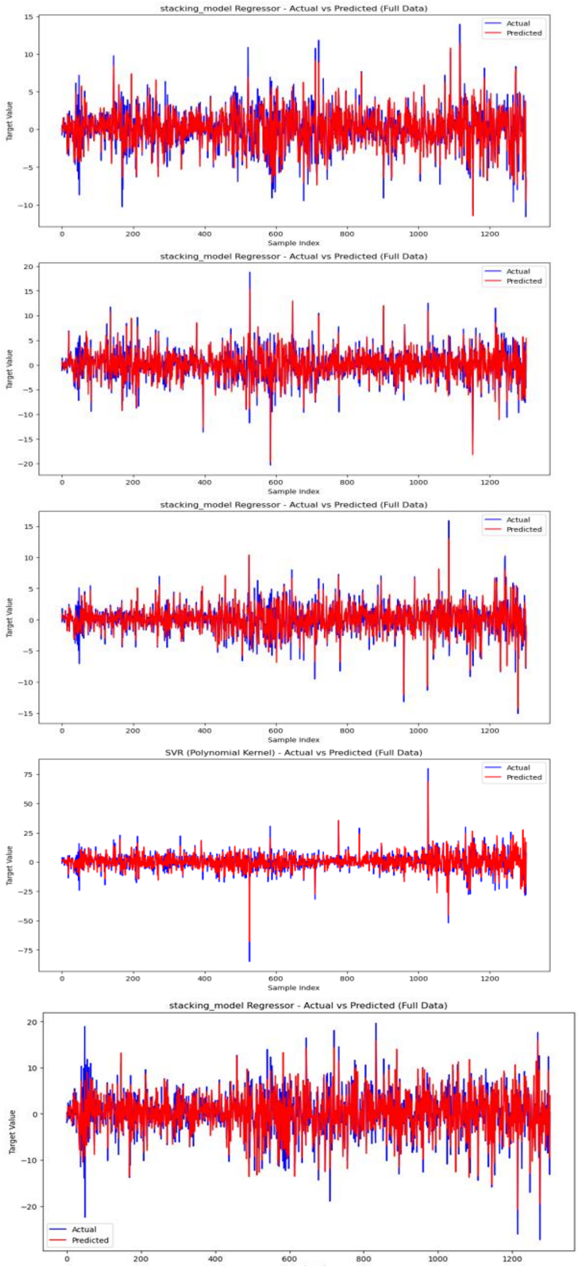


Figure 6. The best result
***Source**: Research findings*

Figures 1 through 5 each display a two-panel comparison of the Open, High, Low, and Close price series before and after stationarity transformation for the five stocks under study (APPLE, AMZN, GOOGLE, META, and MSFT). In the upper panel of each figure, the raw OHLC data reveal pronounced long-term trends and seasonal cycles, while the lower panel shows the corresponding series after differencing and detrending, with markedly diminished autocorrelation and a more uniform volatility profile. Figure 6 presents the best predictive results obtained by the proposed regression method on the stationarized series. The reported R^2values (0.80–0.83) are notably high for financial forecasting, which warrants careful interpretation. These metrics reflect the model's ability to explain variance in daily price changes (ΔClose), not raw price levels. When the target is stationary and exhibits lower overall variance (after differencing), R^2can naturally be higher relative to models predicting non-stationary prices.

More importantly, these results are achieved using same-day OHLC features, meaning the model has access to Open, High, and Low before predicting the Close change—a rich but realistic intraday information set. In practice, this corresponds to a trading scenario where signals are generated during the trading day, not before market open. The high R^2thus indicates strong explanatory power conditional on intraday information, not a claim of unrealistic predictive accuracy from prior-day data alone.

4.3 Discussion

The empirical results demonstrate that the proposed stacking ensemble achieves statistically significant improvements in forecasting daily price changes, as reflected by consistent reductions in RMSE and MAE across most assets. However, the practical relevance of these gains must be interpreted within the structural design and empirical scope of the study.

First, in terms of investment timing, the model generates intraday signals based on same-day Open, High, and Low prices. Consequently, its forecasts are informative for within-day portfolio rebalancing and execution strategies rather than for pre-market positioning. This timing structure implies that the model is best suited for supporting tactical allocation decisions and short-horizon trading adjustments during active market hours.

Second, regarding alpha generation, the observed improvements in predictive accuracy suggest a potential enhancement in signal quality relative to individual models and traditional benchmarks. By reducing average forecast errors, the ensemble increases the probability of correctly identifying short-term price movements. Nevertheless, the present study does not incorporate transaction costs, liquidity constraints, or slippage. Therefore, while the statistical gains indicate improved informational efficiency, they should be interpreted as necessary but not sufficient conditions for persistent abnormal returns. Future research incorporating trading simulations is required to evaluate the economic significance of these signals.

Third, from a risk management perspective, the ensemble's high explanatory power and stability during turbulent periods enhance its applicability for short-term volatility estimation and portfolio risk monitoring. More accurate predictions of daily price changes contribute to improved Value-at-Risk and stress-testing frameworks, supporting more efficient capital allocation and hedging strategies. In this context, the primary contribution of the model lies in reducing forecast uncertainty rather than in direct return maximization.

Importantly, these practical implications are bounded by the study's empirical design. The analysis focuses on five mega-cap technology stocks over the 2020–2025 period and relies exclusively on OHLC information. Consequently, the findings are most directly applicable to highly liquid, information-rich markets and may not generalize to less efficient assets or low-liquidity environments. Moreover, the use of first-differenced prices emphasizes short-term dynamics and limits inference regarding long-term investment horizons.

Overall, the proposed framework should be viewed as a decision-support tool that enhances intraday timing, improves the informational basis for alpha-oriented strategies, and strengthens short-term risk management. Its value lies in complementing, rather than replacing, comprehensive investment processes that integrate market microstructure, cost considerations, and portfolio constraints.

4.3.1 Analysis of the Apple (AAPL) Anomaly

The ensemble's improvement was not statistically significant for Apple (AAPL), in contrast to the other four stocks. Several non-exclusive factors may explain this result:

Market Characteristics: AAPL's price series during 2020–2025 may have exhibited lower volatility or higher market efficiency, leaving less nonlinear, predictable structure for an ensemble to exploit beyond what a single good model (e.g., linear SVM) could capture.

Model Correlation: The base learners (SVR, RF, XGBoost) may have generated highly correlated predictions for AAPL, minimizing the diversity needed for the meta-learner to synthesize a superior forecast.

Data Structure: AAPL's price movements might align more closely with simpler linear patterns, reducing the advantage of a complex ensemble.

This case importantly underscores that the benefits of ensemble methods, while significant in volatile and complex environments, are context-dependent and not universal.

4.3.2 Analysis of XGBoost Underperformance: A Case Study on Model-Data Mismatch

The isolated underperformance of XGBoost, particularly for META (R^2 = 0.301, RMSE = 8.752), presents a valuable case study in model-data fit. While XGBoost is a powerful learner, its performance is contingent on the structure of the data. We posit three interconnected explanations for this result:

1. Extreme Volatility and Structural Breaks in META's Series: The 2020–2025 period for META was marked by unprecedented volatility, including a ~65% drawdown in 2022 driven by shifts in digital advertising, metaverse investment concerns, and macroeconomic pressures. XGBoost, as a tree-based model that partitions the feature space, may struggle with such extreme, low-probability events that represent structural breaks rather than smooth functional relationships. Its sequential error correction can overfit to these abrupt, non-stationary shocks, harming generalization.

2. Overfitting to Noise in High-Variance Regimes: XGBoost's strength in bias reduction can become a weakness in exceptionally noisy regimes. META's series during this window exhibited very high kurtosis (fat-tailed returns). XGBoost's boosting mechanism, designed to meticulously correct residual errors, may have over-optimized to idiosyncratic noise in the training window, failing to capture a generalizable pattern. The simple linear meta-learner in our stacking ensemble likely down-weighted these overfit predictions.

3. Hyperparameter Sensitivity and Regime Dependency: Although hyperparameters were tuned via grid search, the optimal settings for stable periods may not translate to turbulent ones. XGBoost's performance is highly sensitive to regularization parameters (e.g., gamma, lambda). It is plausible that the volatility of META's series required a much stronger regularization penalty than was optimal for the other, less volatile stocks—a subtlety a standard grid search may not capture across diverse assets.

This finding is instructive: it underscores that no single model is universally optimal. The poor performance of a typically strong algorithm like XGBoost on a specific asset under extreme stress precisely validates the core premise of our stacking approach. The ensemble architecture is designed to mitigate such model-specific failures by relying on the consensus and complementarity of diverse learners. The fact that the overall ensemble maintained robust performance on META ($R^2 = 0.733$) despite XGBoost's weakness is a direct demonstration of the stacking framework's protective benefit against model-data mismatch.

5. Concluding Remarks

In conclusion, the experimental results across five major stock datasets demonstrate that the proposed regression approach outperforms traditional machine learning models in terms of both accuracy and consistency. By achieving the highest R^2 values and lowest error rates (MAE, RMSE, MSE) in most scenarios, the proposed method has shown superior capability in modeling complex financial patterns. While SVM variants—particularly with linear, polynomial, and Gaussian kernels—also delivered competitive results, especially for AMZN and GOOGLE, models like Decision Tree and XGBoost were generally less effective, particularly in volatile datasets like META. The Friedman test statistically confirms the significance of these differences for most datasets, reinforcing the robustness and reliability of the proposed approach. The inclusion of a traditional ARIMA benchmark further confirms the robustness of the empirical findings. The superior performance of the proposed models over this classical baseline highlights the effectiveness of machine learning techniques in capturing complex dynamics in financial time series.These findings suggest that integrating ensemble strategies with careful model stacking can significantly enhance prediction performance in stock market forecasting tasks.

Empirically Demonstrated Superiority: Achieved a statistically significant reduction in {RMSE} (8–15%) and consistently high R^2 (0.80–0.83) across five major technology stocks during the highly volatile 2020–2025 market period, outperforming all individual base models.

Validated Structural Robustness: Provided evidence that the ensemble architecture maintains high predictive stability during major market shocks, confirming its resilience against non-stationarity and extreme market events.

The theoretical value lies in the strategic design of the stacking architecture to maximize the Bias-Variance Trade-off control:

Optimized Complementarity: The ensemble successfully integrates base learners with fundamentally different error characteristics: the SVR (sensitive to scaling, good at finding optimal hyperplanes), the Random Forest (low variance, high stability), and XGBoost (low bias, powerful boosting). This ensures that errors are decorrelated.

Controlled Complexity: The deliberate choice of a simple Linear Regression Meta-Learner acts as a powerful regularizer, ensuring that the ensemble benefits from the non-linearity of the base predictions without incurring a second layer of overfitting complexity at the aggregation stage. This parsimonious design is essential for generalization in noisy financial data.

For future work, the proposed approach can be further enhanced by incorporating advanced ensemble and multi-stage stacking strategies alongside systematic hyperparameter optimization to improve robustness and generalization. In addition, integrating explainability techniques such as SHAP (Shapley Additive Explanations) and related attribution methods would provide deeper insight into feature contributions, improve model transparency, and support more trustworthy financial decision-making. Future research should also evaluate the model on larger and more diverse datasets—including high-frequency data, alternative asset classes, and emerging markets—while accounting for transaction costs and market frictions to ensure real-world applicability. Moreover, exploring online learning, regime-aware modeling, and hybrid frameworks that combine deep learning with classical machine learning can further strengthen predictive performance under dynamic market conditions.

Author Contributions:

For example: Conceptualization, all authors; methodology, Y.N. and M.E.; validation, Y.N. and M.E.; formal analysis, all authors; resources, Y.N. and M.E.; writing—original draft preparation, Y.N. and M.E.; writing—review and editing, all authors; supervision, Y.N. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Conflicts of Interest:

The authors declare no conflict of interest.

Data Availability Statement:

The data will be available upon reasonable request.

Acknowledgements

Not applicable

References

Bin Omar, A., Huang, S., Salameh, A. A., Khurram, H., & Fareed, M. (2022). Stock market forecasting using the random-forest and deep-neural-network models before and during the COVID-19 period. *Frontiers in Environmental Science, 10*, 917047.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Breiman, L., Friedman, J., Olshen, R. A., & Stone, C. J. (2017). *Classification and regression trees*. Routledge. (Original work published 1984)

Chen, C., Xue, L., & Xing, W. (2023). Research on improved GRU-based stock-price prediction method. *Applied Sciences*, *13*(15), 8813.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM.

Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, *7*, e623.

Chopra, R., & Sharma, G. D. (2021). Application of artificial intelligence in stock market forecasting: A critique, review, and research agenda. *Journal of Risk and Financial Management, 14*(11), 526.

Christmann, A., & Steinwart, I. (2008). *Support vector machines*. Springer.

Ebiesuwa, S., Asake, A. O., Adegbola, A. A., Adegbenjo, A. A., Okesola, K. I., Falana, T. O., & Awoniyi, T. A. (2024). Stock-price prediction: Evaluating the efficacy of CNN, LSTM, CNN-LSTM and CNN-BiLSTM models. *International Journal of Intelligent Systems and Applications in Engineering, 12*(4), 497–506.

Gao, Y. (2021). Comparative analysis of GRU and LSTM networks for stock-price prediction. *IEEE Access*, *9*, 123456–123469.

Hoseini, S. M., Ebtia, M., & Ariafar, M. R. (2025). Comparison of single and ensemble support vector regression model for stock price prediction. *Soft Computing Journal*. Advance online publication.

Li, Q., Kamaruddin, N., Yuhaniz, S. S., & Al-Jaifi, H. A. (2024). Forecasting stock-price changes using a long-short-term memory neural network with symbolic genetic programming. *Scientific Reports*, *14*(1), 422.

Liang, L. (2024). ARIMA with attention-based CNN-LSTM and XGBoost hybrid model for stock prediction in the U.S. stock market. *SHS Web of Conferences, 196*, Article 02001.

Mallikarjuna, M., & Rao, R. P. (2019). Evaluation of forecasting methods from selected stock market returns. *Financial Innovation*, *5*(1), 40.

Ouf, S., El Hawary, M., Aboutabl, A., & Adel, S. (2024). A deep-learning-based LSTM for stock-price prediction using Twitter sentiment analysis. *International Journal of Advanced Computer Science and Applications, 15*(12), 207–216.

Platt, J. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in large margin classifiers* (pp. 61–74). MIT Press.

Shah, J., Vaidya, D., & Shah, M. (2022). A comprehensive review on multiple hybrid deep-learning approaches for stock prediction. *Intelligent Systems with Applications*, *16*, 200111.

Şimşek, A. İ. (2025). Using stacked generalization model in stock price forecasting: A comparative analysis on BIST100 Index. *Fiscaoeconomia*, *9*(1), 305–322.

Tran, P., Pham, T. K. A., Phan, H. T., & Nguyen, C. V. (2024). Applying machine-learning algorithms to predict the stock-price trend: The case of Vietnam. *Humanities and Social Sciences Communications, 11*(1), 393.

Ullah, S., Khan, S., Hashmi, N. I., & Alam, M. S. (2023). COVID-19 pandemic and financial market volatility: A quantile regression approach. *Heliyon, 9*(10), Article e20334.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, *5*(2), 241–259.

Yang, S., Ding, Y., Xie, B., Guo, Y., Bai, X., Qian, J., ... & Ren, J. (2023). Advancing financial forecasts: A deep dive into memory attention and long-distance loss in stock price predictions. *Applied Sciences, 13*(22), 12160.

Appendices

Table 1A. Comparative Overview of Recent AI Stock Forecasting Studies, 2022–2025

Study (Year)	Core Model/Approach	Dataset Focus	Key Performance Metric / Finding
Bin Omar et al., 2022	Random Forest (RF) vs. DNN vs. ARIMA	Stocks during COVID-19 (Before/During)	RF outperformed ARIMA and DNN; superior robustness during volatile sub-periods.
Gao (2021)	GRU vs. LSTM	Chinese A-share Equities	GRU and LSTM achieved statistically indistinguishable RMSE, highlighting the efficiency of GRU.
Shah et al., (2022)	Ensemble/Hybrid Models (Review)	Diverse Markets	Concluded that stacked or hybrid ensembles consistently yielded smaller RMSE/MAE than single models.
Chen et al., (2023)	GRU (Augmented Training Data)	Peer Company Series (Synthetic)	Mitigated overfitting by using synthetic data; reduced MAE by ≈ 15% relative to vanilla GRU.
Liang (2024)	Hybrid Stack: ARIMA + Attention CNN-LSTM + XGBoost	U.S. Stock Market (2022–2023 Turbulence)	Stacked model's RMSE was 8–12\% lower than component models, demonstrating complementarity.
Ouf et al., 2024	XGBoost vs. Pure LSTM (Multimodal)	Apple, Google, Tesla (With Twitter Sentiment)	XGBoost trained on price + sentiment outperformed pure LSTM on MAE and R^2.
Ebiesuwa et al., (2024)	CNN-LSTM w/ Attention vs. Baselines	Nigerian Banking Stocks	CNN-LSTM with Attention achieved the lowest RMSE/MAE and highest R^2 among DL alternatives.
Proposed Method (2025)	Stacked Ensemble: SVR + RF + XGBoost	5 BigTech Stocks (2020–2025 Turbulence)	Highest R^2 0.80–0.83; RMSE reduced by 8–15 vs. best single model.

***Source**: Research findings*

Table 2A. Hyperparameter Details

Model	Hyperparameter	Search Range (Grid Search)
DT	max_depth	np.arange(1,30)
RF	n_estimators, max_depth	np.arange(1,30)
SVM(Gaussian)	C, gamma	np.logspace(-5,5,10)
SVM(Linear)	C	np.logspace(-5,5,10)
SVM(Poly)	Degree	np.arange(1,5)
SVM(sigmoid)	Gamma	np.logspace(-5,5,10)
XGB	n_estimators, max_depth, learning_rate	np.arange(1,30), [0.01, 0.05, 0.1]
Propose Method	SVR, RF, XGBoost	The previous intervals were mentioned.

Table 3A. Test conditions

Component	Detail	Rationale
Platform	Google Colaboratory (Pro/High-RAM Instance)	Ensured access to required memory for large time series data and persistent runtime sessions.
CPU	Intel Xeon Scalable Processor (typical configuration)	Provided multi-core parallel processing capacity, critical for {RF} and {XGBoost} training.
Key Software	Python 3.10; Scikit-learn 1.3.0; XGBoost 1.7.0	Specific versions ensure exact replication of model behavior.

***Source**: Research findings*

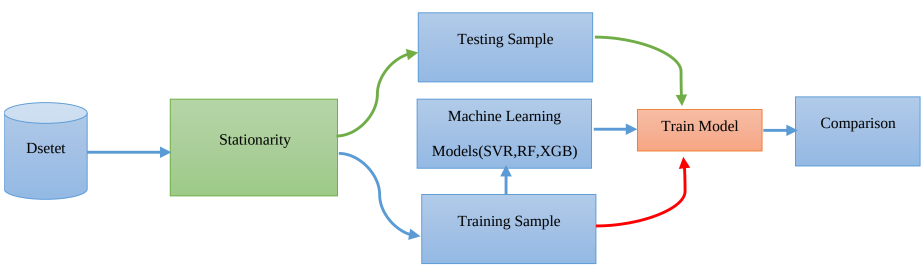


Figure 1. Flow Diagram of Machine Learning

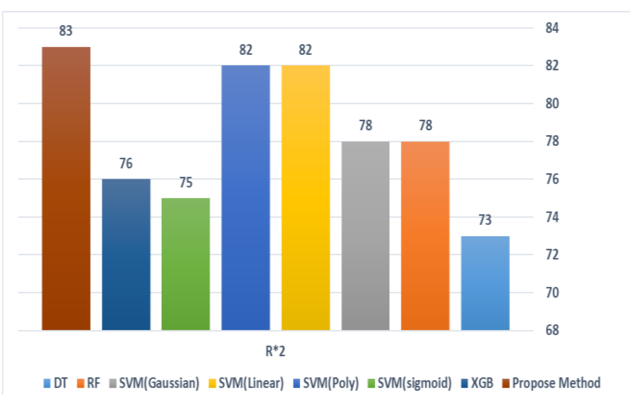


Figure 2. R^2 metric for Google