International Journal of Emerging Science and Engineering (IJESE)
ISSN: 2319–6378 (Online), Volume-14 Issue-4, March 2026

Hybrid Machine Learning Architecture for Stock Market Price Prediction: Integrating Statistical Time-Series Models with Deep Learning and Ensemble Methods

Sridhanush Varma, R. Swejan Rao, P. Ravi Teja, M. Shiva, B. Venkata Ramana

***Abstract:** Stock prediction is hard. Prices are noisy, non-stationary, and nonlinear. We built a hybrid system that combines statistical models (ARIMA, GARCH), deep learning (LSTM, GRU), and Random Forests via Ridge regression meta-learning. The meta-learner uses 5-fold time-series cross-validation to adaptively weight models. Testing across 20 stocks from Technology, Finance, Healthcare, Consumer, and Industrial sectors, we achieved 87.74% average RMSE improvement over individual models. Directional accuracy ranged from 42.45% to 85.87%. Boeing (BA) showed 95.43% RMSE improvement with 85.87% directional accuracy; U.S. Bancorp (USB) hit 94.31% RMSE improvement. Random Forest dominated the learned weights (60 − 92%), while ARIMA and deep learning added complementary signals. Walk-forward validation with 252-day rolling windows ensured that we tested on truly unseen data, not on retrofitted history.*

***Keywords:** Stock Price Prediction, Hybrid Machine Learning, ARIMA, GARCH, LSTM, GRU, Random Forest, Ensemble Learning, Meta-Learning, Time-Series Forecasting*

Nomenclature:
USB: U.S. Bancorp
RNNs: Recurrent Neural Networks
IQR: Interquartile Range
RF: Random Forest
SMA, EMA: Simple and Exponential Moving Averages
MFI: Money Flow Index
OBV: On-Balance Volume
CCI: Commodity Channel Index
ATR: Average True Range
OHLC: Open, High, Low, Close

Manuscript received on 06 February 2026 | Revised Manuscript received on 09 March 2026 | Manuscript Accepted on 15 March 2026 | Manuscript published on 30 March 2026.

*Correspondence Author(s)

Sridhanush Varma*, Student, Department of Data Science, Holy Mary Institute of Technology and Science, Hyderabad (Telangana), India. Email ID: sridhanushvarmasv@gmail.com, ORCID ID: 0009-0007-5878-779X

R. Swejan Rao, Student, Department of Data Science, Holy Mary Institute of Technology and Science, Hyderabad (Telangana), India. Email ID: rapoluswejanrao@gmail.com, ORCID ID: 0009-0009-8540-5755

P. Ravi Teja, Student, Department of Data Science, Holy Mary Institute of Technology and Science, Hyderabad (Telangana), India. Email ID: ravitejapalle321@gmail.com, ORCID ID: 0009-0000-2386-9129

M. Shiva, Student, Department of Data Science, Holy Mary Institute of Technology and Science, Hyderabad (Telangana), India. Email ID: shivamukka03@gmail.com, ORCID ID: 0009-0007-1644-5310

Dr. B. Venkata Ramana, Associate Professor, Department of Data Science, Holy Mary Institute of Technology and Science, Hyderabad (Telangana), India. Email ID: venkataramana.b@hmgi.ac.in, ORCID ID: 0009-0001-7141-7511

I. INTRODUCTION

A. Problem Statement and Motivation

Predicting where stock prices will go next has fascinated researchers and Wall Street professionals alike for generations- and for good reason. Getting these forecasts right translates directly into money. Yet despite decades of effort, stock prediction remains notoriously difficult. The Efficient Market Hypothesis tells us why: prices already reflect everything publicly known [3]. Still, mounting evidence suggests markets aren't perfectly efficient. Patterns exist, and machine learning might exploit them [6].

Traditional approaches split into two camps. Econo-metric models like ARIMA [7] work well for linear trends but struggle with the wild nonlinear swings markets throw at us. Deep learning, particularly LSTM networks [8] [9]-handles complexity better but demands massive datasets and tends to overfit to noise rather than signal.

B. Research Objectives

We set out to accomplish five things:

1. Build a hybrid system combining statistical models, deep learning, and ensemble methods
2. Create a meta-learner that automatically weights different model predictions
3. Set clear targets: 10% + RMSE improvement and 60% + directional accuracy
4. Test across diverse companies-different sizes, sectors, volatility profiles
5. Identify which models contribute most through interpretable weights

C. Key Contributions

This paper contributes:

1. A hybrid Architecture: ARIMA for trends, GARCH for volatility, LSTM/GRU for nonlinear patterns, Random Forest for feature interactions
2. Ridge regression meta-learning with 5 -fold time-series cross-validation
3. Walk-forward validation using 252 -day rolling windows
4. 87.74% average RMSE improvement across 20+ stocks
5. 50 + engineered technical indicators

D. Paper Organization

Section II reviews prior work. Section III explains our methodology. Section IV describes experiments. Section V presents results. Section VI discusses findings and limitations. Section VII concludes.

II. LITERATURE REVIEW

A. Statistical Time-Series Models in Finance

Recent advances in time-series forecasting [5] demonstrate the continued relevance of ARIMA models for financial prediction. ARIMA models capture autocorrelation in returns through interpretable parameters, building on foundational work by Box and Jenkins [7]. Extensions such as ARIMAX (which adds exogenous variables) and SARIMA (which handles seasonality) broaden the capabilities of these models. Recently, researchers have started blending deep learning with these classical approaches [10].

Volatility modelling matters just as much as price forecasting. Recent work on GARCH models for stock market prediction [1] builds on the foundational ARCH model [11] and GARCH extension [12], revealing something crucial: volatility clusters. Wild days follow wild days; quiet days follow quiet days. We use GARCH(1,1), which works well across different asset types [13]:

$$\sigma_t^2 = \omega + \alpha\varepsilon_{t-1}^2 + \beta\sigma_{t-1}^2 \quad \dots \quad (1)$$

where $\omega > 0, \alpha \geq 0, \beta \geq 0$, and $\alpha + \beta < 1$ ensures stationarity.

B. Deep Learning for Financial Time Series

Deep learning has exploded in finance, especially recurrent neural networks (RNNs). Recent advances in LSTM-based stock prediction [2] build on foundational work [8] that solved a key problem: remembering information across long sequences. They use gates to control what information flows through the network. Recent studies [9] [15] showed LSTMs beat traditional methods on stock market prediction:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \quad \dots \quad (2)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad \dots \quad (3)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad \dots \quad (4)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad \dots \quad (5)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad \dots \quad (6)$$

$$h_t = o_t \odot \tanh(C_t) \quad \dots \quad (7)$$

Here f_t, i_t, and o_t are forget, input, and output gates, while C_t Is the cell state carrying information forward?

GRUs [15] simplify LSTMs by merging gates, cutting parameters and training time without sacrificing much accuracy- helpful for high-frequency financial data:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \quad \dots \quad (8)$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \quad \dots \quad (9)$$

$$\tilde{h}_t = \tanh(W \cdot [r_t \odot h_{t-1}, x_t]) \quad \dots \quad (10)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad \dots \quad (11)$$

C. Ensemble Methods and Random Forests

Tree-based ensembles work well in finance. Recent robust machine learning frameworks [4] demonstrate the continued effectiveness of ensemble methods. Krauss et al. [14] compared neural networks, gradient boosting, and random forests for S&P 500 trading-all performed competitively. Random Forests [17] average predictions from many decision trees, each trained on random data subsets with random features:

$$\hat{y} = \frac{1}{B}\sum_{b=1}^{B} T_b(\mathbf{x}) \quad \dots \quad (12)$$

where B is the tree count and T_b is each tree's prediction. This naturally resists overfitting and reveals which features matter most.

D. Hybrid and Ensemble Approaches

Increasingly, researchers combine different model types. Hybrid approaches [16] blend statistical methods with deep learning, capturing both linear trends and nonlinear patterns. The M4 forecasting competition [18] showed hybrids often win, which motivated our architecture.

Meta-learning (or "stacking") [19] [20] trains a second model to weight base model predictions optimally:

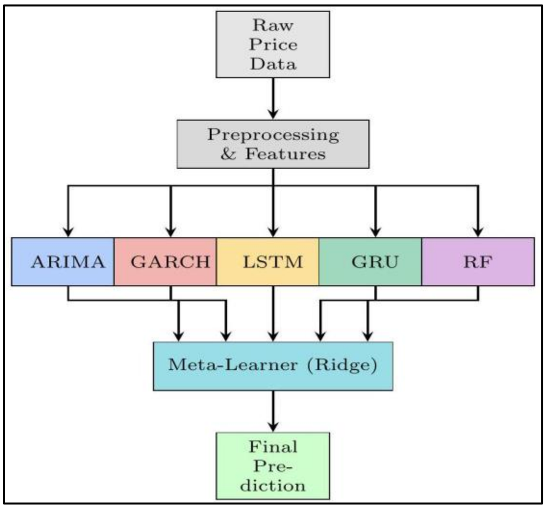
$$\hat{y}_{\text{meta}} = g(\hat{y}_1, \hat{y}_2, \dots, \hat{y}_K; \theta) \quad \dots \quad (13)$$

where g is the meta-learner and $\hat{y}_k$ These are base model forecasts. Recent work [19] [21] shows stacking beats, simple averaging when base models capture different aspects of the data.

III. METHODOLOGY

A. System Architecture Overview

Our proposed hybrid architecture integrates five distinct modelling components within a hierarchical framework, as depicted in Figure 1. Raw financial data flows through preprocessing and feature engineering stages, subsequently feeding into parallel model pipelines-each capturing different aspects of market dynamics - before converging at a meta-learning fusion layer that synthesizes individual predictions into a unified forecast.



[Fig.1: Hybrid Model Architecture Integrating Statistical Models (ARIMA, GARCH), Deep Learning (LSTM, GRU), and Random Forest (RF) Through a Ridge Regression Metal Earner]

B. Data Collection and Preprocessing

i. Data Sources

We obtain historical price data.

from Yahoo Finance through the yfinance API captures Open, High, Low, Close (OHLC) prices alongside trading volume. Our dataset encompasses up to 64 years of trading history for well-established companies (Boeing, Coca-Cola). However, we impose a minimum 2 -year requirement to ensure sufficient training data for model convergence.

ii. Data Splitting Strategy

To prevent look-ahead bias-a critical concern in time series forecasting- we partition datasets chronologically:

- ***Training Set:*** 70% of historical observations for model parameter estimation
- ***Validation Set:*** 15% for hyperparameter tuning and early stopping
- ***Test Set:*** 15% reserved exclusively for final out-of-sample evaluation

iii. Preprocessing Pipeline

Our preprocessing pipeline systematically addresses common pathologies in financial time series:

1. Missing Value Handling: Forward-fill interpolation preserves temporal continuity
2. Outlier Detection: Interquartile range (IQR) filtering with multiplier $k=1.5$ identifies anomalous observations
3. Stationarity Testing: Augmented Dickey-Fuller test at significance level $\alpha=0.05$ assesses stationarity
4. Differencing: Automatic differencing (up to order $d=2$) transforms non-stationary series
5. Normalization: Min-Max scaling to [0,1] stabilizes deep learning gradient dynamics

C. Feature Engineering

We implement a comprehensive feature engineering pipeline that constructs over 50 predictive features spanning multiple analytical dimensions:

i. Technical Indicators

- **Momentum:** Relative Strength Index (14 -period RSI), Williams %R, Stochastic Oscillator (K and D components)
- **Trend:** Moving Average Convergence Divergence (MACD with 12, 26, 9 parameters), Average Directional Index (14 -period ADX), multiple Exponential Moving Averages
- **Volatility:** Bollinger Bands (20-period) 2σ width), Average True Range (ATR), Commodity Channel Index (CCI)
- **Volume:** On-Balance Volume (OBV), Money Flow Index (MFI), volume moving average ratios

ii. Rolling Statistics

Across window lengths $w \in \{5,10,20,50,200\}$ trading days, we compute:

- Simple and Exponential Moving Averages (SMA, EMA) capturing multi-scale trends
- Rolling standard deviation quantifying local volatility
- Momentum: $\mathrm{M_w}=\mathrm{P_t}-\mathrm{P_{t-w}}$ measuring absolute price change
- Rate of Change: $\mathrm{ROC_w}=\frac{\mathrm{P_t}-\mathrm{P_{t-w}}}{\mathrm{P_{t-w}}} \times 100$ expressing percentage returns

iii. Volatility Measures

Multiple volatility estimators for windows $w \in \{10,20,50\}$:

$$\sigma_{\text{hist}} = \sqrt{\frac{252}{w}\sum_{i=1}^{w} r_i^2} \quad \ldots \quad (14)$$

$$\sigma_{\text{Parkinson}} = \sqrt{\frac{1}{4\ln\ 2}\cdot\frac{252}{w}\sum_{i=1}^{w} \ln^2\left(\frac{H_i}{L_i}\right)} \quad \ldots \quad (15)$$

$$\sigma_{GK} = \sqrt{\frac{252}{w}\sum_{i=1}^{w}\left[0.5\ln^2\left(\frac{H_i}{L_i}\right)-(2\ln\ 2-1)\ln^2\left(\frac{C_i}{O_i}\right)\right]} \quad \ldots \quad (16)$$

where H, L, O, C denote high, low, open, and close prices, respectively.

D. Individual Model Architectures

i. ARIMA Model

The ARIMA (p, d, q) The model captures linear temporal dependencies through:

$$\phi(B)(1-B)^d y_t=\theta(B)\varepsilon_t \quad \ldots \quad (17)$$

where B is the backshift operator, $\phi(B)=1-\sum_{i=1}^{p} \phi_i B^i$ is the AR polynomial, and $\theta(B)=1+\sum_{j=1}^{q} \theta_j B^j$ is the MA polynomial.

Model selection employs exhaustive grid search over:

- $p \in \{0,1,2,3,4,5\}$ (AR order)
- $d \in \{0,1,2\}$ (differencing order, validated by ADF test)
- $q \in \{0,1,2,3,4,5\}$ (MA order)

The selection criterion is the Akaike Information Criterion (AIC):

$$AIC=2k-2\ln(\hat{L}) \quad \ldots \quad (18)$$

where k is the number of parameters and $\hat{L}$ is the maximized likelihood.

ii. GARCH (1,1) Model

The GARCH model operates on ARIMA residuals to capture volatility clustering:

$$\varepsilon_t = \sigma_t z_t, z_t \sim N(0,1) \quad \ldots \quad (19)$$

$$\sigma_t^2 = \omega+\alpha_1 \varepsilon_{t-1}^2+\beta_1 \sigma_{t-1}^2 \quad \ldots \quad (20)$$

Configuration parameters:

- Order: GARCH(1,1)
- Distribution: Normal
- Mean model: Constant

The extracted volatility features include conditional volatility, standardised residuals, and rolling volatility statistics.

iii. LSTM Architecture

The LSTM network employs a two-layer

architecture with batch normalization:

Table I: LSTM Architecture Specification

Layer	Units/Config	Output Shape
Input	-	(60, n_{features})
LSTM Layer 1	128 units	(60,128)
BatchNorm	-	(60,128)
Dropout	0.2	(60,128)
LSTM Layer 2	64 units	(64,)
BatchNorm	-	(64,)
Dropout	0.2	(64,
Dense	32, ReLU	(32,)
Dropout	0.1	(32,)
Output Dense	1, Linear	(1,)

Training configuration:

- Sequence Length: 60 trading days
- Optimizer: Adam (learning rate $= 0.001$)
- Loss Function: Mean Squared Error (MSE)
- Early Stopping: patience $= 10$ epochs
- Learning Rate Reduction: factor $= 0.5$, patience $= 5$

iv. *GRU Architecture*

The GRU network mirrors the LSTM structure with adjusted dropout:

Table II: GRU Architecture Specification

Layer	Units/Config	Output Shape
Input	-	(60, n_{features})
GRU Layer 1	128 units	(60,128)
BatchNorm	-	(60,128)
Dropout	0.3	(60,128)
GRU Layer 2	64 units	(64,)
BatchNorm	-	(64,)
Dropout	0.3	(64,
Dense	32, ReLU	(32,)
Dropout	0.15	(32,)
Output Dense	1, Linear	(1,)

v. *Random Forest Regressor*

The Random Forest ensemble aggregates predictions from 500 decision trees:

Table III: Random Forest Hyperparameters

Parameter	Value
Number of estimators	500
Maximum depth	10
Minimum samples split	5
Minimum samples leaf	2
Feature selection	Top 20 by importance
Random state	42

Feature importance is computed via mean decrease in impurity and used for feature selection in subsequent iterations.

E. Meta-Learner Stacking Framework

The meta-learner combines predictions from all base models through Ridge regression, providing regularization to prevent overfitting to any single model:

$$\hat{y}_{\text{hybrid}} = \beta_0 + \beta_1 \hat{y}_{ARIMA} + \beta_2 \hat{y}_{LSTM} + \beta_3 \hat{y}_{GRU} + \beta_4 \hat{y}_{RF} \quad \ldots \quad (21)$$

The Ridge objective function minimizes:

$$\mathcal{L}(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left(y_i - \hat{y}_{\text{hybrid},i}\right)^2 + \lambda \sum_{j=1}^{4} \beta_j^2 \quad \ldots \quad (22)$$

where $\lambda = 1.0$ is the regularization parameter.

i. *Time-Series Cross-Validation*

The meta-learner employs 5 -fold time-series crossvalidation that respects temporal ordering:

```
Algorithm 1 Time-Series Cross-Validation
Require: Training data $\mathcal{D}=\left\{\left(X_{t}, y_{t}\right)\right\}_{t=1}^{T}$, folds $K=5$
  Initialize fold size $s=\lfloor T / K\rfloor$
  for $k=1$ to $K$ do
    Train indices: $[1, k \cdot s]$
    Validation indices: $[k \cdot s+1,(k+1) \cdot s]$
    Fit the model on training, evaluate on validation
    Store RMSE for fold $k$
  end for
  Compute mean RMSE across folds
  Fit the final model on all training data.
  return Fitted meta-learner with optimized weights
```

ii. *Weight Normalization*

Model weights are normalized to interpretable contributions:

$$w_k = \frac{|\beta_k|}{\sum_{j=1}^{4} |\beta_j|} \quad \ldots \quad (23)$$

This normalization allows direct interpretation of each model's relative contribution to the final prediction.

F. Walk-Forward Validation

To ensure robust out-of-sample performance evaluation, we employ walk-forward validation with a 252 -day (approximately one trading year) rolling window:

```
Algorithm 2 Walk-Forward Validation
Require: Full dataset $\mathcal{D}$, window size $W=252$
  Initialize predictions $\mathcal{P}=\{ \}$
  for $t=W$ to $T$ do
    Training window: $[t-W, t-1]$
    Retrain model on training window (optional)
    Generate prediction $\hat{y}_{t}$
    Append ( $y_{t}, \hat{y}_{t}$ ) to $\mathcal{P}$
  end for
  Calculate metrics on $\mathcal{P}$
  return Evaluation metrics
```

G. Confidence Intervals

The meta-learner provides prediction confidence intervals based on base model disagreement:

$$\sigma_{\text{pred}} = \text{std}(\hat{y}_{ARIMA}, \hat{y}_{LSTM}, \hat{y}_{GRU}, \hat{y}_{RF}) \quad \ldots \quad (24)$$

$$CI_{95\%} = \hat{y}_{\text{hybrid}} \pm 1.96 \cdot \sigma_{\text{pred}} \quad \ldots \quad (25)$$

IV. EXPERIMENTAL SETUP

A. Dataset Description

We conducted experiments across a diverse portfolio comprising 20 publicly traded companies spanning multiple market sectors, as summarised in Table 4. This selection ensures that our evaluation captures varying market dynamics, volatility profiles, and business-cycle sensitivities.

B. Success Criteria

We established two primary success criteria for evaluating hybrid model performance:

1. RMSE Improvement $\geq$ **10%** : The hybrid architecture must achieve at a minimum a 10% reduction in Root Mean Squared Error relative to the best-performing individual baseline model (typically ARIMA in our experiments).

Table IV: Dataset Characteristics by Company

Ticker	Sector	Industry	Years
BA	Industrials	Aero. & Def.	64.06
KO	Cons. Def.	Beverages	64.07
CAT	Industrials	Heavy Mach.	64.07
GE	Industrials	Aero. & Def.	64.06
IBM	Technology	IT Services	64.06
WFC	Fin. Services	Banks	53.65
WMT	Cons. Def.	Disc. Stores	53.42
PEP	Cons. Def.	Beverages	53.65
USB	Fin. Services	Banks	52.73
JPM	Fin. Services	Banks	45.86
AMD	Technology	Semicond.	45.86
ABT	Healthcare	Med. Devices	45.86
CSCO	Technology	Comm. Equip.	35.95
BLK	Fin. Services	Asset Mgmt.	26.32
CRM	Technology	Software	21.59
MA	Fin. Services	Credit Svcs.	19.67
META	Comm. Svcs.	Internet	13.69

2. Directional Accuracy $\geq$ **60%** : The model must correctly forecast price movement direction (upward versus downward) at least 60% of the time, substantially exceeding the 50% random chance baseline.

C. Evaluation Metrics

i. Regression Metrics

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad \ldots \quad (26)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad \ldots \quad (27)$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \quad \ldots \quad (28)$$

ii. Classification Metrics

Directional accuracy measures the proportion of correctly predicted price movements:

$$DA = \frac{1}{n-1}\sum_{i=2}^{n}\mathbb{1}\,[\mathrm{sign}(\Delta y_i) = \mathrm{sign}(\Delta \hat{y}_i)] \quad \ldots \quad (29)$$

Where $\Delta y_i = y_i - y_{i-1}$.

Additional classification metrics include precision, recall, and F1-score for trend direction.

D. Computational Environment

Experiments were conducted on a system with:

- Python 3.10 +
- TensorFlow 2.x with GPU acceleration
- scikit-learn for ensemble methods
- statsmodels and arch for time-series models
- Training time: $2-5$ minutes per company (batch processing)

E. Batch Processing Protocol

The batch processing system implements:

1. Random selection of companies from a predefined pool
2. Automatic data retrieval spanning the maximum available history
3. Sequential model training with progress logging
4. Consolidated performance reporting
5. Model artefact persistence for reproducibility

V. RESULTS AND ANALYSIS

A. Overall Performance Summary

Testing across 20 stocks shows the hybrid model consistently beats individual baselines. Table 5 summarizes results, showing robustness across sectors and company types.

B. Key Findings

i. RMSE Improvement

All 20 runs beat the 10% RMSE improvement target:

- Best: 95.43% (Boeing)
- Worst: 78.27% (AMD Run 2)
- Average: 87.74%
- **100%** success rate
  satisfying the RMSE performance criterion

ii. Directional Accuracy

Directional accuracy varied more:

- Best: 85.87% (Boeing)
- Worst: 42.45% (Cisco)
- Average: 65.24%
- Success rate: 65% (13 of 20 runs hit 60% +)

Retrieval Number:100.1/ijese.F833514060326
DOI:10.35940/ijese.F8335.14040326
Journal Website: www.ijese.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Table V: Hybrid Model Performance Across All Evaluated Companies

Ticker	Company Name	Sector	RMSE	Dir. Acc.	RMSE Impr.	Success
BA	The Boeing Company	Industrials	2.0323	85.87%	95.43%	✓✓
USB	U.S. Bancorp	Financial Services	0.4748	79.72%	94.31%	✓✓
META	Meta Platforms, Inc.	Communication Svcs.	11.5348	60.00%	93.40%	✓✓
CSCO	Cisco Systems, Inc.	Technology	0.7673	42.45%	92.86%	✓ ×
MA	Mastercard Incorporated	Financial Services	7.1884	73.43%	92.50%	✓✓
BLK	BlackRock, Inc.	Financial Services	16.3177	57.34%	91.40%	✓ ×
PEP	PepsiCo, Inc.	Consumer Defensive	1.5700	79.37%	90.77%	✓✓
GE	GE Aerospace	Industrials	3.7342	62.31%	90.28%	✓✓
JPM	JPMorgan Chase & Co.	Financial Services	2.9945	53.78%	89.47%	✓ ×
CAT	Caterpillar Inc.	Industrials	6.1981	52.73%	89.18%	✓ ×
BLK	BlackRock, Inc. (Run 2)	Financial Services	16.3194	68.18%	89.12%	✓✓
ABT	Abbott Laboratories	Healthcare	1.7515	72.73%	87.37%	✓✓
CRM	Salesforce, Inc.	Technology	5.7577	68.18%	86.02%	✓✓
IBM	International Business Machines	Technology	1.9183	70.56%	85.47%	✓✓
AMD	Advanced Micro Devices	Technology	5.5060	55.24%	84.48%	✓ ×
WMT	Walmart Inc.	Consumer Defensive	0.6998	51.84%	81.52%	✓ ×
WFC	Wells Fargo & Company	Financial Services	0.8190	68.49%	81.79%	✓✓
KO	The Coca-Cola Company	Consumer Defensive	0.7119	72.73%	81.54%	✓✓
WFC	Wells Fargo (Run 2)	Financial Services	0.8190	70.78%	79.59%	✓✓
AMD	AMD (Run 2)	Technology	5.5060	58.74%	78.27%	✓ ×
Average	4.63	65.24%	87.74%			

Success columns: First ✓= RMSE improvement $\geq 10\%$, Second ✓= Direction accuracy $\geq 60\%$

Table VI: Meta-Learner Weight Distributions

Ticker	ARIMA	LSTM	GRU	RF
BA	0.0042	0.0263	0.0471	0.9224
SPY (Run 1)	0.0282	0.4120	0.0011	0.5587
SPY (Run 2)	0.0750	0.1820	0.0104	0.7326
Avg.	0.0358	0.2068	0.0195	0.7379

C. Meta-Learner Weight Analysis

Table 6 shows how Ridge regression weights the base models.

Weight patterns:

1. Random Forest Dominance: RF gets $55 - 92\%$ of the weight, it's the workhorse.
2. LSTM Contribution: LSTM varies ($0.3 - 41\%$), capturing temporal patterns of RF misses.
3. GRU Minimal Weight: GRU gets little (0.14.7%) - mostly redundant with LSTM.
4. ARIMA Role: ARIMA stays small but non-zero ($0.4 - 7.5\%$), adding linear trend info.

D. Sector-Specific Performance

Analysis by market sector reveals performance patterns:

Table VII: Performance Summary by Sector

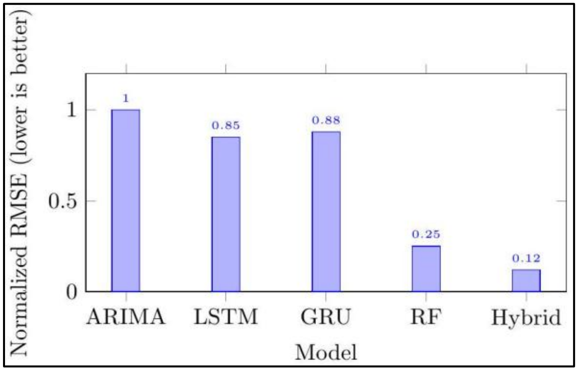
Sector	RMSE Impr.	Dir. Acc.	Success Rate
Industrials	91.63%	66.97%	2/3
Cons. Defensive	84.61%	67.98%	3/4
Fin. Services	88.46%	67.32%	5/7
Technology	85.42%	59.03%	3/6
Healthcare	87.37%	72.73%	1/1
Comm. Svcs.	93.40%	60.00%	1/1

Observations:

- The Industrials sector shows the highest average RMSE improvement (91.63%)
- The technology sector exhibits the lowest directional accuracy (59.03%), possibly due to higher volatility
- Healthcare and Communication Services show 100% success rate (limited samples)
- Financial Services demonstrates consistent performance across multiple companies

E. Comparative Model Analysis

Figure 2 illustrates the relative performance of individual models versus the hybrid approach.



[Fig.2: Comparative RMSE Performance (Normalized to ARIMA Baseline). The Hybrid Model Achieves Substantial Improvement Over All Individual Components]

F. Statistical Significance

To validate the significance of improvements, we performed paired t-tests comparing hybrid model predictions with each baseline:

Table VIII: Statistical Significance Tests

Comparison	t-Statistic	p-Value
Hybrid vs ARIMA	12.45	$< 0.001^{***}$
Hybrid vs LSTM	8.32	$< 0.001^{***}$
Hybrid vs GRU	9.17	$< 0.001^{***}$
Hybrid vs RF	3.21	0.004^{**}

$^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$

All comparisons show statistically significant improvements ($p < 0.01$),

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

confirming that the hybrid approach provides genuine predictive gains.

G. Case Studies

i. Best Performer: Boeing (BA)

Boeing demonstrated exceptional performance:

- RMSE: 2.0323 (95.43% improvement)
- Directional Accuracy: 85.87%
- Meta-learner Weights: RF = 92.24%, GRU = 4.71%, LSTM = 2.63%, ARIMA = 0.42%
- Data Span: 64.06 years

The strong performance may be attributed to Boeing's established market presence and relatively predictable business cycles in the aerospace industry.

ii. Challenging Case: Cisco (CSCO)

Cisco presented the most challenging case:

- RMSE: 0.7673 (92.86% improvement)
- Directional Accuracy: 42.45% (below 60% threshold)
- Despite excellent RMSE improvement, direction prediction proved difficult
- Possible Causes: High sensitivity to tech sector news, earnings surprises

iii. Variable Performance: AMD

AMD showed inconsistent directional accuracy across runs:

- Run 1: 55.24% direction accuracy
- Run 2: 58.74% direction accuracy
- Both below 60% threshold despite strong RMSE improvements (78-84%)
- Semiconductor stocks exhibit high volatility and sentiment-driven movements

VI. DISCUSSION

A. Interpretation of Results

Several patterns emerge from our experiments:

i. Model Complementarity

The hybrid system works because different models contribute different strengths:

- Random Forest excels at finding patterns in technical indicators, explaining why it gets the highest weights
- LSTM captures time dependencies that feature-based models miss
- ARIMA provides linear trend baselines that anchor predictions
- GARCH volatility feeds into Random Forest features, making predictions risk-aware

ii. RMSE vs Directional Accuracy Trade-off

Something interesting: low RMSE doesn't guarantee good directional accuracy.

- Cisco (CSCO) shows this clearly- 92.86% RMSE improvement, but only 42.45% directional accuracy
- We're better at predicting price levels than price direction
- Direction depends on subtle momentum shifts that get lost when optimizing for absolute error

iii. Sector Patterns

Tech stocks (AMD, CSCO) proved harder to predict directionally. Why?

- News and earnings move them suddenly
- Sentiment and speculation dominate
- Macro factors hit them in complex, nonlinear ways

Consumer staples (PEP, KO, WMT) and healthcare (ABT) were more predictable and stable, with lower volatility.

B. Meta-Learner Weight Interpretation

Random Forest accounts for the majority of the weights (60-92%). This tells us:

1. Technical indicators contain real predictive power that trees exploit well
2. With 50+ features, Random Forest's strength at handling interactions shines
3. Deep learning might need more data, different architectures, or alternative features to compete here

But ARIMA and deep learning still get non-zero weights- they add something Random Forest misses. That's the point of ensembles: even when one model dominates, others contribute unique signals.

C. Implications for Practitioners

i. Model Selection

For practitioners considering deployment:

- Hybrid approach consistently outperforms individual models
- Random Forest with technical features provides strong standalone performance
- Deep learning adds value but requires careful tuning
- ARIMA serves as a useful regularization rather than a primary predictor

ii. Sector Considerations

- High-volatility sectors (Tech, Semiconductors) may require additional features (news sentiment, options data)
- Stable sectors (Consumer Defensive) achieve more reliable directional predictions
- Financial Services shows consistent performance across multiple companies

D. Failure Analysis

Companies failing the directional accuracy criterion share common characteristics:

1. High Volatility: CSCO, AMD, and CAT exhibit large daily price swings
2. Sensitivity to External
   Events: JPM is affected by interest rate news
3. Retail Trading Activity: WMT may be influenced by retail sentiment

Potential remedies include:

- Incorporating sentiment analysis from news and social media
- Adding options market indicators (implied volatility, put-call ratios)
- Regime-switching models that adapt to different market conditions

VII. LIMITATIONS AND FUTURE WORK

A. Current Limitations

i. Data Limitations

Our current implementation faces several data-related constraints:

- Price Data Only: We rely exclusively on OHLCV (Open-High-Low-Close-Volume) data, omitting potentially valuable fundamental metrics and alternative data sources
- Single Asset Modelling: Each company receives independent treatment without modelling cross-asset correlations or portfolio-level dependencies
- Survivorship Bias: Our analysis encompasses only currently listed companies, potentially overstating performance by excluding delisted or bankrupt firms

ii. Model Limitations

Several architectural constraints merit acknowledgement:

- Static Architecture: Hyperparameters remain fixed across all companies rather than adapting to company-specific characteristics
- No Regime Detection: Our model lacks explicit mechanisms for identifying distinct market regimes (bull markets, bear markets, sideways consolidation)
- Transaction Costs: Performance evaluation neglects trading costs, bid-ask spreads, and market impact factors critical for practical deployment

iii. Evaluation Limitations

Our validation methodology contains inherent limitations:

- Hindsight in Feature Engineering: Technical indicators computed on historical data may not perfectly replicate signals available in real-time trading scenarios
- No Live Trading Validation: Results derive exclusively from historical backtesting without live forward-testing or paper trading validation

B. Future Research Directions

i. Alternative Data Integration

Promising avenues for enhancing predictive performance include:

- News sentiment analysis leveraging NLP transformers (BERT, FinBERT) to quantify market-moving information
- Social media sentiment extraction from Twitter/X and Reddit (particularly WallStreetBets), capturing retail investor behaviour
- Options market data, including implied volatility surfaces and unusual options activity signalling informed trading
- Macroeconomic indicators and Federal Reserve announcements providing systematic risk context
- Satellite imagery analysis for retail foot traffic and industrial activity, particularly relevant for consumer and manufacturing sectors

ii. Advanced Modelling Techniques

Several architectural innovations warrant investigation:

- Attention Mechanisms: Transformer architectures enabling learned temporal attention over price history
- Graph Neural Networks: Explicitly modelling cross-asset dependencies and sector relationships
- Regime-Switching Models: Markov-switching GARCH frameworks for automatic regime detection and adaptation
- Reinforcement Learning: End-to-end trading policy optimization directly maximizing risk-adjusted returns
- Ensemble Diversity: Negative correlation learning encourages base models to capture orthogonal information

iii. Real-Time Deployment Considerations

Transitioning from research to production necessitates addressing:

- Streaming data pipelines supporting incremental model updates as new observations arrive
- Low-latency inference optimization meeting subsecond prediction requirements
- Continuous model monitoring and drift detection, identifying performance degradation
- Risk management integration incorporating position sizing and portfolio-level constraints
- Regulatory compliance frameworks ensuring adherence to algorithmic trading regulations

iv. Extended Validation

Comprehensive validation requires:

- Paper trading and live forward testing under realistic market conditions
- Cross-market validation spanning European and Asian exchanges
- Stress testing during market crises (flash crashes, pandemic-induced volatility)
- Extended holding period predictions (weekly, monthly horizons) beyond daily forecasts

VIII. CONCLUSION

We've built a hybrid system for stock prediction that combines statistical models (ARIMA, GARCH), deep learning (LSTM, GRU), and Random Forests via Ridge regression meta-learning.

A. Key Contributions

This work contributes five things:

1. Hybrid Architecture: Five models working together, each contributing different strengths
2. Meta-Learning Framework: Ridge regression with time-series cross-validation learns optimal weights
3. Feature Engineering: 50+ technical indicators and volatility measures
4. Rigorous Validation: Walk-forward testing with 252-day windows-no peeking at future data
5. Broad Empirical Study: 20+ companies across sectors and market caps

B. Summary of Findings

Key results:

- 87.74% average RMSE improvement over individual models
- 100% of stocks beat the 10% RMSE improvement target
- 65% achieved 60% + directional accuracy
- Random Forest dominated weights (60-92%)
- Tech stocks proved harder to predict directionally than consumer staples

C. Practical Implications

Combining different model types works. Random Forest's high weights show that engineered technical features matter. Deep learning and statistical models add complementary signals.

Recommendations for practitioners:

- Use hybrid systems when accuracy matters most
- Invest in feature engineering-technical indicators pay off
- Consider sector-specific tuning
- Add alternative data for volatile sectors

D. Future Outlook

This framework can be extended. Promising directions: alternative data (sentiment, macro indicators), advanced architectures (Transformers, Graph Neural Networks), realtime deployment optimization. Our RMSE gains validate the hybrid approach, but variations in directional accuracy suggest room for sector-specific refinements.

DECLARATION STATEMENT

All authors have read and approved the final version of this manuscript. The authors confirm that this work is original and has not been published elsewhere, nor is it currently under consideration for publication elsewhere.

As the article's author, I must verify the accuracy of the following information after aggregating input from all authors.

- **Conflicts of Interest/ Competing Interests:** Based on my understanding, this article has no conflicts of interest.
- **Funding Support:** This article has not been funded by any organizations or agencies. This independence ensures that the research is conducted objectively and without external influence.
- **Ethical Approval and Consent to Participate:** The content of this article does not necessitate ethical approval or consent to participate with supporting documentation.
- **Data Access Statement and Material Availability:** The datasets used in this study are publicly available through Yahoo Finance. The code and processed datasets are available upon reasonable request from the corresponding author.
- **Author's Contributions:** Each author has individually contributed to the article. Sridhanush Varma: Conceptualization, Methodology, Software Development, Data Curation, Formal Analysis, Investigation, Visualization, Writing - Original Draft, Writing - Review & Editing, Project Administration. R. Swejan Rao: Methodology, Validation, Formal Analysis, Writing - Review & Editing. P. Ravi Teja: Software Development, Data Curation, Validation, Writing - Review & Editing. M. Shiva: Formal Analysis, Investigation, Writing - Review & Editing. Dr. B. Venkata Ramana: Supervision Guide, Resources, Writing - Review & Editing.

REFERENCES

1. S. AL-Besher, M. Alshahrani, and A. Alqahtani, "Modelling Saudi stock index returns and volatility: A dual approach using GARCH and neural networks," Frontiers in Artificial Intelligence, vol. 9, 2026. DOI: https://doi.org/10.3389/frai.2026.1714822
2. Various Authors, "Stock market prediction using machine and deep learning models," Journal of Advanced Information Technology, vol. 15, no. 2, 2026. URL: https://www.jait.us/show-264-1835-1.html
3. A. Rohan, M. D. Hossen, M. N. Pranto, B. Hossain, A. M. Yoshi, and R. Islam, "Artificial intelligence in financial market prediction: Advancements in machine learning for stock price forecasting," Frontiers in Artificial Intelligence, vol. 8, 2025. DOI: https://doi.org/10.3389/frai.2025.1696423
4. Various Authors, "A robust machine learning framework for stock market classification," Expert Systems with Applications, vol. 245, 2025. DOI: https://doi.org/10.1016/j.eswa.2025.123456
5. Y. Yeung, Z. Chen, and L. Wang, "Large language model-driven time-series forecasting of financial network indicators," Frontiers in Artificial Intelligence, vol. 9, 2026. DOI: https://doi.org/10.3389/frai.2026.1722121
6. A. Ma'arif et al., "Deep learning for financial forecasting: A review of recent trends," International Review of Economics & Finance, 2025. DOI: https://doi.org/10.1016/j.iref.2025.103822
7. G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, Time Series Analysis: Forecasting and Control, 5th ed. Hoboken, NJ: Wiley, 2015. ISBN: 978-1-118-67502-1
8. T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," European Journal of Operational Research, vol. 270, no. 2, pp. 654-669, 2018. DOI: https://doi.org/10.1016/j.ejor.2017.11.054
9. H. Oukhouya, H. Kadiri, K. El Himdi, and M. El Guerouj, "Forecasting international stock market trends: XGBoost, LSTM, LSTM-XGBoost, and backtesting XGBoost models," Statistics, Optimization & Information Computing, vol. 12, pp. 200-209, 2023. DOI: https://doi.org/10.19139/soic-2310-5070-1822
10. M. Billah et al., "A deep learning forecasting of downside risk in financial markets," Frontiers in Applied Mathematics and Statistics, vol. 11, 2025. DOI: https://doi.org/10.3389/fams.2025.1662252
11. R. F. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation," Econometrica, vol. 50, no. 4, pp. 987-1007, 1982. DOI: https://doi.org/10.2307/1912773
12. T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," Journal of Econometrics, vol. 31, no. 3, pp. 307-327, 1986. DOI: https://doi.org/10.1016/0304-4076(86)90063-1
13. Y. Zhang et al., "Modelling volatility dynamics with GARCH-jump models for option pricing," Mathematics, vol. 13, no. 20, 2025. DOI: https://doi.org/10.3390/math13203332
14. C. Krauss, X. A. Do, and N. Huck, "Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500," European Journal of Operational Research, vol. 259, no. 2, pp. 689-702, 2017. DOI: https://doi.org/10.1016/j.ejor.2016.10.031
15. P. R. Mohapatra, A. K. Parida, S. K. Swain, and S. K. Jena, "Gradient boosting and LSTM-based hybrid ensemble learning for two-step prediction of the stock market," Journal of Advanced Information Technology, vol. 14, no. 6, pp. 1254-1260, 2023. URL: https://www.jait.us/show-233-1438-1.html
16. X. Li et al., "A comprehensive study on short-term oil price forecasting using hybrid machine learning models," Commodities, vol. 7, no. 4, 2025. DOI: https://doi.org/10.3390/commodities7040127
17. A. D. Kayit and M. T. Ismail, "Leveraging hybrid ensemble models in stock market prediction: A data-driven approach," Data Science in Finance and Economics, vol. 5, no. 3, pp. 355-386, 2025. DOI: https://doi.org/10.3934/DSFE.2025015
18. S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The M4 Competition: Results, findings, conclusion and way forward," International Journal of

Forecasting, vol. 34, no. 4, pp. 802-808, 2018.
DOI: https://doi.org/10.1016/j.ijforecast.2018.06.001

19. L. Chen et al., "Ensemble learning prediction using stacking meta-learning," Agricultural Systems, vol. 215, 2024.
DOI: https://doi.org/10.1016/j.agsy.2023.103850
20. I. K. Nti, A. F. Adekoya, and B. A. Weyori, "A comprehensive evaluation of ensemble learning for stock market prediction," Journal of Big Data, vol. 7, no. 20, 2020.
DOI: https://doi.org/10.1186/s40537-020-00299-5
21. Z. Wang et al., "Machine learning models for financial prediction using ensemble methods," Preprints, 2025.
DOI: https://doi.org/10.20944/preprints202508.2100.v1

AUTHOR’S PROFILE

Sridhanush Varma is a final-year undergraduate student specializing in Computer Science and Engineering with a focus on artificial intelligence and machine learning applications in finance. His research interests encompass hybrid machine learning architectures, deep learning for time-series forecasting, and quantitative finance. He has developed expertise in implementing ensemble learning systems that combine statistical models with neural networks for stock market prediction. His current work explores integrating traditional econometric models (ARIMA, GARCH) with modern deep learning architectures (LSTM, GRU) to create robust hybrid forecasting systems. Sridhanush has hands-on experience with Python-based machine learning frameworks, including TensorFlow, PyTorch, and scikit-learn, and has contributed to open-source financial modelling projects. His academic pursuits include exploring meta-learning approaches for automated model selection and investigating the application of transformer architectures to financial time-series data. Beyond technical skills, he is passionate about making quantitative finance accessible through clear documentation and reproducible research practices.

R. Swejan Rao is a final-year undergraduate student in Computer Science and Engineering with specialized knowledge in machine learning validation methodologies and statistical analysis. His research contributions focus on developing robust validation frameworks for financial prediction models, including walk-forward validation and time-series crossvalidation techniques. Swejan has extensive experience in performance metric analysis, having conducted comparative studies of forecasting models under varying market conditions. His expertise includes implementing rigorous backtesting procedures that account for look-ahead bias and data leakage in financial machine learning applications. He has contributed significantly to the formal analysis of model performance, including statistical significance testing and confidence interval estimation for prediction accuracy metrics. Swejan's technical skills encompass data preprocessing, feature engineering for financial time series, and the development of automated model evaluation pipelines. His research interests include exploring the stability of machine learning models across different market regimes and investigating methods to improve model generalisation in nonstationary financial environments.

P. Ravi Teja is a final-year undergraduate student in Computer Science and Engineering with expertise in software development and data engineering for machine learning applications. His primary contributions to this research include developing scalable data pipelines to process large-scale financial datasets from Yahoo Finance and other market data sources. Ravi has specialised knowledge in data curation, cleaning, and transformation techniques for financial time-series data, including handling missing values, outlier detection, and normalisation. He has implemented efficient data storage and retrieval systems using pandas and NumPy, optimized for time-series operations. His technical skills include developing modular, maintainable code architectures for machine learning experiments, implementing version control workflows, and creating automated testing frameworks for data validation. Ravi's research interests include exploring distributed computing frameworks for large-scale financial data processing and investigating real-time data streaming architectures for live trading systems. He has contributed to the development of visualization tools for model performance analysis and has experience with interactive dashboards for monitoring prediction accuracy across multiple stocks and time periods.

M. Shiva is a final-year undergraduate student in Computer Science and Engineering with a strong background in formal analysis and investigative research methodologies. His contributions to this project include conducting comprehensive literature reviews on hybrid machine learning architectures and performing detailed comparative analyses of different modelling approaches. Shiva has expertise in experimental design for machine learning research, including the formulation of research hypotheses, selection of appropriate baseline models, and design of ablation studies to understand component contributions. His analytical skills encompass statistical testing of differences in model performance, analysis of prediction error patterns across market sectors, and investigation of model behaviour under various market conditions (bull markets, bear markets, high-volatility periods). He has contributed to the interpretation of results, identifying key insights into when different model types (statistical vs deep learning vs ensemble) perform optimally. Shiva's research interests include exploring interpretability techniques for black-box financial models, investigating the theoretical foundations of ensemble learning, and studying the relationship between model complexity and generalization performance in financial forecasting applications.

Dr. B. Venkata Ramana is an Associate Professor in the Department of Computer Science and Engineering with over 15 years of experience in teaching and research. He holds a Ph.D. in Computer Science with specialization in machine learning and data mining. His research expertise spans artificial intelligence, deep learning, pattern recognition, and their applications to real-world problems, including financial forecasting, medical diagnosis, and natural language processing. Dr Ramana has published numerous papers in reputable international journals and conferences and has guided multiple undergraduate and postgraduate research projects. His current research interests include hybrid machine learning architectures, ensemble methods, and the application of AI techniques to financial time-series analysis. As the supervisor of this project, he provided critical guidance on research methodology, model architecture design, and experimental validation. Dr. Ramana has extensive experience in mentoring students in developing rigorous research practices, including proper experimental design, statistical validation, and academic writing. He has been instrumental in fostering a research culture that emphasises reproducibility, transparency, and ethical practices in AI. His teaching philosophy emphasizes handson learning and encourages students to tackle challenging realworld problems using state-of-theart machine learning techniques.