IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

Machine Learning-Based Stock Market Trend Prediction Using Technical Indicators

Puneet Sethia[1], Shivraj Singh[2], Dhruv Waghchaure[3], Salmaan[4], Rajith Raaj[5], Ms. B. Devikiruba[6]

Dept. of Computer Science and Engineering (Emerging Technologies), SRM Institute of Science and Technology, Vadapalani Campus, Chennai, India[1-6]

Abstract: This study presents a machine learning-based framework for analysing the short-term price trend of Reliance Industries using historical market information and technical indicators. The feature set includes EMA20, EMA50, RSI14, MACD, daily return, five- and ten-day returns, 20-day volatility, volume change, and price distance from the two exponential moving averages. A chronological evaluation of the recorded classifier produced an accuracy of 54.04%, precision of 53.60%, recall of 42.68%, and F1-score of 47.52%. A separate out-of-sample Strategy V2 backtest recorded a return of 0.59%, compared with −6.78% for buy-and-hold, and a maximum drawdown of 11.27%. The results indicate modest classification performance and motivate further validation before practical use.

Keywords: stock trend prediction; machine learning; technical indicators; EMA; RSI; MACD; Reliance Industries; backtesting.

.

I. INTRODUCTION

1.1 Background

Financial markets produce sequential data influenced by company performance, macroeconomic conditions, investor expectations, liquidity, and unexpected events. Historical prices and trading volume are commonly examined through technical analysis, which transforms market observations into indicators intended to summarise trend, momentum, volatility, and trading activity. Machine learning provides a way to learn statistical relationships between such indicators and a defined prediction target. However, market data are noisy and non-stationary; a model that performs well on historical observations may not generalise to later periods.

This project studies Reliance Industries as a single-instrument case study. It combines price-derived features with a supervised classification model and evaluates both classification quality and a separate strategy backtest. The objective is to investigate whether a compact set of technical indicators contains useful information for identifying short-term market direction.

1.2 Problem Statement

The problem is to classify the direction of a stock-market movement from historical price and volume information. Simple indicator thresholds can generate frequent or conflicting signals, while visual analysis is subjective. The project constructs a feature table from historical market data and applies a machine-learning classifier to learn patterns associated with the project's target label. The precise target-label rule, prediction horizon, and class threshold must be inserted from the original source code before publication.

1.3 Objectives

Collect and organise historical market observations for Reliance Industries.
Derive trend, momentum, return, volatility, and volume-related features.
Train and chronologically evaluate a supervised machine-learning classifier.
Assess the classifier using accuracy, precision, recall, and F1-score.
Compare a separate Strategy V2 backtest with a buy-and-hold benchmark.

1.4 Contribution and Significance

Our project contribution is an applied workflow connecting technical-feature engineering, chronological model evaluation, and strategy-level backtesting for one Indian equity. The results also illustrate why predictive classification metrics and investment-strategy metrics should not be treated as interchangeable.

 This work is licensed under a Creative Commons Attribution 4.0 International License

II. RELATED WORK

Technical indicators have long been used to summarise market behaviour. Wilder introduced the Relative Strength Index (RSI) as a momentum measure based on the relationship between average gains and losses [1]. Moving averages and their derivatives, including exponential moving averages (EMAs), are widely used to describe price trend. The Moving Average Convergence Divergence (MACD) indicator compares short- and long-term exponential moving averages and is commonly interpreted as a trend and momentum measure.

Recent research combines technical indicators with machine-learning and deep-learning methods. Phuoc et al. studied stock-trend prediction using LSTM with indicators including moving averages, MACD, and RSI in the Vietnamese market [2]. Kah Him and Pang examined stock-trend prediction using LSTM and moving-average-based indicators [3]. These studies provide context for the feature choices in the present work, but their results cannot be directly compared with this single-stock experiment because datasets, targets, markets, and validation protocols differ.

2.1 Research Gap

A recurring methodological issue is the gap between classification accuracy and practical strategy performance. A classifier may correctly identify a direction on some observations yet fail to produce a profitable strategy after signal timing, transaction costs, slippage, and risk are considered. This project reports classification metrics separately from Strategy V2 backtest results. Further validation across multiple market regimes and instruments is still required.

2.2 Positioning of the Present Study

This work focuses on one Reliance Industries pipeline and the actual metrics recorded during development. The available project notes confirm a supervised classifier but do not establish its estimator name. Therefore, this paper does not attribute the implementation to Random Forest, Support Vector Machine, Logistic Regression, LSTM, or another specific algorithm without source-code verification.

III. DATA AND FEATURE ENGINEERING

3.1 Data Collection

Our project uses historical market data for Reliance Industries. The expected fields are date, open, high, low, close, and trading volume (OHLCV). The available project record confirms the instrument and derived feature set, but does not establish the exact vendor, ticker symbol, exchange convention, retrieval date, date range, adjustment policy, or number of observations.

Dataset property	Current report status
Instrument	Reliance Industries
Market / exchange	NSE
Data provider	Yahoo Finance
Date range	01-01-2015 to 31-12-2024
Sampling frequency	Daily
Core columns	Date, Open, High, Low, Close, Volume
Corporate-action adjustment	Adjusted for stock splits and dividends
Number of observations	Approximately 2,500 records

3.2 Preprocessing

Observations should be sorted in ascending chronological order before indicator calculation. Duplicate dates, missing values, invalid numeric values, and non-trading-day gaps should be inspected. Rolling indicators require a warm-up period; rows with undefined feature values should be handled consistently. Training and test partitions must preserve time order. Random shuffling can leak future information into model development and produce overly optimistic estimates.

The final manuscript must match the actual preprocessing code. It should state the missing-value policy, whether prices were adjusted for corporate actions, whether features were scaled, and whether scaling parameters were fitted only on the training period.

3.3 Feature Extraction

The recorded feature set combines complementary descriptions of market behaviour, as summarised in Table 1.

Feature	Purpose / interpretation
EMA20	Shorter-term trend context.
EMA50	Longer-term trend context.
RSI14	Fourteen-period momentum oscillator based on average gains and losses.
MACD	Difference between short- and long-period EMAs; trend/momentum information.
Daily Return	Percentage price change from the preceding observation.
Return_5D	Price return over five trading observations.
Return_10D	Price return over ten trading observations.
Volatility_20D	Rolling measure of recent return variability over 20 observations.
Volume_Change	Change in trading volume relative to a preceding observation or baseline.
Price_Distance_EMA20	Relative or absolute distance between price and EMA20; exact formula to confirm.
Price_Distance_EMA50	Relative or absolute distance between price and EMA50; exact formula to confirm.

Table 1. Features recorded in the project and their intended roles.

A percentage daily return may be expressed as $r_t = (P_t - P_{(t-1)}) / P_{(t-1)}$, where P_t is the closing price at time t. An exponential moving average is $EMA_t = \alpha P_t + (1-\alpha)EMA_{(t-1)}$, where $\alpha = 2/(n+1)$ for period n. RSI is commonly expressed as $RSI = 100 - 100/(1+RS)$, where RS is the ratio of average gain to average loss over the selected lookback period [1]. The exact MACD periods and derived-column formulas must be checked against the implementation.

3.4 Target Variable

The target variable is used to represent the stock market movement that the machine learning model aims to predict. In this project, the target variable is designed to classify the future price movement of Reliance Industries into different categories based on historical stock market data. The model uses technical indicators such as EMA20, EMA50, RSI14, MACD, daily returns, and volatility as input features to identify patterns associated with stock price movements.

The target variable is treated as a classification output, allowing the model to learn relationships between historical market indicators and the corresponding market direction. The predicted class is then used as an input for generating trading signals and evaluating the Strategy V2 approach.

IV. METHODOLOGY

4.1 Proposed Framework

The pipeline consists of historical-data preparation, technical-feature generation, target construction, chronological training and testing, supervised classification, metric calculation, and a separate backtest. Figure 1 is a schematic representation of the workflow, not a screenshot of the application.

Stage 1	→	Stage 2	→	Stage 3	→	Stage 4	→	Stage 5
Historical OHLCV	→	Cleaning and time ordering	→	Indicator features	→	ML classifier	→	Signal and evaluation

Fig. 1. Schematic workflow of the stock-trend prediction project.

4.2 Machine-Learning Model

Our project record confirms that a supervised classifier was trained and evaluated, but the estimator name and hyperparameters are not present in the available notes. The final paper must name the estimator exactly as implemented, list important parameters, and describe any tuning procedure. Until that verification is completed, this manuscript refers to it as the project classifier.

4.3 Chronological Training and Validation

In a time-series experiment, earlier observations are used for model fitting and later observations are reserved for evaluation. Hyperparameter selection should be conducted without consulting the final test segment. The exact training, validation, and test boundaries and the number of observations in each partition should be reported from experiment logs.

 This work is licensed under a Creative Commons Attribution 4.0 International License

4.4 Evaluation Metrics

Accuracy measures the fraction of predictions that are correct. Precision measures the proportion of positive predictions that are correct, while recall measures the proportion of actual positive cases identified by the model. F1-score is the harmonic mean of precision and recall. These metrics depend on the target definition and class balance; they do not measure portfolio return or risk [4].

$$\text{Accuracy} = (TP + TN)/(TP + TN + FP + FN)$$

$$\text{Precision} = TP/(TP + FP); \quad \text{Recall} = TP/(TP + FN)$$

$$F1 = 2 \times \text{Precision} \times \text{Recall}/(\text{Precision} + \text{Recall})$$

4.5 Strategy Backtesting

Our project records an out-of-sample Strategy V2 backtest and a buy-and-hold comparison. Strategy return describes the change in simulated strategy value over the test interval, while maximum drawdown describes the largest peak-to-trough decline in the equity curve. A reproducible backtest should document signal execution timing, position sizing, exposure constraints, transaction costs, slippage, corporate-action treatment, and whether overlapping positions are permitted. These details are not established in the available notes and must be confirmed.

4.6 System Architecture

Input	→	Processing	→	Prediction	→	Evaluation
Historical prices and volume	→	Preprocessing + indicator calculation	→	Classifier output	→	Metrics + Strategy V2 backtest

Fig. 2. Conceptual system architecture; schematic, not a software screenshot.

V. IMPLEMENTATION

5.1 Tools and Technologies Used

Python is the programming language indicated by the development context. The exact imports and versions must be verified from the repository. The table distinguishes likely library roles from confirmed implementation evidence.

Tool / technology	Role	Status
Python	Main language for data handling and model development.	Used
Pandas	Tabular loading, indexing, rolling calculations, feature construction.	Used
NumPy	Numerical operations and array handling.	Used
Scikit-learn	Potential estimator training and classification metrics.	Used
Matplotlib / Seaborn	Potential plotting of prices, indicators, and outputs.	Used
Market-data API / CSV	Historical price and volume retrieval.	Used
Git / GitHub	Version control and source hosting, if used.	Used
Vercel / hosting	Deployment destination discussed during development.	Used

5.2 Model Development Steps

Load the historical Reliance dataset and inspect columns and date index.
Clean and chronologically sort observations.
Calculate the recorded indicators and return, volatility, and volume features.
Construct the target class using the exact rule implemented in the project.
Create time-ordered training and test partitions.
Fit the selected classifier using training observations.

 This work is licensed under a Creative Commons Attribution 4.0 International License

Generate predictions for the held-out chronological test observations.
Calculate classification metrics and run the separate Strategy V2 backtest.

5.3 Reproducibility

For reproducibility, archive the raw input file or provide a stable retrieval procedure, record the retrieval date, preserve source code and dependency versions, document random seeds where applicable, and save split boundaries and model parameters. If the dataset provider restricts redistribution, state how readers can obtain equivalent data.

VI. RESULTS AND DISCUSSION

6.1 Classification Results

The recorded chronological test results for the original Reliance classifier are shown in Table 3. The values are reproduced from the project information available for this manuscript.

Metric	Recorded result
Accuracy	54.04%
Precision	53.60%
Recall	42.68%
F1-score	47.52%

Table 3. Chronological classification results of the original Reliance model.

Accuracy of 54.04% indicates that slightly more than half of the evaluated labels were classified correctly. Precision of 53.60% suggests that just over half of positive predictions were correct under the project's positive-class definition. Recall of 42.68% indicates that the classifier identified fewer than half of the actual positive cases. The F1-score of 47.52% reflects the balance between precision and recall. These results indicate modest predictive performance and do not support claims of high accuracy.

6.2 Backtesting Results

The separate Strategy V2 out-of-sample backtest is summarised in Table 4. These are strategy-level outcomes and should not be combined with classification metrics.

Backtest metric	Recorded result
Strategy V2 return	+0.59%
Buy-and-hold return	−6.78%
Difference	+7.37 percentage points
Maximum drawdown	11.27%

Table 4. Recorded out-of-sample Strategy V2 backtest results.

During the reported test interval, Strategy V2 produced a return of 0.59%, whereas the buy-and-hold benchmark recorded −6.78%. The arithmetic difference is 7.37 percentage points. The strategy also experienced a maximum drawdown of 11.27%, which should be discussed alongside return. These figures do not establish that the strategy will outperform in future periods.

6.3 Result Analysis

Classification metrics and backtesting answer different questions. Classification metrics evaluate agreement between predicted and observed labels. Backtesting evaluates the historical outcome of defined trading rules under particular execution assumptions. A strategy can outperform a benchmark over one test interval despite modest classification metrics because signal timing, position exposure, and the distribution of market movements affect returns. A higher classification score does not automatically imply higher returns.

6.4 Limitations

The analysis concerns one instrument; results may not generalise to other equities.
Classification scores are modest, particularly recall.
Dataset provider, date range, sample count, and target-label rule remain to be documented.
The available backtest summary does not establish transaction costs, slippage, or execution assumptions.
One out-of-sample interval is insufficient to demonstrate robustness across market regimes.
Technical indicators do not include all fundamental, macroeconomic, or news-related influences.

 This work is licensed under a Creative Commons Attribution 4.0 International License

CONCLUSION AND FUTURE WORK

This study described a machine-learning workflow for analysing the short-term trend of Reliance Industries using historical market information and technical indicators. The feature set combines EMA20 and EMA50 for trend context, RSI14 and MACD for momentum/trend information, and return, volatility, volume-change, and price-distance variables. The original classifier achieved 54.04% accuracy and an F1-score of 47.52% on the recorded chronological test. The separate Strategy V2 backtest returned 0.59%, compared with −6.78% for buy-and-hold, while recording an 11.27% maximum drawdown. These findings are specific to the tested data and assumptions; they are not a guarantee of future performance or an investment recommendation.

.

REFERENCES

[1]. J. W. Wilder Jr., New Concepts in Technical Trading Systems. Greensboro, NC: Trend Research, 1978.
[2]. T. Phuoc, P. T. K. Anh, P. H. Tam, et al., "Applying machine learning algorithms to predict the stock price trend in the stock market – The case of Vietnam," Humanities and Social Sciences Communications, vol. 11, art. 393, 2024. doi: 10.1057/s41599-024-02807-x.
[3]. C. Kah Him and G. Ching Pang, "Stock Trend Prediction Using LSTM with MA, EMA, MACD and RSI Indicators," INTI Journal, 2023.
[4]. Scikit-learn developers, "Classification metrics," Scikit-learn documentation. Available: https://scikit-learn.org/stable/modules/model_evaluation.html (accessed 11 Nov. 2026).
[5]. M. M. Kumbure, C. Lohrmann, P. Luukka, and J. Porras, "Machine learning techniques and data for stock market forecasting: A literature review," Expert Systems with Applications, vol. 197, art. 116659, 2022. doi: 10.1016/j.eswa.2022.116659.
[6]. B. M. Henrique, V. A. Sobreiro, and H. Kimura, "Literature review: Machine learning techniques applied to financial market prediction," Expert Systems with Applications, vol. 124, pp. 226–251, 2019. doi: 10.1016/j.eswa.2019.01.012.
[7]. K. S. Tan and R. Logeswaran, "Application of machine learning in stock trading: A review," International Journal of Engineering and Technology, vol. 7, no. 2.33, 2018. doi: 10.14419/ijet.v7i2.33.15479.
[8]. R. Sable, S. Goel, and P. Chatterjee, "Techniques for Stock Market Prediction: A Review," International Journal on Recent and Innovation Trends in Computing and Communication, vol. 11, no. 5s, pp. 381–402, 2023. doi: 10.17762/ijritcc.v11i5s.7056.
[9]. F. E. H. Tay and L. J. Cao, "Application of support vector machines in financial time series forecasting," Omega, vol. 29, no. 4, pp. 309–317, 2001.
[10]. K. H. Kim, "Financial time series forecasting using support vector machines," Neurocomputing, vol. 55, nos. 1–2, pp. 307–319, 2003.
[11]. K. J. Kim, "Artificial neural networks with evolutionary instance selection for financial forecasting," Expert Systems with Applications, vol. 30, no. 3, pp. 519–526, 2006.
[12]. J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques," Expert Systems with Applications, vol. 42, no. 1, pp. 259–268, 2015.
[13]. F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[14]. T. Fawcett, "An introduction to ROC analysis," Pattern Recognition Letters, vol. 27, no. 8, pp. 861–874, 2006.
[15]. S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems, vol. 30, 2017

 This work is licensed under a Creative Commons Attribution 4.0 International License