Dean&Francis

ISSN 2959-6130

Machine Learning Algorithms for Predicting ETF Directional Movements Using Technical Indicators and VIX

Xin Liu

The University of Sydney, Sydney, Australia
Xinliu.uni@gmail.com

Abstract:

Predicting exchange-traded funds (ETFs) is challenging due to their diversified portfolios, exposure to market volatility, and short-term noise. This study compares eight machine learning models—Logistic Regression (LR), Linear Discriminant Analysis (LDA), Support Vector Machine (SVM), Random Forest (RF), Gradient Boosting Decision Tree (GBDT), Extreme Gradient Boosting (XGBoost), Long Short-Term Memory (LSTM), and Naïve Bayes (NB)—for forecasting the direction of daily ETF returns. Two horizons, next-day (T+1) and five-day (T+5), are examined. The analysis also tests the incremental value of incorporating the CBOE Volatility Index (VIX). Results show that performance varies across ETFs and horizons: linear models and LSTM perform best on large-cap indices (SPY, DIA), while RF leads for the small-cap index (IWM). At T+1, top models reach 56–58% accuracy; at T+5, LSTM improves markedly to 64.5% on SPY and remains strongest on QQQ and DIA, while most other models decline. Overall, horizon effects are model-specific rather than uniformly positive, and adding VIX provides only marginal, statistically insignificant gains (<2%). The findings suggest that model selection should depend on ETF and horizon, while broad volatility indicators such as VIX offer limited value for short-term forecasts.

Keywords: Exchange-Traded Funds (ETFs); Machine Learning; Directional Prediction; Technical Indicators; Volatility Index (VIX)

1 Introduction

Exchange-traded funds (ETFs) have become integral to modern portfolio management. Their liquidity and popularity make forecasting ETF price movements a key challenge for both traders and researchers. Yet short-term ETF movements remain difficult to predict due to the complex interplay among index components, market microstructure, and external volatility shocks.

Recent advances in machine learning (ML) have renewed interest in financial forecasting. Studies show that ML and deep learning (DL) models can outperform traditional econometric approaches. Ayyildiz and Iskenderoglu report that artificial neural networks (ANNs) generally outperform competing models across G7 indices [1]. Strader et al. likewise find that ANNs tend to excel in regression tasks, whereas SVMs are more effective for classification [2]. Deep architectures such as long short-term memory (LSTM) networks capture nonlinear and temporal dependencies in financial data [3,4]. Other surveys indicate that supervised methods often outperform unsupervised or linear techniques [5,6]. Building on these findings, recent studies propose hybrid models achieving higher predictive accuracy [7,8].

Despite these advances, ETF-focused research remains limited. Most prior work examines individual equities or broad market indices, offering little systematic evidence on ETFs. Existing ETF studies, such as Piovezan et al. and Shih et al., are narrow in scope, focusing on specific markets or a few model types [9,10]. Although volatility indices such as VIX are widely recognized as sentiment measures, their value in ML-based ETF prediction remains underexplored. Econometric studies have shown strong links between VIX and ETF returns, but this relationship has yet to be integrated into ML frameworks [11]. The role of prediction horizon is also seldom addressed; few studies compare near-term (T+1) and multi-day (T+5) forecasts.

To address these gaps, this paper conducts a comparative study of eight ML models for short-term ETF direction prediction. Features are constructed from standard technical indicators, with and without the CBOE Volatility Index (VIX). Using four major U.S. ETFs over 2015–2024, the study benchmarks model performance within a unified framework, examines model–ETF–horizon interactions, and evaluates the incremental contribution of volatility through time-aware validation and formal statistical testing.

2. Experiment

2.1 Data

The dataset used in this study includes four major exchange-traded funds (ETFs) and the CBOE Volatility Index (VIX) as an additional feature, providing a balanced representation of U.S. market exposures [12]. Daily data were obtained through the yfinance Python library for the period January 2015 to December 2024. For each instrument, daily open, high, low, close, adjusted close, and trading volume were collected. The VIX was incorporated as an explanatory variable capturing market-wide volatility expectations. Table 1 summarizes the selected ETFs, listing their tickers, full names, benchmark indices, and market segments.

Table 1. Summary of ETF selected

Ticker	Full Name	Index Tracked	Market Segment
SPY	SPDR S&P 500 ETF	S&P 500	Large-cap broad market
QQQ	Invesco QQQ	NASDAQ-100	Technology-focused
DIA	SPDR Dow Jones Industrial Average ETF	Dow Jones Industrial Average (DJIA)	Blue-chip stocks
IWM	iShares Russell 2000 ETF	Russell 2000	Small-cap

Data preprocessing involved several steps to ensure data quality and stationarity. After cleaning, log returns were calculated from daily closing prices to stabilize variance and approximate continuous compounding. Augmented Dickey–Fuller (ADF) tests confirmed strong stationarity for all ETF and VIX return series, with p-values below 0.01, indicating their suitability for predictive modeling. Finally, ETF data were merged with the VIX to create aligned panels, ensuring that all features and target variables corresponded to the same dates.

2.2 Feature Engineering

A comprehensive set of technical indicators was computed as model inputs, with parameters chosen according to standard conventions appropriate for short-term prediction. The VIX daily close was also included in the “tech + VIX” feature variant as a proxy for market sentiment. Table 2 summarizes the indicators used in this study, outlining their categories, parameter settings, and intended purposes.

ISSN 2959-6130

Table 2. Technical indicators and parameters selected

Indicator	Category	Parameters	Purpose
MA	Trend	60, 120, 240 days	medium/long-term trend
EMA	Trend	5, 10, 20 days	Short-term Weighted trend
RSI	Momentum	14 days	Overbought/oversold
Stochastic Oscillator	Momentum	K=14, D=3, smoothing=3	Momentum relative to high/low range
Williams %R	Momentum	14 days	Overbought/oversold
ROC	Momentum	5 days	Rate of change
CCI	Momentum	14 days	Price deviation from average
MACD	Trend/Volatility	Fast=12, Slow=26, Signal=9	Trend + momentum crossover
Bollinger Bands	Trend/Volatility	20 days, ±2σ	Volatility band
ATR	Volatility	14 days	Average range of price movement
OBV	Volume		Volume trend

2.3 Label Construction

The prediction task is formulated as a binary classification problem in which the target variable represents the direction of ETF price movement. For each ETF on day t, the label is set to 1 when the next-day return (t + 1) is non-negative and 0 otherwise. For the weekly horizon, the label equals 1 if the cumulative return from t + 1 to t + 5 is non-negative and 0 if negative. This design reflects a trading-oriented perspective, focusing on whether an ETF is expected to rise or fall over the next day and the following week.

2.4 Train–Test Splitting

Given the temporal nature of financial data, chronological splits were used to avoid look-ahead bias. For each ETF and forecast horizon, roughly 80% of observations were allocated for training and 20% for testing. A time-based rolling-window approach was adopted to emulate real forecasting conditions and follow best practices in time-series modeling.

2.5 Experimental Workflow

The experiment was carried out in eight main stages. The first seven steps involve data preprocessing and model development, and the final step focuses on evaluating the models and conducting statistical analyses. The complete workflow is illustrated in Fig. 1.

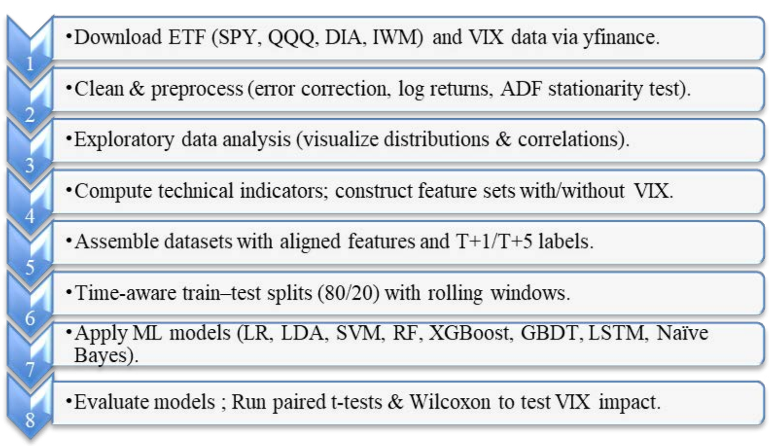

Fig. 1 Experimental workflow (Photo/Picture credit: Original).

3 Methodology

This study evaluates eight machine learning models for ETF return prediction. The models represent five major families widely used in financial forecasting, including linear classifiers (LR, LDA), probabilistic methods (NB), kernel-based approaches (SVM), tree ensembles (RF, GBDT, XGBoost), and deep learning (LSTM). This design allows systematic comparison across levels of model sophistication, from traditional statistical techniques to modern deep learning architectures.

3.1 LR

LR is one of the most common methods for binary classification in data analysis [13]. It estimates the probability that an observation belongs to a particular class, producing values between 0 and 1 that represent the likelihood of the target variable taking the value one [13]. The model is simple, interpretable, and computationally efficient, and has been extensively applied in areas such as credit scoring and return prediction [14].

3.2 LDA

LDA assumes that each class follows a Gaussian distribution with a shared covariance matrix and identifies a linear combination of features that best separates the classes [15,16]. It produces linear decision boundaries and performs reliably even with relatively small samples [15,16]. LDA's advantages include efficiency, interpretability, and robustness to correlated predictors, though its dependence on normality and equal covariance assumptions limits flexibility when those conditions are not met [15,16].

3.3 NB

NB is a probabilistic classifier derived from Bayes' theorem, assuming conditional independence among features given the class. Under this assumption—typically with Gaussian likelihoods for continuous inputs—the model estimates the posterior probability of each class by multiplying the conditional probabilities of feature values [15–17]. NB classifiers are fast and perform well on small datasets, but their independence assumption rarely holds in financial data, which can limit predictive accuracy [15–17].

3.4 SVM

SVM is a margin-based classifier that identifies the hyperplane maximizing class separation. Using a soft-margin formulation it allows limited misclassification and can apply kernel functions to model non-linear boundaries while performing regularized risk minimization [1]. SVMs often excel in high-dimensional classification tasks but are computationally expensive and sensitive to hyperparameter choices such as regularization strength and kernel type [17].

3.5 RF

RF combines many decision trees built through bootstrap sampling and random feature selection. Each tree contributes a vote, and the ensemble's majority decision forms the final prediction. This bagging strategy, together with feature randomization, reduces variance and mitigates overfitting [17]. RF captures complex non-linear relationships and provides feature importance estimates, though at the cost of interpretability. When relationships are largely linear, simpler models may perform comparably, and very large forests can increase computational cost [17].

3.6 GBDT

GBDT constructs an additive ensemble of shallow trees trained sequentially, with each new tree correcting residual errors of the current ensemble. A small learning rate updates the model gradually, improving stability [14,15]. By iteratively minimizing prediction errors, GBDT achieves strong performance on structured data but can be slow and sensitive to hyperparameter settings such as tree depth, learning rate, and number of trees [14,15,17].

3.7 XGBoost

XGBoost extends gradient boosting with additional regularization and system-level optimization. It penalizes model complexity and employs column subsampling and parallel computation to enhance speed and generalization [18,19]. The algorithm is scalable and effective with sparse data, often outperforming standard GBDT. However, it shares ensemble models' drawbacks—numerous hyperparameters and reduced interpretability [18,19].

3.8 LSTM

LSTM is a recurrent neural network architecture designed to capture long-term temporal dependencies via gated memory cells. Information flow is regulated by input, forget, and output gates that determine how the internal state evolves over time [20,21]. This gating mechanism enables LSTMs to learn complex temporal and nonlinear patterns, proving effective in financial forecasting. Nevertheless, they demand extensive training data and computational resources, and their internal structure limits interpretability.

4 Results

The following results summarize the performance of eight machine learning algorithms tested on four ETFs (SPY, QQQ, DIA, and IWM) over two forecast horizons (T+1 and T+5).

4.1 Model Performance per ETF

At the T+1 horizon, model accuracies peak around 57–58% for SPY and QQQ, achieved mainly by LDA, LR, and LSTM, while IWM proves more difficult, with RF performing best at about 52.5%. Over the T+5 horizon, accuracy improves for all ETFs with LSTM attaining 64.5% on SPY and remaining top on QQQ (62.2%) and DIA (60.1%), whereas RF leads on IWM (59.4%). Table 3 summarizes the best-performing ETF for each model using only technical indicators, and Table 4 presents the top

ISSN 2959-6130

models for each ETF across both horizons.

Table 3. Best ETF prediction per model using only technical indicators as input (Tech Only). Each cell shows the ETF with the highest test accuracy (%).

Model	Best ETF @ T+1 (Acc.)	Best ETF @ T+5 (Acc.)
LR	SPY (58.02%)	DIA (58.59%)
LDA	SPY (58.02%)	DIA (58.59%)
SVM	QQQ (57.36%)	IWM (53.96%)
RF	IWM (52.53%)	IWM (59.41%)
GBDT	IWM (50.92%)	IWM (54.19%)
XGBoost	IWM (49.89%)	IWM (54.60%)
LSTM	SPY (57.93%)	SPY (64.52%)
NB (Gaussian)	IWM (50.11%)	IWM (49.12%)

Table 4. Top 3 models by test accuracy for each ETF at T+1 and T+5(Tech Only).

ETF	Horizon	1st (Model, Acc.)	2nd (Model, Acc.)	3rd (Model, Acc.)
SPY	T+1	LDA / Logistic (58.02%)	LSTM (57.93%)	RF (48.07%)
SPY	T+5	LSTM (64.52%)	LDA (53.30%)	LR (47.14%)
QQQ	T+1	LR (58.02%)	LSTM (57.93%)	LDA (57.80%)
QQQ	T+5	LSTM (62.21%)	LR (58.37%)	LDA (58.15%)
DIA	T+1	LDA / LR/ SVM (56.04%)	LSTM (55.63%)	RF (48.27%)
DIA	T+5	LSTM (60.14%)	LDA (58.59%)	LR (58.59%)
IWM	T+1	RF (52.53%)	LR (52.31%)	LSTM (51.26%)
IWM	T+5	RF (59.41%)	RF (57.49%)	LDA (56.39%)

4.2 Horizon Comparison

Averaging across ETFs, LSTM gains +7.11 percentage points when moving from T+1 to T+5, while LDA and RF gain +1.90 and +1.34 points, respectively. In contrast, SVM (−16.39 pts) and NB (−9.34 pts) deteriorate at T+5. The horizon effect is quantified in Table 5.

Table 5. Average model accuracy at T+1 vs T+5(Tech Only) and the change (ΔAcc%).

Model	Avg. Acc. T+1	Avg. Acc. T+5	ΔAcc% (T+5 − T+1)
LSTM	55.69%	59.74%	+7.11%
LDA	55.71%	56.61%	+1.90%
RF	47.43%	48.15%	+1.34%
XGBoost	46.85%	46.34%	−1.22%
LR	56.10%	54.68%	−2.29%
GBDT	46.71%	45.40%	−2.97%
NB (Gaussian)	46.54%	42.24%	−9.34%
SVM	53.08%	43.99%	−16.39%

4.3 Impact of VIX

The impact of VIX is shown by the model in Table 6 and by the ETF in Table 7. On average, adding VIX increases accuracy by 0.57 percentage points at T+1 and 2.04 at T+5, though neither improvement is statistically significant. The effect varies across models and ETFs: SVM benefits the most, LR shows moderate gains at T+5, while RF, XGB, and NB exhibit minimal or negative changes. LSTM performance remains largely unaffected.

Table 6. Average accuracy changes from adding VIX, by model (percentage points).

Model	ΔAcc% (T+1)	ΔAcc% (T+5)
SVM	+5.51%	+15.72%
LR	−0.61%	+3.11%
LDA	+0.20%	+0.59%
GBDT	−1.50%	+0.07%
LSTM	0.00%	0.00%
XGBoost	+0.42%	−0.85%
NB (Gaussian)	−0.79%	−0.98%
RF	+1.33%	−1.30%

Table 7. Average accuracy change from adding VIX, by ETF (percentage points).

ETF	ΔAcc% (T+1)	ΔAcc% (T+5)
SPY	+2.84%	+0.92%
QQQ	+0.74%	+3.52%
DIA	+0.03%	+3.27%
IWM	−1.33%	+0.47%

5. Conclusion

This study provides an assessment of machine learning models for short-term ETF directional prediction across four major U.S. ETFs. Several key conclusions emerge:

Model–ETF–horizon interactions are critical. Model performance is highly context-dependent: no single approach dominates across all ETFs and horizons. At T+1, linear models (LDA, LR) and LSTM achieved the best accuracy (~58%). At T+5, LSTM dominated SPY, QQQ, and DIA (60–65%), while RF was most effective for IWM (59%). NB and GBDT consistently ranked lowest.

Horizon effects are model-specific. When extending the horizon from T+1 to T+5, three out of eight models (37.5%) showed accuracy improvements, with LSTM benefiting most (+7.1%). However, the majority of models (62.5%) experienced declines, including SVM (−16.4%) and NB (−9.3%), suggesting that horizon effects are model-specific rather than uniformly positive.

VIX adds little incremental value for short-term prediction in this setting. Average improvements from adding VIX were <2% and statistically insignificant ($p > 0.1$). Some algorithms (e.g., SVM +15% at T+5; QQQ/DIA +3%) benefit from VIX, especially at T+5, while other models and ETFs showed negligible or negative effects, and the overall ranking of the best model–ETF combinations remains unchanged.

Future work could extend this study by testing additional asset classes and regional ETFs to verify the robustness of the results. Expanding feature sets to include sentiment indicators, macroeconomic variables, and cross-asset signals may further improve prediction accuracy. Developing hybrid frameworks that integrate linear, tree-based, and deep learning models, and evaluating their economic value through realistic back-testing and cost-adjusted performance, would help connect model accuracy with practical profitability.

Acknowledgements

The author gratefully acknowledges the academic foundation provided by The University of Sydney, where she completed her degree in 2012. She is currently conducting independent research.

References

[1] Ayyildiz N, Iskenderoglu O. How effective is machine learning in stock market predictions. Heliyon, 2024, 10(2): e24123.

[2] Strader T J, Rozycki J J, Root T H, Huang Y H J. Machine learning stock market prediction studies: review and research directions. Journal of International Technology and Information Management, 2020, 28(4): 1–20.

[3] Nabipour M, Nayyeri P, Jabani H, Mosavi A, Salwana E, S S. Deep learning for stock market prediction. Entropy, 2020, 22(8): 840.

[4] Ali M, Khan D M, Aamir M, Ali A, Ahmad Z. Predicting the direction movement of financial time series using artificial neural network and support vector machine. Complexity, 2021, 2021: 2906463.

ISSN 2959-6130

[5] Pahwa N, Khalfay N, Soni V, Vora D. Stock prediction using machine learning: a review paper. International Journal of Computer Applications, 2017, 163(5): 36–43.
[6] Soni P, Tewari Y, Krishnan D. Machine learning approaches in stock price prediction: a systematic review. Journal of Physics: Conference Series, 2022, 2161(1): 012065.
[7] Sun Y, Mutalib S, Omar N, Tian L. A novel integrated approach for stock prediction based on modal decomposition technology and machine learning. IEEE Access, 2024, 12: 95209–95222.
[8] Guo Y, Li Y, Xu Y. Study on the application of LSTM-LightGBM model in stock rise and fall prediction. MATEC Web of Conferences, 2021, 336: 05011.
[9] Piovezan R P B, de Andrade Junior P P, Ávila S L. Machine learning method for return direction forecast of exchange traded funds (ETFs) using classification and regression models. Computational Economics, 2024, 63(5): 1827–1852.
[10] Shih K H, Wang Y H, Kao I C, Lai F M. Forecasting ETF performance: a comparative study of deep learning models and the Fama-French three-factor model. Mathematics, 2024, 12(19): 3158.
[11] Chang C L, Hsieh T L, McAleer M. Connecting VIX and stock index ETF. Facultad de Ciencias Económicas y Empresariales. Instituto Complutense de Análisis Económico (ICAE), 2017.
[12] Yahoo Finance. Historical data for SPY, QQQ, DIA, IWM, and VIX. Yahoo Finance, 2015–2024. Accessed Sept. 1, 2025. Available: https://finance.yahoo.com.
[13] Acito F. Logistic regression. In: Predictive Analytics with KNIME: Analytics for Citizen Data Scientists. Cham: Springer Nature Switzerland, 2023: 125–167.
[14] Hosmer D W, Lemeshow S, Sturdivant R X. Applied Logistic Regression (3rd ed.). Hoboken, NJ: Wiley, 2013: 35–47.
[15] James G, Witten D, Hastie T, Tibshirani R, Taylor J. An Introduction to Statistical Learning: With Applications in Python. Springer, 2023.
[16] Géron A. Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (2nd ed.). O'Reilly Media, 2019.
[17] Scikit-learn Developers. Supervised learning — scikit-learn 1.5.2 documentation. Scikit-learn, 2024. Available: https://scikit-learn.org/stable/supervised_learning.html.
[18] Chen T, Guestrin C. XGBoost: a scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016: 785–794.
[19] NVIDIA Corporation. What is XGBoost? NVIDIA Glossary, 2024. Available: https://www.nvidia.com/en-au/glossary/xgboost/.
[20] Hochreiter S, Schmidhuber J. LSTM can solve hard long time lag problems. Proceedings of the 9th International Conference on Neural Information Processing Systems (NIPS'96), 1996: 473–479. Cambridge, MA: MIT Press.
[21] Gers F A, Schmidhuber J, Cummins F. Learning to forget: continual prediction with LSTM. Neural Computation, 2000, 12(10): 2451–2471.