Predictive models and directional strategies for the USD/COP pair

Pedro Elías Pérez Salazar[1], Héctor Hugo Mora[2], Luis Alejandro Fonseca Páez[3]

[1]Ingeniero Electrónico. Universidad de los Llanos (Unillanos), Colombia.
pedro.perez@unillanos.edu.co ORCID: https://orcid.org/0009-0008-8975-9182

[2]Administrador de Empresas. Especialista en Alta Gerencia de Mercadotecnia, Universidad Pedagógica y Tecnológica de Colombia. Magíster en Mercadeo Agroindustrial, Universidad Jorge Tadeo Lozano. Doctor en Administración de Negocios, Escuela Superior de Economía y Administración de Empresas, Buenos Aires, Argentina.
Universidad Pedagógica y Tecnológica de Colombia (UPTC), Colombia.
hector.mora@uptc.edu.co ORCID: https://orcid.org/0009-0006-7711-4913

[3]Ingeniero en Minas. Magíster en Metalurgia y Ciencia de los Materiales. Doctor en Ingeniería y Ciencia de los Materiales, Universidad Pedagógica y Tecnológica de Colombia.
Universidad Pedagógica y Tecnológica de Colombia (UPTC), Colombia.
alejandro.fonseca@uptc.edu.co
ORCID: https://orcid.org/0009-0001-4983-1521

Abstract

This paper presents a comparative study of machine learning models for the directional prediction of financial time series, taking the USD/COP pair as a case study. From 29 macrofinancial and market variables, nine models belonging to three families were evaluated: regularized linear models, gradient boosting models and hybrid models aimed at mitigating the collapse of variance over seven prediction horizons between 1 and 30 business days. The evaluation combined statistical and machine learning metrics with economic performance indicators, prioritizing the directional utility of the signals over the magnitude estimation error. The results show that Ridge obtained the best average performance, with a Direction Accuracy of 52.3 percent, an average Sharpe Ratio of 1.60 and an average Profit Factor of 1.29. Likewise, the pure gradient boosting models presented Variance Ratio values below 0.1 and negative economic performance, which shows relevant limitations in financial contexts with a low signal level. A central finding is that traditional metrics such as the coefficient of determination are not very informative to evaluate strategies based on return prediction, so it is recommended to complement them with directional and economic metrics. The entire analysis is based on a reproducible processing and experimentation flow that integrates intake, training, recording of experiments and generation of predictions.

Keywords: machine learning, financial time series prediction, directional prediction, USD/COP, gradient boosting, regularized linear models, financial evaluation metrics.

1. INTRODUCTION

Financial time series prediction is a complex problem due to the low signal level, the presence of noise, the non-stationarity of markets, and the simultaneous influence of

macroeconomic, financial, and operational variables. In this context, rigorously evaluating the performance of different machine learning models is difficult: manual analysis processes are not very reproducible, depend on the analyst's individual judgment, and complicate systematic comparison between approaches. These limitations motivate careful statistical and experimental treatment, which allows the integration of heterogeneous data sources, training and validating models under comparable conditions, and evaluating their predictions with appropriate metrics.

Machine learning provides a natural framework to address these types of problems, allowing non-linear relationships between multiple predictors to be modeled and the expected direction of asset movement to be estimated. However, in financial domains, comparing models requires careful control for temporal information leakage, hyperparameter selection, and choice of evaluation metrics. To ensure reproducibility and traceability of experiments, the modeling process relies on *Machine Learning Operations* (MLOps) practices, which automate the model lifecycle from data ingestion and transformation to training, versioning, and prediction generation [1], [2].

The main objective of this paper is to compare different modeling approaches for the directional prediction of the USD/COP pair, chosen as a case study for its sensitivity to global and regional macrofinancial variables, such as the U.S. dollar, market volatility, oil price, country risk and other indicators associated with emerging economies. The analysis is carried out over multiple time horizons and is based on a reproducible process and experimentation flow that integrates intake, pre-processing, training, recording of experiments and generation of predictions.

Regularized linear models, *gradient boosting* models and hybrid models aimed at combining directional prediction and magnitude estimation are compared. Unlike an evaluation based solely on traditional regression metrics, the analysis incorporates indicators oriented to the financial use of predictions, such as Direction Accuracy (DA), Sharpe Ratio (SR), Profit Factor (PF) and Variance Ratio (VR).

The main contributions of the work are: i) the experimental comparison of nine predictive models, grouped into three families, over seven time horizons for the directional prediction of the USD/COP pair; ii) the analysis of the relevance of directional and economic metrics against traditional metrics such as *R2* in return prediction problems; iii) the characterization of the phenomenon of variance collapse in pure *gradient boosting* models and its partial mitigation by means of hybrid models; and iv) the implementation of a reproducible processing and experimentation flow applied to the case study.

I. DATA AND EXPERIMENTAL METHODOLOGY

This section describes the case study, the data sources used, the models evaluated, and the metrics used to measure the performance of the strategies generated. The evaluation focused on the directional prediction of the USD/COP pair over different time horizons, considering both machine learning metrics and indicators linked to the performance of financial strategies.

A. *Data sources*

The dataset was constructed from public and free sources, including FRED, TwelveData, Banco de la República de Colombia, and Investing.com [3]–[6]. From these sources, financial, macroeconomic, and regional variables were integrated, with different update frequencies and publication lags. In total, the system considered 29 explanatory variables associated with the analyzed asset and relevant external factors.

The historical data were organized as time series and used in a format compatible with financial analysis, including prices and derived variables. The experimental period covered approximately ten years of historical information, from 2015 to 2025, with around 2500 daily observations. The separation of data was carried out respecting the temporal order, with differentiated sets for training, validation and testing, which avoids random partitions that could introduce temporal information leakage.

B. *Models evaluated*

Nine models grouped into three main families were evaluated: regularized linear models, *grad-dient boosting* models, and hybrid models. The linear models included Ridge, Bayesian Ridge, and ARD Regression [7]–[9]. These models were used as interpretable and robust baselines against low-signal level scenarios.

The second family corresponded to *gradient boosting* models based on decision trees: XGBoost, LightGBM and CatBoost [10]–[12]. These algorithms allow capturing nonlinear relationships and interactions between variables, making them suitable for tabular problems with multiple predictors. However, in financial contexts with a low predictive signal, they can tend to generate predictions close to the target mean, a phenomenon associated with the collapse of variance.

The third family was composed of hybrid models: Hybrid XGBoost, Hybrid LightGBM, and Hybrid CatBoost [13], [14]. These models combined a directional classification component with a regression component for the magnitude of the expected return. The goal of this approach was to separate the market direction decision from the estimation of the movement size, in order to mitigate the variance collapse observed in pure boosting models.

Seven horizons were evaluated: 1, 5, 10, 15, 20, 25 and 30 business days. This configuration allowed analyzing the performance of the models both in short horizons, associated with frequent rebalancing decisions, and in intermediate and long horizons, more linked to strategies such as swing trading or position trading.

C. *Evaluation Metrics*

The evaluation of financial models cannot be limited to traditional error metrics, as a low difference between actual and predicted values does not guarantee financial utility. In trading or risk management applications, the direction of movement and risk-adjusted return may be more relevant than the exact estimation of the magnitude of the return. For this reason, metrics aimed at both predictive performance and economic performance were used.

The first metric considered was DA, which measures the proportion of times in which the model correctly corrects the direction of the future movement of the asset [15]. It is defined as:

$$DA = \frac{1}{n}\sum_{t=1}^{n} \mathbb{I}\left[\text{sign}(A_t) = \text{sign}(F_t)\right], \qquad (1)$$

where *At* represents the actual return at time *t*, *Ft* corresponds to the return predicted by the model, I[·] is the indicator function, and *n* is the total number of observations. This metric is suitable when the main objective is to decide whether to hold, buy, sell, or rebalance a position.

The SR was also used, which evaluates the return obtained per unit of risk assumed [16]. It is calculated as:

$$SR = \frac{\mathbb{E}[R_p - R_f]}{\sigma_p}, \qquad (2)$$

where *Rp* represents the return of the strategy or portfolio, *Rf* corresponds to the risk-free rate and σp is the standard deviation of the strategy's returns. Higher values indicate a higher reward per unit of risk assumed. In addition, the FP was used, defined as the ratio between accumulated gains and accumulated losses [17]:

$$PF = \frac{\sum_{i=1}^{N_w} G_i}{\sum_{j=1}^{N_l} L_j}, \qquad (3)$$

where Gi represents the profits made on winning trades, Lj the losses recorded on losing trades, *Nw* the number of winning trades, and Nl the number of losing trades. Values greater than one indicate that the strategy generates more gains than losses. Finally, VR was incorporated, which compares the variance of predictions with the variance of actual returns [18]:

$$VR = \frac{\text{Var}(F_t)}{\text{Var}(A_t)}. \qquad (4)$$

This metric allows detecting cases in which a model generates almost constant predictions. Very low values of VR indicate collapse of variance and loss of discriminative capacity, a phenomenon especially relevant in financial series with a low signal level.

Although the R2 coefficient of determination was also analyzed, its interpretation was carried out with caution. This metric is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \qquad (5)$$

where yi represents the actual value, $y\hat{}i$ the prediction of the model, and $y\bar{}$ the mean of the actual values. In financial distress, a model may perform poorly in terms of quadratic error and still generate operationally useful directional signals. Therefore, the comparison between models focused on the combination of directional, economic, and predictive stability metrics.

II. IMPLEMENTATION AND EXPERIMENTAL FLOW

This section describes the experimental flow and implementation decisions that allow transforming the time series into inputs suitable for training, validation, and prediction generation. The execution and automation environment, the data processing, the definition of the target variable, the search for hyperparameters, and the proposed hybrid model are detailed.

A. Runtime and Automation Environment

To ensure the reproducibility, automation, and traceability of the experiments, the flow was executed on a containerized microservices environment [19], with separate responsibilities and communication through defined interfaces. This approach decouples the main components of the system, facilitates its maintenance, and enables the future incorporation of new data sources, models, or monitoring mechanisms.

The system integrates Apache Airflow for recording and tracking experiments, PosrgreSQL for structured storage of results and metadata, MinIO as a repository of S3-compatible artifacts, a REST API developed in Python to expose predictions and metrics, and visualization dashboards for querying results [20] – [24]. The design of the flow follows principles of separation of responsibilities, low coupling, high cohesion and fault tolerance [25].

Fig. 1 summarizes the proposed experimental flow, from data acquisition to the exposure of predictions and metrics through decoupled services.

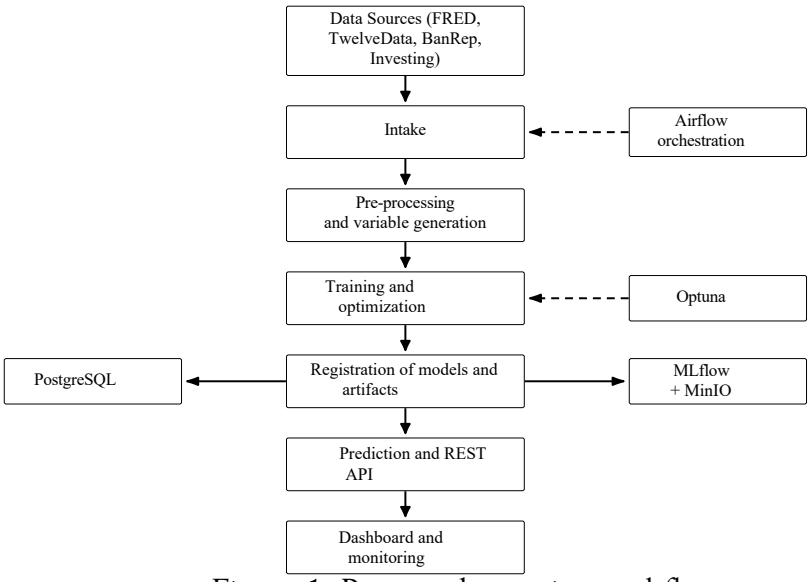


Figure 1. Proposed experimental flow.

The flow of information begins with the automatic acquisition of data from external sources. Then, the transformation processes consolidate the time series, apply the treatment of missing values, and generate the input variables. Subsequently, the training services adjust the models and register their artifacts. Finally, the prediction service exposes the results using the REST API, while the dashboards consume these endpoints to visualize forecasts, metrics, and operational alerts.

Table I summarizes the functional responsibilities of the main components implemented.

Table I: responsibilities of major components

Service	Primary Responsibility
Data ingestion	Automatic external data acquisition, initial validation and anomaly detection.
Processing	Cleaning, transformation, normalization and generation of variables.
Training	Model tuning, hyperparameter optimization, and temporal validation.
Model registration	Model versioning, artifact storage, and metadata management.
Prediction	Forecasting, metric calculation, and exposure using REST APIs.
Monitoring	Monitoring data quality, model performance, and service availability.

B. Operational flow and temporal separation

The experimental flow was designed to avoid temporal information leakage and to preserve the logic of a financial application. For this reason, the separation between training, validation, and test was made respecting the chronological order of the observations. This decision is essential in time series, since a random partition could indirectly incorporate future information during training or model selection.

From an operational point of view, the system was aimed at generating useful signals for rebalancing decision-making. Instead of prioritizing only the exact estimation of the future price, the problem was formulated around the expected direction of movement. Under this logic, a prediction is useful when it helps to decide whether to hold, buy, sell, or rebalance a position. The passive buy-and-hold strategy [26] was considered as a conceptual reference, so that the signals generated by the models were analyzed in terms of their operational utility against this reference.

The frequency of signal generation was defined considering the trade-off between trading opportunity and transaction costs. Too high a frequency can increase the impact of commissions, spreads and *slippage* [27]; conversely, too low a frequency reduces the number of adjustment opportunities. For this reason, the system evaluated horizons between 1 and 30 business days, with special interest in windows compatible with medium-term decisions.

C. Operational data processing

The sources used have different frequencies of updating and publication lags. During the exploratory analysis, it was observed that the missing values were mainly concentrated at the beginning of the time series, a pattern associated with differences in historical availability and institutional lags, rather than a random absence of data.

Different strategies for treating missing values were evaluated, summarized in Table II. The objective was to preserve as many observations as possible without introducing artificial biases or temporal information leakage.

Table II: processing missing values

Strategy	Description	Observation
A	Forward fill + deletion of incomplete rows.	Low risk of leakage; reduces samples.
B	Treatment according to model family.	Balance between data usage and robustness.
C	Forward fill + fillna(0).	You can enter extreme values.
D	Elimination of the initial incomplete section.	Simple and conservative.
E	Subsets of variables by availability.	Increased maintenance complexity.

From this comparison, a specific strategy was adopted for each type of model. The tree-based models took advantage of their native capacity to handle missing values, applying forward imputation to preserve temporal consistency. On the other hand, the linear models required discarding the initial observations affected by publication lags, until they had a complete input matrix.

Direct imputation with zero was discarded as a general strategy, since it can introduce observations artificially far from the original distribution when the variables were not previously normalized. In macroeconomic variables with positive means, replacing missing values with zero can generate values equivalent to multiple standard deviations from the mean, which distorts training and validation.

After the treatment of missing values, the variables were normalized by z-score. For each variable xj, the normalized value was calculated as:

$$z_{t,j} = \frac{x_{t,j} - \mu_j^{train}}{\sigma_j^{train}}, \qquad (6)$$

where xt,j represents the original value of the variable j at time t, μtrainj is the mean calculated over the training set and $_{Uj}\sigma_{train}$ is its standard deviation. These parameters were estimated only on the training data and then applied to validation and testing, in order to avoid data leakage [28].

D. Target variable and hyperparameters

The target variable was defined from the future logarithmic return of the asset for a horizon h. For each instant t, the target return was calculated as:

$$r_{t,h} = \ln\left(\frac{P_{t+h}}{P_t}\right), \qquad (7)$$

where Pt represents the price of the asset at time t and Pt+h the price observed after h business days. From this return, a binary directional label was also defined:

$$y_{t,h} = \begin{cases} 1, & \text{si } r_{t,h} \geq 0, \\ 0, & \text{si } r_{t,h} < 0. \end{cases} \qquad (8)$$

This coding allows the evaluation to be focused on the direction of movement, an aspect that is especially relevant for strategies based on rebalancing or positioning.

The models were optimized by Bayesian hyperparameter search using Optuna [29]. The evaluated ranges were defined based on recommendations from the literature and preliminary tests. In addition, an overfit penalty was incorporated when the difference between training and validation performance exceeded a predefined threshold. Table III summarizes the search space used.

E. Hybrid model

In addition to linear and pure gradient boosting models, a hybrid model was implemented aimed at mitigating variance collapse. In financial series with low signal level, tree-based models can converge towards predictions close to the target mean, which generates almost constant outputs that are not very useful for decision-making.

To reduce this effect, the hybrid model separates the problem into two components: a directional classification stage and a magnitude estimation stage. The former predicts whether the expected move will be bullish or bearish, while the latter estimates the magnitude of the expected return. The final prediction is obtained by combining both outputs:

Table III: HYPERPARAMETER SPACE EVALUATED.

Model	Parameter	Range	Scale
Ridge	Alpha	[0,1–100]	Log
XGBoost	n_estimators	[50–200]	Whole
XGBoost	max_depth	[2–5]	Whole
XGBoost	learning_rate	[0,01–0,1]	Log
XGBoost	reg_alpha	[0,1–10]	Log
XGBoost	reg_lambda	[0,1–10]	Log
LightGBM	num_leaves	[15–63]	Whole
LightGBM	learning_rate	[0,01–0,1]	Log
LightGB	feature_fractio	[0,6–1,0]	Conti

M	n		nuous
CatBoost	Depth	[4–8]	Whol e
CatBoost	learning_rate	[0,01–0,1]	Log
CatBoost	l2_leaf_reg	[1–10]	Conti nuous

$$\hat{r}_{t,h}^{\text{hyb}} = (2\hat{y}_{t,h} - 1)\,|\hat{m}_{t,h}|\,, \qquad (9)$$

where $\hat{y}_{t,h}$ is the directional class estimated by the classifier and $\hat{m}$ is the magnitude estimated by the regression component. In this way, the model decouples the direction decision from the motion size estimate, preserving greater variability in the predictions and reducing the tendency to generate constant outputs.

III. RESULTS AND DISCUSSION

This section presents the experimental design used to evaluate the models, the results obtained, the ranking of the most relevant characteristics and a discussion of the limitations of the coefficient of determination as a main metric in financial prediction problems.

A. *Experimental design*

The experiments were carried out on the USD/COP pair, using historical information corresponding to the period 2015–2025. The dataset included approximately 2500 daily observations and 29 explanatory variables of financial, macroeconomic and regional origin. The separation between training, validation and testing was carried out respecting the temporal order, with the aim of avoiding future information leakage during the process of adjustment and selection of models.

Nine models were evaluated grouped into three families: regularized linear models, pure *gradient boosting* models and hybrid models. Each model was trained and evaluated on seven prediction horizons: 1, 5, 10, 15, 20, 25 and 30 working days. This configuration allowed the behavior of the models to be analyzed in short, intermediate and long horizons, maintaining comparable conditions between approaches.

The evaluation was carried out using metrics oriented to both predictive performance and the operational utility of the signals generated. In particular, DA, SR, PF and VR were considered. In addition, the coefficient of determination was analyzed for diagnostic purposes, although it was not used as the main criterion of comparison due to its limitations in financial contexts with a low signal level.

Table IV: average performance of the models evaluated.

Model	DA(%)	SR	PF	VR
Ridge	52,3	1,60	1,29	0,85
Bayesian Ridge	50,9	1,15	1,22	0,82
ARD	49,9	0,87	1,13	0,78
Hybrid LightGBM	50,3	0,59	1,09	0,65
Hybrid XGBoost	49,9	0,50	1,09	0,68
Hybrid CatBoost	48,8	-0,03	1,00	0,72

CatBoost Pure	46,5	-1,11	0,89	0,09
XGBoost Pure	45,1	-1,44	0,87	0,08
LightGBM Pure	44,4	-1,75	0,85	0,06

B. *Results obtained*

Table IV summarizes the average performance of the models evaluated. The average values of the four metrics mentioned above are reported considering the seven prediction horizons.

The Ridge model performed the best average, with a DA of 52.3%, an SR of 1.60, and a PF of 1.29. While the directional improvement from the 50% random threshold is moderate, in financial distress small systematic differences can be operationally relevant when applied consistently.

Linear models generally performed better than pure *gradient boosting* models. Ridge and Bayesian Ridge were the models with the highest overall stability, while ARD showed a more variable behavior between horizons. In particular, ARD achieved a DA of 56.0% over the 30-day horizon, although its average performance was lower than Ridge's.

The pure *gradient boosting* models presented unfavorable results. XGBoost, LightGBM and CatBoost obtained average DA values below 50%, negative SR and PF less than one. This behavior indicates that the signals generated by these models were not sufficient to construct operationally useful strategies in the case analyzed. The VR analysis allows us to interpret part of this result. The pure *boosting* models presented values below 0.1, which shows a collapse of variance in their predictions. In these cases, the outputs of the model tend to concentrate on values close to the mean of the target, losing discriminative capacity. On the other hand, the linear models and the hybrid models maintained considerably higher VR values, which indicates a greater effective variability in the signals generated.

The hybrid models partially mitigated the collapse of variance observed in the pure *boosting* models. However, they were unable to outperform the regularized linear models. This result suggests that, for the USD/COP case and under the experimental configuration evaluated, the greater complexity of the models does not necessarily translate into better operational performance.

C. *Feature Ranking*

Table V presents the most relevant variables identified during the modeling process. The ranking shows that the characteristics associated with the U.S. dollar, global volatility, interest rates, oil, country risk, and regional variables had a greater impact on the predictions.

Table V: ranking of variables according to importance

Position	Feature	Importance	Category
1	dxy_z	High	Dollar

			Index
2	vix_z	High	Volatility
3	dxy_change	High	Dollar Index
4	curve_slope	Medium-High	US Rates
5	wti_z	Media	Oil
6	embi_z	Media	Country risk
7	usdmxn_ret	Media	Regional
8	col_us_spread	Media	Carry trade

The dxy_z and dxy_change variables reflect the influence of the Dollar Index on the dynamics of the USD/COP pair. Its presence among the most important variables is consistent with the sensitivity of emerging currencies to changes in the global strength of the US dollar.

The vix_z variable, associated with the implied volatility of the US stock market, also appears among the main explanatory factors. This suggests that episodes of global risk aversion or appetite have a relevant impact on the behavior of the exchange rate analyzed.

At a second level of importance, variables associated with interest rates, oil, country risk, and regional dynamics appear. The presence of wti_z is consistent with the relevance of oil for the Colombian economy, while embi_z provides information related to the international perception of risk. Finally, variables such as usdmxn_ret and col_us_spread indicate that regional and carry trade factors also contribute to explaining part of the observed dynamics.

D. Limitations of R2 as a Financial Evaluation Metric

One of the relevant findings of the analysis was that the coefficient of determination is not a sufficient metric to evaluate financial prediction models when the predictions are used to generate trading signals. Although *R2* allows the ability of a model to be measured to explain the variance of a target variable, its interpretation can be misleading when the operational objective is to correctly anticipate the direction of movement and not to estimate with exact precision the magnitude of the return.

In the combinations evaluated between models and horizons, it was observed that 93.7% presented negative values of *R2*. However, 84.1% of these combinations generated positive returns and 87.3% also obtained positive SR. This result shows that a model can present poor performance from the point of view of quadratic error and, even so, generate useful signals from an operational perspective.

Fig. 2 illustrates this discrepancy. Most of the combinations are located in regions with negative *R2*, but with positive returns. This reinforces the idea that traditional regression metrics do not necessarily capture the economic utility of predictions in trading problems.

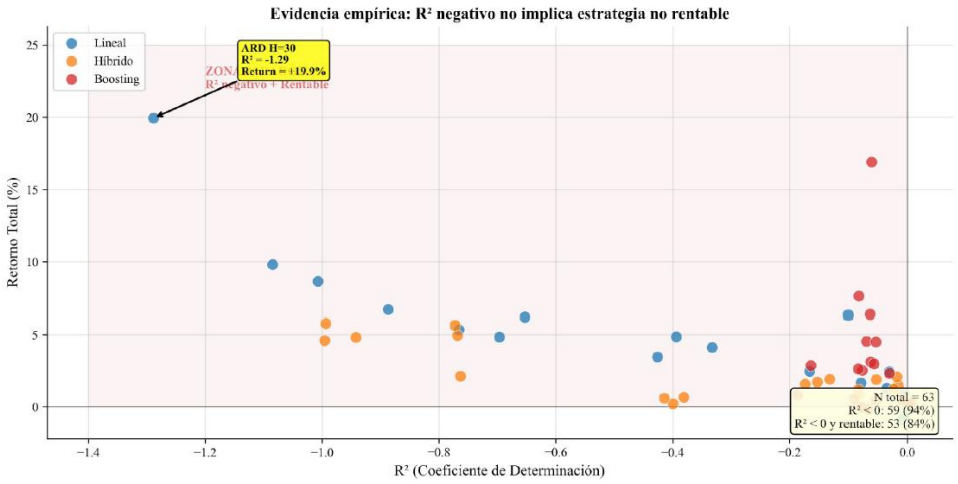


Figure 2. Relationship between the coefficient of determination and the total return of the strategies evaluated.

A representative case corresponds to the ARD model with a horizon of 30 days, which obtained an *R2* of -1.29, but achieved an SR of 2.89, a DA of 56.0% and a PF of 1.39. This example shows that a metric based on square error can penalize models that, although imprecise in magnitude, are useful for anticipating the direction of movement.

Consequently, the evaluation of financial models must complement traditional metrics with directional, economic and predictive stability indicators. In particular, DA, SR, PF and VR are more informative for evaluating the operational utility of strategies based on return prediction.

IV. CONCLUSIONS

This paper presented a comparative study of machine learning models for the directional prediction of financial time series, taking the USD/COP pair as a case study. The analysis was based on a reproducible processing and experimentation flow that integrates intake, training, recording of experiments and generation of predictions, which allowed comparing the different approaches under equivalent conditions.

As a case study, the USD/COP pair was evaluated based on 29 macrofinancial and market variables, considering nine models and seven prediction horizons between 1 and 30 business days. The results obtained show that the regularized linear models presented the best overall performance in the evaluated scenario. In particular, Ridge obtained an average DA of 52.3%, an average SR of 1.60 and an average PF of 1.29, which exceeds pure *gradient boosting* models and hybrid models in most of the metrics considered.

A relevant finding was the presence of variance collapse in pure *gradient boosting* models. XGBoost, LightGBM and CatBoost presented VR values below 0.1, along with negative SR and PF less than unity. This behavior suggests that, in low-signal financial contexts, greater model complexity does not guarantee an improvement in the operational utility of predictions.

Hybrid models made it possible to partially mitigate the collapse of variance, by separating the directional prediction from the magnitude estimate. However, their performance did not exceed that of regularized linear models. This indicates that, for the case analyzed, the

simpler models offered a better combination of stability, interpretability, and economic performance.

The results also highlight the need to complement traditional regression metrics with trading-oriented indicators. In particular, the coefficient of determination can be uninformative when the operational objective is to anticipate the direction of movement and not to accurately estimate the magnitude of the return. For this reason, metrics such as DA, SR, PF and VR are more suitable to evaluate the practical usefulness of predictive models applied to financial strategies.

As limitations, the study was restricted to the USD/COP pair and the historical period 2015–2025. In addition, no intraday data, sentiment signals from news or social media, or explicit modeling of costs such as *spread*, *slippage*, or actual execution fees were incorporated. Therefore, the results should be interpreted as experimental evidence on a specific case study, and not as validation of a trading strategy ready for live trading.

As future work, it is proposed to extend the evaluation to other Latin American currency pairs, incorporate real transaction costs in the backtesting process, explore alternative sources of information and evaluate models capable of integrating textual or sentiment signals. It would also be relevant to analyze more extensive out-of-sample validation schemes and monitoring mechanisms in production that allow detecting performance degradation, regime changes and drift in the distribution of data.

REFERENCES

1. M. Treveil and A. Shukla, *MLOps: Continuous Delivery and Automation Pipelines in Machine Learning*. Sebastopol, CA, USA: O'Reilly Media, 2020.
2. Sculley *et al.*, "Hidden technical debt in machine learning systems," in *Advances in Neural Information Processing Systems*, vol. 28, 2015.
3. Federal Reserve Bank of St. Louis, "FRED: Federal Reserve Economic Data," 2025. [Online]. Available: https://fred.stlouisfed.org/. Accessed: Aug. 9, 2025.
4. Twelve Data, "Twelve Data: Stock, Forex, and Crypto Market Data APIs," 2026. [Online]. Available: https://twelvedata.com/. Accessed: Jul. 2, 2026.
5. Banco de la República, "Economic Statistics," 2026. [Online]. Available: https://suameca.banrep.gov.co/estadisticas-economicas/. Accessed: Jul. 2, 2026.
6. Fusion Media Limited, "Investing.com," 2025. [Online]. Available: https://www.investing.com/. Accessed: Aug. 9, 2025.
7. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67
8. J. C. MacKay, "Bayesian interpolation," *Neural Computation*, vol. 4, no. 3, pp. 415–447, 1992.
9. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

10. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

11. Ke *et al.*, "LightGBM: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

12. L. Prokhorenkova, G. Gusev, A. Vorobev, A. Dorogush, and A. Gulin, "CatBoost: Unbiased boosting with categorical features," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.

13. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.

14. M. Massaoudi *et al.*, "A hybrid Bayesian Ridge regression-CWT-CatBoost model for PV power forecasting," in *Proc. IEEE Kansas Power and Energy Conf.*, 2020.

15. J. Neely, P. A. Weller, and R. Dittmar, "The use of technical analysis in the foreign exchange market," *Journal of International Money and Finance*, vol. 16, no. 1, pp. 51–71, 1997.

16. W. F. Sharpe, "The Sharpe ratio," *The Journal of Portfolio Management*, vol. 21, no. 1, pp. 49–58, 1994.

17. P. J. Kaufman, *Trading Systems and Methods*, 5th ed. Hoboken, NJ, USA: Wiley, 2013.

18. Lo and A. C. MacKinlay, "Stock market prices do not follow random walks: Evidence from a simple specification test," *Review of Financial Studies*, vol. 1, no. 1, pp. 41–66, 1988.

19. Docker, Inc., "Docker: Accelerated container application development," 2025. [Online]. Available: https://www.docker.com/. Accessed: Aug. 9, 2025.

20. Apache Software Foundation, "Apache Airflow," 2025. [Online]. Avai-lable: https://airflow.apache.org/. Accessed: Aug. 9, 2025.

21. Databricks, "MLflow: Open source platform for the machine learning lifecycle," 2025. [Online]. Available: https://mlflow.org/. Accessed: Aug. 9, 2025.

22. PostgreSQL Global Development Group, "PostgreSQL: The world's most advanced open source relational database," 2025. [Online]. Avai-lable: https://www.postgresql.org/. Accessed: Aug. 9, 2025.

23. MinIO, Inc., "MinIO: High performance, Kubernetes native object storage," 2025. [Online]. Available: https://min.io/. Accessed: Aug. 9, 2025.

24. Grafana Labs, "Grafana: The open observability platform," 2025. [On-line]. Available: https://grafana.com/. Accessed: Aug. 9, 2025.

25. S. Brown, The C4 Model for Software Architecture. Leanpub, 2020. [Online]. Available: https://c4model.com/. Accessed: Aug. 9, 2025.

26. Malkiel, *A Random Walk Down Wall Street*, 12th ed. New York, NY, USA: W. W. Norton, 2019.

27. Y. Zhang *et al.*, *The Rebalancing Edge: Optimizing Target-Date Fund Rebalancing Through*

Threshold-Based Strategies. Vanguard Research Report, Dec. 2024.

28. B. Kaufman and S. Rosset, *Data Preparation for Analytics Using SAS*. Cary, NC, USA: SAS Institute, 2014.

29. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next- generation hyperparameter optimization framework," in *Proc. 25th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2019, pp. 2623–2631, doi: 10.1145/3292500.3330701.