Intraday Price-Movement Prediction Using Technical Indicators with Feature Selection and Genetic Algorithm-Based Hyperparameter Optimization

1st Vicenzo F. Copetti
Federal University of Rio Grande (FURG) (FURG)
Rio Grande, Brazil
0009-0001-5357-3828

2nd Bruno L. Dalmazo
Federal University of Rio Grande (FURG)
Rio Grande, Brazil
0000-0002-6996-7602

2rdViviane Leite Dias de Mattos
Federal University of Rio Grande (FURG)
Rio Grande, Brazil
0000-0002-3512-6290

4th Richard F. Pinto
Federal University of Rio Grande (FURG)
Rio Grande, Brazil
0009-0007-0176-3383

5th Diego Renan Bruno
São Paulo State University (UNESP)
Rio Grande, Brazil
0009-0008-1806-0278

6th Eduardo N. Borges
Federal University of Rio Grande (FURG)
Rio Grande, Brazil
0000-0003-1595-7676

7th Giancarlo Lucca
Federal University of Pelotas (UFPel)
Pelotas, Brazil
0000-0002-3776-0260

8th Fabian C. Cardoso
University of Rio Verde (UniRV)
Rio Verde, Brazil
0000-0002-2842-0387

9th Rafael A. Berri
Federal University of Rio Grande (FURG)
Rio Grande, Brazil
0000-0002-3812-4186

***Abstract*—Intraday financial-market modeling remains a challenging task. In this context, technical indicators derived from price, volume, volatility, and trend have long been employed as proxies to capture different market behaviors and trading pressure. However, the large number of available technical indicators often leads to high-dimensional feature spaces, increasing redundancy and negatively impacting model generalization. This work proposes an end-to-end machine learning framework for binary next-interval price-movement prediction using intraday data, with a particular emphasis on the systematic selection of relevant technical indicators. Using the BovDBv2 dataset with 5-minute bars, a total of 61 indicators are computed and normalized, encompassing volume-based, volatility-based, and trend-based information. A Sequential Forward Selection (SFS) procedure, guided by a balanced Random Forest classifier and the binary F1-score, is applied to identify a compact and informative subset of features. And then, a Genetic Algorithm (GA) with binary encoding is employed to optimize the hyperparameters of two classification models: Random Forest (RF) and Multi-Layer Perceptron (MLP). Experimental results indicate that the GA-optimized MLP achieves an accuracy of 0.6269 and a F1-score of 0.6242, outperforming the GA-optimized RF, which attains an accuracy of 0.4786 and a F1-score of 0.4331. Overall, the findings suggest that combining feature selection over technical indicators with GA-based hyperparameter tuning improves generalization performance in intraday price movement prediction tasks.**

***Index Terms*—feature selection, hyperparameter optimization, intraday prediction, technical indicators, genetic algorithm.**

I. Introduction

Intraday financial forecasting constitutes a central problem in quantitative finance and algorithmic trading, yet it remains inherently challenging [1]. Short-horizon price movements are driven by microstructure noise, regime shifts, and pronounced non-stationary dynamics, which collectively hinder predictive stability and degrade model generalization. In practical settings, forecasting pipelines commonly rely on technical indicators computed from intraday price and volume data. Although informative, these indicators typically form high-dimensional and strongly correlated feature sets, increasing redundancy, amplifying overfitting risk, and rendering model performance sensitive to preprocessing and evaluation strategies.

Within this context, two fundamental challenges arise. First, effective feature selection is required to control dimensionality and enhance model robustness, particularly in the presence of correlated technical indicators. Second, predictive performance is highly dependent on model hyperparameters—such as tree depth and sampling strategies in Random Forests(RFs), or network architecture and regularization in Multi-Layer Perceptrons—while exhaustive search is computationally prohibitive due to the size and mixed discrete–continuous nature of the parameter space.

Evolutionary computation provides a suitable framework to address these challenges [2]. Genetic Algorithms (GAs) enable efficient exploration of complex, non-convex hyperparameter spaces without relying on gradient information, while naturally accommodating discrete design choices and bounded continuous parameters through flexible encoding schemes [3].When coupled with rigorous validation protocols, GA-based optimization can yield competitive model configurations while mitigating selection bias and improving generalization in intraday forecasting tasks.

In this work, we investigate a workflow for intraday binary classification on BovDBv2 (5-minute bars) [4], [5]. We compute 61 indicators, normalize features, and apply a SFS procedure [6] to obtain a compact set of three indicators (*OBV*, *ADOSC_3_5*, *TRIMA_9*). We then employ a binary-encoded GA to optimize RF and MLP hyperparameters, reporting convergence behavior through fitness evolution and final performance via confusion matrices and standard metrics.

The main contributions of this paper are:

- The proposal of a fully reproducible intraday learning workflow integrating technical-indicator construction, data normalization, feature selection, and Genetic Algorithm (GA)-based hyperparameter optimization, evaluated on the *BovDBv2* dataset.
- The formulation of a binary-encoded GA with explicitly defined search spaces for Random Forest and Multi-Layer Perceptron models, including a detailed specification of evolutionary operators to ensure reproducibility.
- A comprehensive comparative analysis of GA-optimized Random Forest and Multi-Layer Perceptron classifiers, encompassing fitness evolution dynamics and confusion-matrix-based evaluation on a strictly held-out test set.

The remainder of this paper is organized as follows: Section II reviews related work. Section III details the methodology. Section IV reports results. Section V concludes the paper and outlines future work.

II. Related Work

Machine learning techniques have been extensively applied to financial time series prediction due to their capacity to model nonlinear relationships and complex interactions inherent in market data. Seminal studies in this area have shown that combining learning algorithms with technical indicators yields more informative representations than raw price series, particularly in short-term forecasting settings [7]. Indicator-based representations are designed to capture trend, momentum, volatility, and volume dynamics that are not readily observable from price information alone.

Temporal neural models such as LSTMs are relevant baselines for directional prediction because they explicitly model temporal dependencies. However, simpler architectures such as MLPs may still be competitive when feature engineering already provides a compact representation of short-term market dynamics through technical indicators [8].

Tree ensembles such as RF are common in financial forecasting, but their performance depends strongly on representation and tuning, and no single classifier consistently dominates across markets and horizons [9].

Feature selection is a central issue when working with large collections of technical indicators, as redundancy and multicollinearity can degrade model generalization. Genetic Algorithms (GAs) have been successfully applied to feature selection problems due to their ability to search large combinatorial spaces efficiently. Santana et al. [10] and Canuto and Nascimento [11] proposed GA-based approaches for selecting informative feature subsets in ensemble-learning contexts, demonstrating improved predictive performance and reduced overfitting compared with unfiltered feature spaces.

Beyond feature selection, GAs have also been extensively employed for hyperparameter optimization in neural and hybrid models. Fang et al. [12] showed that GA-optimized neural networks outperform manually tuned configurations in stock market prediction tasks. Similar findings were reported for MLP-based models, where evolutionary optimization helps balance model complexity and generalization performance [13]. These results highlight the suitability of GAs for navigating mixed discrete and continuous search spaces typical of machine learning hyperparameters.

Several studies further emphasize the importance of combining feature selection and hyperparameter tuning for financial prediction. Naik and Mohan [14] demonstrated that selecting compact subsets of technical indicators significantly improves short-term prediction accuracy, while Ding et al. [15] showed that GA-based optimization enhances neural network convergence and stability. The theoretical foundations underlying these approaches are rooted in the seminal works of Holland [16] and Goldberg [3], which established Genetic Algorithms as general-purpose optimization methods for complex, multimodal problems.

Despite these advances, a substantial portion of the literature addresses feature selection and hyperparameter optimization as independent stages or restricts the analysis to a single model family. Moreover, relatively few studies investigate evolutionary optimization within a unified intraday learning framework that enforces strict control of data leakage and adopts time-aware evaluation protocols. This work addresses these limitations by jointly integrating systematic feature selection over technical indicators with binary-encoded Genetic Algorithm–based hyperparameter optimization for both Random Forest and Multi-Layer Perceptron models, evaluated on intraday data from the *BovDBv2* dataset.

III. A Machine Learning Framework for Intraday Price-Movement Prediction

This section presents the methodological framework adopted in this study. The procedure is organized as:

- Dataset description and construction of the technical-indicator representation;
- Min–Max normalization of the indicator set;
- Feature selection and the resulting indicator subset;
- GA-based hyperparameter optimization; and

- Definition of the search spaces for Random Forest (RF) and Multi-Layer Perceptron (MLP), followed by evaluation on a held-out test set.

A. Dataset and Technical Indicators

The experiments are conducted using the *BovDBv2* dataset, which provides intraday market data organized as fixed-interval bars with 5-minute aggregation for the Brazilian market. Each record corresponds to one intraday bar containing price and traded-volume information associated with a timestamp.

The learning task is formulated as a binary classification problem aimed at predicting the direction of the price movement in the subsequent intraday interval. Let c_t denote the closing price at interval t. We define two classes: C_1 (upward movement) when $c_{t+1} > c_t$, and C_2 (non-upward movement) when $c_{t+1} \leq c_t$. In the implementation, these classes are encoded as $y = 1$ for C_1 and $y = 0$ for C_2. The final observation of each series is discarded to preserve temporal consistency due to the one-step-ahead formulation.

From these series, a set of technical indicators is computed, resulting in **61 derived indicators**. Indicator-based representations are widely adopted in IJCNN studies on market modeling and prediction [7], [8]. The indicator set comprises multiple families, including *trend/overlap* measures (e.g., moving averages), *volatility* measures (e.g., band- and range-based indicators), and *volume/flow* measures (e.g., accumulation/distribution and volume pressure). Several indicators are computed with different window lengths; suffixes denote the lookback window in number of bars (e.g., `TRIMA_9`). For indicators parameterized by two windows, such as `ADOSC_3_5`, the pair $(3, 5)$ represents the fast and slow smoothing windows (in bars) used to compute the oscillator.

B. Min–Max Normalization

To reduce scale disparities and ensure numerical comparability across indicators, Min–Max normalization is applied to the technical-indicator columns. Given an indicator value x, the normalized value $\tilde{x}$ is computed as:

$$\tilde{x} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \tag{1}$$

where $x_{\min}$ and $x_{\max}$ are the minimum and maximum values observed for that indicator over the considered sample, respectively.

Leakage control. To prevent information leakage, normalization parameters are estimated exclusively on the training partition and then applied to the held-out test partition using the same $x_{\min}$ and $x_{\max}$. This is particularly important in intraday settings, where temporal dependence can inflate test performance if global statistics are used.

C. Feature Selection and Selected Indicators

A Feature Selection procedure is adopted to obtain a compact representation and mitigate redundancy among predictors. The method follows a greedy SFS strategy: starting from an empty subset, at each iteration the candidate feature that yields the largest improvement under a validation criterion is added to the current subset, and the process repeats until a predefined subset size K is reached. Although evolutionary approaches can also be used for feature subset selection [10], [11], this study fixes a compact subset before GA-based model tuning to reduce the dimensionality of the optimization problem and to keep the subsequent hyperparameter search focused.

Wrapper evaluator and selection criterion. In our implementation, adopted in this study, subset quality is evaluated using a Random Forest classifier with balanced class weights to mitigate imbalance during selection. Candidate subsets are scored using the binary F1-score on a validation split, and the feature that maximizes the score is added at each iteration.

A subset size of $K = 3$ was adopted, yielding `OBV`, `ADOSC_3_5`, and `TRIMA_9`. This choice favors a compact representation, reducing redundancy among technical indicators and making the subsequent GA-based hyperparameter optimization more tractable. In addition, limiting the feature subset helps control model complexity in a noisy intraday setting, where overly large representations may increase overfitting risk. Although this design choice is consistent with the objective of balancing informativeness and parsimony, no sensitivity analysis over larger K values is provided.

D. Genetic Algorithm Optimization

After fixing the selected feature subset, a GA is employed to optimize the hyperparameters of the Random Forest and Multi-Layer Perceptron classifiers. Each individual represents a candidate solution encoded as a binary string, which is subsequently decoded into model-specific hyperparameter configurations. Fitness is evaluated exclusively on the *training partition* using a stratified five-fold validation scheme to enhance robustness, while the *test partition* is strictly reserved for final performance assessment. The use of GA-based optimization for financial time series modeling and parameter tuning has been previously investigated in the context of IJCNN studies [12].

Binary encoding and decoding. Let a hyperparameter be represented by b bits, yielding an integer $z \in [0, 2^b - 1]$. For continuous or integer ranges, we map z to the target interval via linear scaling:

$$\theta = \theta_{\min} + \frac{z}{2^b - 1} \left(\theta_{\max} - \theta_{\min} \right), \tag{2}$$

followed by rounding when the parameter is integer-valued. For categorical parameters, the integer z selects an element from a predefined option list (possibly with modulo to handle unmatched codes). This encoding enables the GA to handle mixed discrete/continuous spaces uniformly.

The GA follows the standard evolutionary cycle of population initialization, fitness evaluation, tournament selection, two-point crossover, bit-flip mutation, and elitist retention of the best-performing solution. Table I summarizes the GA parameters adopted in this study.

Fitness definition. For RF, the fitness function is based on cross-validated classification performance computed on the

TABLE I
THE RELATION OF PARAMETERS USED BY THE GENETIC ALGORITHM .

Parameter	Value
Number of generations (N_{gen})	1000
Population size (N_{pop})	20
Tournament size	3
Crossover	Two-point, probability = 0.80
Mutation	Bit-flip, probability = 0.05
Per-bit mutation probability	0.02
Elitism	Hall-of-Fame size = 1
Random seed	42

training partition. For MLP, the fitness function combines training and validation accuracy in each fold as:

$$\text{Fitness}_{MLP} = 0.4 \cdot \text{Acc}_{train} + 0.6 \cdot \text{Acc}_{val}, \quad (3)$$

and is averaged across folds. This weighted scheme emphasizes generalization while still penalizing configurations that severely underfit.

Class imbalance handling. During MLP training, the training folds employ bootstrap oversampling (random resampling with replacement) of the minority class [17], [18], while scaling is performed using a StandardScaler fitted on the training split and applied to validation and test data. For RF, imbalance is partially mitigated through balanced class weights.

IV. RESULTS

This section presents the experimental results obtained with GA-based hyperparameter optimization for the RF and MLP classifiers under the $K = 3$ feature subset. The analysis is organized as follows. First, the fitness evolution across generations is examined to assess the convergence behavior and stability of the optimization process. Next, the final test-set performance of the tuned models is reported.

A. Fitness Evolution

Figures 1 and 2 present the GA fitness evolution for MLP and RF across 1000 generations. The plots track best, mean, and worst fitness values per generation, while the shaded region indicates population spread (worst $\rightarrow$ best), offering insight into convergence and exploration dynamics.

From the convergence perspective, the GA-optimized MLP exhibits a rapid improvement in fitness during the early generations, followed by a clear stabilization phase after approximately the first few hundred generations. This behavior suggests that the evolutionary process is able to quickly identify promising regions of the hyperparameter search space, converging toward stable configurations with limited oscillation in later stages.

In contrast, the GA optimization for the Random Forest shows slower convergence and higher variability across generations. The wider spread between worst, mean, and best fitness values indicates a more irregular fitness landscape, where small hyperparameter changes can lead to substantial performance fluctuations. This behavior is consistent with the sensitivity of tree-based ensembles to depth, sampling, and splitting parameters, particularly when trained on compact feature representations.

Overall, the observed fitness-evolution patterns indicate that the GA provides a more stable and consistent optimization process for the MLP than for the Random Forest under the adopted feature subset ($K = 3$).

B. Test-Set Performance

Table II summarizes the final performance on the reported hold-out sets ($N = 1895$). The MLP optimized with GA achieves 0.6269 accuracy (macro-F1 $= 0.6242$), while the RF optimized with GA achieves 0.4786 accuracy (macro-F1 $= 0.4331$). The RF confusion matrix shows a strong tendency to predict class 1, resulting in a large number of false positives.

TABLE II
TEST-SET PERFORMANCE ($K = 3$, $N = 1895$).

Model	Accuracy	Macro-F1
RF (GA-optimized)	0.4786	0.4331
MLP (GA-optimized)	0.6269	0.6242

C. Confusion Matrices

Tables III and IV report the confusion matrices for RF and MLP, respectively, following the convention of True labels by rows and Predicted labels by columns.

TABLE III
CONFUSION MATRIX ON THE TEST SET FOR RANDOM FOREST (K=3).

	Pred 0	Pred 1
True 0	185	823
True 1	165	722

TABLE IV
CONFUSION MATRIX ON THE TEST SET FOR MLP (K=3).

	Pred 0	Pred 1
True 0	514	457
True 1	250	674

A more detailed inspection of the confusion matrices reveals important differences in the predictive behavior of the two classifiers. The RF exhibits a strong bias toward predicting class 1, resulting in a large number of false positives. Although this behavior yields moderate recall for the positive class, it severely degrades precision for class 1 and leads to poor recall for class 0, which ultimately compromises the macro-F1 score.

In contrast, the MLP achieves a more balanced trade-off between precision and recall across both classes. While false negatives and false positives are still present, the distribution of errors is more symmetric, indicating improved discrimination between upward and downward price movements. This balanced behavior directly contributes to the higher macro-F1 observed for the MLP, confirming its superior generalization ability under the evaluated intraday setting.

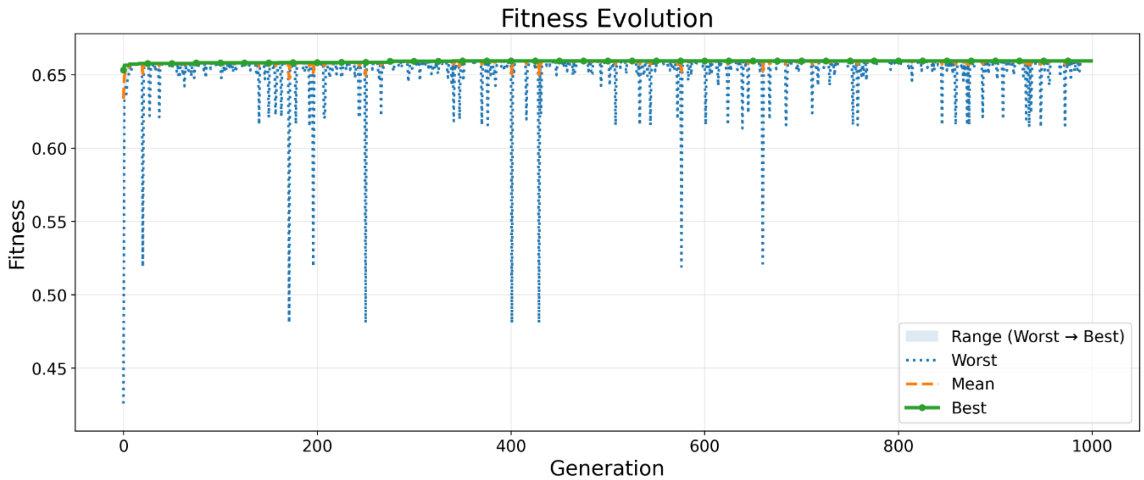

Fig. 1. GA fitness evolution for the MLP (K=3). Best, mean, and worst fitness are tracked over 1000 generations; the shaded area indicates the range (worst → best).

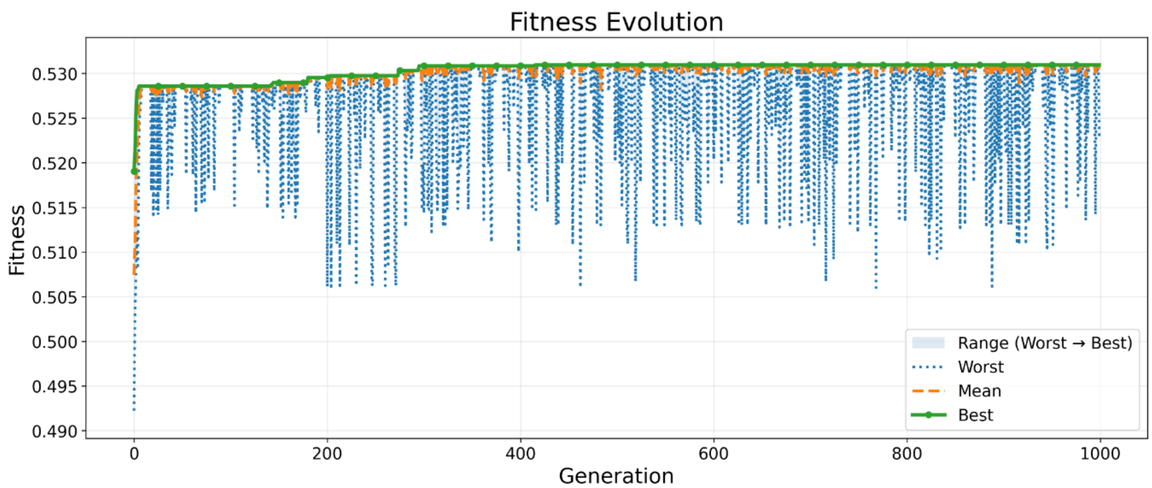

Fig. 2. GA fitness evolution for the Random Forest (K=3). Best, mean, and worst fitness are tracked over 1000 generations; the shaded area indicates the range (worst → best).

D. The Best Hyperparameters Found by GA

The best configurations obtained by GA in the reported runs are summarized in Table V. Recall that, in our GA formulation, each individual encodes a candidate hyperparameter set as a chromosome, which is decoded into model-specific settings before training and fitness evaluation. These values correspond to the decoded parameters of the best individuals stored by the optimization notebooks.

The superior performance of the GA-optimized MLP can also be partially explained by the characteristics of the selected feature subset. The indicators `OBV` and `ADOSC_3_5` encode cumulative and short-term volume pressure, capturing information related to market participation and buying/selling imbalance, while `TRIMA_9` provides a smoothed representation of local price trends.

This combination of volume-driven and trend-driven signals yields a compact yet informative representation of intraday market dynamics. Such a representation appears to favor the nonlinear modeling capacity of the MLP, which can learn continuous interactions among indicators more effectively than tree-based partitioning mechanisms. As a result, the MLP

TABLE V
BEST HYPERPARAMETERS SELECTED BY GA FOR $K = 3$.

Model	Best Hyperparameters
RF	`n_estimators`=396 `max_depth`=8 `min_samples_split`=49 `min_samples_leaf`=21 `max_features`=0.4

MLP	`hidden_layers`=[10,2]
	`activation=relu`
	`solver=adam`
	$\alpha = 0.001239$
	`learning_rate_init`=0.010846

is better suited to exploit the complementary nature of the selected indicators, leading to improved generalization performance on the test set.

V. Conclusion

Intraday price-movement prediction remains a challenging problem due to the high level of noise, non-stationarity, and rapidly changing dynamics that characterize short-term financial markets. In this context, the widespread use of technical indicators often leads to high-dimensional feature spaces, increasing redundancy and sensitivity to model configuration and evaluation choices. This work addressed these challenges by proposing an end-to-end machine learning workflow that integrates feature selection and evolutionary hyperparameter optimization for intraday classification tasks.

Using 5-minute data from the *BovDBv2* dataset, a broad set of technical indicators was initially computed to capture complementary aspects of market behavior, including trend, volume pressure, and volatility. A SFS procedure was then employed to identify a compact and informative subset of indicators. The resulting reduced representation played a key role in mitigating redundancy, simplifying the learning task, and improving stability in a highly noisy intraday environment.

After fixing the selected feature subset, a binary-encoded Genetic Algorithm was used to optimize the hyperparameters of RF and MLP models. The evolutionary approach proved effective in exploring mixed discrete and continuous search spaces without relying on exhaustive or manual tuning strategies. Fitness evolution analyses indicated consistent convergence behavior, suggesting that the GA was able to identify stable and competitive configurations within the defined search spaces.

Experimental results on a strictly time-ordered hold-out test set showed that the GA-optimized MLP achieved superior generalization performance when compared with the GA-optimized Random Forest. This outcome highlights the importance of model expressiveness when learning non-linear relationships among technical indicators in intraday settings, while also reinforcing the need for proper regularization and validation to avoid overfitting.

Overall, the findings demonstrate that combining systematic feature selection over technical indicators with GA-based hyperparameter optimization can improve robustness and predictive performance in intraday price-movement classification. The proposed workflow is fully reproducible and scalable, offering a practical alternative to ad hoc tuning procedures commonly adopted in financial machine learning studies.

Future work should extend the empirical evaluation in three directions. First, stronger baselines should be included, particularly temporal and boosting-based models, to better contextualize the observed gains. Second, experiments should be repeated across different assets and time periods in order to assess robustness under changing market conditions. Third, predictive results should be linked to economic performance through trading-oriented metrics such as PnL and Sharpe ratio, since classification quality alone does not fully reflect practical usefulness in intraday financial applications.

Acknowledgment

The authors would like to thank FAPERGS (24/2551-0001396-2, 23/2551-0000773-8), CNPq (307416/2025-9), FAPERGS/CNPq (23/2551-0000126-8) and the Fesurv-University of Rio Verde.

References

[1] A. Cartea, S. Jaimungal, and J. Penalva, *Algorithmic and High-Frequency Trading*. Cambridge, U.K.: Cambridge Univ. Press, 2015.

[2] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. Berlin, Heidelberg: Springer, 2003, doi: 10.1007/978-3-662-05094-1.

[3] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Reading, MA, USA: Addison-Wesley, 1989.

[4] F. C. Cardoso, J. A. V. Malska, P. J. Ramiro, G. Lucca, E. N. Borges, V. L. D. de Mattos, and R. A. Berri, "BovDB: a data set of stock prices of all companies in B3 from 1995 to 2020," *Journal of Information and Data Management*, vol. 13, no. 1, 2022, doi: 10.5753/jidm.2022.2345.

[5] A. S. Souza, G. Lucca, E. N. Borges, F. C. Cardoso, B. L. Dalmazo, and R. A. Berri, "Dataset for intraday analysis of B3 stock prices," Harvard Dataverse, 2024, doi: 10.7910/DVN/TMB4IG.

[6] J. Doak, *An Evaluation of Feature Selection Methods and Their Application to Computer Security*. Tech. Rep. CSE-92-18, Department of Computer Science, University of California, Davis, CA, USA, 1992.

[7] M. R. Vargas, C. E. M. dos Anjos, G. L. G. Bichara, and A. G. Evsukoff, "Deep learning for stock market prediction using technical indicators and financial news articles," in *Proc. International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–8, doi: 10.1109/IJCNN.2018.8489208.

[8] D. M. Q. Nelson, A. C. M. Pereira, and R. A. de Oliveira, "Stock market's price movement prediction with LSTM neural networks," in *Proc. International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 1419–1426, doi: 10.1109/IJCNN.2017.7966019.

[9] M. Ballings, D. Van den Poel, N. Hespeels, and R. Gryp, "Evaluating multiple classifiers for stock price direction prediction," *Expert Systems with Applications*, vol. 42, no. 20, pp. 7046–7056, 2015, doi: 10.1016/j.eswa.2015.05.013.

[10] L. E. A. Santana, L. Silva, and A. M. P. Canuto, "Feature selection in heterogeneous structure of ensembles: A genetic algorithm approach," in *Proc. International Joint Conference on Neural Networks (IJCNN)*, 2009, pp. 2784–2791.

[11] A. M. P. Canuto and B. S. Nascimento, "A genetic-based approach to features selection for ensembles using a hybrid and adaptive fitness function," in *Proc. International Joint Conference on Neural Networks (IJCNN)*, 2012, pp. 1–8, doi: 10.1109/IJCNN.2012.6252699.

[12] Y. Fang, J. Fu, and Y. Yao, "Improving the genetic-algorithm-optimized wavelet neural network for stock market prediction," in *Proc. International Joint Conference on Neural Networks (IJCNN)*, 2014, pp. 2287–2293, doi: 10.1109/IJCNN.2014.6889969.

[13] F. Ecer, S. Ardabili, S. S. Band, and A. Mosavi, "Training multilayer perceptron with genetic algorithms and particle swarm optimization for modeling stock price index prediction," *Entropy*, vol. 22, no. 11, p. 1239, 2020, doi: 10.3390/e22111239.

[14] N. Naik and B. R. Mohan, "Optimal feature selection of technical indicator and stock prediction using machine learning technique," in *Proc. International Conference on Emerging Technologies in Computer Engineering (ICETCE)*, 2019, pp. 261–268, doi: 10.1007/978-3-030-30577-2_25.

[15] S. Ding, C. Su, and J. Yu, "An optimizing BP neural network algorithm based on genetic algorithm," *Artificial Intelligence Review*, vol. 36, no. 2, pp. 153–162, 2011, doi: 10.1007/s10462-011-9208-5.

[16] J. H. Holland, "Genetic algorithms and adaptation," in *Adaptive Control of Ill-Defined Systems*, New York, NY, USA: Springer, 1984, pp. 317–333.

[17] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. New York, NY, USA: Chapman & Hall/CRC, 1993.

[18] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009, doi: 10.1109/TKDE.2008.239.