Horizon-Aware Feature Selection and Evolutionary Optimization for Intraday Prediction in Futures Markets

Arthur E. Nunes[1], Bruno L. Dalmazo[1], Viviane Leite Dias de Mattos[1], Richard F. Pinto[1], Diego Renan Bruno[2], Eduardo N. Borges[1], Giancarlo Lucca[1], Fabian C. Cardoso[3], and Rafael A. Berri[1(✉)]

[1] Federal University of Rio Grande (FURG), Rio Grande, Brazil
{arthur.nunes,dalmazo,vivianemattos,richard_pinto,eduardoborges,giancarlo.lucca,rafaelberri}@furg.br
[2] São Paulo State University (UNESP), São José do Rio Preto, Brazil
diego.bruno@unesp.br
[3] University of Rio Verde (UniRV), Rio Verde, Brazil
fabian@unirv.edu.br

Abstract. Intraday financial forecasting remains a challenging task due to noise, non-stationarity, and the dynamic nature of market behavior. To capture these complex dynamics, studies often employ multiple technical indicators, which can lead to high-dimensional datasets, increasing the risk of overfitting and degrading model generalization. This study proposes a methodological workflow for intraday binary classification of the Mini Bovespa Index (WIN) using 5-min data from the BovDBv2 dataset. Feature selection is performed independently for different forecast horizons (1, 5, and 7 candles) using Sequential Forward Selection with distinct estimators. Subsequently, a genetic algorithm is employed to optimize the hyperparameters of three classifiers: multi-layer perceptron, random forest, and K-Nearest Neighbors. Experimental results show that shorter forecasting horizons yield superior predictive performance. For instance, the MLP model with SFS-KNN at horizon 1 achieved a macro F1-score of 0.632, while performance decreased to 0.377 at horizon 7 under the same configuration. Overall, the findings indicate that feature relevance and model performance vary across forecasting horizons and highlight the importance of horizon-aware feature selection for intraday financial prediction.

Keywords: Machine Learning · Intraday Prediction · Futures Markets · Feature Selection · Evolutionary Optimization

1 Introduction

The prediction of price movements in intraday windows is a central problem in quantitative finance [1]. This type of forecasting poses major challenges, since

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16768, pp. 315–332, 2027.
https://doi.org/10.1007/978-3-032-30488-9_20

financial time series are strongly influenced by noise, regime changes, and non-stationary dynamics, factors that hinder the generalization of predictive models [2].

With the aim of mitigating these difficulties, several techniques can be applied, such as: statistical modeling, fundamental analysis and technical analysis. In this study the technical analysis approach is adopted, using technical indicators, which are computed from prices and volumes, to extract relevant information about market behavior. Indicators such as moving averages, oscillators, and trend measures are widely used in practical applications, especially with intraday data.

However, the combined use of multiple technical indicators may result in high-dimensional datasets with strong correlations among attributes. This scenario increases the risk of overfitting, makes model performance sensitive to preprocessing choices, and complicates the identification of the most relevant features for the forecasting task [3].

In this context, feature selection techniques become essential to reduce dimensionality and identify the most informative attributes for each forecasting horizon. By selecting relevant subsets of indicators, it is possible to improve model generalization and reduce computational complexity [4].

Additionally, evolutionary algorithms, particularly Genetic Algorithms (GA) [5], constitute an efficient alternative for dealing with complex and non-convex search spaces. Inspired by the principles of natural evolution, GAs enable hyperparameter optimization without the need for gradient information, being especially suitable for problems involving continuous, discrete, and categorical variables [6].

The objective of this study is to propose and evaluate a methodological workflow for intraday classification using data from the BovDBv2 dataset [7], considering 5-min candles of the Mini Bovespa Index (WIN). Based on price and volume data, technical indicators of trend and momentum are computed and then used as features for the dataset, followed by preprocessing steps. Feature selection is performed using the Sequential Forward Selection (SFS) method, employing different estimators, with the objective of identifying more informative feature subsets for different forecasting horizons.

Subsequently, a GA is used to optimize the hyperparameters of different well-known classifiers, namely Multi-Layer Perceptron (MLP) [8], Random Forest (RF) [9], and K-Nearest Neighbors (KNN) [10], considering different temporal forecasting horizons. Model performance is evaluated using metrics suitable for imbalanced classification problems, allowing the analysis of the combined impact of feature selection, hyperparameter optimization, classifier choice, and forecasting horizon on the predictive capability of the system.

The main contributions of this study can be summarized as follows:

- Proposal of a horizon-aware feature selection strategy for intraday financial prediction.
- Integration of wrapper-based feature selection with evolutionary hyperparameter optimization.

- Empirical evaluation across multiple forecasting horizons using intraday futures market data.
- Analysis of how feature relevance and model configurations change across distinct intraday forecasting horizons.

The remainder of this paper is organized as follows: Sect. 2 presents the related work, Sect. 3 describes the adopted methodology, Sect. 4 discusses the obtained results, and finally, Sect. 5 presents the conclusions and directions for future work.

2 Related Work

The application of machine learning techniques to financial market prediction has been extensively investigated in the literature, particularly in the context of short-term price movements and trend classification. Traditional statistical models have gradually been complemented or replaced by machine learning approaches due to their ability to capture nonlinear relationships and complex patterns present in financial time series.

Several studies have explored the use of technical indicators as input features for predictive models in financial markets. Indicators derived from price and volume data, such as moving averages, momentum oscillators, and trend strength measures, are widely employed to represent market behavior in a structured form. Works such as [11] and [12] highlight that technical indicators remain relevant features for short-term forecasting, especially when combined with nonlinear classifiers.

Neural networks, particularly MLPs, have been frequently applied to financial prediction tasks due to their flexibility and universal approximation capability. Previous studies demonstrate that MLPs can outperform linear models in trend prediction problems when properly configured and trained [8]. However, other machine learning classifiers such as RF [9] and KNN [10] have also been successfully employed in financial forecasting tasks, benefiting from their ability to model nonlinear decision boundaries and complex feature interactions.

Feature selection has also been recognized as a critical step in financial time series modeling, as the inclusion of irrelevant or redundant features may degrade predictive performance and increase computational cost. Wrapper-based methods, such as SFS, have been successfully applied in financial contexts to identify compact and informative subsets of indicators [13]. The choice of estimator within the feature selection process has been shown to influence the resulting feature set, since different models capture distinct data characteristics.

In addition to feature selection, several works have investigated the use of evolutionary and metaheuristic algorithms for hyperparameter optimization in machine learning models. GAs have been widely adopted due to their ability to efficiently explore large and complex search spaces without requiring gradient information [14]. In financial applications, GAs have been employed to optimize neural network architectures, learning parameters, and trading strategies, often leading to improved generalization performance [15].

Despite these advances, many studies focus on a single prediction horizon or rely on a fixed feature set across different temporal scales [8,12]. Fewer works explicitly investigate horizon-dependent feature selection and hyperparameter optimization in an integrated manner. In this context, the present study contributes by combining horizon-specific feature selection using SFS with GA-based hyperparameter optimization of multiple classifiers (MLP, RF, and KNN), allowing the analysis of how feature relevance and model configuration vary across distinct intraday forecast horizons.

3 Methodology

In this section, the proposed method is presented. It is composed of the following subsections: Input Data, describing the data source and dataset construction; Technical Indicators, presenting the indicators used; Feature Selection, where attribute selection and forecast horizons (FH) are defined; GA and Classifiers, describing the use of the GA for hyperparameter optimization and the evaluated classifiers (MLP, RF, and KNN); and Evaluation, where the metric and evaluation procedure are presented. Figure 1 summarizes the overall workflow.

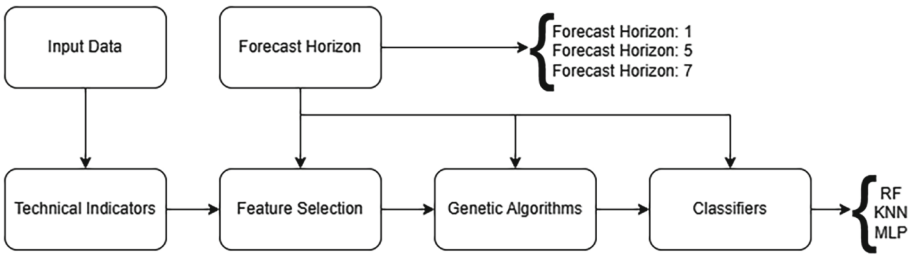


Fig. 1. Methodology Overview

3.1 Input Data

The data were obtained from the BovDB.v2 database [7]. For this study, intraday data from January to June 2024 were considered and the asset analyzed was the Mini Bovespa Index (WIN), selected due to its wide use in short-term trading operations in the Brazilian market and its high liquidity. For the intraday analysis, 5-min interval data were used, allowing the analysis of short FHs and a better understanding of the asset's behavior.

In addition to the information originally available in the database, additional columns containing technical indicators calculated from price data were created, as well as a variable called trend, which assumes the value 0 for a downward trend and 1 for an upward trend, the trend label was defined based on the price

variation between the current candle and the future candle corresponding to the selected forecast horizon. In total, the dataset contains 15,300 instances, 42 features and the class distribution is around 53% upward trends and 47% downward trends.

3.2 Technical Indicators

For this study, several technical indicators were used, belonging to two major groups: trend and momentum [16–18]. The trend indicators employed were: EMA, SMA, DEMA, TRIMA, WMA, TEMA, and SAR, while the momentum indicators were: MOM, ADX, ADXR, CMO, BOP, IMI, MACD, and ROC. All moving averages were computed using periods of 3, 5, 7, and 9 (5-min intraday intervals).

Momentum indicators were used with the objective of capturing the intensity and speed of price movements over time. The Momentum indicator (MOM) was employed to measure the absolute price variation over a given interval, while ADX together with ADXR were used to quantify trend strength, regardless of its direction.

The Chande Momentum Oscillator (CMO) was included to assess the predominance of upward or downward movements, whereas the Balance of Power (BOP) was used to measure the relative dominance between buyers and sellers throughout the trading session. The Intraday Momentum Index (IMI) was computed based on opening and closing prices.

Additionally, the MACD indicator, as well as its signal line and histogram, were used in order to identify changes in momentum and potential trend reversals. Finally, the Rate of Change (ROC) was employed to measure the percentage rate of price variation.

The preprocessing of technical indicators is a fundamental step, as it facilitates feature comparison, reduces the influence of outliers, and improves algorithm performance. Accordingly, different transformation strategies were applied to the technical indicators, such as normalization, rescaling, or standardization. The exception was the Balance of Power (BOP), which by its mathematical definition already has a fixed range of $[-1, 1]$ centered at zero.

The moving averages EMA, SMA, DEMA, TRIMA, WMA, and TEMA were normalized according to the following equation:

$$x_t^{\text{norm}} = \frac{x_t - \min(x)}{\max(x) - \min(x)} \tag{1}$$

where x_t represents the indicator value at time t, and $\min(x)$ and $\max(x)$ correspond to the minimum and maximum values of the indicator within the considered dataset.

For the ADX, IMI, and ADXR indicators, values were divided by 100, since their original scale ranges from 0 to 100. In the case of CMO and ROC, which do not have strict bounds, division by 100 was also adopted in order to reduce

the amplitude of values and maintain numerical consistency with the other indicators, without imposing a normalization that could alter their dynamics or interpretation.

The MACD indicator, its signal line, and its histogram were transformed using z-score standardization, according to the equation:

$$x_t^{\text{std}} = \frac{x_t - \mu_x}{\sigma_x} \tag{2}$$

where x_t is the indicator value at time t, μ_x is the mean of the indicator values, and σ_x is the standard deviation.

This choice is justified by the fact that MACD does not have fixed bounds and is centered around zero, with variability dependent on the asset's price level and volatility.

3.3 Feature Selection

Feature selection is a technique widely used in machine learning with the objective of identifying irrelevant or redundant attributes, reducing data dimensionality, and improving the performance of predictive models [4]. In this study, feature selection was performed using the SFS method [13], a greedy wrapper-based algorithm in which different machine learning models can be employed as estimators to evaluate subsets of features.

SFS is a method in which feature selection is carried out through the iterative evaluation of the performance of a machine learning model trained on different subsets of features [13]. At each iteration, the attribute that provides the greatest improvement according to a chosen evaluation metric is incorporated into the selected set, until a stopping criterion is satisfied.

In this work, SFS was applied using two distinct estimators: the KNN [10] algorithm and RF [9]. Although the selection procedure remains the same, the choice of estimator influences the final set of selected attributes, since different models capture different patterns in the data. The MLP classifier was not employed as an estimator in the SFS procedure due to its higher computational cost and larger number of hyperparameters, which would significantly increase the execution time of the feature selection stage. By using RF and KNN as estimators, it is also possible to analyze how different learning paradigms influence the subset of selected features.

RF was used as an estimator within the SFS feature selection process. It is an ensemble learning method based on multiple decision trees built from different data samples and random subsets of attributes [19]. This property allows RF to provide a robust assessment of feature relevance by capturing nonlinear relationships and feature interactions.

KNN was also used as an estimator in the SFS process. It is an instance-based learning algorithm in which classification is performed according to the k nearest neighbors in the feature space [20]. In this context, feature relevance is evaluated based on similarity relationships among observations.

In addition, this stage was carried out considering different FHs defined as 1, 5 and 7 candles. The FH represents the temporal distance ahead for which the prediction is performed; for example, a horizon of 1 candle (5 min) indicates an attempt to predict the asset's behavior five min into the future. This strategy allows the predictive behavior of the features to be evaluated at different temporal scales.

Thus, the feature selection procedure was performed independently for each forecast horizon, since feature relevance may vary according to the prediction interval. This approach allows horizon-specific patterns to be captured and avoids imposing a single feature set across different temporal dynamics. The resulting feature subsets were then used as input for the GA-based hyperparameter optimization of the classifiers (MLP, RF, and KNN).

3.4 GA and Classifiers

In this study, the GA was employed as an optimization approach for the selection of hyperparameters of machine learning models, considering different FHs. The use of GA enables an efficient exploration of hyperparameter combinations that would be infeasible to evaluate exhaustively, in addition to reducing the risk of convergence to suboptimal solutions. In this way, GA contributes to obtaining configurations that are more suitable for the different forecast scenarios analyzed.

The GA was independently applied to optimize the hyperparameters of three classifiers: MLP, RF, and KNN. For each FH and corresponding feature subset obtained through SFS, a distinct GA optimization process was executed for each classifier, resulting in horizon-specific and model-specific configurations.

The MLP is a machine learning algorithm based on artificial neural networks, composed of multiple fully connected layers of neurons, including an input layer, one or more hidden layers, and an output layer [21]. Table 1 presents the search space of the MLP hyperparameters optimized by the GA.

Table 1. Search space of MLP hyperparameters optimized by the GA

Parameters	Min	Max/options	Type
Number of hidden layers	1	3	Integer
Base number of Neurons	10	200	Integer
Decay factor	0.4	0.9	Real
Learning rate	1×10^{-5}	5×10^{-3}	Real
Activation function	–	ReLU, Tanh, Logistic	Categorical
Optimizer	–	Adam, SGD	Categorical
Regularization (α)	1×10^{-6}	1×10^{-3}	Real

The number of hidden layers and the base number of neurons are directly related to the depth and representational capacity of the MLP. The decay factor

defines the progressive reduction in the number of neurons across the hidden layers.

The learning rate determines the magnitude of weight updates during the training process, affecting both the convergence speed and the stability of the algorithm. The activation function is responsible for introducing nonlinearities into the neural network, while the optimizer defines the strategy used to adjust the model parameters. Finally, the regularization parameter α controls the penalty applied to the network weights.

Similarly, the GA was used to optimize the main hyperparameters of the RF and KNN classifiers. Tables 2 and 3 present the respective search spaces considered for these models.

Table 2. Search space of KNN hyperparameters optimized by the GA

Parameters	Min	Max/options	Type
Number of neighbors (k)	1	50	Integer
Weight function	–	Uniform, Distance	Categorical
Distance metric	–	Euclidean, Manhattan, Minkowski	Categorical
Minkowski parameter (p)	1	2	Integer

The number of neighbors (k) determines how many closest observations in the feature space are considered when assigning a class to a new sample. The weight function defines whether all neighbors contribute equally (uniform) or whether closer neighbors have greater influence (distance). The distance metric specifies how similarity between observations is computed, while the Minkowski parameter (p) controls the norm used when the Minkowski metric is selected.

Table 3. Search space of RF hyperparameters optimized by the GA

Parameters	Min	Max	Type
Number of trees ($n_estimators$)	50	500	Integer
Maximum depth (max_depth)	5	50	Integer
Minimum samples split	2	20	Integer
Minimum samples leaf	1	20	Integer

The number of trees ($n_estimators$) defines the size of the ensemble. The maximum depth controls the complexity of individual trees. The minimum samples split parameter determines the minimum number of samples required to split an internal node, while the minimum samples leaf specifies the minimum number of samples that must be present in a leaf node.

The GA parameters used in this study are related to the evolutionary process and the genetic operators employed. The parameter NGEN defines the number of generations of the GA, while POPSIZE corresponds to the population size. The crossover probability (CXPB) determines the frequency with which two individuals are combined to generate new solutions. The mutation probability per gene (indpb) controls the chance of each individual parameter undergoing a random change, and the tournament size defines the number of individuals compared during the selection process. Finally, the Hall of Fame size indicates the number of best global solutions preserved throughout the execution of the algorithm. Table 4 presents the basic configuration of the GA parameters adopted in this study.

Table 4. The Relation of GA Parameters

Parameters ID	Values
NGEN	1000
POPSIZE	15
CXPB	0.8
indpb	0.05
Tournament Size	3
HOF size	1

This configuration was kept constant across all FHs and classifiers to ensure comparability of the optimization process.

3.5 Evaluation

The predictive performance of the models was assessed using the macro F1-score as the main evaluation metric. This metric corresponds to the unweighted average of the F1-score computed for each class, ensuring that both classes contribute equally to the final performance measure. This choice is particularly appropriate for financial prediction tasks, where imbalanced class distributions may occur.

During the model development stage, cross-validation was employed on the training data to guide both feature selection and hyperparameter optimization. This procedure allows the models to be evaluated on multiple data partitions, reducing the risk of overfitting and providing a more reliable estimate of generalization performance.

Finally, the selected configurations were evaluated on an independent test set. The results obtained with GA-based optimization were compared with two baseline configurations: a baseline without hyperparameter optimization and a baseline without feature selection. This comparison enables the assessment of the individual contributions of feature selection and hyperparameter optimization to the overall predictive performance.

4 Results

In this section, the obtained results are discussed, considering the distinct FHs (FH = 1, 5, and 7 candles) previously defined. The section is organized into three parts. The first subsection addresses the feature subsets selected by the SFS method. The second subsection presents the classifiers optimized by the GA. Finally, the third subsection compares the performance of the optimized classifiers across the different FHs.

4.1 Feature Selection

Figures 2 and 3 illustrate the relationship between the macro F1-score and the number of selected features obtained through the SFS. The analysis was conducted using two different estimators, namely KNN and RF, across three FHs (FH = 1, 5, and 7 candles).

As shown in Fig. 2, when KNN is used as the estimator, the optimal number of selected features varies significantly across FHs. For FH = 1, the best macro F1-score (0.6333) was achieved with 20 selected features out of the 42 available. However, for FH = 5 and FH = 7, peak performance was obtained with substantially smaller subsets, consisting of only 4 features, yielding macro F1-scores of 0.5476 and 0.5178, respectively. These results suggest that KNN-based SFS tends to favor more compact feature representations for longer FHs.

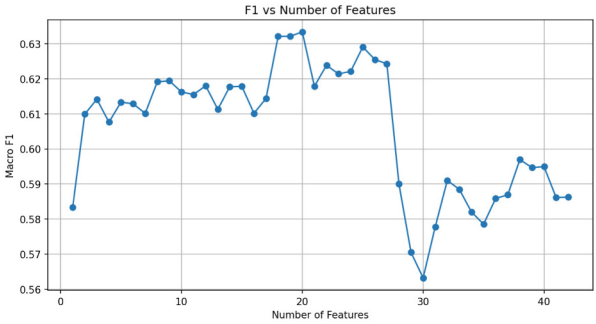
In contrast, Fig. 3 shows that RF-based SFS generally retains a larger number of features. For FH = 1, the highest macro F1-score (0.6611) was achieved with 38 selected features. Similarly, for FH = 5 and FH = 7, the best results (0.5407 and 0.5210, respectively) were obtained with 39 and 34 features. This behavior reflects the ability of RF to handle high-dimensional input spaces and capture nonlinear interactions without substantial performance degradation.

Overall, the results indicate that the choice of estimator significantly influences the dimensionality of the selected feature subsets. While KNN-based SFS tends to produce more compact representations—particularly for longer FHs—RF-based SFS preserves a larger portion of the original feature space, leveraging its ensemble structure to manage model complexity effectively.

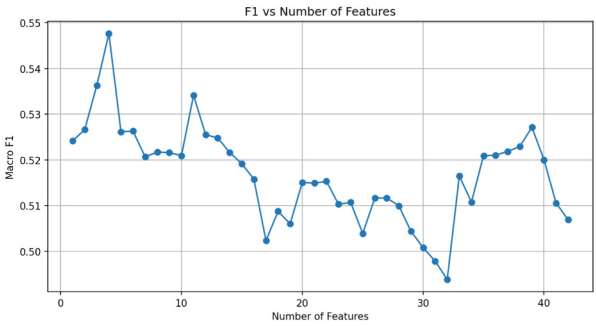
It is important to emphasize that the feature subsets obtained through SFS were used consistently in the subsequent classification stage. For each FH, the MLP was evaluated using both subsets derived from SFS-KNN and SFS-RF, allowing the analysis of how the estimator choice during feature selection influences neural network performance. In contrast, RF and KNN classifiers were trained using the feature subsets selected by their respective estimators, ensuring coherence between the selection mechanism and the inductive bias of each model.

4.2 Classifiers Optimized by GA

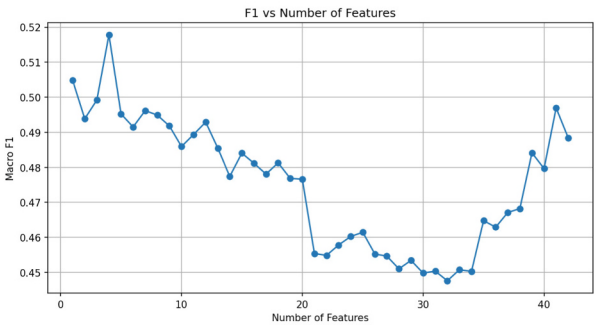
The GA was applied to optimize the hyperparameters of the MLP, KNN, and RF classifiers. For each FH and feature selection configuration, an independent



(a) FH = 1



(b) FH = 5

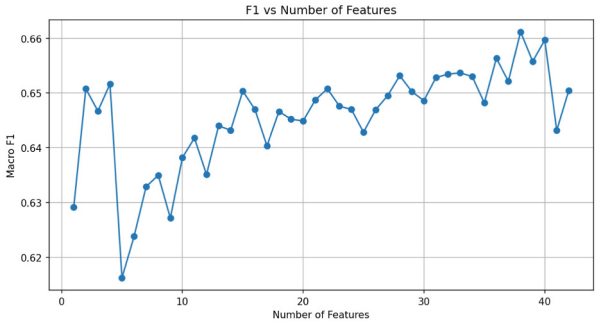


(c) FH = 7

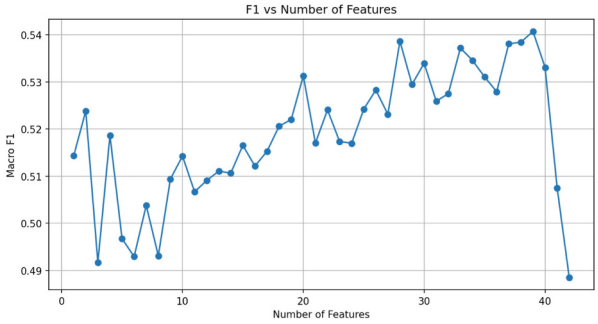
Fig. 2. Macro F1-score versus number of selected features using SFS with KNN as estimator.

GA instance was executed, resulting in distinct optimization processes and final configurations.

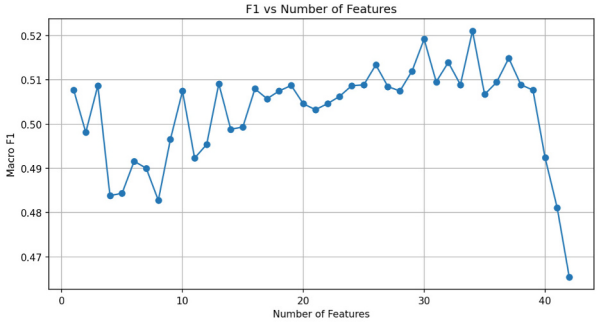
Due to space limitations, Figs. 4 and 5 present the convergence behavior of the GA only for the MLP classifier, considering feature subsets obtained through SFS with KNN and RF estimators, respectively. The convergence patterns observed



(a) FH = 1



(b) FH = 5



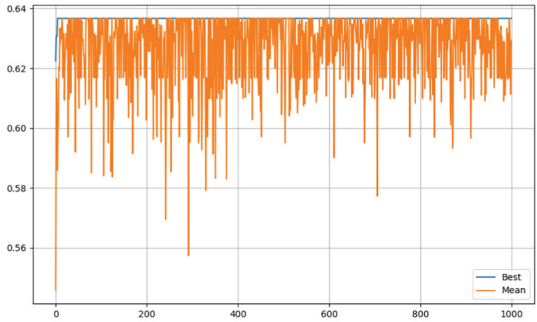
(c) FH = 7

Fig. 3. Macro F1-score versus number of selected features using SFS with RF as estimator.

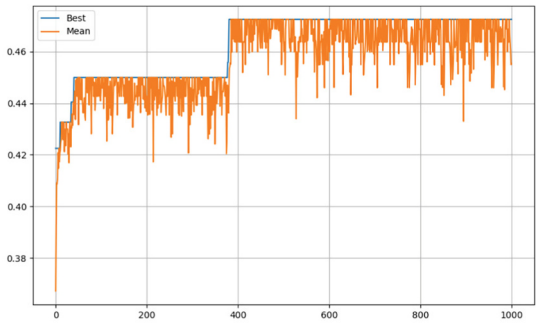
for KNN and RF exhibited similar stabilization behavior and are therefore omitted for brevity.

Tables 5, 6, and 7 summarize the best hyperparameter configurations identified by the GA for each classifier and FH.

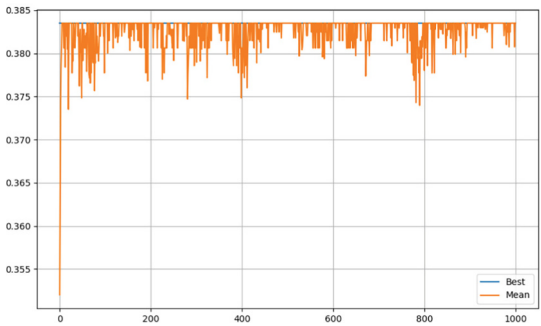
For the MLP, substantial architectural variation can be observed across horizons and feature selection strategies. When using SFS-KNN features, deeper



(a) FH = 1 candle

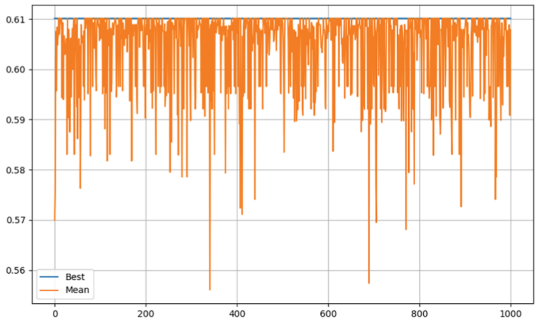


(b) FH = 5 candles

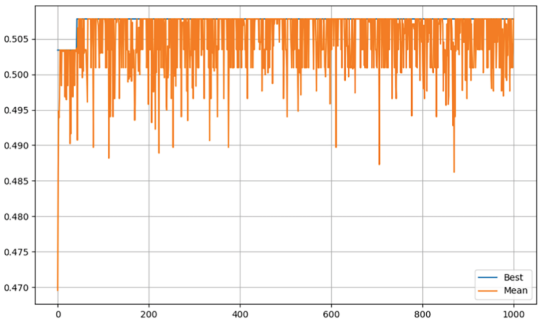


(c) FH = 7 candles

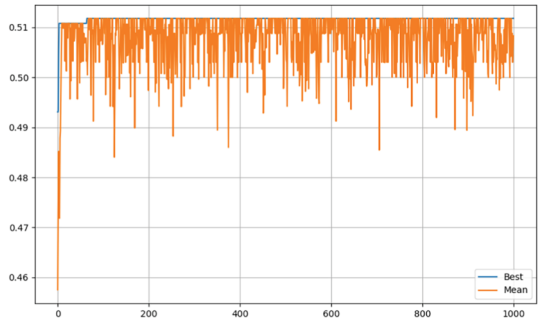
Fig. 4. GA convergence during MLP hyperparameter optimization using SFS + KNN feature subsets.



(a) FH = 1 candle



(b) FH = 5 candles



(c) FH = 7 candles

Fig. 5. GA convergence during MLP hyperparameter optimization using SFS + RF feature subsets.

Table 5. Best MLP hyperparameters obtained by the GA for each configuration

SFS Estimator	FH	Layers	Hidden units	LR	α	Activation	Solver
KNN	1	3	(172, 134, 104)	0.00137	1.29×10^{-4}	ReLU	Adam
KNN	5	3	(139, 119, 102)	0.00107	1.00×10^{-3}	ReLU	Adam
KNN	7	1	(155)	0.00500	1.00×10^{-6}	Tanh	Adam
RF	1	2	(65, 57)	0.00052	3.81×10^{-4}	Tanh	SGD
RF	5	1	(10)	0.00390	1.00×10^{-6}	ReLU	Adam
RF	7	1	(63)	0.00155	1.63×10^{-4}	ReLU	Adam

Table 6. Best KNN hyperparameters obtained by the GA for each FH

Horizon	k	Weights	Metric	p
1	35	Distance	Euclidean	1
5	1	Uniform	Euclidean	1
7	14	Uniform	Manhattan	1

architectures were favored for FH = 1 and FH = 5, whereas a simpler single-layer structure was selected for FH = 7. In contrast, when SFS-RF features were adopted, the GA generally selected more compact architectures, suggesting that the larger feature subsets may reduce the need for deeper representations.

Regarding KNN, the optimal number of neighbors varied significantly across horizons, indicating sensitivity to temporal scale. While FH = 1 favored a larger neighborhood ($k = 35$), longer horizons resulted in smaller or moderate values of k, reflecting differences in local structure within the feature space.

For RF, the GA selected distinct combinations of tree depth and regularization-related parameters. Shorter horizons tended to favor shallower trees, whereas longer horizons required deeper structures and adjusted splitting constraints, highlighting horizon-dependent model complexity.

Overall, the results reinforce the importance of horizon-specific hyperparameter optimization. Even under a fixed GA configuration, the search process consistently identified distinct model structures and parameter settings tailored to each predictive scenario.

Table 7. Best RF hyperparameters obtained by the GA for each FH

Horizon	n_estimators	max_depth	min_samples_split	min_samples_leaf
1	269	5	5	1
5	257	6	20	13
7	328	10	19	4

4.3 Classifiers Performance

This subsection evaluates the predictive performance of the proposed models across different forecasting horizons. In addition to the GA-optimized configurations, two baseline scenarios are considered in order to assess the impact of hyperparameter optimization and feature selection on the final performance.

Table 8 presents the macro F1-score obtained by the GA-optimized classifiers, together with two baseline configurations: a baseline model using the same feature subsets but without GA optimization, and a baseline without feature selection.

Table 8. Performance comparison between GA-optimized models and baseline configurations

Model	SFS	FH	F1 GA	F1 Baseline	F1 No FS
MLP	KNN	1	0.632	0.644	0.583
MLP	KNN	5	0.466	0.338	0.497
MLP	KNN	7	0.377	0.337	0.488
MLP	RF	1	0.613	0.594	0.583
MLP	RF	5	0.497	0.512	0.497
MLP	RF	7	0.480	0.483	0.488
RF	RF	1	0.668	0.640	0.591
RF	RF	5	0.537	0.523	0.487
RF	RF	7	0.518	0.502	0.477
KNN	KNN	1	0.672	0.633	0.587
KNN	KNN	5	0.523	0.548	0.505
KNN	KNN	7	0.508	0.518	0.481

For FH = 1, all models achieved their best performance. The GA-optimized KNN obtained the highest macro F1-score (0.672), closely followed by RF (0.668). The MLP models exhibited slightly lower performance, with F1-macro values of 0.632 when using SFS-KNN features and 0.613 with SFS-RF features. These results indicate that, for short-term prediction, instance-based and ensemble methods provide more effective decision boundaries than the neural network configuration explored.

As the FH increases (FH = 5 and FH = 7), a consistent performance degradation can be observed across all classifiers. This behavior suggests that longer-term predictions introduce greater uncertainty and reduce the separability of the classes. Although RF maintained a slight advantage over KNN for FH = 5 and FH = 7, the differences among the models become less pronounced as the horizon grows.

Regarding the MLP, the use of SFS-RF features generally resulted in more stable performance across horizons compared to SFS-KNN features, particularly

for FH = 7. Nevertheless, the neural network did not outperform the simpler models in any scenario, even after GA-based hyperparameter optimization. This behavior may be partially explained by the feature selection procedure. Since SFS was conducted using KNN and RF as estimators, the resulting feature subsets are likely biased toward these models, which may disadvantage the MLP during the classification stage. This result also shows how the choice of estimator in SFS can influence the relative performance of different classifiers.

The comparison with the baseline configurations also highlights the importance of feature selection and hyperparameter optimization. In particular, the RF showed consistent improvements over its baseline configuration across all forecasting horizons when GA-based optimization was applied. In most cases, the models trained without feature selection achieved lower F1 scores, indicating that the reduction of the feature space contributed to more robust decision boundaries.

Overall, the results demonstrate that model performance is strongly horizon-dependent. Short-term forecasting (FH = 1) yields significantly higher predictive quality, whereas longer horizons lead to near-random performance levels for all evaluated classifiers. Among the tested approaches, the GA-optimized KNN achieved the best overall result, while RF showed competitive and stable behavior across different temporal scales.

5 Conclusions

Intraday financial forecasting remains a challenging task due to the noisy and dynamic nature of market data. In this context, selecting relevant features and properly configuring predictive models are key aspects for improving classification performance.

This work proposed a methodological workflow for intraday classification that combines technical indicators, horizon-dependent feature selection using SFS, and GA-based hyperparameter optimization applied to different classifiers (MLP, RF, and KNN). The feature selection stage was performed independently for each forecasting horizon, allowing the models to adapt to different temporal dynamics.

The experimental results indicate that predictive performance is strongly dependent on the forecasting horizon. The best results were consistently obtained for FH = 1, while performance gradually degraded as the horizon increased. Among the evaluated models, the GA-optimized KNN achieved the highest overall performance, with RF also presenting competitive and stable results. In contrast, the MLP did not outperform the simpler approaches despite the evolutionary optimization, suggesting that model complexity does not necessarily mean better performance under the adopted feature selection strategy.

Overall, the findings suggest that horizon-aware feature selection plays a central role in predictive performance, often having a greater impact than increases in model complexity. Future works may explore alternative feature selection strategies, different hyperparameter optimization methods, and the adoption of

modern deep learning models along with more comprehensive evaluation protocols, including real-world performance and statistical validation.

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. Wilmott, P.: Paul Wilmott Introduces Quantitative Finance. John Wiley & Sons (2013)
2. Malkiel, B.G.: The efficient market hypothesis and its critics. J. Econ. Perspect. **17**(1), 59–82 (2003)
3. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York (2009)
4. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**, 1157–1182 (2003)
5. Whitley, D.: A genetic algorithm tutorial. Stat. Comput. **4**(2), 65–85 (1994)
6. Yao, X.: Evolving artificial neural networks. Proc. IEEE **87**(9), 1423–1447 (1999)
7. Souza, A.S., Lucca, G., Borges, E.N., Cardoso, F.C., Dalmazo, B.L., Berri, R.: Dataset for intraday analysis of b3 stock prices (BovDB.v2) (2024). https://doi.org/10.7910/DVN/TMB4IG
8. Zhang, G.P., Patuwo, B.E., Hu, M.Y.: Forecasting with artificial neural networks: the state of the art. Int. J. Forecast. **14**(1), 35–62 (1998)
9. Cutler, A., Cutler, D.R., Stevens, J.R.: Random forests. In: Ensemble Machine Learning, pp. 157–175. Springer (2012)
10. Peterson, L.E.: K-nearest neighbor. Scholarpedia **4**(2), 1883 (2009)
11. Loh, W.-Y.: Feature selection for classification of high-dimensional data. Stat. Sin. **19**(1), 1–21 (2009)
12. Atsalakis, G.S., Valavanis, K.P.: Surveying stock market forecasting techniques-Part II: soft computing methods. Expert Syst. Appl. **36**(3), 5932–5941 (2009)
13. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artif. Intell. **97**(1–2), 273–324 (1997)
14. Goldberg, D.E.: Genetic Algorithms in Search. Optimization and Machine Learning. Addison-Wesley, Reading (1989)
15. Brabazon, A., O'Neill, M., Maringer, D.: Natural Computing in Computational Finance. Springer (2015)
16. Murphy, J.J.: Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications. New York Institute of Finance, New York (1999)
17. Chande, T.S., Kroll, S.: The New Technical Trader. John Wiley & Sons (1994)
18. Kaufman, P.J.: Trading Systems and Methods. John Wiley & Sons (2013)
19. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
20. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory **13**(1), 21–27 (1967)
21. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**(6088), 533–536 (1986)