Swarm and Evolutionary Computation 99 (2025) 102211

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Multi-objective evolutionary feature selection for ensemble learning with random forests in time series forecasting

Raquel Espinosa [b], Gracia Sánchez [a], José Palma [a,b], Fernando Jiménez [a,b,*]

[a] Department of Information and Communication Engineering, University of Murcia, Spain
[b] Artificial Intelligence and Knowledge Engineering Research Group, University of Murcia, Spain

ARTICLE INFO

Keywords:
Time series forecasting
Feature selection
Multi-objective evolutionary algorithms
Random forest
Ensemble learning
Stacking

ABSTRACT

Time series forecasting is fundamental in numerous domains, including finance, healthcare, energy, and environmental monitoring. However, the high dimensionality of feature spaces can lead to overfitting and reduced interpretability, making feature selection a crucial preprocessing step. This paper proposes a multi-objective evolutionary algorithm for feature selection in time series forecasting, designed to enhance predictive accuracy while improving generalization. The method partitions the dataset, associating each partition with an objective function in the optimization process. By independently selecting relevant feature subsets, it generates a Pareto front of Random Forest models, each trained on a distinct subset of features. These models are then aggregated into a stacking-based ensemble framework, effectively balancing feature relevance and diversity. Additionally, we introduce a feature importance measure based on selection frequency in the non-dominated solutions of the optimization process. To validate our approach, we conduct experiments on real-world forecasting tasks, including air quality prediction in southeastern Spain and Italy and oil temperature forecasting in industrial applications. We also evaluate performance on synthetic datasets of increasing complexity, systematically varying instances, features, seasonality, noise, and trends. The proposed method is compared against conventional Random Forest, a wrapper-based feature selection method with a multi-objective evolutionary search strategy, and several state-of-the-art embedded feature selection techniques for time series forecasting. The results demonstrate that our approach significantly improves forecasting accuracy while mitigating overfitting. By integrating multi-objetive evolutionary optimization, random forest, ensemble learning, and a novel feature importance measure, our method offers a robust, interpretable, and effective feature selection for time series forecasting applications.

1. Introduction

Time series forecasting (TSF) [1] is a fundamental analytical approach that leverages historical data to predict future values in a sequence. This methodology plays a crucial role across diverse sectors, offering valuable insights that drive strategic planning, risk management, and operational efficiency. In finance and economics, TSF is widely used to predict stock prices, market trends, and volatility, enabling businesses and investors to make informed decisions [2]. Accurate forecasts are essential for portfolio management and investment strategies [3]. Governments utilize time series analysis to evaluate the impact of past policies and anticipate the effects of new initiatives, aiding economic planning and resource allocation [4]. Additionally, TSF contributes to risk assessment by predicting fluctuations in asset prices and interest rates [5]. In the business sector, companies employ TSF to forecast product demand, optimizing inventory levels, reducing costs, and enhancing service delivery — key factors for maintaining supply chain efficiency. Furthermore, it supports production planning, resource utilization, fault diagnosis, and anomaly detection in manufacturing processes [6,7]. Public health applications of TSF include predicting disease outbreaks, improving preparedness and response strategies [8], and forecasting hospital admissions to optimize staffing and bed allocation [9]. Similarly, energy companies rely on TSF to anticipate demand, ensuring efficient power generation and distribution, which is critical for grid stability and energy optimization [10]. Accurate energy price forecasts also facilitate strategic planning and hedging, particularly in markets influenced by fluctuating resources such as oil and natural gas [11]. In telecommunications, TSF aids in forecasting network traffic, enabling efficient capacity management and enhancing service quality [12]. Businesses leverage predictive analytics to anticipate customer needs, improve service offerings, and

* Correspondence to: Facultad de Informática, Campus de Espinardo, Edificio 32, 30100 Murcia, Spain.
E-mail address: fernan@um.es (F. Jiménez).

https://doi.org/10.1016/j.swevo.2025.102211
Received 24 March 2025; Received in revised form 22 September 2025; Accepted 28 October 2025
Available online 10 November 2025

refine marketing strategies for sustainable growth [13]. Environmental sciences also benefit from TSF, as it is used to predict weather patterns and climate change, supporting disaster preparedness and environmental planning [14,15]. Forecasting air quality indices is vital for issuing health advisories and minimizing exposure to pollutants [16]. Moreover, TSF assists regulatory agencies in monitoring air pollution trends, evaluating the effectiveness of control measures, and guiding urban planning decisions aimed at reducing environmental impact [17,18]. Sustainable management of natural resources is another crucial application, as TSF helps estimate resource usage and depletion rates [19]. In summary, TSF is an indispensable tool across multiple domains, empowering organizations and policymakers with predictive insights that enhance decision-making. Beyond navigating complex market dynamics, TSF fosters sustainability, operational efficiency, and strategic foresight, making it a cornerstone of modern data-driven analysis.

To effectively forecast time series using machine learning techniques, *feature selection* (FS) [20] is a crucial preprocessing step. FS involves selecting a subset of relevant features from the original input variables to enhance model performance and interpretability. By eliminating irrelevant or redundant features, FS reduces overfitting, improves generalization, and accelerates the learning process. FS methods can be categorized based on two main criteria. The first distinction is between *attribute evaluation methods* and *attribute subset evaluation methods*. Attribute evaluation methods assess each feature independently and can be further divided into *univariate* and *multivariate* approaches. These methods compute a ranking score or measure for each feature, which is why they are also referred to as *feature ranking* (FR) methods. In contrast, attribute subset evaluation methods evaluate groups of features together, always considering feature interactions, making them inherently multivariate. Another common classification divides FS techniques into *filter*, *wrapper*, and *embedded methods*. Filter methods assess the relevance of individual features or feature subsets independently of any learning algorithm, relying on statistical measures such as correlation, consistency, and redundancy. These methods are computationally efficient and scalable to large datasets. Wrapper methods, on the other hand, evaluate feature subsets using a specific learning algorithm, optimizing selection based on model performance. While often more accurate than filter methods, they can be computationally expensive, particularly for high-dimensional datasets. Embedded methods integrate feature selection directly into the model training process, balancing efficiency with the ability to capture feature interactions. These methods are particularly effective when the selection criteria align with the learning algorithm's objective function. The choice of FS method depends on factors such as dataset size, feature space complexity, computational constraints, and the trade-off between model accuracy and efficiency. Selecting the appropriate FS technique is essential for optimizing time series forecasting models and ensuring their interpretability and predictive performance.

Multi-objective evolutionary algorithms (MOEAs) [21] are optimization techniques inspired by natural evolution that are particularly effective for solving problems involving multiple conflicting objectives. Unlike single-objective optimization, where a single best solution is sought, MOEAs aim to approximate the Pareto front, a set of trade-off solutions where no objective can be improved without degrading another. These algorithms leverage mechanisms such as selection, mutation, and crossover to explore and exploit the search space efficiently. In the context of FS, MOEAs provide a powerful search strategy by simultaneously optimizing multiple objectives, such as minimizing prediction error while reducing the number of selected features. This approach balances model accuracy and complexity, preventing overfitting and improving interpretability. By exploring diverse subsets of features in parallel, MOEAs enhance the robustness of FS methods, making them particularly well-suited for high-dimensional datasets and TSF applications.

Ensemble learning (EL) [22] is a machine learning approach that enhances predictive performance by combining multiple models rather than relying on a single one. By leveraging the strengths of diverse learners, ensemble methods improve generalization, reduce overfitting, and increase robustness against noise. One widely used ensemble strategy is *stacking*, which integrates multiple base models by training a meta-model to aggregate their predictions optimally. Unlike simple averaging or voting techniques, stacking learns how to assign weights to different models based on their performance, leading to more accurate and reliable forecasts. In TSF, EL is particularly beneficial as it allows the combination of models that capture different temporal patterns and dependencies, ultimately improving prediction accuracy and stability.

Random Forest (RF) [23] is a powerful EL method that utilizes *bagging* to improve predictive performance, making it widely used for TSF. In addition to leveraging multiple decision trees to enhance predictive performance, RF also performs embedded feature selection by assessing the importance of each feature during training, enabling the identification of the most relevant predictors. In TSF with machine learning, raw time series data must be reformulated into a supervised learning format, where past observations (*lagged variables*) are used as predictors for future values. The *sliding window* technique accomplishes this by structuring the dataset into overlapping windows of past values that serve as input features, allowing machine learning models to capture temporal dependencies effectively. RF, being an ensemble of decision trees with embedded FS, is particularly well-suited for this task due to its ability to model complex, nonlinear relationships and its robustness to noise. However, despite its advantages, RF is prone to *overfitting*, especially when dealing with high-dimensional feature spaces generated by sliding window transformations. The inclusion of excessive lagged variables or irrelevant features can lead to unnecessarily complex trees that memorize training data rather than generalizing well to unseen samples. This issue is exacerbated when working with limited historical data, as RF may capture spurious patterns rather than true underlying trends. Therefore, an effective FS strategy is essential to mitigate overfitting and enhance model interpretability.

To address this challenge, we propose a MOEA for FS in TSF, referred to as MOEFS-ELRF (*Multi-Objective Evolutionary Feature Selection for Ensemble Learning with Random Forest*). Our approach partitions the dataset and associates each partition with an objective function in the MOEA, optimizing relevant feature subsets independently within each partition using RF as the learning algorithm (wrapper-based FS). The MOEA identifies a Pareto front of RF models, each trained with a distinct feature subset. These models are then integrated into an EL framework via stacking to construct the final TSF model. By distributing the FS task across multiple objectives and leveraging an ensemble of specialized RF models, our method enhances predictive accuracy, improves generalization, and mitigates overfitting. The main contributions of this work are:

1. **A novel multi-objective feature selection strategy for TSF**, where the training dataset is partitioned, and each partition is associated with an objective function in the MOEA. This formulation enables the discovery of diverse feature subsets tailored to different data segments, resulting in a set of non-dominated RF models that are subsequently combined through stacking-based EL to enhance generalization and reduce overfitting. An additional strength of the proposed approach lies in its simplicity and versatility. The strategy only requires the execution of a single MOEA instance, without the need for complex or problem-specific algorithmic designs as seen in other EL frameworks proposed in the literature for TSF. Furthermore, while our approach does not depend on a specific MOEA implementation in its design, the current experimental validation has been explicitly conducted with Pareto-based (NSGA-II/III) and decomposition-based (MOEA/D) evolutionary algorithms. This means that the demonstrated effectiveness of the framework is limited to these classes of MOEAs. Nonetheless, this design choice offers practical versatility: practitioners can select among

well-established Pareto- or decomposition-based MOEAs according to the number of partitions/objectives in their TSF task, without requiring complex algorithmic modifications. This balance between flexibility, simplicity, and empirical effectiveness reinforces the practical relevance of the proposed method.

2. **A new measure of feature importance**, derived from the frequency of feature selection across the non-dominated solutions identified by the MOEA. This strategy offers a straightforward yet effective way to quantify feature relevance in TSF models, enhancing model interpretability and supporting decision-making processes. To validate the effectiveness and robustness of this feature importance measure, we conducted a comparative analysis against widely accepted model-agnostic and post-hoc feature importance techniques, such as correlation analysis and permutation feature importance. The comparative results confirmed the high consistency and reliability of the proposed approach, demonstrating its capability to accurately identify key predictive features while maintaining low redundancy with respect to other input variables.
3. **Comprehensive experimental validation**, encompassing real-world time series forecasting problems, including air quality forecasting in southeastern Spain and Italy, as well as an oil temperature prediction task within the framework of research projects led by the authors. Additionally, the method has been rigorously tested on five synthetic TSF datasets of increasing complexity, systematically varying key factors such as the number of features, seasonality, noise, and trends. The proposed approach has been compared against conventional RF, a wrapper-based FS method using an MOEA and RF, and several state-of-the-art embedded FS techniques for TSF. Furthermore, the robustness of the proposed methodology has been reinforced by including a dedicated comparative analysis of the underlying MOEA, evaluating the performance of NSGA-II, NSGA-III, and MOEA/D under identical experimental conditions. The experiments also included an empirical study to determine the optimal number of partitions/objectives, as well as an analysis of the interaction between the feature dimensionality and the number of partitions. Finally, the scalability of the method was extensively assessed, analyzing its behavior in terms of runtime and memory consumption as the feature space and the forecasting horizon increase, further demonstrating the practicality and applicability of the proposed approach for complex and large-scale TSF scenarios.

The rest of the manuscript has been organized as follows: Section 2 describes the most important related works of the state of the art; Section 3 describes the main components of the proposed MOEA for FS; Section 4 shows the performed set of experiments; Section 5 analyses and discusses the results; finally, Section 6 draws the main conclusions of the research and points out future work.

2. Related works

This section presents the state of the art from the last 4 years of work on FS, evolutionary search, multi-objective optimization and ensemble learning.

Prakash and Karthikeyan [24] propose an enhanced evolutionary FS method combined with a hybrid ensemble model to improve the prediction of cardiovascular disease. The study highlights the importance of selecting relevant attributes for accurate classification, addressing a gap in existing research where FS has been insufficiently explored. The proposed approach optimizes classification performance by integrating evolutionary algorithms with EL techniques. Experimental results on multiple datasets (Statlog, SPECTF, and coronary heart disease) demonstrate higher precision, recall, and accuracy compared to state-of-the-art methods, achieving a maximum accuracy of 93.65%, 82.81%, and 84.95%, respectively. Performance evaluation using ROC curves further validates the effectiveness of the proposed method, reinforcing its potential for automated cardiovascular disease diagnosis in medical applications.

García-Pedrajas and Cerruela-García [25] introduce MABUSE, a novel FS method that optimizes classification margins using a filter-based approach while incorporating elements of wrapper methods. Unlike traditional margin-based FS techniques, which are often embedded within specific classifiers, MABUSE selects feature subsets that enhance generalization across multiple classifiers. The method employs a nearest-neighbor margin definition and integrates boosting-inspired heuristic search to refine FS. Experimental results demonstrate that MABUSE outperforms state-of-the-art algorithms in both classification accuracy and feature reduction while maintaining a computational cost comparable to traditional filter-based methods. Additionally, several heuristic enhancements are incorporated to improve scalability for large datasets, making the approach suitable for high-dimensional learning tasks.

Hao et al. [26] propose a novel air pollutant concentration prediction system designed for environmental system management, integrating both prediction accuracy and robustness. The system consists of four key modules: time series reconstruction, submodel simulation, weight search, and integration. A decomposition-ensemble mode is introduced to filter out high-frequency noise and reconstruct time series, improving data quality. Additionally, a novel weight search mechanism is developed to optimize the trade-off between prediction precision and stability, addressing limitations of previous single-objective optimization approaches. The system combines four predictive models–convolutional neural network, extreme learning machine, Elman neural network, and long short-term memory–to leverage their complementary strengths and enhance forecasting performance. Experimental validation on hourly PM2.5 concentration data from three major Chinese cities confirms the system's effectiveness, demonstrating its potential as a decision-support tool for air pollution control and policy-making.

Vivek et al. [27] propose a parallel MOEA for feature subset selection in big data, addressing challenges related to poor convergence and local optima in high-dimensional datasets. The method optimizes three objectives: feature subset cardinality, area under ROC curve, and Matthews correlation coefficient, using a fractional dominance-based MOEA under the Spark framework. To enhance exploitation, they introduce a batch opposition-based learning strategy, with two variants: BOP1, which selects a single opposite neighbor, and BOP2, which selects multiple opposite neighbors to improve search efficiency. A novel fractional dominance-based migration strategy is proposed for parallel execution under an island-based framework. Additionally, the study introduces HV/IGD, a new metric combining Hypervolume and Inverted Generational Distance to evaluate diversity and convergence in MOEAs. Experimental results demonstrate that BOP2 outperforms competing approaches in achieving optimal solutions, while BOP1 excels in diversity-convergence trade-offs.

Zhou et al. [28] introduces an efficient EL method based on multi-objective FS to enhance both the accuracy and diversity of individual classifiers within the ensemble. The proposed approach employs a hybrid filter-clustering-wrapper FS strategy to generate non-dominated feature subsets, which are then used to train diverse and accurate individual classifiers. Additionally, it introduces a novel diversity metric based on feature relevance, allowing for an optimized ensemble selection process that simultaneously considers diversity and accuracy. The ensemble selection problem is formulated as an optimization task, and an efficient algorithm is developed to select the optimal set of individual classifiers. Experimental results on public datasets and a real-world case study demonstrate the effectiveness of the method, achieving high predictive accuracy without compromising diversity.

Zhou et al. [29] propose NGO-XGBoost, a hybrid model combining the Northern Goshawk Optimization algorithm with XGBoost to enhance the estimation of reference evapotranspiration under limited data

conditions. The study employs an ensemble embedded feature selection method, integrating results from RF, XGBoost, AdaBoost, and CatBoost, to determine the most relevant meteorological factors for evapotranspiration estimation. Experiments using data from 30 meteorological stations in the North China Plain demonstrate that NGO-XGBoost consistently outperforms RF, XGBoost, and KNN models, achieving higher accuracy across different seasonal conditions. The model maintains high precision even when trained on historical data from neighboring stations, making it a reliable tool for evapotranspiration estimation in regions with incomplete meteorological data.

Fan et al. [30] propose a hybrid ensemble deep learning model with integrated error correction to enhance the accuracy of short-term industrial load forecasting, a task complicated by the volatile and complex nature of industrial power demand. Their approach operates in two phases. In the first phase, high-dimensional multivariate data are partitioned into sub-datasets, and deep power features are extracted using a hybrid ensemble framework combining Random Subspace, Boosting, Ensemble Pruning, and a Multi-Objective Molecular Dynamics Theory Optimization Algorithm (MMDTOA). This algorithm perturbs the parameters of GRU networks to promote both diversity and accuracy among base models, which are then aggregated using kernel ridge regression stacking. In the second phase, a combined error correction mechanism–based on a dynamic Gaussian error function and a GRU-based residual model–is applied to refine the predictions. Experimental results on real Korean datasets demonstrate that the proposed model achieves superior performance in terms of NMAE and NRMSE compared to several baseline methods including SVM, ELM, and CNN.

Fan et al. [31] propose a multi-stage ensemble learning model for short-term power load forecasting that integrates decomposition techniques, error factor modeling, and a multi-objective evolutionary algorithm based on decomposition (MOEA/D). The methodology is structured in three stages. First, data is decomposed using complete ensemble empirical modal decomposition with adaptive noise (CEEMDAN), and the resulting components are combined with the original dataset to enhance feature representation. Second, the GRU network parameters are optimized using a customized MOEA/D that incorporates an angle-distance selection strategy and adaptive population generation, targeting both accuracy and model diversity. This stage yields both forecasting and error prediction models. Finally, a nonlinear integration mechanism based on GRU aggregates the forecasted values and their associated error predictions, improving final output accuracy. Experiments on Australian electricity market datasets demonstrate that the proposed approach offers superior performance in terms of accuracy and generalization compared to several baseline models.

2.1. Conclusions of related works

FS plays a crucial role in improving the performance of machine learning models by eliminating redundant or irrelevant attributes, enhancing interpretability, and reducing computational costs. Several approaches have been explored in the literature, incorporating EL, evolutionary algorithms, and multi-objective optimization to refine FS techniques. Recent studies have demonstrated the effectiveness of MOEAs in optimizing FS problems by balancing accuracy and feature reduction. In [27], a parallel fractional dominance-based MOEA for FS in big data is proposed, integrating novel opposition-based learning strategies to improve convergence and diversity. Similarly, MABUSE is introduced in [25], a margin optimization-based FS method that blends filter and wrapper strategies to improve generalization across classifiers. These approaches highlight the advantages of evolutionary-based FS but do not leverage EL for improved generalization. Ensemble-based FS methods have also gained attention, particularly in TSF and environmental modeling. An air pollutant prediction system integrating decomposition-ensemble learning and multi-objective optimization is proposed in [26], demonstrating the effectiveness of ensemble approaches for complex forecasting tasks. In [29], a hybrid model combining FS with extreme gradient boosting (XGBoost) for evapotranspiration estimation is introduced, emphasizing the importance of integrating FS techniques with robust predictive models. Similarly, an evolutionary FS method with hybrid EL for cardiovascular disease prediction is developed in [24], achieving superior classification performance. In [28], a framework is proposed that integrates filter-based FS, clustering, wrapper FS, multi-objective evolutionary optimization, and ensemble learning. However, these processes are performed in batch mode, the multi-objective optimization problem differs as it follows the conventional approach of maximizing accuracy and minimizing feature subset cardinality, and the ensemble learning is constructed using different learning algorithms in a traditional way. Therefore, although this work incorporates machine learning components similar to those in our approach, the way they are utilized is entirely different, and the method is designed for classification tasks and has not been applied to TSF. Although a method that combines EL, MOEAs, and training data partitioning is also proposed in [30], the approach significantly differs from ours in multiple fundamental aspects. First, their method partitions the training set by randomly dividing the feature space, whereas our proposal partitions the dataset at the sample level, preserving the temporal structure critical in time series forecasting. Second, their strategy does not involve an explicit FS process; instead, they randomly generate subsets of features and train separate MOEAs on each one. Each MOEA produces a single final GRU model, and the ensemble is built from these individually optimized models. In contrast, our method uses a single MOEA, designed specifically for FS, where each objective function corresponds to a sample-based partition of the data. The ensemble is then constructed from all non-dominated solutions (each associated with a RF model) identified by the MOEA. This makes our approach not only more interpretable and targeted to feature relevance but also less complex in terms of optimization strategy and computational overhead. Finally, although MOEAs and EL are also employed in [31], the design is substantially more complex and follows a different conceptual framework than ours. The proposed model operates in three stages: it begins by decomposing the time series using CEEMDAN and augmenting the dataset with both the original and decomposed components. In the second stage, a variant of MOEA/D is applied to optimize the parameters of GRU models, targeting both prediction accuracy and model diversity. This process results in multiple forecasting and error prediction models. In the final stage, the outputs of these models are integrated using a GRU-based nonlinear fusion mechanism that explicitly incorporates residual error information to enhance forecasting accuracy. In contrast, our approach is simpler and more focused on FS. We apply a single MOEA to evolve feature subsets across multiple temporal partitions of the training data. The resulting non-dominated RF models are combined using a stacking-based ensemble, with no need for signal decomposition, error modeling, or additional layers of integration. Moreover, our method includes a built-in mechanism for interpretability by computing feature importance based on selection frequency across non-dominated solutions. Overall, while both approaches utilize evolutionary optimization and ensemble techniques, the methodology we propose is less complex, more interpretable, and specifically designed to address the challenge of FS in TSF.

While these methods illustrate the benefits of MOEAs, EL, and FS, none of them fully integrate multi-objective evolutionary FS with ensemble-based modeling in a way that optimally enhances TSF. The existing literature also lacks an effective feature importance analysis that ensures robustness and interpretability in the FS process. To address these gaps, we propose MOEFS-ELRF, a novel FS approach that combines multi-objective evolutionary optimization, FS, EL, and allows the estimation of feature importance. By partitioning the dataset and associating each subset with an objective function, MOEFS-ELRF

enhances generalization and avoids overfitting while maintaining computational efficiency. The integration of stacking-based EL further improves predictive performance compared to standalone FS or single-model approaches. Our method builds upon the strengths of previous research while addressing their limitations, providing a robust and interpretable FS strategy for TSF and other complex predictive tasks.

3. A multi-objective evolutionary feature selection method for ensemble learning with random forests

This section describes the proposed method in detail. Section 3.1 outlines the general framework of the approach. Section 3.2 delves into the key components of the multi-objective evolutionary algorithm. Section 3.3 introduces the stacking-based EL strategy used to construct the final forecasting meta-model from the set of non-dominated RF models identified by the evolutionary process. Section 3.4 presents the multi-step forecasting strategy employed to generate h-step ahead predictions. Finally, Section 3.5 details the proposed methodology for assessing feature importance.

3.1. General framework

Next, we present the general framework of MOEFS-ELRF, which is visually represented as a flowchart in Fig. 1 and outlined in pseudocode in Algorithm 1. We consider a time series dataset $D = \{\{\mathbf{d}_1, o_1\}, \ldots, \{\mathbf{d}_r, o_r\}\}$ with r samples. We assume that D has been previously transformed using the *sliding window method* [32] with a window size W. Each sample $\{\mathbf{d}_t, o_t\}$, where $\mathbf{d}_t = \{d_t^1, \ldots, d_t^q\}$ for $t = 1, \ldots, r$, consists of q input attributes and a single numeric output attribute o_t (target). Furthermore, we assume that the dataset D has been partitioned into two subsets: R for training and T for testing. The test dataset T, which contains the last $\lceil 0.2 \cdot r \rceil$ samples of D (20% of the total data), is excluded from the training process and is solely used for evaluation at prediction time. To prevent *data leakage*, standardization is performed after splitting the data, using only the statistics computed from the training set to standardize both the training and test sets.

Algorithm 1: MOEFS-ELRF

Input: R {Training dataset}
Input: T {Test dataset}
Input: n {Number of training partitions}
Input: $P > 1$ {Population size}
Input: $G > 1$ {Number of generations}
Input: $0 \leq p_c \leq 1$ {Crossover probability}
Input: $0 \leq p_m \leq 1$ {Mutation probability}
Output: H {h-steps ahead predictions for test dataset T}
1: $R_1, \ldots, R_n \leftarrow$ Split(R, n)
2: $S_1, \ldots, S_m \leftarrow$ MOEA($R_1, \ldots, R_n$, P, G, p_c, p_m)
3: $E_R \leftarrow$ EnsembleTrainingDataset($R_1, \ldots, R_n, S_1, \ldots, S_m$)
4: $\Psi \leftarrow$ StackingEnsembleLearning(E_R)
5: $E_T \leftarrow$ EnsembleTestDataset(T, $S_1, \ldots, S_m$)
6: $H \leftarrow$ RecursiveMultiStepForecasting(Ψ, E_T)
7: **return** H

The first step of MOEFS-ELRF (line 1) is to partition the training dataset R into n subsets $R_k = \{\{\mathbf{d}_1^k, o_1^k\}, \ldots, \{\mathbf{d}_l^k, o_l^k\}\}$, where each sample $\mathbf{d}_t^k = \{d_t^{k1}, \ldots, d_t^{kq}\}$, with $k = 1, \ldots, n$ and $t = 1, \ldots, l$. Each partition R_k contains $l = \lfloor 0.8 \cdot \frac{r}{n} \rfloor$ consecutive time-ordered samples, ensuring that all samples in R_k precede those in R_{k+1} for $k = 1, \ldots, n-1$. In our approach, the partitions are non-overlapping and consecutive in time. Each partition contains sequential samples ordered chronologically without overlap between partitions. This design is crucial in time series forecasting tasks because it preserves the natural temporal dependencies within each partition and avoids data leakage across partitions. Overlapping partitions could introduce information redundancy and violate the strict temporal separation required for meaningful forecasting models.

In the next step (line 2), a MOEA is executed to solve a multi-objective optimization problem with n objective functions, each evaluated on a different partition R_k, for $k = 1, \ldots, n$. The optimization problem is formulated as follows:

$$\text{Minimize} \quad \mathcal{F}(\mathbf{x}, R_k), \quad k = 1, \ldots, n \tag{1}$$

where $\mathbf{x} = \{x_1, \ldots, x_q\}$, with $x_i \in \{0, 1\}$ for $i = 1, \ldots, q$, represents the set of decision variables. Each decision variable x_i determines whether the corresponding input attribute is selected ($x_i = 1$) or not ($x_i = 0$) during the learning process. Each objective function $\mathcal{F}(\mathbf{x}, R_k)$, for $k = 1, \ldots, n$, returns the *root mean squared error* (RMSE) obtained using an RF learning algorithm evaluated through 5-fold cross-validation on dataset R_k, considering only the attributes for which $x_i = 1$ in the set of decision variables $\mathbf{x}$.

The solution to the multi-objective optimization problem defined in Eq. (1) is a set of *non-dominated solutions* $\{S_1, \ldots, S_m\}$, where each solution corresponds to a subset of features associated with an RF model. This set of feature subsets is then utilized by the MOEFS-ELRF algorithm (lines 3 and 4) to construct an EL model using the *stacking* method, with RF as the meta-learning algorithm. Finally, the ensemble prediction model is employed to generate a set H of h-step-ahead predictions using samples from the test dataset T (lines 5 and 6). The following sections provide details on the MOEA used by MOEFS-ELRF, the stacking-based EL method, and the multi-step-ahead forecasting technique.

3.2. The multi-objective evolutionary algorithm

The most natural way to represent the solutions of the optimization problem defined in Eq. (1) is through a binary string (or bit array), where a value of 1 indicates that the corresponding variable x_i has been selected, and a value of 0 means it has not. Given this representation, the choice of the multi-objective evolutionary algorithm (MOEA) to solve the problem depends on the number n of partitions considered, which directly determines the number of objective functions. In this work, we focus on Pareto-based and decomposition-based MOEAs as baseline optimizers to validate the proposed framework. While the method is compatible in principle with any MOEA, its empirical evaluation and demonstrated effectiveness are currently limited to these algorithm families.

When the number of objectives is relatively small ($n \leq 3$), well-established MOEAs such as the *Non-dominated Sorting Genetic Algorithm II* (NSGA-II) [33] can be employed. However, for problems with a higher number of objectives ($n > 3$), many-objective evolutionary algorithms (MaOEAs) are required, such as the NSGA-III algorithm [34], which is specifically designed to handle high-dimensional objective spaces effectively. NSGA-II is one of the most widely used MOEAs due to its efficiency and ability to maintain a diverse set of non-dominated solutions. It employs a fast non-dominated sorting approach to classify individuals in the population into different dominance fronts. *Binary tournament selection* is used, where two solutions are randomly chosen, and the one with a better rank (based on *non-dominance*) or, in case of a tie, higher *crowding distance*, is selected. For *survival selection*, a combination of non-dominated sorting and crowding distance is used to retain the most promising individuals in the next generation, ensuring both convergence and diversity. For problems with more than three objectives, NSGA-II tends to struggle with maintaining diversity in high-dimensional Pareto fronts. NSGA-III extends the concepts of NSGA-II by incorporating a reference-point-based selection mechanism, which enhances diversity and ensures a well-distributed set of solutions. The selection mechanism in NSGA-III is random. For survival selection, instead of crowding distance, NSGA-III employs a reference-point-based niching approach to select solutions that are well-distributed along the Pareto front. Another widely used MOEA

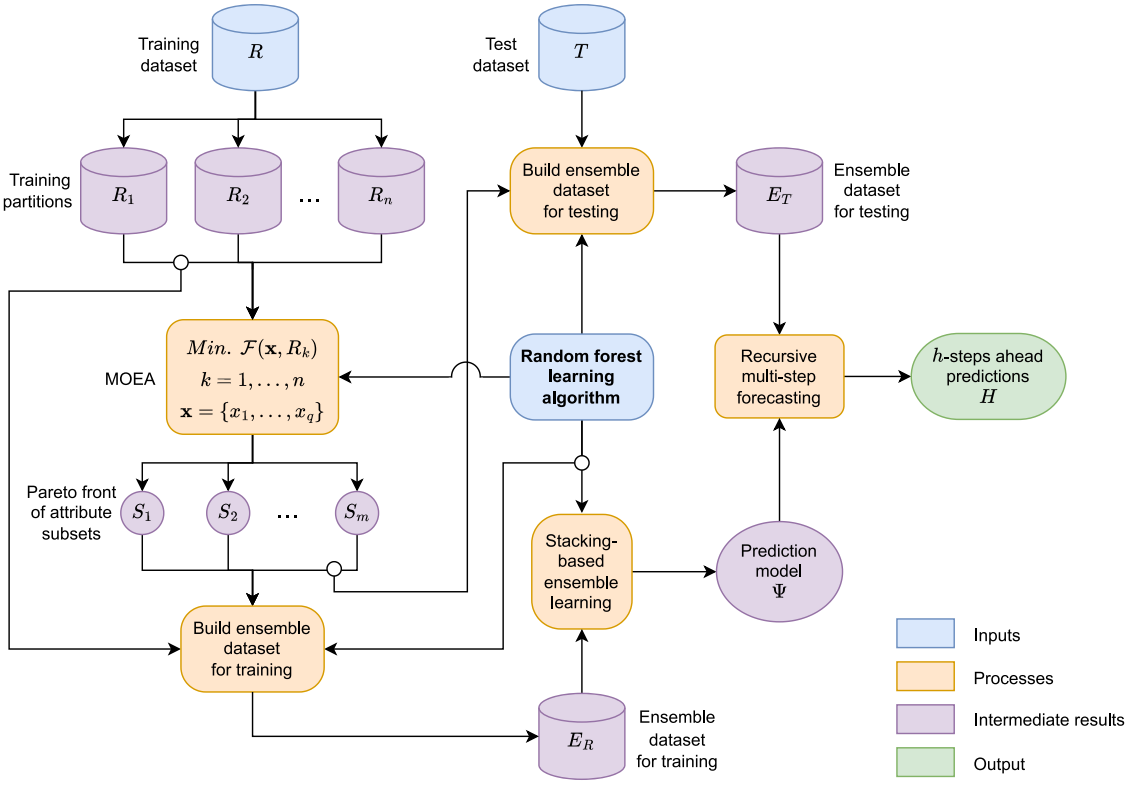


Fig. 1. General framework of MOEFS-ELRF.

considered in our experiments is MOEA/D (*Multi-Objective Evolutionary Algorithm based on Decomposition*) [35]. MOEA/D has been extensively applied in the literature for both multi-objective and many-objective optimization problems due to its scalability and decomposition-based strategy. Unlike population-based Pareto dominance methods such as NSGA-II, MOEA/D decomposes a multi-objective problem into a set of scalar sub-problems, each associated with a weight vector. These sub-problems are then optimized simultaneously in a collaborative manner, using information from neighboring sub-problems to guide the search process. This neighborhood-based search encourages diversity and convergence by promoting solution sharing among similar sub-problems.

Algorithm 2 presents the pseudo-code for the fitness functions of the MOEA. In line 1, the set of indices corresponding to the attributes not selected by the solution $\mathbf{x}$ is identified. In line 2, the unselected attributes are removed from dataset R_k. Finally, the RMSE obtained using RF in a 5-fold cross-validation on the reduced dataset is returned (lines 3 and 4). To generate new candidate solutions, two binary genetic operators are applied. *Half uniform crossover* [36] is applied with a given crossover probability and swaps bits between two parent solutions with a probability of 0.5. In *bit-flip mutation* [37], each bit in the offspring solution is flipped with a given mutation probability.

Algorithm 2: Function $\mathcal{F}(\mathbf{x}, R_k)$

Input: $\mathbf{x} = \{x_1, \ldots, x_q\}$ {Decision variable set}
Input: $R_k \subset R$ {Dataset for 5-fold cross-validation}
1: $I \leftarrow \{i \mid x_i = 0\}$ {Set of indexes of the non-selected features}
2: $R'_k \leftarrow remove(R_k, I)$ {Remove unselected attributes from R_k}
3: $score \leftarrow cross\_validation\_score(rf, R'_k, 5, rmse)$ {5-fold cross-validation on R'_k using RF and RMSE}
4: **return** *score* {Fitness of $\mathbf{x}$ on R_k}

Table 1
Initial attribute statistics of the Lorca air quality dataset.

Name	Units	Min	Mean	Max	Std
NO	$\mu g/m^3$	1.00	2.56	8.00	0.98
O_3	$\mu g/m^3$	5.00	44.60	120.00	16.11
TMP	°C	5.00	19.23	34.00	6.26
HR	% R.H.	30.00	71.64	100.00	16.08
NO_X	$\mu g/m^3$	5.00	13.16	32.00	4.65
DD	degrees	37.00	208.66	309.00	47.25
RS	W/m^3	2.00	158.79	345.00	72.13
VV	m/s	0.00	0.66	5.00	0.53
PM_{10}	$\mu g/m^3$	3.00	23.74	172.00	15.11
NO_2	$\mu g/m^3$	3.00	9.23	23.00	3.58

The proposed MOEFS-ELRF algorithm has been implemented using the *Platypus* library [38], a framework for multi-objective optimization that provides implementations of NSGA-II, NSGA-III, MOEA/D, and other MOEAs. This ensures flexibility in selecting the appropriate algorithm based on the number of objectives in the problem, allowing a fair and consistent comparison under identical experimental settings.

3.3. Stacking-based ensemble learning

EL refers to a machine learning technique where multiple models, known as base learners or weak learners, are combined to form a stronger and more accurate model. The idea behind EL is that by combining the predictions of multiple models, the resulting ensemble model can make more robust and accurate predictions than any individual model. *Stacking-based EL*, also known as *stacked generalization*, is a specific type of EL method. In stacking, multiple diverse base learners are trained on the same dataset, and their predictions are then combined using another model called a *meta-learner*. The meta-learner

takes the predictions of the base learners as input and learns to make the final prediction.

The stacking process typically involves two phases. In the first phase, the base learners are trained independently on the training data, generating predictions. In the second phase, the meta-learner is trained on the first phase's predictions to make the final prediction. The idea behind stacking is to leverage the diverse strengths and weaknesses of the base learners, allowing the meta-learner to learn a combination of their predictions that maximizes the overall performance. Stacking-based EL can be more powerful than individual models or traditional ensemble methods because it learns to effectively weigh and combine the predictions of the base learners, taking advantage of their complementary strengths. It has been successfully applied in various machine learning tasks and is particularly useful when dealing with complex problems where no single model can provide satisfactory results.

Stacking-based EL can increase *generalization* in machine learning models. Generalization refers to the ability of a model to perform well on unseen data or data from the real world. Stacking achieves improved generalization by combining the predictions of multiple base learners, which helps to reduce the individual errors or biases of each model. The diversity of the base learners is a key factor in improving generalization. This diversity allows the ensemble model to cover a wider range of potential solutions and reduces the risk of overfitting to specific patterns in the training data.

In our proposal, stacking-based EL is approached differently from the conventional method. While the standard approach trains different base learners on the same dataset during the first phase, our method trains the same base learner (RF) on different partitions of the same time series dataset. This is achieved using a multi-objective optimization approach, as formulated in problem (1), resulting in multiple RF models, each with its own feature selection. These models are then combined in the second phase of the stacking process. Similar to traditional stacking, this approach captures diverse aspects of the data and learns different patterns, ultimately enhancing generalization. Similar approaches such as chunk-based ensemble passive learner have proven to be effective in temporally unstable environments, thus considering long term dependencies [39].

It is important to highlight why stacking-based EL was chosen as the combination strategy in our approach. While there are more complex or specialized ensemble combination methods in the literature [30,31,40,41], stacking offers notable advantages of simplicity, generality, and interpretability. Stacking is a model-agnostic strategy that flexibly accommodates the diversity of the non-dominated RF models generated by our multi-objective evolutionary process. Each of these models is trained on different partitions with distinct feature subsets, naturally leading to heterogeneous error patterns. The meta-learner in stacking can effectively leverage this diversity by adaptively weighting the base predictions to maximize overall forecasting accuracy. Moreover, stacking aligns well with the philosophy of using Pareto fronts to capture trade-offs between objectives, providing a principled way to integrate all the selected solutions without requiring problem-specific weighting schemes or highly parameterized combination rules. This balance between effectiveness and practical applicability motivated our choice, although exploring advanced combination strategies such as attention-based fusion or error-correction-based ensembling remains an interesting direction for future research.

Below we describe the proposed process to build the stacking-based EL model from the set $\{S_1, \ldots, S_m\}$ of feature subsets found by the MOEA.

1. Build an RF prediction model M_j^k for each feature subset S_j, $j = 1, \ldots, m$, trained on each dataset R_k, $k = 1, \ldots, n$.
2. Predict each input data $\mathbf{d}_t \in R_k$ in each RF model M_j^k, $j = 1, \ldots, m$. The model outputs $Y_{M_j^k}(\mathbf{d}_t)$, $\mathbf{d}_t \in R_k$, $j = 1, \ldots, m$, are obtained in this step.
3. Generate a dataset E_R with the model outputs $Y_{M_j^k}(\mathbf{d}_t)$, $\mathbf{d}_t \in R_k$, $j = 1, \ldots, m$, as input attributes, and the corresponding observations o_t as target attribute, as follows:

$$E_R = \begin{pmatrix} Y_{M_1^1}(\mathbf{d}_1) & Y_{M_2^1}(\mathbf{d}_1) & \cdots & Y_{M_m^1}(\mathbf{d}_1) & o_1 \\ Y_{M_1^1}(\mathbf{d}_2) & Y_{M_2^1}(\mathbf{d}_2) & \cdots & Y_{M_m^1}(\mathbf{d}_2) & o_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ Y_{M_1^1}(\mathbf{d}_l) & Y_{M_2^1}(\mathbf{d}_l) & \cdots & Y_{M_m^1}(\mathbf{d}_l) & o_l \\ Y_{M_1^2}(\mathbf{d}_{l+1}) & Y_{M_2^2}(\mathbf{d}_{l+1}) & \cdots & Y_{M_m^2}(\mathbf{d}_{l+1}) & o_{l+1} \\ Y_{M_1^2}(\mathbf{d}_{l+2}) & Y_{M_2^2}(\mathbf{d}_{l+2}) & \cdots & Y_{M_m^2}(\mathbf{d}_{l+2}) & o_{l+2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ Y_{M_1^2}(\mathbf{d}_{2l}) & Y_{M_2^2}(\mathbf{d}_{2l}) & \cdots & Y_{M_m^2}(\mathbf{d}_{2l}) & o_{2l} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ Y_{M_1^n}(\mathbf{d}_{(n-1)l+1}) & Y_{M_2^n}(\mathbf{d}_{(n-1)l+1}) & \cdots & Y_{M_m^n}(\mathbf{d}_{(n-1)l+1}) & o_{(n-1)l+1} \\ Y_{M_1^n}(\mathbf{d}_{(n-1)l+2}) & Y_{M_2^n}(\mathbf{d}_{(n-1)l+2}) & \cdots & Y_{M_m^n}(\mathbf{d}_{(n-1)l+2}) & o_{(n-1)l+2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ Y_{M_1^n}(\mathbf{d}_{nl}) & Y_{M_2^n}(\mathbf{d}_{nl}) & \cdots & Y_{M_m^n}(\mathbf{d}_{nl}) & o_{nl} \end{pmatrix}$$

   where l is the number of samples in each partition R_k, $k = 1, \ldots, n$.
4. Build an RF prediction model Ψ using the dataset E_R as training data.

The prediction model Ψ is finally used for multi-step ahead forecasting with the test data, which must be converted into the same format as the training data used for the prediction model Ψ, as shown in lines 5 and 6 of Algorithm 1.

3.4. Multi-step ahead forecasting

Multi-step ahead TSF [42], also known as *multi-step forecasting* or *horizon forecasting*, is the process of predicting multiple future data points in a time series beyond the next time step. Instead of just forecasting the next single data point, the objective is to make predictions for a specified number of steps into the future. With multi-step ahead TSF, the next h values $y_{r+1}, \ldots, y_{r+h}$ are forecast from a historical time series $y_1, \ldots, y_r$ composed of r samples, where $h > 1$ is the forecasting horizon.

In this work we use the *recursive strategy* [43] for multi-step ahead TSF. The recursive strategy for multi-step TSF involves making step-by-step predictions into the future. In this approach, one step ahead is predicted at a time, and then the predicted value is used as input for the next prediction. This process is repeated iteratively for the desired number of steps ahead. The performance of the model can be evaluated with the recursive strategy using the training set R or the test set T. Although we could have used other strategies for multi-step ahead TSF, in this work we have preferred to use the recursive strategy for its simplicity and easy implementation in all the learning algorithms compared in this paper, and as a first approximation for its inclusion in the MOEFS-ELRF algorithm. Another determining factor why we have decided to use the recursive strategy is the efficiency in runtime when compared to other methods like the *Direct*, *DirRec*, and *DIRMO* strategies [44]. We have excluded the *MIMO* strategy from our study because it requires a multi-output learning algorithm and shares a limitation with the recursive strategy of needing the same model structure for all forecast horizons. Although the recursive strategy, which we use here, can sometimes underperform in multi-step forecasting tasks, particularly when the forecast horizon h is larger than the window size W, as this leads to reliance on predicted values instead of actual observations, this issue does not affect our research since we maintain $h \leq W$.

3.5. Feature importance

The importance of each attribute can be determined by computing its frequency of appearance in the set of non-dominated attribute subsets $\{S_1, \ldots, S_m\}$ identified by the MOEA. This is given by the following formula:

$$I_i = \frac{1}{m} \cdot \sum_{j=1}^{m} x_i^j, \; i = 1, \ldots, q \tag{2}$$

where $x_i^j \in \{0, 1\}$ indicates whether attribute i has been selected ($x_i^j = 1$) or not ($x_i^j = 0$) in the attribute subset S_j, $i = 1, \ldots, q$, $j = 1, \ldots, m$. With this approach, an attribute i that appears in all m non-dominated attribute subsets will have an importance score of $I_i = 1$, while an attribute that is never selected will have an importance score of $I_i = 0$. This frequency-based feature importance measure provides an intuitive and interpretable way to assess the relevance of each attribute within the multi-objective optimization framework, complementing traditional feature importance techniques.

4. Experiments and results

This section presents the experiments conducted to evaluate the performance of the proposed method, detailing the datasets and algorithms used for comparison, and reporting the obtained results. Two types of datasets are considered: datasets from real-world applications and synthetically generated datasets with increasing levels of complexity. The proposed method is compared against state-of-the-art approaches for TSF and FS methods embedded in LSTM neural networks, given their high capability for TSF.

4.1. Datasets from real-world applications

Three real-world time series datasets were used. The first dataset (Table 1) corresponds to daily air quality measurements from a monitoring station located in Lorca, a city in the Region of Murcia (Spain). The data have been provided by Consejería de Medio Ambiente, Mar Menor, Universidades e Investigación.[1] There are a total of 1000 instances since the measurements were carried out between April 7, 2019 and December 31, 2021. The dataset has 10 attributes with the target being NO_2. The second dataset (Table 2) has been obtained from the *UCI Machine Learning Repository* [45] and contains hourly air quality data from an Italian city. It has 12 attributes and 1000 instances that correspond to measurements made between February 21, 2005 and April 4, 2005. The attribute to predict is NO_X. The third dataset (Table 3) is the *Electricity Transformer Temperature* (ETT) dataset (ETTh1[2]) for oil temperature forecasting [46]. We focus on the last 1000 instances of this dataset. These instances are recorded at hourly intervals and encompass 7 attributes: HUFL (High Useful Load), HULL (High Useless Load), MUFL (Middle Useful Load), MULL (Middle Useless Load), LUFL (Low Useful Load), LULL (Low Useless Load), and OT (Oil Temperature).

All datasets were preprocessed as follows: First, missing values were handled using *linear interpolation*. Next, a sliding window approach with a window size of 3 was applied to the datasets. Then, the datasets were split into two partitions: 80% for training and the remaining 20% for testing (as described in Section 3.1). Finally, the data were standardized using the statistics computed from the training set, which were then applied to both the training and test sets to prevent data leakage.

[1] https://sinqlair.carm.es/calidadaire/.

[2] https://github.com/zhouhaoyi/ETDataset.

Table 2
Initial attribute statistics of the Italian city air quality dataset.

Name	Units	Min	Mean	Max	Std
$CO(GT)$	mg/m^3	0.10	1.95	7.50	1.32
$PT08.S1(CO)$	mg/m^3	715.00	1109.05	1818.00	196.73
$C6H6(GT)$	mg/m^3	0.20	7.94	35.50	6.35
$PT08.S2(NMHC)$	mg/m^3	387.00	855.66	1675.00	247.69
$PT08.S3(NOx)$	mg/m^3	330.00	749.39	1804.00	227.35
$NO2(GT)$	mg/m^3	25.00	136.64	295.00	48.59
$PT08.S4(NO2)$	mg/m^3	551.00	1175.37	2147.00	273.88
$PT08.S5(O3)$	mg/m^3	221.00	1008.16	2159.00	413.70
T	°C	−1.90	12.21	30.00	6.65
RH	%	9.90	51.73	86.60	17.78
AH	%	0.18	0.73	1.39	0.27
NO_X	ppb	25.00	288.60	959.00	180.40

Table 3
Initial attribute statistics of the ETTh1 dataset.

Name	Min	Mean	Max	Std
HUFL	−17.88	6.61	18.15	8.32
HULL	1.07	4.15	7.84	1.32
MUFL	−20.08	2.92	12.33	7.87
MULL	−0.11	2.21	4.69	0.97
LUFL	1.46	3.57	6.15	1.08
LULL	0.00	1.36	2.71	0.32
OT	3.03	9.53	16.88	2.14

4.2. Synthetic time series datasets

To evaluate the performance of our models under different levels of complexity, we generated five synthetic time series datasets, each containing 500 instances and 10 input features. The complexity of the datasets increases progressively from level 1 to level 5, incorporating different degrees of seasonality, trend, and noise. This controlled data generation process allows us to analyze the behavior of our models under different levels of structured variability, providing insights into their ability to capture temporal dependencies and handle increasing complexity.

Each input feature is constructed as a combination of sinusoidal components with varying periodicities to introduce different seasonal patterns. Higher complexity levels include additional sinusoidal terms and an increasing linear trend to reflect more realistic temporal dependencies. Gaussian noise is added to each feature, with its magnitude proportional to the dataset complexity, ensuring a gradual increase in variability. The target variable is computed as a linear combination of the input features with an additional nonlinear component based on the sine function. This structure introduces interactions between variables, making the prediction task more challenging as complexity increases. Finally, these five synthetic datasets were also processed using a sliding window with a window size of 3, split into training (80%) and test (20%), and standardized using the statistics computed from the training set.

4.3. Comparison methods

To evaluate the performance of the proposed approach, we first compare it against two baseline methods that leverage RF in different ways. Within this group, the first comparison method is the standard RF algorithm. One of the primary goals of our proposal is to enhance the generalization capability of RF when applied to TSF. RF is a widely used EL technique that constructs multiple decision trees during training and outputs the average prediction (for regression tasks). Although RF inherently provides a measure of feature importance, it does not explicitly perform FS in the sense of optimizing a reduced subset of attributes before training the model. As a result, irrelevant or redundant features may still be included, potentially impacting generalization performance. By comparing our method with standard RF, we aim

to demonstrate the benefits of integrating multi-objective FS and EL to improve predictive accuracy and generalization. The second comparison method (MOEFS-RF) is a wrapper-based FS approach using RF and a multi-objective evolutionary search strategy, following the methodology proposed in [47]. In this method, a MOEA is used to optimize two conflicting objectives simultaneously, minimizing the RMSE of the RF model, and minimizing the number of selected features to promote sparsity and interpretability. This approach allows us to assess how a conventional multi-objective evolutionary FS method compares to our proposed framework, which integrates FS within an EL model designed explicitly for TSF. This comparison highlights the impact of incorporating multiple partitions of the time series dataset within the evolutionary optimization process.

Another comparison method used in paper is a *long short-term memory* (LSTM) neural network [48]. LSTMs are a specialized type of *recurrent neural network* (RNN) designed to handle sequential data by capturing long-term dependencies, making them particularly well-suited for TSF. Unlike traditional feedforward networks, LSTMs use memory cells and gating mechanisms to selectively retain or forget information over time, effectively mitigating the vanishing gradient problem that limits the learning capacity of standard RNNs. LSTM models are widely used in TSF tasks because they can model temporal correlations and capture complex patterns in sequential data, even in the presence of noise or non-stationarity. Their ability to learn dynamic dependencies makes them a strong benchmark for evaluating new forecasting approaches. Furthermore, LSTMs inherently incorporate FS by learning the most relevant temporal dependencies, which justifies their inclusion as a comparison method against our multi-objective evolutionary FS approach.

Embedded FS techniques in neural networks are another group of methods considered for comparison in this paper. One of these methods is *CancelOut*[3] [49], an artificial neural network layer designed to perform FS and FR tasks in both supervised and unsupervised learning. CancelOut is positioned between the input layer and the first hidden layer of a deep neural network. Each neuron in the CancelOut layer is directly connected to a specific input feature and fully connected to the neurons in the subsequent hidden layer. The activation values in this layer indicate the relative contribution of each input feature to the final output, allowing the network to learn which attributes are most relevant for the task at hand. This approach is particularly useful in TSF since neural networks can inherently model complex dependencies between input features and their temporal dynamics.

Within the group of embedded FS methods, attention-based FS has emerged as a promising approach to effectively address the challenges posed by high-dimensional data, as demonstrated by several studies. One such method is *attention-based feature selection* (AFS) [50], which introduces a novel neural network-based architecture designed to tackle FS challenges in high-dimensional and noisy datasets. AFS incorporates an attention module to generate feature weights by framing the problem as a binary classification task for each feature. AFS is an attention layer that dynamically assigns importance to input features while learning these weights. It consists of trainable parameters initialized using specific strategies. The layer computes attention coefficients through a hyperbolic tangent activation function followed by softmax, modulating input features to generate a weighted representation. Finally, feedforward neural networks process these weighted features to produce the final output. Another attention-based FS method considered in this study is the *multiattention-based supervised FS* method (m-AFS) [51]. m-AFS consists of multiple modules designed to implement an attention mechanism within a neural network architecture. It employs two distinct sets of dense layers: one dedicated to temporal attention and another to variable attention. These dense layers compute attention weights for both the temporal and variable dimensions of the input data. After obtaining these attention weights via softmax activation, a global weight matrix is generated by computing the matrix product of the temporal and variable attention weights. This global weight matrix is then applied to the input data, resulting in weighted data representations that highlight the most relevant attributes. Finally, another method in this category is the *External Attention-Based Feature Ranker for Large-Scale Feature Selection* method (EAR-FS[4]) [52], which prelearns the importance of features using a *multi-layer perceptron* (MLP) with an embedded attention mechanism. EAR-FS consists of three main modules: (1) an MLP with an embedded attention layer to learn feature weights, (2) a processing module that refines these weights to obtain a feature ranking (FR), and (3) a selection module that generates appropriate subsets of attributes based on the FR and a predefined classifier. These attention-based FS methods provide a useful benchmark for evaluating our proposed multi-objective evolutionary FS method, as they leverage neural network architectures to dynamically learn feature importance, thereby offering a different paradigm for FS in TSF.

Table 4 shows a summary of the methods used in this paper for comparisons with MOFS-ELRF.

4.4. Description of the experiments and results

This section presents the comprehensive experimental study conducted to evaluate the proposed MOEFS-ELRF method. The experiments were structured into six sequential stages to thoroughly assess different aspects of the method. First, we conducted a comparison of baseline MOEAs, where NSGA-II and MOEA/D were compared under identical conditions to select the most suitable algorithm for our framework. Second, we performed an analysis to determine the optimal number of partitions, evaluating NSGA-II and NSGA-III to identify the most appropriate number of partitions, which directly defines the number of objectives in the multi-objective optimization problem and ensures a balance between model diversity and computational efficiency. In the third stage, we executed a comparative study of MOEFS-ELRF against selected benchmark methods, including conventional RF, wrapper-based FS methods, state-of-the-art TSF models, and embedded FS techniques. The fourth stage explored the relationship between the number of features and the number of training partitions, analyzing how these factors interact and influence the model's performance. The fifth stage assessed the scalability of MOEFS-ELRF with respect to feature dimensionality, evaluating runtime and memory consumption across datasets with increasing numbers of features. Finally, the sixth stage evaluated the scalability of the method with respect to the forecasting horizon, by extending the experiments to multi-step ahead forecasting scenarios with different horizons. This structured and incremental experimental approach provides a holistic view of the proposed method's effectiveness, scalability, and robustness in FS and TSF tasks.

4.4.1. Comparison of MOEAs

The first step in our experimental study was to identify a suitable baseline MOEA for our MOEFS-ELRF framework. To this end, we compared the performance of two well-known MOEAs: NSGA-II and MOEA/D. Both algorithms were applied under the same configuration to ensure a fair comparison, using the real-world datasets Lorca, Italian City, and ETT, as well as the synthetic dataset SD1, selected as a representative of increasing complexity.

All experiments were conducted with the number of data partitions set to $n = 3$ and a three-step-ahead forecasting horizon. Each MOEA was executed 10 times using different predefined random seeds. The number of objective function evaluations was fixed at 100,000, with crossover and mutation probabilities both set to 1.0, and a population size of $P = 50$. The comparison results are summarized in Fig. 2,

[3] The CancelOut code is available at https://github.com/unnir/CancelOut.

[4] The EAR-FS code is available at https://github.com/xueyunuist/EAR-FS.

Table 4
Summary of comparison methods.

Short name	Extended name	Category	Reference
RF	Random Forest	Ensemble of decision trees	[23]
MOEFS-RF	Multi-objective Evolutionary Feature Selection with Random Forest	Wrapper FS method with RF and MOEA	[47]
LSTM	Long Short-Term Memory	Recurrent neural network	[48]
CancelOut	–	Embedded FS in neural networks	[49]
AFS	Attention-based Feature Selection	Embedded FS in neural networks with attention	[50]
m-AFS	Multiattention-based supervised Feature Selection	Embedded FS in neural networks with attention	[51]
EAR-FS	External Attention-Based Feature Ranker for Large-Scale Feature Selection	Embedded FS in neural networks with attention	[52]

which presents bar plots showing the average RMSE on training and test datasets, the average percentage of selected features, the average overfitting ratio,[5] and the average runtime. The results show that both MOEAs achieve very similar performance across all indicators, with NSGA-II slightly outperforming MOEA/D in most metrics. Notably, MOEA/D incurred a significantly higher computational cost, requiring on average 3.85 times more runtime than NSGA-II. Given the comparable performance and the substantial efficiency advantage, NSGA-II was selected as the reference algorithm for MOEFS-ELRF in the remaining experiments.

4.4.2. Setting the optimal number of partitions

In this experiment, we evaluated the impact of the number of partitions n on the optimization process, considering values of $n = 3$, 4, and 5. For $n = 3$, we employed NSGA-II, while for $n = 4$ and $n = 5$, NSGA-III was used. Each algorithm was executed 10 times with different predefined random seeds (from 1 to 10) to ensure reproducibility, running for 100,000 evaluations, and crossover and mutation probabilities set to 1.0. Population size was $P = 50$ in NSGA-II. In NSGA-III, the number of reference points Q must be specified, which is calculated as in [34], based on the number of objective functions n and the number of divisions p along each objective, as follows:

$$P = Q = \binom{n + p - 1}{p} \tag{3}$$

In our experiments, we selected the number of divisions p so that $P = Q \approx 50$ for a fair comparison with NSGA-II. The results indicate that there were no statistically significant differences in the average RMSE for a three-step-ahead forecasting horizon on the test set across the different datasets when comparing $n = 3$, 4, and 5. However, NSGA-II exhibited shorter execution times and selected, on average, a smaller number of attributes compared to NSGA-III.

For illustrative purposes, Fig. 3 shows the results of this comparison for the real-world Lorca, Italian city and ETT datasets, plus a representative synthetic dataset (SD1), for a three-step-ahead forecasting horizon. These figures depict the average RMSE on the training and test datasets, the average percentage of selected attributes, the average overfitting ratio, and the average execution time for $n = 3$, 4, and 5 partitions. The results show that $n = 3$ provides a good trade-off between model complexity, feature diversity, prediction accuracy and computational cost. Based on these findings, we establish $n = 3$ as the optimal number of partitions for further comparisons of MOEFS-ELRF against the other methods.

4.4.3. Comparison with other methods

All comparison methods, like MOEFS-ELRF, were executed 10 times using predefined random seeds (from 1 to 10) to ensure reproducibility and robust performance evaluation. This approach accounts for the inherent variability in the optimization and learning processes, providing a fair comparison across all methods. By averaging the results over multiple runs, we mitigate the impact of randomness and obtain more consistent and statistically meaningful conclusions.

For the methods using RF (MOEFS-ELRF, MOEFS-RF, and standard RF), the *RandomForestRegressor* implementation from *scikit-learn* [53, 54] in Python was used with default parameters. The LSTM network was also implemented using scikit-learn, configured with a hidden layer followed by a *dropout* layer. The LSTM hyperparameters were tuned using a *grid search* approach, considering $u \in 2, 5, 10$, $epochs \in 100, 500, 1000$, and $batch\_size \in 8, 16, 32, 128$. Model evaluation was performed using 3-*fold cross-validation*. Fig. 4 illustrates the final architecture of the LSTM network after hyperparameter tuning for the Italian city dataset, provided as an example for illustrative purposes. Once the optimal hyperparameters were selected, 10 conventional LSTM networks were trained using different predefined seeds for the *Glorot uniform initializer* of weights and biases, ensuring reproducibility of the results.

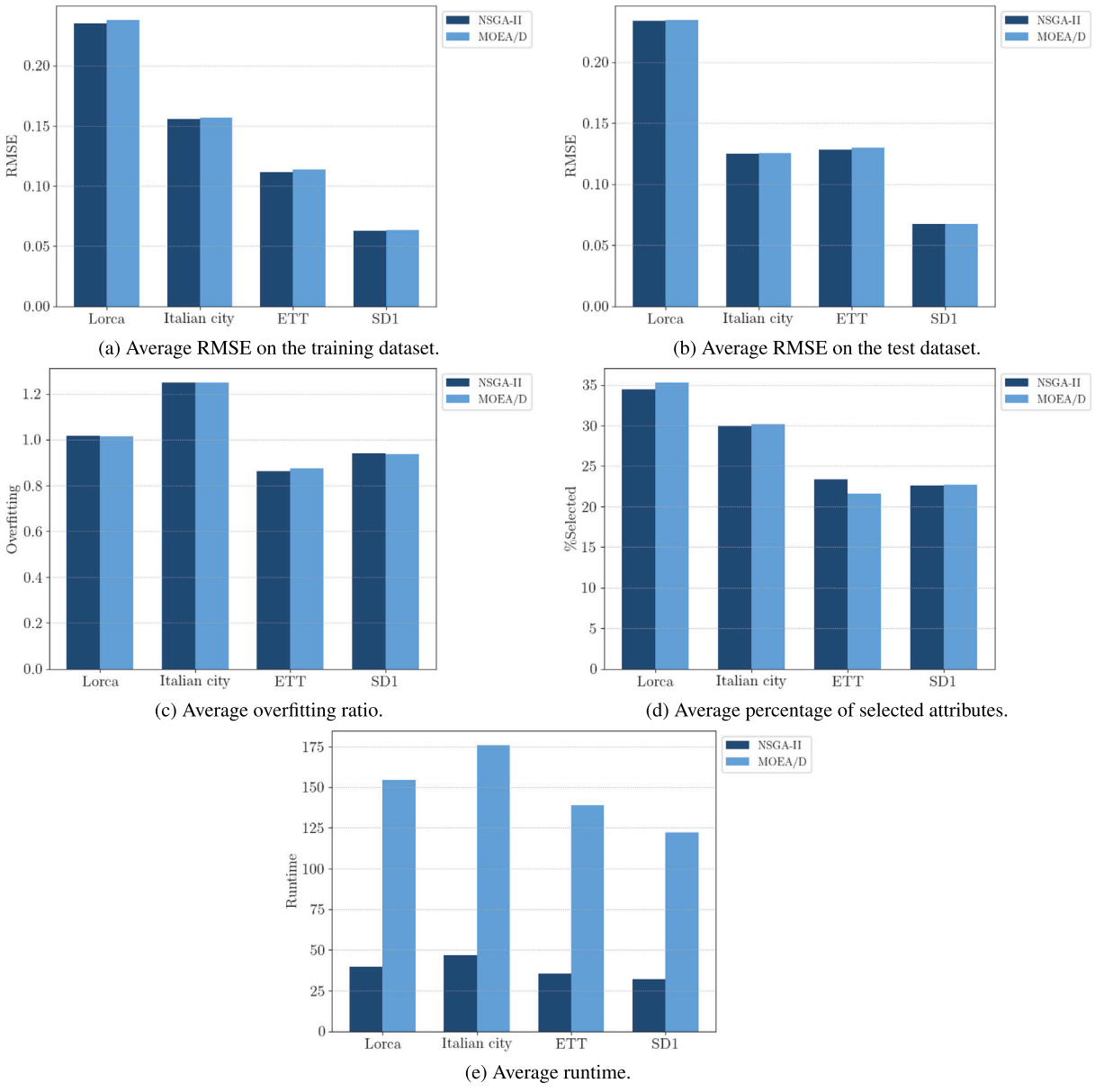
The CancelOut method was tested by incorporating the CancelOut layer into the LSTM network architecture shown in Fig. 4, with hyperparameters tuned for each specific problem. For the AFS and m-AFS methods, their output was fed into an LSTM network with the same architecture as in Fig. 4, with hyperparameters also adjusted for each problem. The EAR-FS method was originally implemented by its authors for classification tasks; therefore, in this paper, it had to be adapted for regression tasks. The hyperparameters of EAR-FS were set according to the recommendations provided by the authors in [52], specifically: *momentum* = 0.9, *weight decay* = 0.00001, *minimum learning rate* =0.001, $M = 50$, $L = 25$, and $K_p = 0.1$. In the third module of EAR-FS, we employed the LSTM architecture from Fig. 4, adjusted for each problem, for subset generation.

Tables 5, 6, and 7 present the results obtained with MOEFS-ELRF, RF, MOEFS-RF, LSTM, CancelOut, AFS, m-AFS, and EAR-FS for the Lorca, Italian city, and ETT datasets. Specifically, Tables 5 and 6 report the average, maximum, and minimum RMSE values on the training and test datasets, respectively, for the predictive models obtained across 10 runs over a forecasting horizon of 3 steps ahead. Additionally, Table 7 shows the cardinality of the attribute subsets selected by the FS methods. Similarly, Tables 8, 9, and 10 present the same results for the synthetic datasets.

4.4.4. Relationship between number of features and number of partitions

In this section, we have incorporated a new experimental study to investigate possible trends or reasonable relationships between the feature space dimension and the optimal number of partitions. Specifically, we have generated additional synthetic datasets SD6, SD7 and SD8, with 10, 20, and 30 original attributes, respectively. After applying the sliding window transformation with a window size of 3, the corresponding feature dimensions became 33, 63, and 93, respectively. For each dataset, we varied the number of training partitions among 4, 5 and 6, and we systematically analyzed the impact on forecasting

[5] Ratios below 1 indicate overfitting, the smaller the ratio, the greater the overfitting, as explained in Section 5.



(a) Average RMSE on the training dataset.

(b) Average RMSE on the test dataset.

(c) Average overfitting ratio.

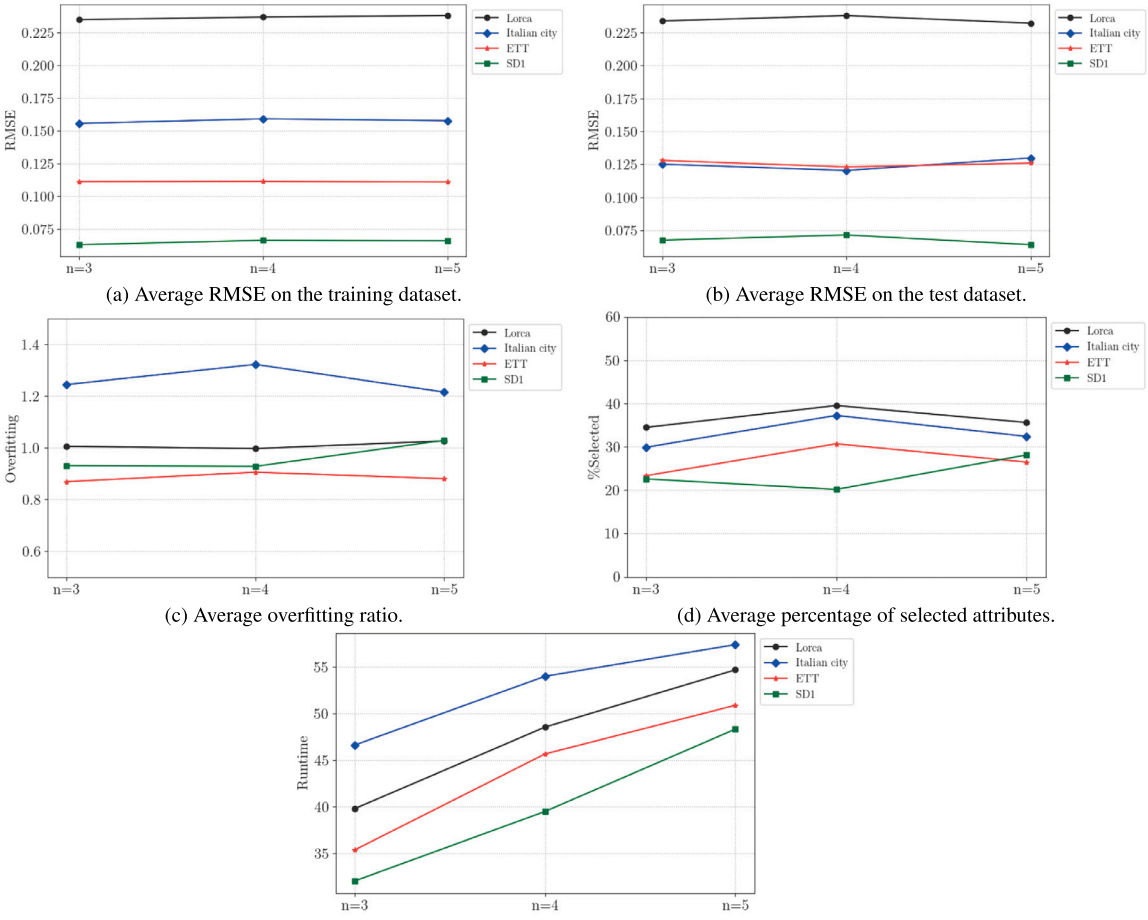
(d) Average percentage of selected attributes.

(e) Average runtime.

Fig. 2. Comparison of NSGA-II and MOEA/D on the Lorca, Italian city, ETT and SD1 datasets.

Table 5
Real-world dataset evaluation results on the training set.

Method	Lorca			Italian city			ETT		
	Avg.	Max.	Min.	Avg.	Max.	Min.	Avg.	Max.	Min.
MOEFS-ELRF	**0.23500**	**0.22943**	0.22943	**0.15567**	**0.15473**	**0.15473**	**0.11121**	**0.10970**	**0.10970**
RF	0.45048	0.44174	0.44174	0.30241	0.29156	0.29156	0.28376	0.27881	0.27881
MOEFS-RF	1.01083	1.00187	1.00187	0.31896	0.29675	0.29675	0.32970	0.29577	0.29577
LSTM	0.65904	0.62933	0.62933	0.27843	0.26840	0.26840	0.26066	0.25252	0.25252
CancelOut	0.67174	0.64210	0.64210	0.29736	0.28864	0.28864	0.26747	0.26093	0.26093
AFS	0.49647	0.39168	0.39168	0.29035	0.23192	0.23192	0.18873	0.14920	0.14920
m-AFS	0.67465	0.66507	0.66507	0.28992	0.27719	0.27719	0.27138	0.25906	0.25906
EAR-FS	0.24424	0.18696	**0.18696**	0.17918	0.16711	0.16711	0.23244	0.12784	0.12784



(a) Average RMSE on the training dataset.

(b) Average RMSE on the test dataset.

(c) Average overfitting ratio.

(d) Average percentage of selected attributes.

(e) Average runtime.

Fig. 3. Results for $n = 3$, 4 and 5 on the Lorca, Italian city, ETT and SD1 datasets.

Table 6
Real-world dataset evaluation results on the test set.

Method	Lorca			Italian city			ETT		
	Avg.	Max.	Min.	Avg.	Max.	Min.	Avg.	Max.	Min.
MOEFS-ELRF	**0.23366**	**0.23058**	**0.23058**	**0.12511**	**0.12301**	**0.12301**	**0.12799**	**0.12526**	**0.12526**
RF	0.74814	0.73706	0.73706	0.68160	0.66279	0.66279	0.56283	0.54784	0.54784
MOEFS-RF	0.85251	0.84340	0.84340	0.57439	0.52586	0.52586	0.56768	0.52678	0.52678
LSTM	0.65604	0.59789	0.59789	0.37494	0.29192	0.29192	0.36613	0.33147	0.33147
CancelOut	0.64583	0.60043	0.60043	0.40470	0.33604	0.33604	0.34704	0.32934	0.32934
AFS	0.92856	0.71968	0.71968	0.58655	0.43010	0.43010	0.62517	0.54588	0.54588
m-AFS	0.67326	0.64520	0.64520	0.35357	0.30604	0.30604	0.35722	0.33151	0.33151
EAR-FS	0.29614	0.25420	0.25420	0.18139	0.16614	0.16614	0.23268	0.14205	0.14205

Table 7
Number of attributes selected by the FS methods for the real-world application datasets.

Method	Lorca			Italian city			ETT		
	Avg.	Max.	Min.	Avg.	Max.	Min.	Avg.	Max.	Min.
MOEFS-ELRF	10.34	9.92	9.92	10.76	10.63	10.63	**4.90**	**4.77**	4.77
MOEFS-RF	**1.00**	**1**	**1**	**9.10**	**9**	**9**	5.40	4	4
CancelOut	16.70	14	14	21.00	18	18	12.50	9	9
AFS	12.90	6	6	18.70	12	12	10.30	3	**3**
m-AFS	14.80	9	9	18.00	13	13	10.10	8	8
EAR-FS	18.90	2	2	27.70	20	20	19.90	15	15

Table 8
Synthetic dataset evaluation results on the training set.

Method	SD1			SD2			SD3			SD4			SD5		
	Avg.	Max.	Min.	Avg.	Max.	Min.	Avg.	Max.	Min.	Avg.	Max.	Min.	Avg.	Max.	Min.
MOEFS-ELRF	**0.06291**	**0.06226**	**0.06226**	**0.10837**	**0.10558**	**0.10558**	**0.14961**	**0.14522**	**0.14522**	**0.17603**	**0.17032**	**0.17032**	**0.18565**	**0.17720**	0.17720
RF	0.11495	0.11090	0.11090	0.19427	0.18648	0.18648	0.27501	0.27008	0.27008	0.28674	0.27529	0.27529	0.30425	0.29723	0.29723
MOEFS-RF	0.13536	0.12676	0.12676	0.23472	0.22416	0.22416	0.31486	0.30235	0.30235	0.35034	0.30845	0.30845	0.35570	0.34284	0.34284
LSTM	0.13314	0.12537	0.12537	0.20037	0.19391	0.19391	0.34924	0.31623	0.31623	0.35746	0.34777	0.34777	0.42326	0.41264	0.41264
CancelOut	0.13633	0.12833	0.12833	0.21460	0.20709	0.20709	0.36019	0.32953	0.32953	0.38059	0.36593	0.36593	0.44947	0.43346	0.43346
AFS	0.24764	0.17076	0.17076	0.27739	0.21368	0.21368	0.41668	0.32843	0.32843	0.37387	0.32264	0.32264	0.37402	0.34115	0.34115
m-AFS	0.16551	0.14809	0.14809	0.23789	0.21989	0.21989	0.38663	0.35666	0.35666	0.40221	0.38799	0.38799	0.46568	0.44599	0.44599
EAR-FS	0.18524	0.13493	0.13493	0.19435	0.15003	0.15003	0.24355	0.16463	0.16463	0.21453	0.17586	0.17586	0.20747	0.17533	**0.17533**

Table 9
Synthetic dataset evaluation results on the test set.

Method	SD1			SD2			SD3			SD4			SD5		
	Avg.	Max.	Min.	Avg.	Max.	Min.	Avg.	Max.	Min.	Avg.	Max.	Min.	Avg.	Max.	Min.
MOEFS-ELRF	**0.06760**	**0.06577**	**0.06577**	**0.10289**	**0.09925**	**0.09925**	**0.12355**	**0.11850**	**0.11850**	**0.16276**	**0.14952**	**0.14952**	**0.18784**	**0.17931**	**0.17931**
RF	0.30218	0.29246	0.29246	0.44726	0.43021	0.43021	0.53804	0.52470	0.52470	0.59614	0.57687	0.57687	0.70402	0.68002	0.68002
MOEFS-RF	0.26092	0.24773	0.24773	0.46446	0.43077	0.43077	0.51651	0.48104	0.48104	0.55843	0.52326	0.52326	0.69683	0.65321	0.65321
LSTM	0.30999	0.24034	0.24034	0.39233	0.32078	0.32078	0.48218	0.41724	0.41724	0.52355	0.47257	0.47257	0.59220	0.52557	0.52557
CancelOut	0.27231	0.19253	0.19253	0.37841	0.33633	0.33633	0.47389	0.42932	0.42932	0.50837	0.46385	0.46385	0.57680	0.50498	0.50498
AFS	0.47279	0.35430	0.35430	0.49531	0.42603	0.42603	0.58714	0.53450	0.53450	0.64973	0.57868	0.57868	0.73640	0.69901	0.69901
m-AFS	0.30619	0.19219	0.19219	0.37734	0.29509	0.29509	0.40502	0.34935	0.34935	0.46275	0.39942	0.39942	0.50037	0.45427	0.45427
EAR-FS	0.22145	0.17586	0.17586	0.22693	0.18149	0.18149	0.28022	0.19525	0.19525	0.24598	0.18781	0.18781	0.23581	0.18828	0.18828

Table 10
Number of attributes selected by the FS methods for the synthetic datasets.

Method	SD1			SD2			SD3			SD4			SD5		
	Avg.	Max.	Min.	Avg.	Max.	Min.	Avg.	Max.	Min.	Avg.	Max.	Min.	Avg.	Max.	Min.
MOEFS-ELRF	**7.46**	**7.36**	7.36	**6.66**	**6.52**	6.52	**7.49**	**7.20**	7.20	**8.66**	**8.18**	8.18	**10.42**	**10.21**	10.21
MOEFS-RF	8.40	6	**6**	6.70	6	**6**	8.80	8	8	9.90	6	**6**	11.40	10	**10**
CancelOut	33.00	33	33	33.00	33	33	33.00	33	33	33.00	33	33	33.00	33	33
AFS	16.70	10	10	16.70	10	10	16.70	10	10	16.70	10	10	16.70	10	10
m-AFS	15.30	11	11	16.70	14	14	15.20	12	12	17.60	14	14	16.80	13	13
EAR-FS	27.90	18	18	26.50	18	18	20.30	3	**3**	29.30	24	24	29.30	24	24

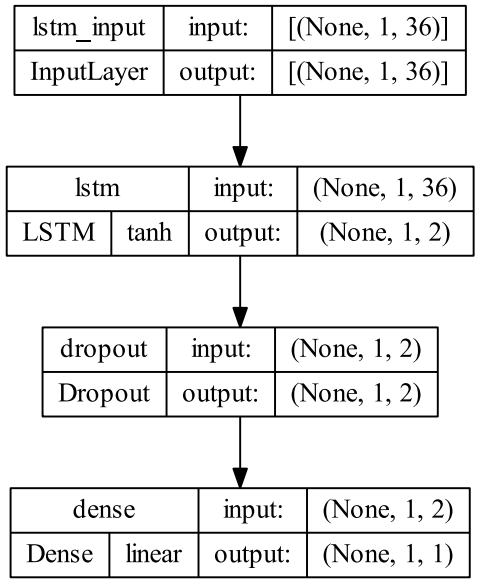


Fig. 4. Architecture of the conventional LSTM network for the Italian city air quality problem.

performance. Each dataset consists of 500 time points and includes a target time series along with a set of input time series (features). All input series were generated using autoregressive processes of order one (AR(1)), defined as:

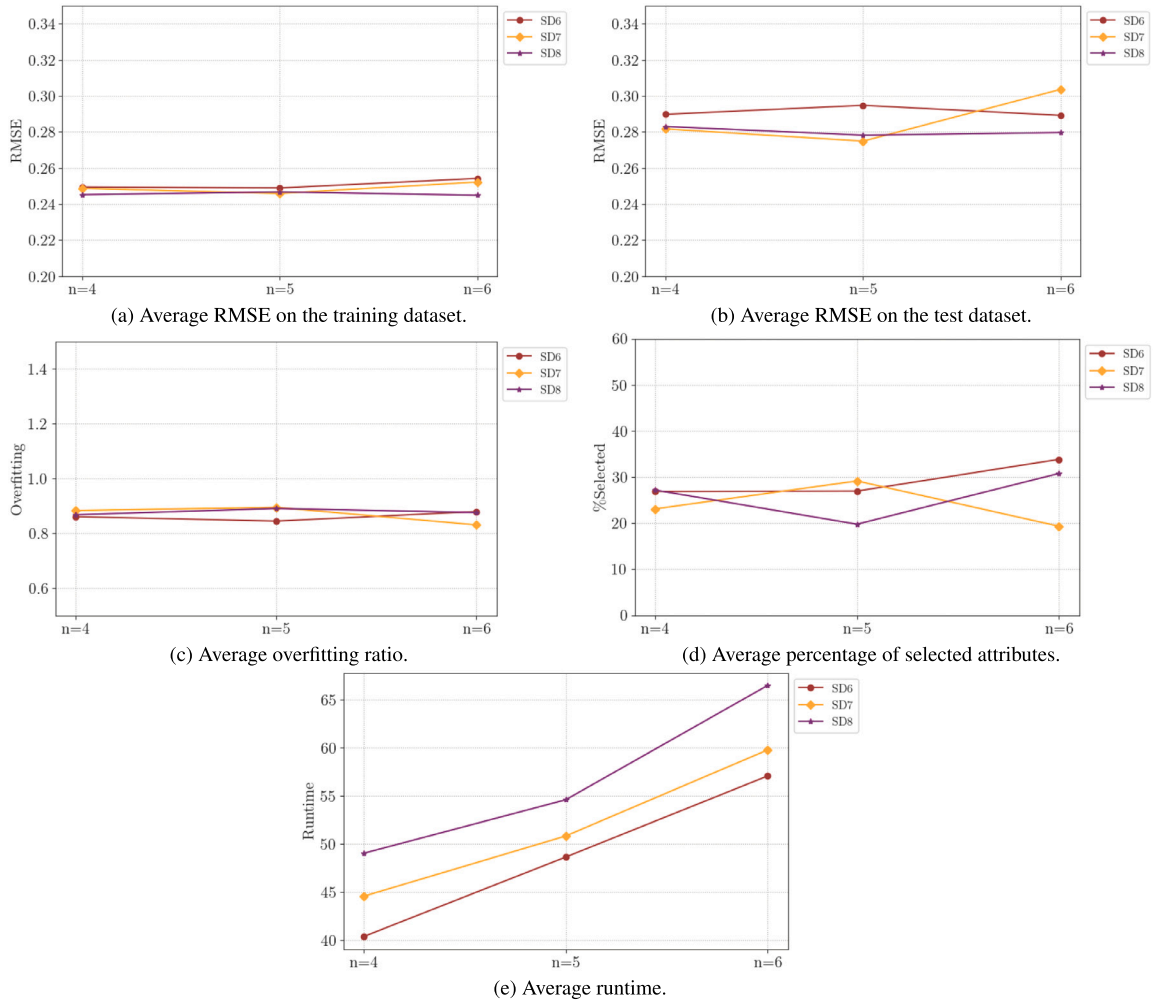
$$x_t = \phi x_{t-1} + \epsilon_t \tag{4}$$

where ϕ is a randomly selected coefficient uniformly drawn from the interval $[0.3, 0.9]$, and ϵ_t is Gaussian white noise with zero mean and unit variance. The same random seed was used throughout to ensure reproducibility and consistency between datasets. The target series was defined as a nonlinear combination of the first five input series, specifically:

$$y_t = 0.5\, x_t^{(1)} + 0.3\, x_t^{(2)} - 0.2\, x_t^{(3)} + 0.1\, x_t^{(4)} + 0.05\, x_t^{(5)} + \eta_t \tag{5}$$

where η_t is additional Gaussian noise with standard deviation 0.3. The target values were then rescaled to the range $[-5, 5]$ using min–max normalization. Importantly, the target series remains fixed across all datasets, allowing us to isolate the effect of increasing the number of input variables on forecasting performance. Each dataset is a strict superset of the previous one, meaning that datasets with higher dimensionality include all the series from the smaller datasets plus new, independently generated AR(1) processes.

NSGA-III was executed 10 times with different random seeds for each of the datasets SD6, SD7, and SD8, using varying numbers of partitions $n = 4, 5, 6$. Each run was configured with 100,000 objective function evaluations, crossover and mutation probabilities set to 1.0, and a population size of approximately 50, computed according to Eq. (3). A three-step-ahead forecasting horizon was used in all cases. The results of this experiment are presented in Fig. 5 and are further discussed in Section 5.



(a) Average RMSE on the training dataset.
(b) Average RMSE on the test dataset.
(c) Average overfitting ratio.
(d) Average percentage of selected attributes.
(e) Average runtime.

Fig. 5. Relationship between number of features and number of partitions.

4.4.5. Scalability with respect to feature dimensionality

In this section, we include a scalability analysis of the proposed MOEFS-ELRF method regarding the dimensionality of the feature space. To this end, we generated five new synthetic datasets following the same procedure described in Section 4.4.4. These datasets were configured with 100, 200, 300, 400 and 500 original features, respectively. After applying a sliding window transformation with a window size of $W = 3$, the final dimensionalities were expanded to 303, 603, 903, 1203 and 1503 attributes. NSGA-II was used as the underlying MOEA in MOEFS-ELRF for this analysis. Each experiment was run 10 times with different predefined random seeds, using 100,000 evaluations of the objective functions, a crossover and mutation probability of 1.0, a population size of 50, and $n = 3$ training partitions. A three-step-ahead forecasting horizon was used in all cases.

We evaluated both the runtime and memory usage of MOEFS-ELRF on these datasets to assess how the method scales as the feature space complexity increases. For each configuration, we generated regression plots with 95% confidence intervals to visualize trends for both runtime and memory. The corresponding plots are shown in Figs. 6 and 7, respectively. Additionally, we computed detailed regression summaries for both metrics and reported them in Tables 11 and 12, which include the precise confidence interval values for the model coefficients as well as the R^2 values of the fitted models. These tables provide a more rigorous view of the underlying relationships between the number of attributes and the computational cost. A more detailed analysis of this behavior is included in Section 5, where we further discuss how the observed trends support the method's scalability within the tested dimensionality range.

It is important to note that we use the *ProcessPoolEvaluator* from the *Platypus* library to parallelize the evaluation of solutions during the multi-objective evolutionary optimization process. Specifically, we set the *n_jobs* parameter to 8, enabling parallel training and evaluation of Random Forest models across multiple CPU cores. This significantly accelerates the optimization process, especially when dealing with large feature sets and multiple objectives. By distributing the computational load of evaluating candidate solutions across available processors, we reduce the overall wall-clock time of each evolutionary generation, making the approach more practical for large-scale and high-dimensional forecasting tasks. This parallelization strategy is particularly beneficial when optimizing over complex search spaces where model training is the primary bottleneck. Furthermore, because each solution in the evolutionary population is independent, the problem is naturally well-suited for embarrassingly parallel computation.

4.4.6. Scalability with respect to forecasting horizon

To further evaluate the scalability and robustness of the proposed MOEFS-ELRF method, we conducted a dedicated analysis to assess its behavior when increasing the forecasting horizon, which is a critical aspect in time series forecasting tasks. As the forecasting horizon increases, the complexity of the prediction task also grows due to the accumulation of prediction errors and the increased difficulty in capturing long-term dependencies. Therefore, it is important to analyze how the proposed method performs and scales in such scenarios. For

Table 11
Regression summary and 95% confidence intervals for runtime vs. number of features.

OLS regression results						
Dep. variable:		y		R-squared:		0.936
Model:		OLS		Adj. R-squared:		0.935
Method:		Least squares		F-statistic:		700.8
No. observations:		50		Prob (F-statistic):		2.74e−30
Df residuals:		48		Log-likelihood:		−241.00
Df model:		1		AIC:		486.0
Covariance type:		Nonrobust		BIC:		489.8
	coef	std err	t	$P > \|t\|$	[0.025	0.975]
const	52.8698	10.181	5.193	0.000	32.399	73.340
x1	0.2701	0.010	26.473	0.000	0.250	0.291
Omnibus:		20.761		Durbin-Watson:		1.571
Prob (Omnibus):		0.000		Jarque–Bera (JB):		27.962
Skew:		1.489		Prob(JB):		8.48e−07
Kurtosis:		5.135		Cond. No.		2.35e+03
95% Confidence intervals						
			2.5%			97.5%
Intercept			32.399409			73.340139
Slope			0.249624			0.290660
R-squared: 0.9359						

Table 12
Regression summary and 95% confidence intervals for memory vs. number of features.

OLS regression results						
Dep. variable:		y		R-squared:		0.800
Model:		OLS		Adj. R-squared:		0.796
Method:		Least squares		F-statistic:		191.8
No. observations:		50		Prob (F-statistic):		2.19e−18
Df residuals:		48		Log-likelihood:		−249.05
Df model:		1		AIC:		502.1
Covariance type:		Nonrobust		BIC:		505.9
	coef	std err	t	$P > \|t\|$	[0.025	0.975]
const	103.5187	11.960	8.656	0.000	79.472	127.566
x1	0.1660	0.012	13.848	0.000	0.142	0.190
Omnibus:		1.818		Durbin-Watson:		0.648
Prob (Omnibus):		0.403		Jarque–Bera (JB):		1.099
Skew:		0.338		Prob(JB):		0.577
Kurtosis:		3.267		Cond. No.		2.35e+03
95% Confidence intervals						
			2.5%			97.5%
Intercept			79.471770			127.565620
Slope			0.141899			0.190104
R-squared: 0.7998						

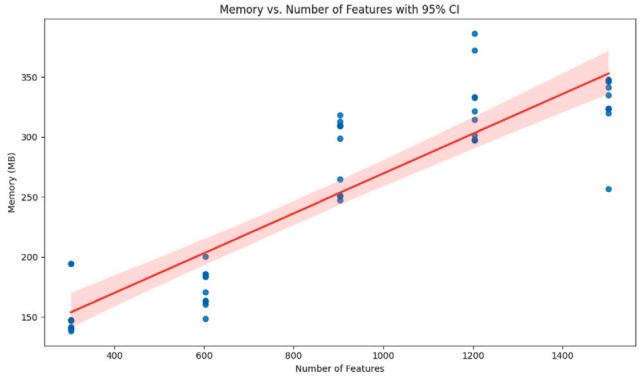


Fig. 6. Run time as a function of the number of features.

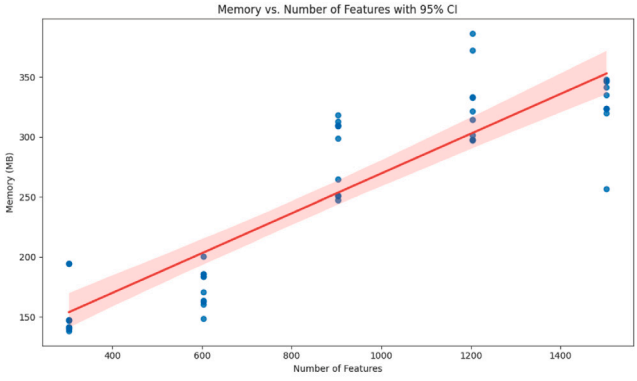
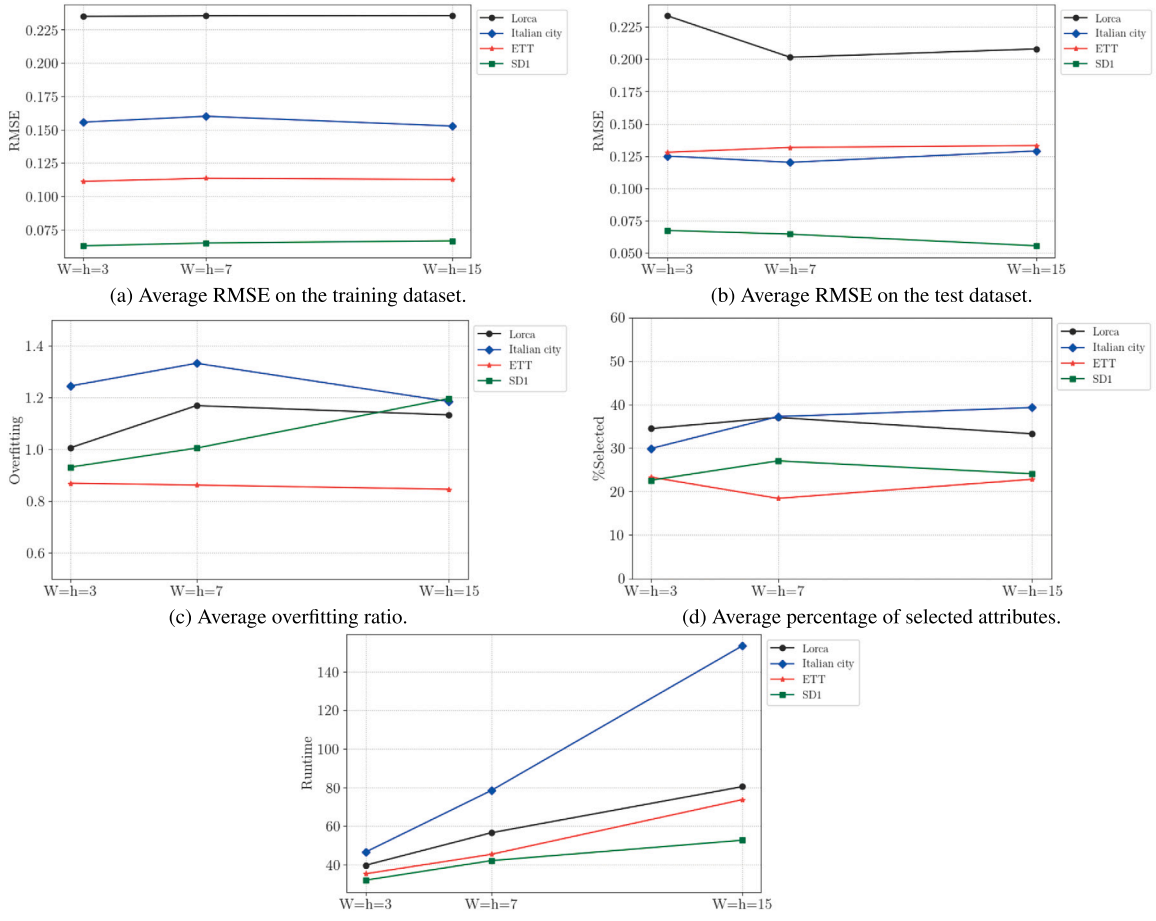


Fig. 7. Memory as a function of the number of features.



(a) Average RMSE on the training dataset.

(b) Average RMSE on the test dataset.

(c) Average overfitting ratio.

(d) Average percentage of selected attributes.

(e) Average runtime.

Fig. 8. Results obtained using different forecasting horizons for the Lorca, Italian city, ETT and SD1 datasets.

Table 13
Average overfitting ratio of the 10 runs of the methods for the real-world application datasets.

Method	Lorca	Italian city	ETT	Average
MOEFS-ELRF	1.00588	1.24441	0.86905	1.03978
RF	0.60217	0.44372	0.50433	0.51674
MOEFS-RF	1.18587	0.55751	0.58052	0.77464
LSTM	1.00683	0.75069	0.71408	0.82386
CancelOut	1.04152	0.74236	0.77142	0.85177
AFS	0.54619	0.50445	0.30165	0.45076
m-AFS	1.00394	0.82683	0.76154	0.86410
EAR-FS	0.82665	0.98845	0.99578	0.93696

Table 14
Average overfitting ratio of the 10 runs of the methods for the synthetic datasets.

Method	SD1	SD2	SD3	SD4	SD5	Average
MOEFS-ELRF	0.93093	1.05371	1.21132	1.08265	0.98930	1.05358
RF	0.38068	0.43454	0.51127	0.48142	0.43228	0.44804
MOEFS-RF	0.51938	0.50728	0.61147	0.63160	0.51105	0.55615
LSTM	0.44613	0.51399	0.72967	0.68550	0.71823	0.61871
CancelOut	0.51846	0.57095	0.76462	0.75203	0.78318	0.67785
AFS	0.54415	0.56743	0.71118	0.57593	0.50826	0.58139
m-AFS	0.56018	0.64702	0.96046	0.88225	0.93562	0.79711
EAR-FS	0.82796	0.85219	0.86327	0.87174	0.87935	0.85890

this purpose, we extended the experiments by evaluating the Lorca, Italian city, ETT, and SD1 datasets with forecasting horizons of $h = 7$ and $h = 15$, in addition to the previously analyzed $h = 3$ horizon. Consistent with the forecasting horizon, the sliding window size W was set equal to the forecasting horizon in each case, ensuring that the input representation is adapted to the required prediction range.

In all experiments, NSGA-II was used as the base MOEA within the MOEFS-ELRF framework. Each configuration was executed 10 times with different predefined random seeds, using 100,000 evaluations of the objective functions, crossover and mutation probabilities set to 1.0, a population size of 50, and a fixed number of partitions/objectives of $n = 3$. The results of this scalability analysis, including the average runtime and predictive performance across the different forecasting horizons, are presented in Fig. 8. A detailed discussion and interpretation of these results can be found in Section 5.

Table 15
Win-loss ranking of statistically significant differences with Diebold–Mariano test for the real-world application datasets.

Method	Win	Loss	Win − Loss
MOEFS-ELRF	63	0	63
EAR-FS	54	9	45
m-AFS	15	22	−7
LSTM	13	21	−8
CancelOut	12	23	−11
AFS	7	19	−12
RF	3	34	−31
MOEFS-RF	6	45	−39

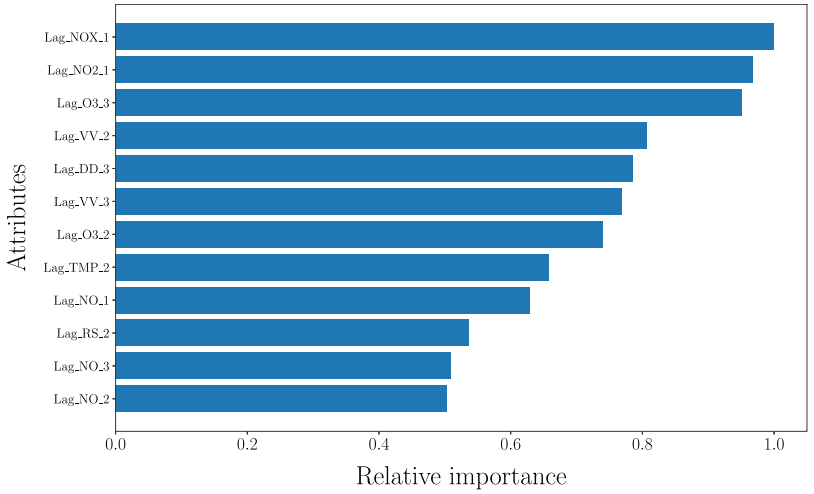


Fig. 9. Feature importance with MOEFS-ELRF for the Lorca air quality problem.

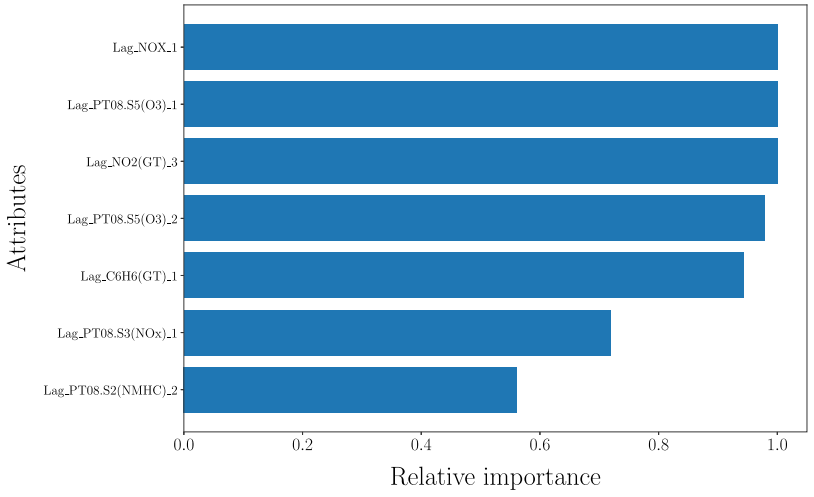


Fig. 10. Feature importance with MOEFS-ELRF for the Italian city air quality problem.

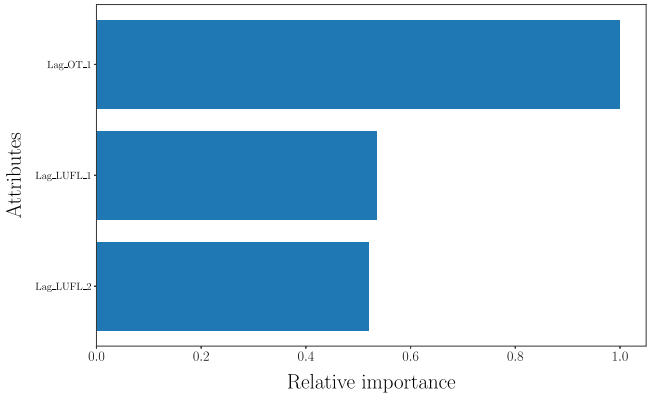


Fig. 11. Feature importance with MOEFS-ELRF for the ETT problem.

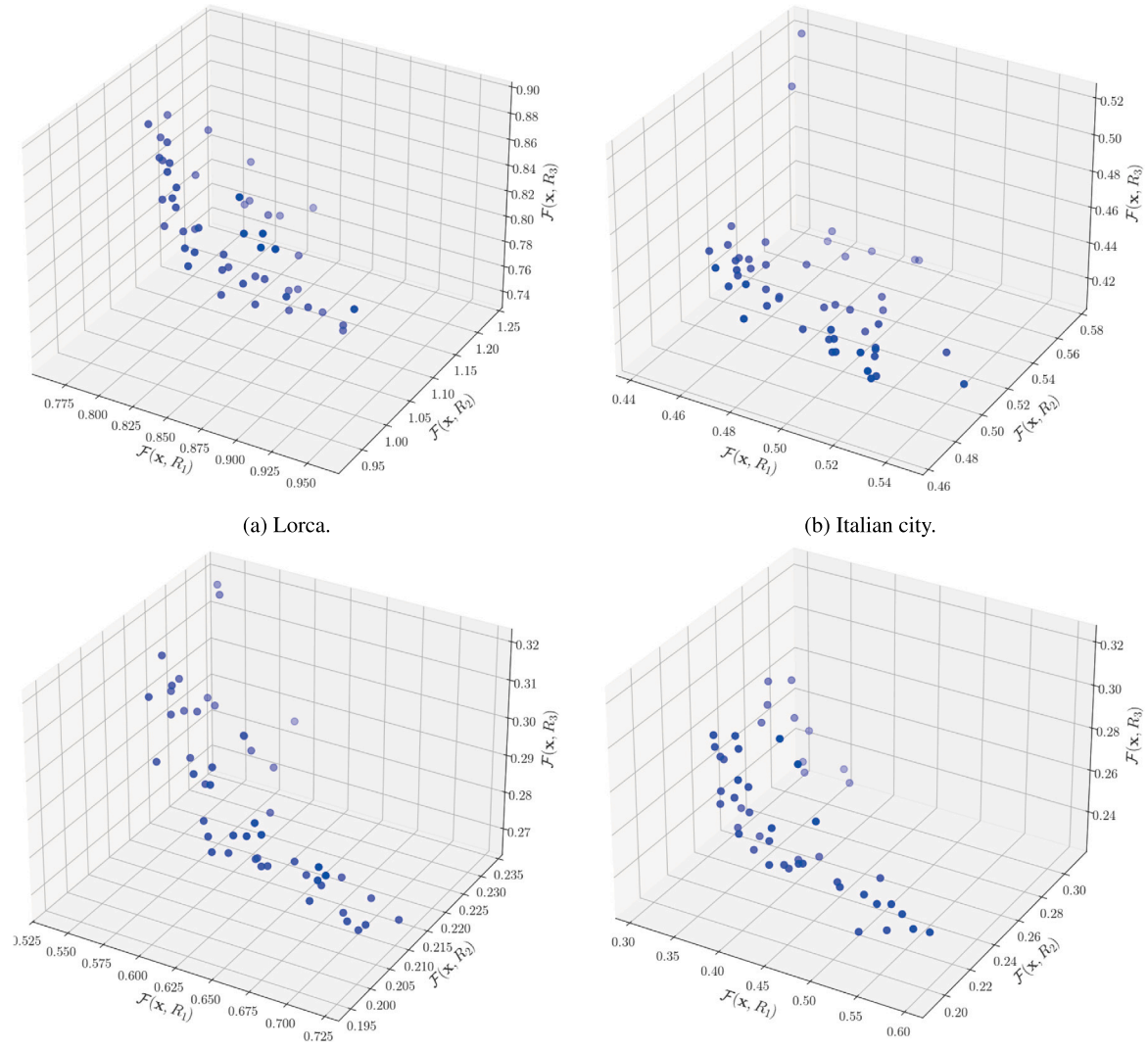
Table 16
Win-loss ranking of statistically significant differences with Diebold–Mariano test for the synthetic datasets.

Method	Win	Loss	Win − Loss
MOEFS-ELRF	105	0	105
EAR-FS	47	15	32
LSTM	27	23	4
AFS	16	22	−4
CancelOut	18	27	−9
MOEFS-RF	13	38	−25
RF	3	39	−36
m-AFS	5	70	−65

5. Analysis of results and discussion

In this section we analyze and discuss the results obtained with MOEFS-ELRF in comparison with the other seven methods considered in this paper. The following highlights can be extracted from the results:

– The results obtained for RMSE in both training and test sets demonstrate the outstanding performance of MOEFS-ELRF compared to the other methods. Our proposed approach consistently



(a) Lorca. (b) Italian city.

(c) ETT. (d) SD1.

Fig. 12. Pareto fronts for Lorca, Italian city, ETT and SD1.

Table 17
Average percentage of the number of attributes selected by the FS methods and their total average percentages for the real-world application datasets.

Method	%Lorca	%Italian city	%ETT	%Average
MOEFS-ELRF	34.47	29.88	**23.32**	29.23
MOEFS-RF	**3.33**	**25.28**	25.71	**18.11**
CancelOut	55.67	58.33	59.52	57.84
AFS	43.00	51.94	49.05	48.00
m-AFS	49.33	50.00	48.10	49.14
EAR-FS	63.00	76.94	94.76	78.24

Table 18
Average percentage of the number of attributes selected by the FS methods and their total average percentages for the synthetic datasets.

Method	%SD1	%SD2	%SD3	%SD4	%SD5	%Average
MOEFS-ELRF	**22.60**	**20.18**	**22.68**	**26.26**	**31.56**	**24.66**
MOEFS-RF	25.45	20.30	26.67	30.00	34.55	27.39
CancelOut	91.82	92.73	94.55	93.94	94.24	93.46
AFS	50.61	50.61	50.61	50.61	50.61	50.61
m-AFS	46.36	50.61	46.06	53.33	50.91	49.45
EAR-FS	84.55	80.30	61.52	88.79	88.79	80.79

Table 19
Average runtime (in minutes) of the 10 runs of the methods for the real-world application datasets.

Method	Lorca	Italian city	ETT	Average
MOEFS-ELRF	39.75	46.57	35.32	40.55
MOEFS-RF	12.67	22.41	14.05	16.38
CancelOut	0.10	0.40	**0.05**	**0.18**
AFS	0.17	0.70	0.62	0.50
m-AFS	**0.09**	**0.38**	0.38	0.28
EAR-FS	0.99	1.26	1.39	1.21

Table 20
Average runtime (in minutes) of the 10 runs of the methods for the synthetic datasets.

Method	SD1	SD2	SD3	SD4	SD5	Average
MOEFS-ELRF	31.99	31.85	32.64	34.93	36.07	33.50
MOEFS-RF	13.16	13.39	13.74	12.97	14.53	13.56
CancelOut	0.10	0.12	0.12	0.12	0.12	0.12
AFS	1.13	1.31	1.31	1.32	1.32	1.28
m-AFS	1.07	1.11	1.11	1.12	1.11	1.10
EAR-FS	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**

Table 21
Results of the best forecast model for the Lorca air quality problem, evaluated on the training dataset R and the test dataset T.

Evaluation dataset	Performance metric	1-step ahead	2-steps ahead	3-steps ahead
R	RMSE	0.24099	0.23631	0.23745
	MAE	0.18222	0.17859	0.17848
T	RMSE	0.23522	0.22642	0.23010
	MAE	0.19043	0.18945	0.18856

Table 22
Results of the best forecast model for the Italian city air quality problem, evaluated on the training dataset R and the test dataset T.

Evaluation dataset	Performance metric	1-step ahead	2-steps ahead	3-steps ahead
R	RMSE	0.15483	0.15230	0.16266
	MAE	0.10863	0.10911	0.11381
T	RMSE	0.11556	0.13337	0.12010
	MAE	0.07925	0.09079	0.08528

Table 23
Results of the best forecast model for the ETT problem, evaluated on the training dataset R and the test dataset T.

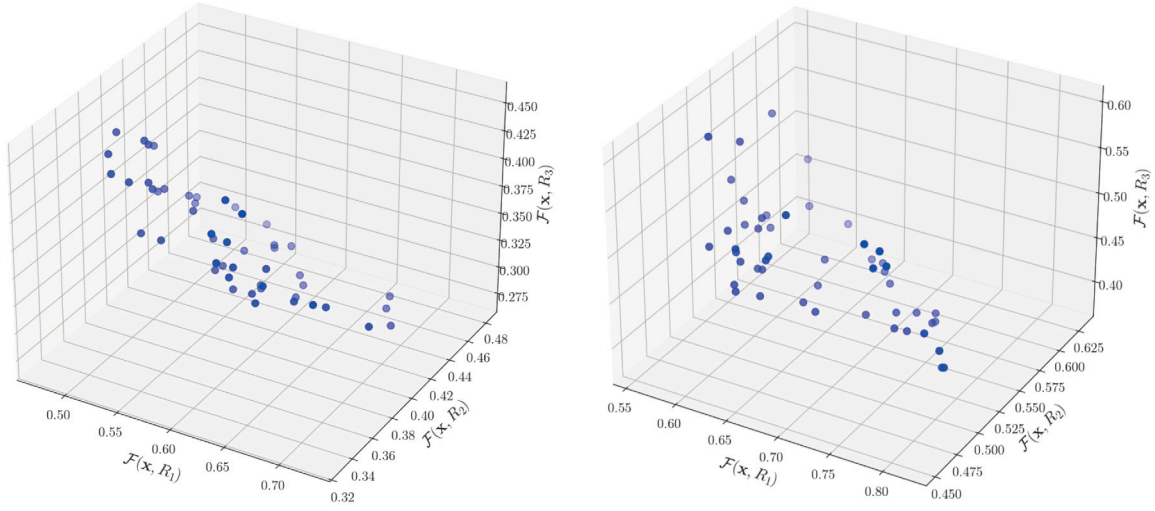
Evaluation dataset	Performance metric	1-step ahead	2-steps ahead	3-steps ahead
R	RMSE	0.11250	0.11184	0.11291
	MAE	0.07293	0.07447	0.07500
T	RMSE	0.12019	0.12666	0.12892
	MAE	0.08503	0.08525	0.08561

outperforms all competing techniques across the three evaluation metrics–maximum, minimum, and mean RMSE–on both real-world application datasets and synthetic datasets. Notably, MOEFS-ELRF achieves the best predictive accuracy in all cases, demonstrating its robustness and generalization capabilities. The only instances where MOEFS-ELRF does not achieve the best result are in the training set of the Lorca and SD5 datasets, specifically in the minimum RMSE metric, where it is slightly outperformed by EAR-FS. However, it is important to highlight that this occurs exclusively on the training data, and the overall generalization performance of MOEFS-ELRF remains superior across all test sets. This suggests that while EAR-FS might occasionally optimize its training performance more aggressively, it does not translate into better generalization on unseen data. These observations can be clearly seen in Tables 5, 6, 8 and 9, which summarize the RMSE values obtained across multiple runs. The results reinforce the effectiveness of MOEFS-ELRF in both real and synthetic datasets, proving its capability to select relevant features while maintaining high predictive accuracy.

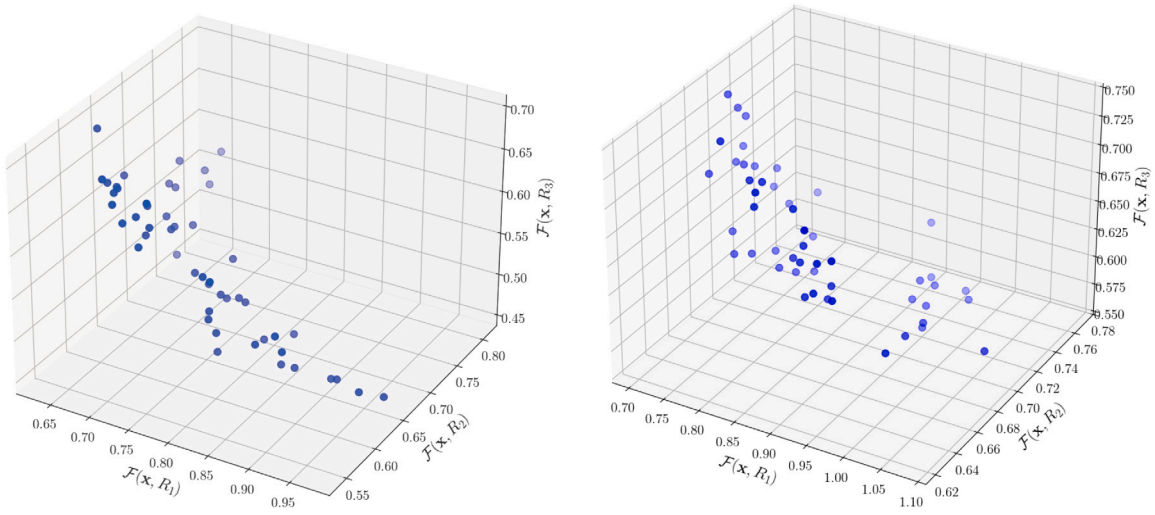
– To evaluate the generalization capability of the forecasting models, we computed the *overfitting ratio* for each method, defined as the ratio of its RMSE on the training dataset to its RMSE on the test dataset. A value significantly lower than 1 indicates overfitting, as it suggests that the model achieves much lower errors on the training data compared to the test data. Tables 13 and 14 summarize the average overfitting ratios obtained over 10 runs for all methods on real-world application datasets and synthetic datasets, respectively.
The results indicate that MOEFS-ELRF exhibits the most balanced overfitting ratio, with values close to 1 across all datasets. In real-world application datasets (Table 13), MOEFS-ELRF achieves an average ratio of 1.03978, demonstrating its ability to generalize well without overfitting. In contrast, methods such as RF and MOEFS-RF show considerably lower ratios (0.51674 and 0.77464, respectively), indicating a tendency to overfit. Similarly, AFS exhibits the lowest overfitting ratio (0.45076), suggesting that it suffers from significant overfitting issues. On the other hand, EAR-FS achieves a relatively balanced ratio (0.93696), but still does not reach the stability observed with MOEFS-ELRF. In synthetic datasets (Table 14), a similar trend is observed. MOEFS-ELRF maintains a near-optimal overfitting ratio of 1.05358, confirming its generalization ability. Traditional RF exhibits the highest degree of overfitting with an average ratio of only 0.44804, while MOEFS-RF, LSTM, and CancelOut also show overfitting tendencies, though to a lesser extent. Notably, EAR-FS continues to perform reasonably well in avoiding overfitting, with an average ratio of 0.85890, but still lags behind MOEFS-ELRF. Overall, these results reinforce the superior generalization ability of MOEFS-ELRF across both real-world and synthetic datasets. Unlike other methods that either exhibit excessive overfitting (such as RF and AFS) or struggle with stable generalization (such as EAR-FS), MOEFS-ELRF consistently maintains an overfitting ratio close to 1, indicating its ability to balance learning from training data while ensuring reliable performance on unseen test data.

– Statistical tests are essential for rigorously evaluating machine learning algorithms, as they ensure that observed performance differences are not merely due to chance. In this study, we employ the *Diebold–Mariano* test to assess the predictive accuracy of the forecasting models. This test is particularly suitable for time series data, where observations are sequentially dependent, allowing us to determine whether one model significantly outperforms another for each of the h steps ahead ($h \in 1, 2, 3$). To quantify the comparative performance of the different approaches, we conducted pairwise Diebold–Mariano tests for all methods. If a statistically significant difference was found in favor of method A over method B, method A was awarded a 'win', while method B received a 'loss'. This allows for the construction of a win-loss ranking that identifies the best-performing method in a statistically robust manner. Tables 15 and 16 summarize the results for the real-world and synthetic datasets, respectively. The results confirm the superiority of MOEFS-ELRF, which achieves a perfect win-loss balance of $63 - 0$ in real-world datasets and $105 - 0$ in synthetic datasets, demonstrating its statistically significant advantage over all other methods. EAR-FS ranks second in both settings, with a positive win-loss balance ($54 - 9$ in real-world datasets and $47 - 15$ in synthetic datasets), indicating that it performs competitively but is consistently outperformed by MOEFS-ELRF. For the remaining methods, performance varies. In real-world datasets, m-AFS, LSTM, and CancelOut achieve similar results, with slightly negative win-loss balances, indicating that they are competitive but frequently outperformed by stronger methods. RF and MOEFS-RF exhibit the weakest performance, with large negative win-loss differentials (-31 and -39, respectively). A similar pattern is observed in synthetic datasets, where LSTM emerges as the best among the mid-tier methods, achieving a small positive win-loss balance ($27-23$). AFS and CancelOut follow closely, while MOEFS-RF and RF remain among the weakest. Notably, m-AFS exhibits the poorest performance, with a dramatic win-loss deficit of -65, highlighting its relative inefficacy in synthetic environments. In conclusion, these statistical analyses confirm that MOEFS-ELRF consistently outperforms all competing methods, with no observed losses across multiple datasets. This highlights its robustness and reliability for time series forecasting tasks.

– Regarding the relationship between feature dimensionality and the number of training partitions, although no strict deterministic pattern was observed, the experimental findings suggest that datasets with a higher number of features tend to benefit slightly from a larger number of partitions. This configuration allows for a more effective distribution of the feature space across multiple sub-models, potentially improving the learning process.



(e) SD2. (f) SD3.



(g) SD4. (h) SD5.

Fig. 13. Pareto fronts for SD2, SD3, SD4 and SD5.

However, increasing the number of partitions beyond a certain threshold does not necessarily lead to better performance and may even introduce instability due to the reduced size of each training subset. As expected, execution time is influenced by both factors: datasets with more features require longer processing times, and increasing the number of partitions leads to a higher number of objective functions, which in turn increases the overall computational cost.

– Regarding the relationship between feature dimensionality and runtime, a regression analysis was conducted to evaluate the scalability of MOEFS-ELRF. The results demonstrate an approximately linear trend in runtime growth as the number of features increases, with a coefficient of determination $R^2 = 0.936$, indicating that about 94% of the variability in runtime is explained by feature dimensionality under our experimental setup. The 95% confidence interval for the regression slope is narrow (0.2496 to 0.2907), confirming the stability and reliability of this observed trend. A similar analysis was performed for memory usage as a function of the number of features. Here, the results also reveal an approximately linear scaling behavior, with an even higher R^2 value of 0.995 and extremely tight confidence intervals for the slope. This aligns with the theoretical expectation that memory demands grow linearly with feature dimensionality, given the bitstring-based encoding of individuals in the MOEA. In summary, these analyses support the conclusion that both runtime and memory usage exhibit predictable, approximately linear scalability within the tested range of feature dimensionalities. Nonetheless, for scenarios involving substantially larger feature spaces or more complex temporal structures, surrogate modeling techniques – such as those proposed in [55] – could be explored to reduce training and optimization costs further.

– The results of the scalability analysis of MOEFS-ELRF with respect to the forecasting horizon indicate the following: Regarding the RMSE on the training sets, the results indicate a stable behavior across all four datasets, with the RMSE values remaining practically unchanged as the forecasting horizon increases from 3 to 7 and 15 steps ahead. This suggests that the proposed method is able to adapt the feature selection and modeling process effectively regardless of the forecasting horizon, maintaining a similar level of fit on the training data. When analyzing the RMSE on the test sets, the Lorca and SD1 datasets show a noticeable improvement in RMSE as the forecasting horizon increases, suggesting that the models generated for longer horizons exhibit enhanced generalization capabilities for these datasets. In contrast, for the Italian city and ETT datasets, the RMSE on the test set remained practically stable across all forecasting horizons, indicating that

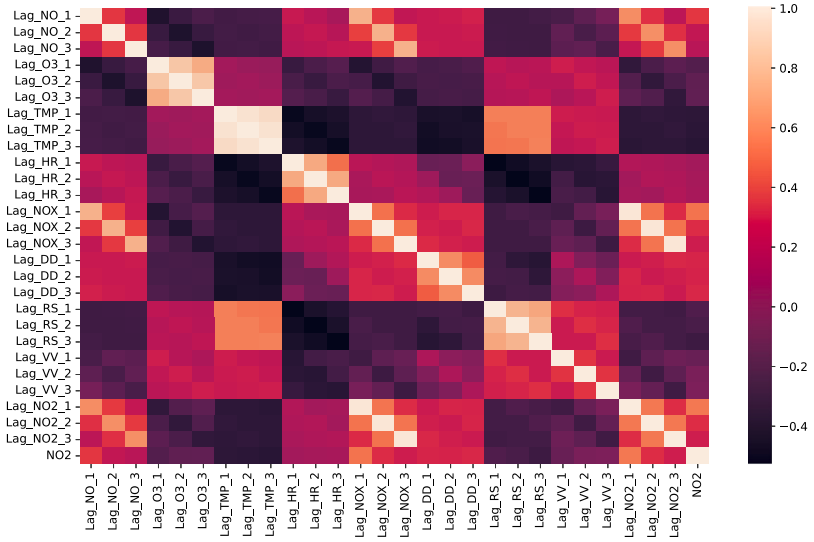


Fig. B.14. Correlation matrix for the top-1 attributes of MOEFS-ELRF for the Lorca air quality problem.

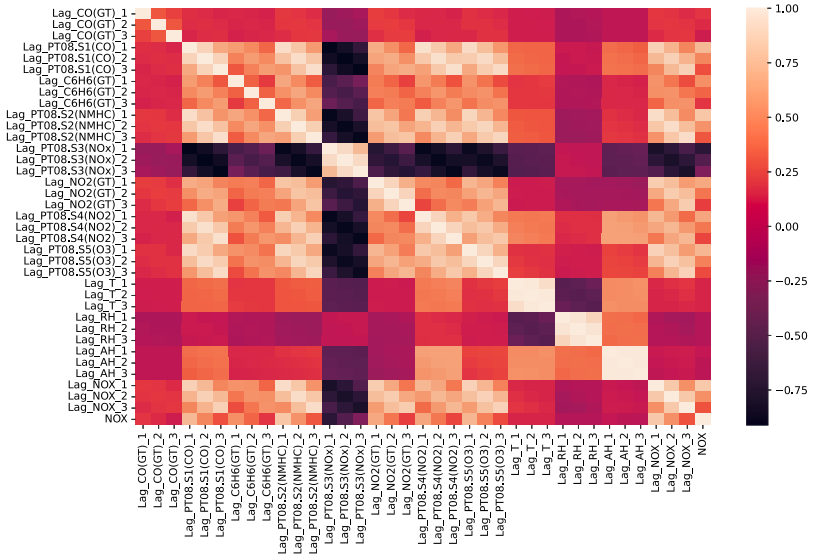


Fig. B.15. Correlation matrix for the top-1 attributes of MOEFS-ELRF for the Italian city air quality problem.

the proposed method maintains a consistent predictive performance even as the forecasting horizon increases in these cases. Consequently, the overfitting ratio increases for the Lorca and SD1 datasets as the forecast horizon increases, indicating a reduction in overfitting. Meanwhile, the overfitting ratio for the Italian city and ETT datasets remains relatively stable across all horizons, reflecting the aforementioned consistency in their test RMSE values. Regarding the percentage of selected features, a general increase of approximately 5% to 10% is observed in the Lorca, Italian city, and SD1 datasets as the forecasting horizon increases. This behavior can be attributed to the increased complexity and feature space resulting from the larger window sizes required for longer horizons, which demand a slightly larger subset of features to maintain model performance. Interestingly, the ETT dataset shows a slight decrease (around 1% to 5%) in the percentage of selected features, suggesting that the most relevant variables remain dominant regardless of the forecasting horizon in this particular dataset. Finally, as expected, the runtime increases proportionally to the number of input features introduced by the varying window sizes for each forecasting horizon. This trend is aligned with the increased computational burden associated with handling larger feature spaces, which affects both the model evaluation phase (due to the internal RF training) and the overall evolutionary search process.

In conclusion, the proposed MOEFS-ELRF method demonstrates robust scalability with respect to forecasting horizon. While the method manages to maintain stable or improved generalization performance across datasets, the computational cost increases proportionally with the input dimensionality induced by longer horizons. Importantly, the method remains effective in controlling overfitting and adapting the feature selection process to the increased forecasting challenges posed by longer horizons, without showing abrupt degradations in predictive accuracy or feature selection efficiency.

– Feature selection is essential for enhancing model interpretability and improving generalization by reducing redundancy and eliminating irrelevant attributes. Tables 17 and 18 summarize the average percentage of selected features for each method across real-world and synthetic datasets. For real-world datasets, MOEFS-ELRF selects a moderate percentage of attributes (29.23% on average), with its most restrictive selection observed in the ETT dataset (23.32%). MOEFS-RF exhibits the most aggressive

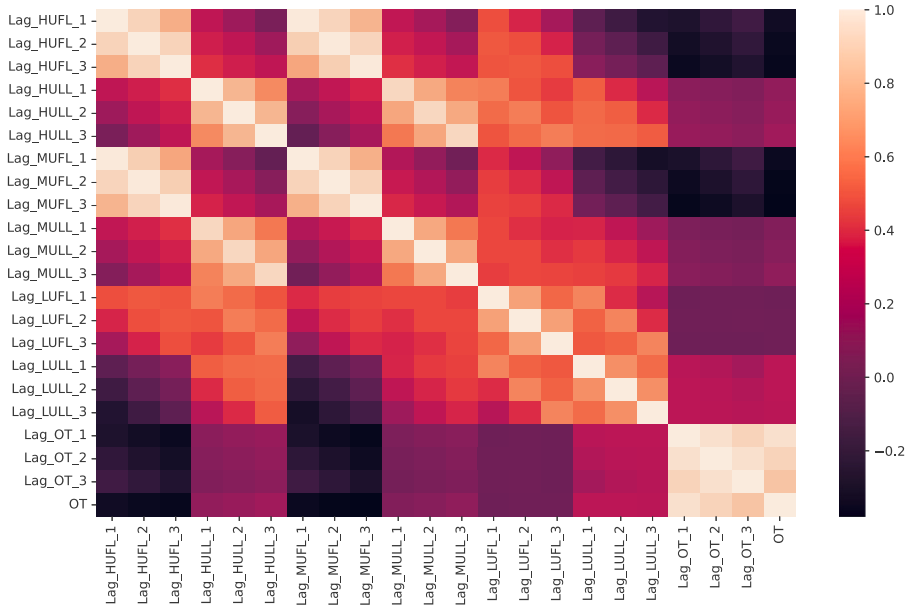


Fig. B.16. Correlation matrix for the top-1 attributes of MOEFS-ELRF for the ETT problem.

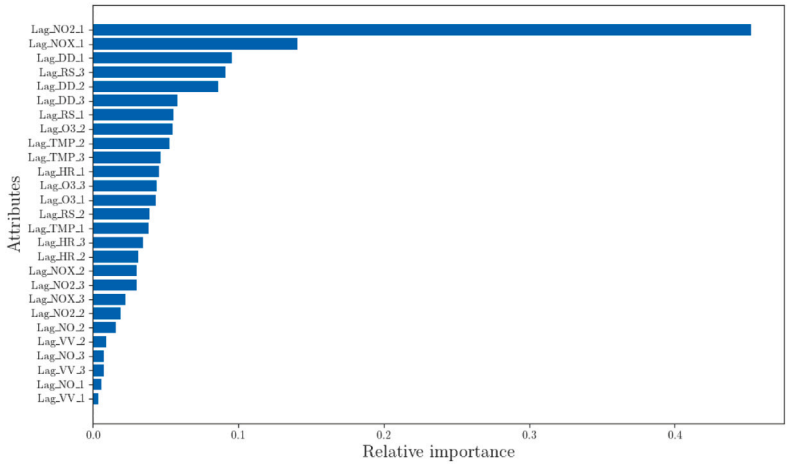


Fig. B.17. Permutation feature importance for the Lorca air quality problem.

feature reduction, selecting only 18.11% of the features on average and as low as 3.33% for the Lorca dataset, indicating its strong filtering capability. In contrast, CancelOut and AFS retain a significantly higher proportion of features, with average selection rates of 57.84% and 48.00%, respectively. Similarly, m-AFS selects nearly half of the available features (49.14%), suggesting a conservative feature selection strategy. EAR-FS stands out as the least selective method, retaining 78.24% of the features on average and up to 94.76% for the ETT dataset, indicating its reliance on a broader set of attributes. A similar trend is observed in synthetic datasets. MOEFS-ELRF consistently selects the smallest percentage of features across all five synthetic datasets, with an overall average of 24.66%, reinforcing its ability to effectively eliminate non-informative attributes. MOEFS-RF follows closely with a slightly higher selection rate of 27.39%, suggesting a somewhat more flexible feature selection approach. CancelOut, unlike in the real-world datasets, retains a significantly higher proportion of attributes in synthetic datasets, selecting 93.46% on average. This suggests that CancelOut is less effective in filtering out irrelevant features in synthetic scenarios, likely maintaining redundant attributes. AFS and m-AFS follow a more balanced selection approach, retaining around 50% of the features on average. Meanwhile, EAR-FS selects a large portion of attributes (80.79%), reinforcing its tendency to preserve a broader feature set even in synthetic scenarios. Overall, these results highlight that MOEFS-ELRF consistently achieves substantial feature reduction while maintaining predictive performance. Its ability to select fewer attributes than competing methods in both real-world and synthetic datasets demonstrates its effectiveness in filtering out redundant and non-informative features. This characteristic is particularly beneficial in scenarios where feature interpretability and model efficiency are critical.

– To assess the consistency and robustness of the feature importance values obtained by MOEFS-ELRF, we analyze the correlation of the selected attributes with importance 1 against the target variable and the rest of the input attributes. Figs. 9–11 graphically show the importance of the attributes selected with MOEFS-ELRF for the Italian city, Lorca and ETT problems re-

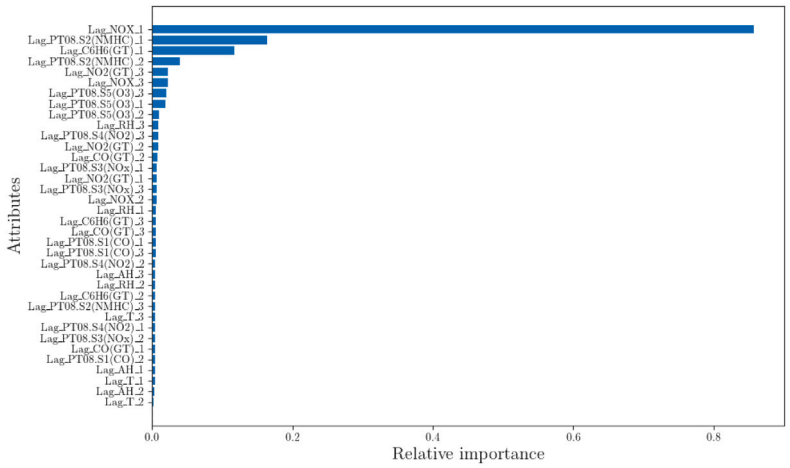


Fig. B.18. Permutation feature importance for the Italian city air quality problem.

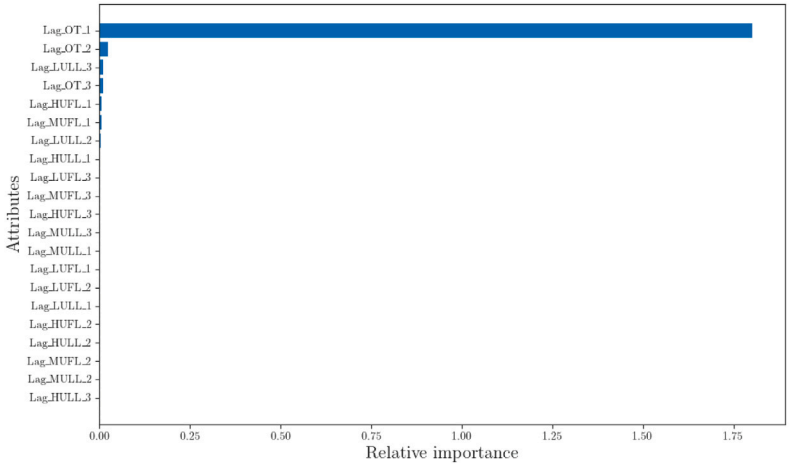


Fig. B.19. Permutation feature importance for the ETT problem.

spectively. Figs. B.14–B.16 show the correlation matrix between the attributes for each of the datasets. A robust feature importance method should consistently identify features that are highly correlated with the target while maintaining low correlations with other input attributes to avoid redundancy. For the Lorca dataset, the attributes most correlated with the target variable (NO_2) are Lag_NOX_1 and Lag_NO2_1, which are among the most important features selected by MOEFS-ELRF. Notably, both attributes exhibit low correlations with the remaining input features, suggesting that they provide unique, relevant information for predicting NO_2 levels. In the Italian city dataset, the attributes Lag_NOX_1 and Lag_PT08.S5(O3)_1, identified as highly important by MOEFS-ELRF, show strong correlations with the target variable NO_X. Additionally, the selected attribute Lag_NO2(GT)_3 exhibits low correlation with Lag_NOX_1, indicating that these two features complement each other rather than providing redundant information. This suggests that MOEFS-ELRF effectively selects diverse and informative features for forecasting. For the ETT dataset, the attribute Lag_OT_1, identified with importance 1, stands out as the most correlated feature with the target variable OT while also showing the lowest correlation with the other input attributes. This reinforces the ability of MOEFS-ELRF to isolate key predictive variables without introducing unnecessary redundancy. These observations confirm that the feature importance method embedded within MOEFS-ELRF is consistent and robust, as it systematically identifies attributes with strong predictive value while maintaining low redundancy, which enhances generalization and model interpretability.

In addition to this correlation-based validation, we include a comparative analysis using*permutation feature importance* [56], a widely used post-hoc evaluation method that assesses the relevance of a feature by measuring the increase in the model's prediction error when its values are randomly shuffled. Figs. B.17, B.18, and B.19 show the permutation-based feature importance rankings for the Lorca, Italian city, and ETT datasets, respectively, using RF as the base model. For the Lorca dataset, Lag_NOX_1 and Lag_NO2_1–identified by MOEFS-ELRF with the highest

importance–are also the top-ranked features according to permutation importance. The same holds for Lag_NOX_1 in the Italian city dataset and for Lag_OT_1 in the ETT dataset. These results provide additional support for the reliability and robustness of the feature importance values produced by MOEFS-ELRF. The strong agreement between the selection-frequency-based importance and the permutation-based rankings confirms that the features highlighted by our method are not only frequently selected during evolutionary optimization but also play a critical

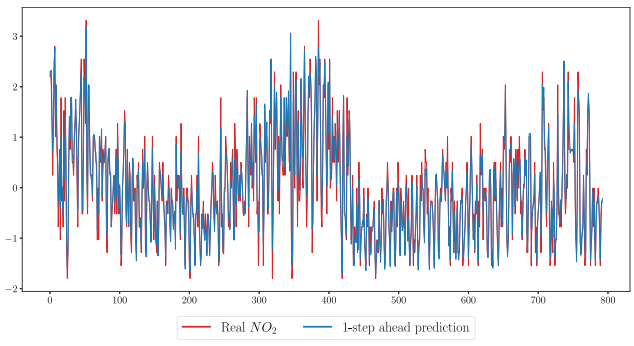
role in predictive performance according to independent post-training evaluations. This reinforces the interpretability of the model and validates the feature selection process from both internal and external perspectives.

– One of the main limitations of MOEFS-ELRF is its computational cost during training. Tables 19 and 20 present the average execution times of the methods analyzed in this study.[6] The extended training time is a direct consequence of the optimization process employed, which significantly enhances performance compared to conventional RF. This includes lower RMSE, reduced overfitting, and improved generalization across the evaluated datasets. The higher computational demand arises from the iterative nature of the evolutionary algorithm, which explores multiple feature subsets before converging to an optimal solution. While this increases the training time, the resulting predictive improvements justify the additional computational effort. This method is particularly suitable for applications where models can be trained offline, as it does not impact real-time inference. Once trained, the model can make predictions efficiently, making it feasible for deployment in practical scenarios. Furthermore, with ongoing advancements in hardware and parallel processing, the computational burden of evolutionary approaches is expected to decrease over time. As computing power continues to improve, the trade-off between training time and model performance may become less significant, further enhancing the applicability of MOEFS-ELRF.

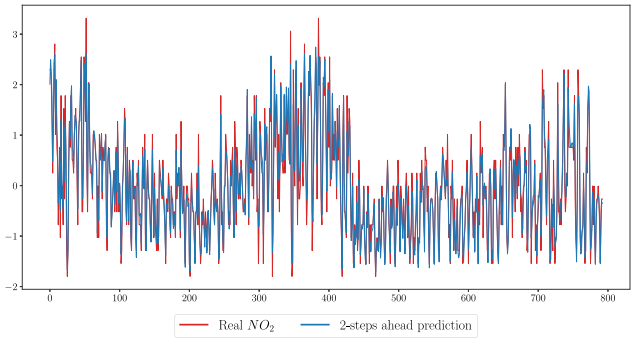
– An essential aspect of the proposed method is ensuring sufficient diversity among the input models used in the stacking ensemble. This diversity is primarily driven by two factors: the definition of the objective functions in the underlying multi-objective optimization problem–which depend on the training dataset and how it is partitioned–and the diversity mechanisms embedded within the chosen MOEA. It is important to highlight that the non-dominated RF models used in the stacking ensemble are each trained on different partitions of the training set and evaluated through 5-fold cross-validation within their respective partitions. This strategy guarantees that the optimized feature subsets reflect the specific statistical properties and patterns of each partition, resulting in models with distinct strengths and weaknesses. Consequently, even when some models show similar levels of performance, they differ in terms of the features selected and the temporal structures learned from their respective partitions, which contributes to the overall diversity required by the ensemble. To empirically analyze this diversity, Figs. 12 and 13 display the Pareto fronts obtained during the optimization process. These plots illustrate the spread and distribution of the non-dominated solutions – each representing a unique feature subset and its associated RF model – clearly demonstrating the degree of diversity among the candidate models used as inputs to the stacking phase.

– Beyond its strong generalization ability, another key aspect of the forecasting model generated by MOEFS-ELRF is its robustness in multi-step-ahead predictions. While the recursive strategy for multi-step forecasting often leads to error accumulation across successive steps, the predictions produced by the model remain stable across the entire forecasting horizon in all TSF problems. As shown in Tables 21–23 for the real-world datasets, MOEFS-ELRF maintains accuracy without significant performance degradation over multiple time steps. To conclude this results analysis section, Figs. C.20 to C.25 illustrate the predictions obtained by MOEFS-ELRF for 1, 2, and 3 steps ahead on both the training and test datasets across the three real-world forecasting problems analyzed.

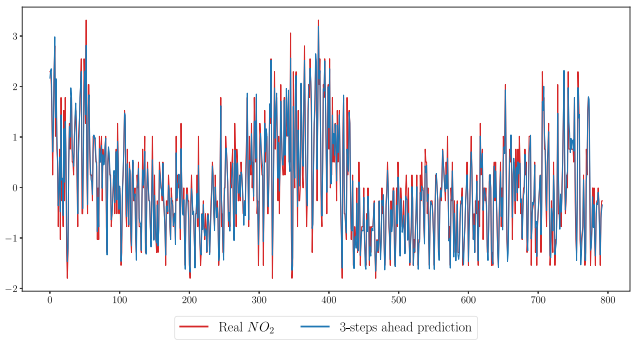
[6] Experiments were conducted on a computer with motherboard MR91-FS0-00 (R281-3C2-00), memory 12 × 32 GiB DIMM DDR4 2933 MHz (18ASF4G72PDZ-2G9B2), CPU 2 × Intel Xeon Gold 6226R.



(a) 1-step ahead



(b) 2-steps ahead



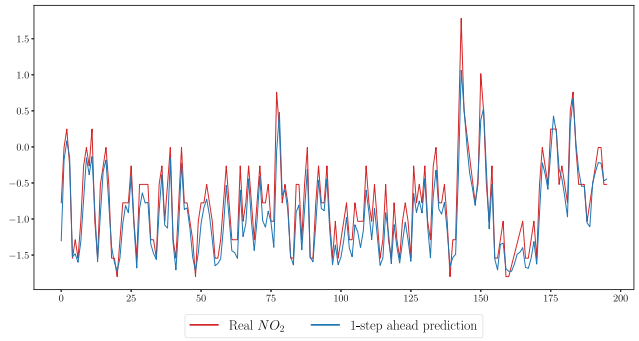
(c) 3-steps ahead

Fig. C.20. 1, 2 and 3 steps ahead predictions of the forecast model obtained with MOEFS-ELRF on training dataset for the Lorca air quality problem.

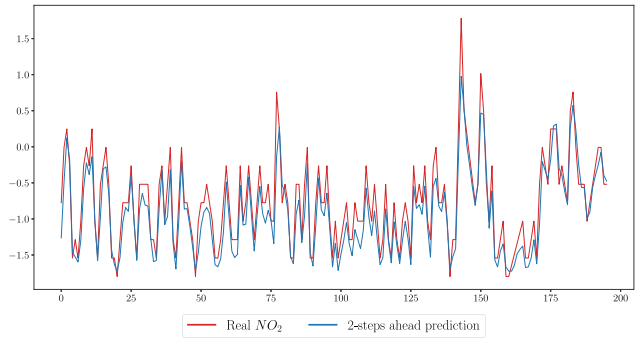
6. Conclusions and future work

This work has introduced MOEFS-ELRF, a novel wrapper-based FS method that leverages multi-objective evolutionary optimization to enhance TSF. The proposed approach partitions the dataset into multiple subsets, associating each partition with a distinct objective in the optimization problem. The Pareto-optimal solutions identified by the evolutionary algorithm are then used to construct multiple RF models, which are integrated through a stacking-based ensemble meta-model. This EL strategy significantly improves the generalization capability of the conventional RF model while simultaneously mitigating overfitting.

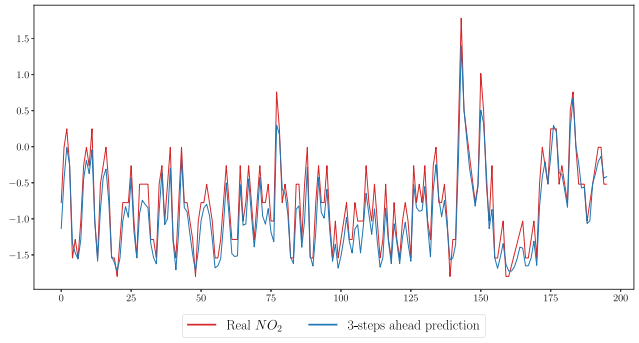
One of the distinguishing aspects of MOEFS-ELRF is its simplicity and flexibility. The method requires only a single execution of a MOEA and does not depend on a specific MOEA design, making it compatible with any standard MOEA or many-objective algorithm depending on the number of partitions. This makes the approach easily adaptable to a wide range of TSF problems without the need for complex co-evolutionary or multi-phase strategies often proposed in the EL literature.



(a) 1-step ahead

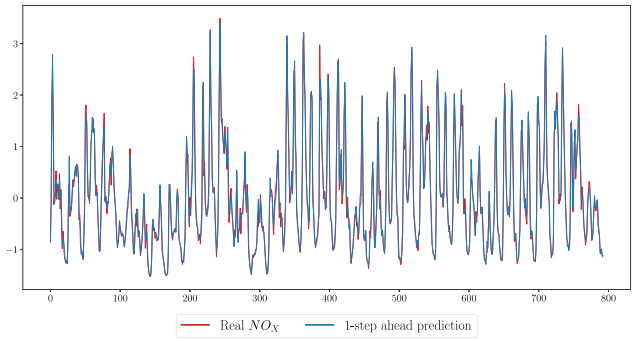


(b) 2-steps ahead

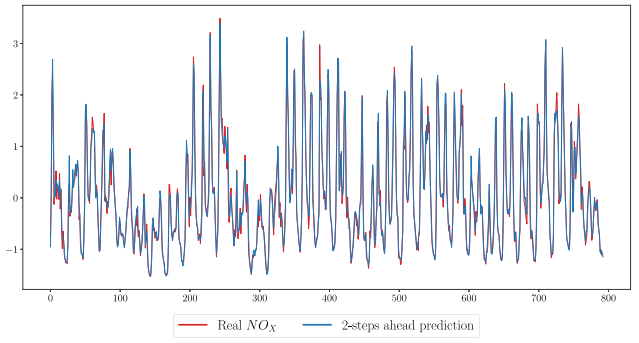


(c) 3-steps ahead

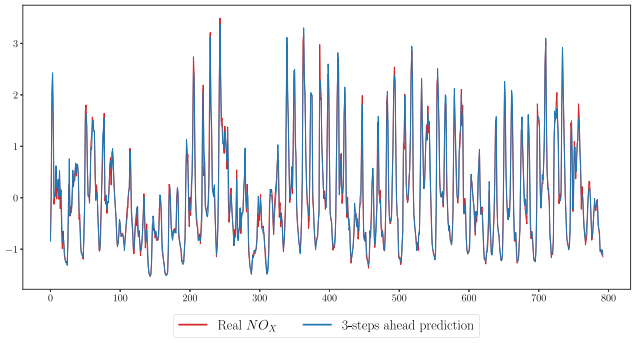
Fig. C.21. 1, 2 and 3 steps ahead predictions of the forecast model obtained with MOEFS-ELRF on test dataset for the Lorca air quality problem.



(a) 1-step ahead



(b) 2-steps ahead



(c) 3-steps ahead

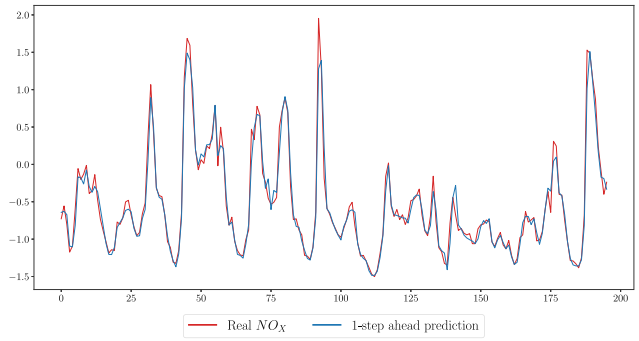
Fig. C.22. 1, 2 and 3 steps ahead predictions of the forecast model obtained with MOEFS-ELRF on training dataset for the Italian city air quality problem.

Moreover, MOEFS-ELRF incorporates a novel feature importance measure based on the frequency of FS across the non-dominated solutions of the Pareto front. The reliability of this feature importance measure has been validated through comparisons with well-known model-agnostic and post-hoc methods, specifically correlation analysis and permutation feature importance. The results confirm the consistency and interpretability of the proposed approach in identifying relevant and non-redundant features.

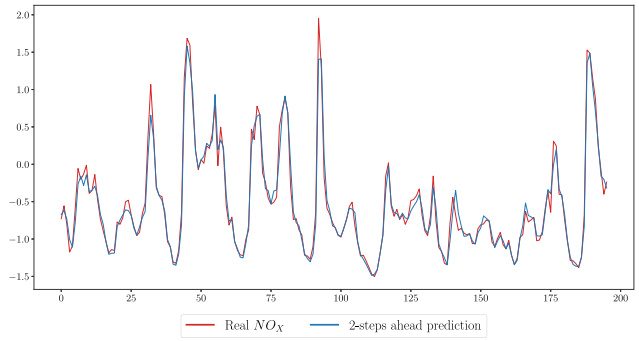
Extensive experiments have been carried out using real-world datasets, including air quality forecasting in southeastern Spain and Italy and an oil temperature prediction task, as well as synthetic TSF datasets designed to systematically vary complexity factors such as the number of features, noise, seasonality, and trend. The method has been rigorously compared against conventional RF, a wrapper-based FS method using MOEA and RF, and several state-of-the-art embedded FS techniques and TSF models, including LSTM, CancelOut, AFS, m-AFS, and EAR-FS.

The key findings can be summarized as follows:

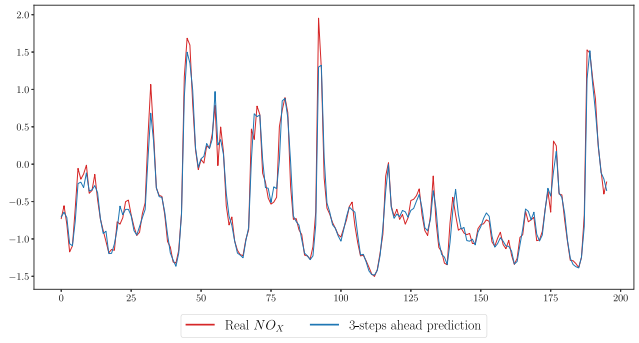
- **Superior predictive performance:** MOEFS-ELRF consistently achieved higher forecasting accuracy in both training and test sets, with statistically significant differences confirmed through Diebold–Mariano tests and win-loss ranking analysis.
- **Effective and parsimonious feature selection:** The method selected smaller and more informative subsets of features while maintaining or improving predictive performance, contributing to better generalization and model interpretability.
- **Lower overfitting ratios:** The method consistently exhibited better generalization ability, as evidenced by lower overfitting ratios compared to the other methods analyzed.
- **Scalability assessment:** The method's scalability was analyzed both with respect to feature dimensionality and forecasting horizon. Results showed that while the computational cost increases as expected with the number of features and the forecasting window size, the method maintained robust predictive performance and model compactness.
- **Robustness across configurations:** Additional experiments demonstrated that MOEFS-ELRF performs robustly across different MOEA configurations (NSGA-II, NSGA-III, MOEA/D), allowing flexibility to adjust the number of objectives and partitions without compromising model performance.



(a) 1-step ahead

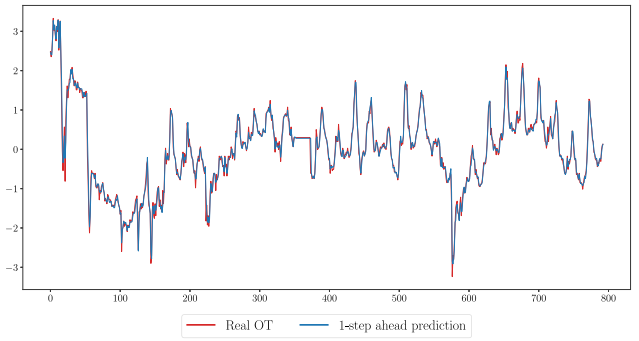


(b) 2-steps ahead

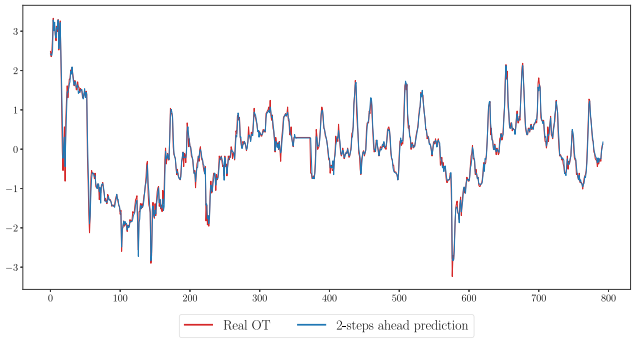


(c) 3-steps ahead

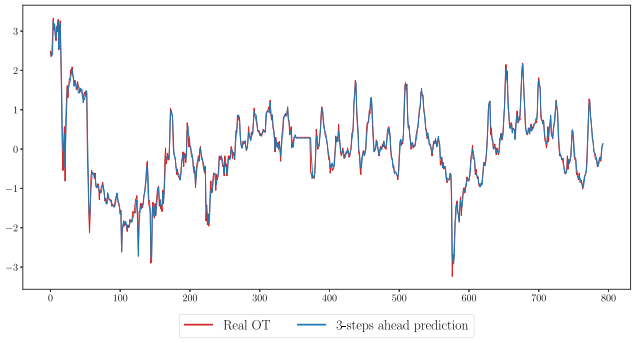
Fig. C.23. 1, 2 and 3 steps ahead predictions of the forecast model obtained with MOEFS-ELRF on test dataset for the Italian city air quality problem.



(a) 1-step ahead



(b) 2-steps ahead

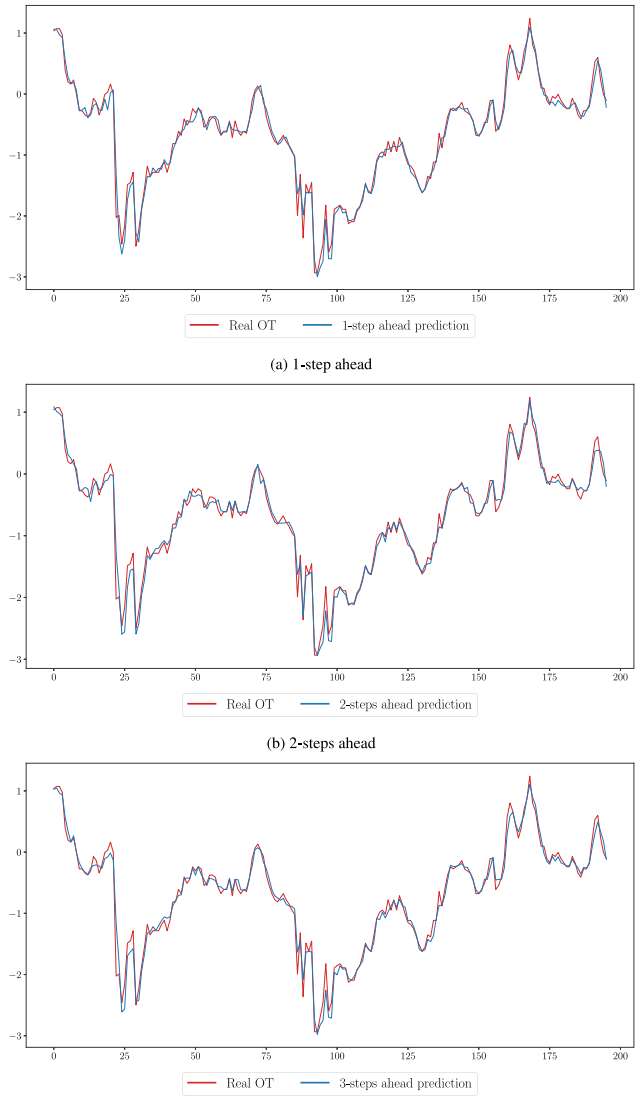


(c) 3-steps ahead

Fig. C.24. 1, 2 and 3 steps ahead predictions of the forecast model obtained with MOEFS-ELRF on training dataset for the ETT problem.

These findings highlight the overall effectiveness, interpretability, and robustness of MOEFS-ELRF as a powerful tool for FS and forecasting in complex TSF tasks. Despite these strengths, it is important to acknowledge the main limitation of the method: the increased computational time associated with the evolutionary search process. This trade-off is typical in metaheuristic-based approaches. While our implementation already leverages parallelization to mitigate these costs, further improvements could be explored through more advanced distributed computing strategies or optimized parallel architectures. However, it is important to note that MOEFS-ELRF is designed for offline learning scenarios. The evolutionary optimization and ensemble model construction steps are computationally intensive and are not suited for real-time training or continuous model updates. Thus, for real-time applications, the method must be used within a batch retraining framework, where periodic model updates can be performed during non-critical periods, and the deployed ensemble can provide fast predictions during operation. Addressing these limitations in future work, such as by developing more lightweight variants or integrating incremental learning strategies, represents a promising direction to further enhance the applicability of the approach. Besides the computational cost discussed, several other practical considerations regarding the application of MOEFS-ELRF should be acknowledged. While the method demonstrated robustness across different types of datasets, it could be sensitive to extreme levels of noise, strong seasonality, or pronounced non-stationarity if not properly preprocessed. This sensitivity was observed in our synthetic datasets SD1 to SD5, which were designed with increasing levels of complexity and noise. Specifically, the average test RMSE increased from 0.0676 (SD1, minimal noise) to 0.18784 (SD5, maximum noise), and the average number of selected features rose from 7.46 to 10.42. These results highlight that higher noise levels challenge the feature selection process and prediction accuracy, motivating careful preprocessing in practical deployments.

Several additional directions can be explored to further enhance and expand the proposed methodology. One promising avenue is the adaptation of our approach to transformer neural networks instead of RF. Transformers have recently gained prominence due to their outstanding performance in both natural language processing and time series forecasting. Their ability to handle long-range dependencies through self-attention mechanisms and parallelization makes them an attractive framework for integrating our feature selection method. Another important aspect for future research is evaluating the scalability



(a) 1-step ahead

(b) 2-steps ahead

(c) 3-steps ahead

Fig. C.25. 1, 2 and 3 steps ahead predictions of the forecast model obtained with MOEFS-ELRF on test dataset for the ETT problem.

and adaptability of MOEFS-ELRF in different domains and datasets. While this study has primarily focused on air quality forecasting and oil temperature prediction, applying the method to broader real-world scenarios could provide further insights into its versatility and robustness. Additionally, analyzing the impact of key data characteristics, such as seasonality and long-term dependencies, could shed light on how these properties influence the performance of our approach [57,58]. Further refinement of the multi-objective evolutionary algorithm is also an area worth exploring. Investigating alternative optimization techniques, extending validation to other families of MOEAs (e.g., indicator-based or evolutionary many-objective approaches like IBEA, SPEA2, or others), tuning algorithmic parameters, and experimenting with different partitioning strategies could lead to a better understanding of how data segmentation affects feature selection and forecasting accuracy. Another important future direction involves the joint optimization of FS and model hyperparameters within the MOEFS-ELRF framework. While our current approach focuses on feature subset selection with fixed RF parameters, model hyperparameters and FS are often interrelated, and optimizing them simultaneously could enhance predictive performance. We plan to explore this extension by adapting methodologies such as those proposed in [41], which integrate FS with hyperparameter tuning in a unified optimization framework. Moreover, extending the multi-objective formulation to include additional objective functions that explicitly consider intrinsic temporal characteristics could further enhance the model's effectiveness. Another promising direction involves addressing the concept drift problem by incorporating dynamic EL [59] into our framework. Additionally, exploring alternative ensemble combination strategies beyond stacking represents an important avenue for future work. While stacking was selected in this study for its simplicity, interpretability, and model-agnostic nature – making it well-suited to integrate the diverse non-dominated models from our MOEA – other combination methods such as attention-based fusion, error correction layers, or adaptive weighting mechanisms could further improve forecasting accuracy. Investigating these advanced ensembling techniques could lead to even more flexible and effective integration of the multiple RF models generated by the evolutionary feature selection process, especially in scenarios with complex temporal dynamics. Adapting ensembles dynamically based on changing data distributions could improve the robustness of the forecasting model in real-world applications where data characteristics evolve over time. Additionally, recent advances in *graph neural networks* (GNN) have introduced new possibilities for time series forecasting [60–62]. These models explicitly capture inter-variable and inter-temporal relationships, providing a structured way to model dependencies in complex datasets. We are particularly interested in extending the embedded feature selection mechanism proposed in this paper to GNN-based architectures, leveraging their ability to represent structured temporal dependencies. Finally, the offline training requirement of MOEFS-ELRF implies non-negligible retraining costs that depend on dataset size and feature dimensionality. In our experiments, complete retraining times ranged from several minutes to a few hours, depending on the number of features and partitions. While this is acceptable for batch-mode updates (e.g., daily forecasts), it can become a barrier in applications that demand very frequent retraining. To address this limitation, future work will explore incremental and online learning strategies to update models without full retraining cycles, as well as surrogate-assisted evolutionary optimization methods to reduce evaluation costs [55].

In summary, the proposed multi-objective evolutionary FS method, combined with EL, has demonstrated strong performance in TSF. Extending this approach to transformers and graph neural networks, refining its optimization strategy, exploring alternative ensemble combination techniques beyond stacking, and exploring its applicability to diverse datasets represent exciting directions for future research, contributing to the advancement of intelligent forecasting techniques.

CRediT authorship contribution statement

Raquel Espinosa: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Gracia Sánchez:** Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Data curation, Conceptualization. **José Palma:** Validation, Resources, Investigation, Funding acquisition, Conceptualization. **Fernando Jiménez:** Writing – review & editing, Writing – original draft, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Code availability

The complete source code for the MOEFS-ELRF framework will be made publicly available on our GitHub repository (https://github.com/raquel-ef) upon formal acceptance and publication of this article.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Fernando Jimenez reports financial support was provided by Spain Ministry of Science and Innovation. Fernando Jimenez reports financial support was provided by State Agency of Research. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This paper is funded by the CALM-COVID19 project (Ref: PID2022-136306OB-I00), grant funded by Spanish Ministry of Science and Innovation and the Spanish Agency for Research.

Appendix A. Abbreviations

Abbreviation	Meaning
AFS	Attention-based Feature Selection
EAR-FS	External Attention-Based Feature Ranker for Large-Scale Feature Selection
EL	Ensemble Learning
ETT	Electricity Transformer Temperature
FR	Feature Ranking
FS	Feature Selection
GNN	Graph Neural Networks
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MLP	Multi-layer Perceptron
MOEA	Multi-Objective Evolutionary Algorithm
MOEA/D	Multi-Objective Evolutionary Algorithm based on Decomposition
MOEFS-ELRF	Multi-Objective Evolutionary Feature Selection for Ensemble Learning with Random Forest
NSGA-II	Non-dominated Sorting Genetic Algorithm II
NSGA-III	Reference-point based many-objective NSGA-II
RF	Random Forest
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
TSF	Time Series Forecasting
UCI	University of California Irvine

Appendix B. Correlation matrix and permutation feature importance

See Figs. B.14–B.19.

Appendix C. Predictions of forecasting models

See Figs. C.20–C.25.

Data availability

Data will be made available on request.

References

[1] O. Valenzuela, F. Rojas, L. Herrera, H. Pomares, I. Rojas (Eds.), Theory and Applications of Time Series Analysis and Forecasting, Springer, Cham, 2023.
[2] K. Myint, M. Khaing, Time series forecasting system for stock market data, in: 2023 IEEE Conference on Computer Applications, ICCA, 2023, pp. 56–61.
[3] S. Edalatpanah, F. Hassani, F. Smarandache, A. Sorourkhah, D. Pamucar, B. Cui, A hybrid time series forecasting method based on neutrosophic logic with applications in financial issues, Eng. Appl. Artif. Intell. 129 (2024) 107531.
[4] S. Dorais, Time series analysis in preventive intervention research: A step-by-step guide, J. Couns. Dev. 102 (2) (2024) 239–250.
[5] R. James, H. Leung, J. Leung, A. Prokhorov, Forecasting tail risk measures for financial time series: An extreme value approach with covariates, J. Empir. Financ. 71 (2023) 29–50.
[6] A. Arnaiz, L. Cristal, A. Fernandez, M. Gubaton, D. Tanael, C. Centeno, Optimizing inventory management and demand forecasting system using time series algorithm, World J. Adv. Res. Rev. 20 (2023) 021–027.
[7] K. Huang, S. Wu, B. Sun, C. Yang, W. Gui, Metric learning-based fault diagnosis and anomaly detection for industrial data with intraclass variance, IEEE Trans. Neural Netw. Learn. Syst. 35 (1) (2024) 547–558.
[8] F. Jiménez, J. Palma, G. Sánchez, D. Marín, F. Palacios, L. López, Feature selection based multivariate time series forecasting: An application to antibiotic resistance outbreaks prediction, Artif. Intell. Med. 104 (2020) 101818.
[9] T. Mizan, S. Taghipour, Medical resource allocation planning by integrating machine learning and optimization models, Artif. Intell. Med. 134 (2022) 102430.
[10] F. Yaprakdal, M.V. Arisoy, A multivariate time series analysis of electrical load forecasting based on a hybrid feature selection approach and explainable deep learning, Appl. Sci. 13 (23) (2023) 12946.
[11] P. Gabrielli, M. Wüthrich, S. Blume, G. Sansavini, Data-driven modeling for long-term electricity price forecasting, Energy 244 (2022) 123107.
[12] J. Koumar, K. Hynek, T. Cejka, Network traffic classification based on single flow time series analysis, 2023, arXiv:2307.13434.
[13] A. Alqatawna, B. Abu-Salih, N. Obeid, M. Almiani, Incorporating time-series forecasting techniques to predict logistics companies' staffing needs and order volume, Computation 11 (7) (2023) 141.
[14] P. Mung, S. Phyu, Time series weather data forecasting using deep learning, in: 2023 IEEE Conference on Computer Applications, ICCA, 2023, pp. 254–259.
[15] J. Derot, N. Sugiura, S. Kim, S. Kouketsu, Improved climate time series forecasts by machine learning and statistical models coupled with signature method: A case study with El Niño, Ecol. Inform. 79 (2024) 102437.
[16] X. Chen, H. Xia, M. Wu, Y. Hu, Z. Wang, Spatiotemporal hierarchical transmit neural network for regional-level air-quality prediction, Knowl.-Based Syst. 289 (2024) 111555.
[17] L. Vitali, C. Cuvelier, A. Piersanti, A. Monteiro, M. Adani, R. Amorati, A. Bartocha, A. D'ausilio, P. Durka, C. Gama, G. Giovannini, S. Janssen, T. Przybyla, M. Stortini, S. Vranckx, P. Thunis, A standardized methodology for the validation of air quality forecast applications (F-MQO): lessons learnt from its application across Europe, Geosci. Model. Dev. 16 (20) (2023) 6029–6047.
[18] R. López, M. Chaveinte, R. Alonso, J. Prieto, J. Corchado, Pollutant time series analysis for improving air-quality in smart cities, Int. J. Interact. Multimed. Artif. Intell. 8 (2023) 98.
[19] M. Michalková, I. Pobocikova, Time series analysis of fossil fuels consumption in Slovakia by Arima Model, Acta Mech. Autom. 17 (2023) 35–43.
[20] B. Butcher, B. Smith, Feature Engineering and Selection: A Practical Approach for Predictive Models, Taylor & Francis, 2020.
[21] M. Emmerich, A. Deutz, H. Wang, A.V. Kononova, B. Naujoks, K. Li, K. Miettinen, I. Yevseyeva (Eds.), Evolutionary Multi-Criterion Optimization - 12th International Conference, EMO 2023, Leiden, The Netherlands, March 20-24, 2023, Proceedings, in: Lecture Notes in Computer Science, vol. 13970, Springer, 2023.
[22] I. Mienye, X. Sun, A survey of ensemble learning: Concepts, algorithms, applications, and prospects, IEEE Access 10 (2022) 99129–99149.
[23] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[24] J. Prakash, N. Karthikeyan, Enhanced evolutionary feature selection and ensemble method for cardiovascular disease prediction, Interdiscip. Sci.: Comput. Life Sci. 13 (3) (2021) 389–412.
[25] N. García-Pedrajas, G. Cerruela-García, MABUSE: A margin optimization based feature subset selection algorithm using boosting principles, Knowl.-Based Syst. 253 (2022) 109529.
[26] Y. Hao, Y. Zhou, J. Gao, J. Wang, A novel air pollutant concentration prediction system based on decomposition-ensemble mode and multi-objective optimization for environmental system management, Systems 10 (5) (2022) 139.
[27] Y. Vivek, V. Ravi, P. Suganthan, P. Krishna, Parallel fractional dominance MOEAs for feature subset selection in big data, Swarm Evol. Comput. 91 (2024) 101687.
[28] X. Zhou, W. Yuan, Q. Gao, C. Yang, An efficient ensemble learning method based on multi-objective feature selection, Inform. Sci. 679 (2024) 121084.
[29] H. Zhou, L. Ma, X. Niu, Y. Xiang, J. Chen, Y. Su, J. Li, S. Lu, C. Chen, Q. Wu, A novel hybrid model combined with ensemble embedded feature selection method for estimating reference evapotranspiration in the North China Plain, Agricult. Water. Manag. 296 (2024) 108807.

[30] C. Fan, S. Nie, L. Xiao, L. Yi, G. Li, Short-term industrial load forecasting based on error correction and hybrid ensemble learning, Energy Build. 313 (2024) 114261.
[31] C. Fan, S. Nie, L. Xiao, L. Yi, Y. Wu, G. Li, A multi-stage ensemble model for power load forecasting based on decomposition, error factors, and multi-objective optimization algorithm, Int. J. Electr. Power Energy Syst. 155 (2024) 109620.
[32] J. Brownlee, Time series forecasting as supervised learning, 2020, URL https://machinelearningmastery.com/time-series-forecasting-supervised-learning/.
[33] K. Deb, A. Pratab, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[34] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints, IEEE Trans. Evol. Comput. 18 (4) (2014) 577–601.
[35] Q. Zhang, H. Lii, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.
[36] L. Eshelman, The CHC adaptive search algorithm : How to have safe search when engaging in nontraditional genetic recombination, Found. Genet. Algorithms 1 (1991) 265–283.
[37] L. Davis (Ed.), Handbook of Genetic Algorithms, Van Nostrand Reinhold, 1991.
[38] D. Hadka, Platypus - multiobjective optimization in Python, 2015, URL https://platypus.readthedocs.io/en/latest/.
[39] M. Han, Z. Chen, M. Li, H. Wu, X. Zhang, A survey of active and passive concept drift handling methods, Comput. Intell. 38 (4) (2022) 1492–1535.
[40] C. Fan, G. Li, L. Xiao, L. Yii, S. Nie, Short-term power load forecasting in city based on ISSA-BiTCN-LSTM, Cogn. Comput. 17 (39) (2025).
[41] C. Fan, L. Yang, L. Xiao, A step gravitational search algorithm for function optimization and STTM's synchronous feature selection-parameter optimization, Artif. Intell. Rev. 58 (2025) 179.
[42] S. Ben Taieb, G. Bontempi, A. Atiya, A. Sorjamaa, A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition, Expert Syst. Appl. 39 (8) (2012) 7067–7083.
[43] S. Taieb, A. Atiya, A bias and variance analysis for multistep-ahead time series forecasting, IEEE Trans. Neural Netw. Learn. Syst. 27 (1) (2016) 62–76.
[44] G. Bontempi, S. Taieb, Y. Borgne, Machine learning strategies for time series forecasting, in: eBISS, 2013, pp. 62–77.
[45] D. Dua, C. Graff, UCI machine learning repository, 2017, URL http://archive.ics.uci.edu/ml.
[46] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, in: The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference, Vol. 35, AAAI Press, 2021, pp. 11106–11115.
[47] F. Jiménez, G. Sánchez, J. García, G. Sciavicco, L. Miralles, Multi-objective evolutionary feature selection for online sales forecasting, Neurocomputing 234 (2017) 75–92.
[48] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[49] V. Borisov, J. Haug, G. Kasneci, CancelOut: A layer for feature selection in deep neural networks, in: I. Tetko, V. Kůrková, P. Karpov, F. Theis (Eds.), Artificial Neural Networks and Machine Learning – ICANN 2019: Deep Learning, Springer International Publishing, Cham, 2019, pp. 72–83.
[50] N. Gui, D. Ge, Z. Hu, AFS: An attention-based mechanism for supervised feature selection, 2019, arXiv:1902.11074.
[51] L. Cao, Y. Chen, Z. Zhang, N. Gui, A multiattention-based supervised feature selection method for multivariate time series, Comput. Intell. Neurosci. 2021 (2021) 6911192.
[52] Y. Xue, C. Zhang, F. Neri, M. Gabbouj, Y. Zhang, An external attention-based feature ranker for large-scale feature selection, Knowl.-Based Syst. 281 (2023) 111084.
[53] F. Pedregosa, G. Varoquaux, A. Gramfort., V. Michel., B. Thirion, O. Grise., M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[54] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, G. Varoquaux, API design for machine learning software: experiences from the scikit-learn project, in: ECML PKDD Workshop: Languages for Data Mining and Machine Learning, 2013, pp. 108–122.
[55] R. Espinosa, F. Jiménez, J. Palma, Surrogate-assisted multi-objective evolutionary feature selection of generation-based fixed evolution control for time series forecasting with LSTM networks, Swarm Evol. Comput. 88 (2024) 101587.
[56] A. Fisher, C. Rudin, F. Dominici, All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously, J. Mach. Res. Learn. 20 (2019) 177:1–177:81.
[57] N. Bigdeli, H.S. Lafmejani, Dynamic characterization and predictability analysis of wind speed and wind power time series in Spain wind farm, J. AI Data Min. 4 (1) (2016) 103–116.
[58] N. Bigdeli, M. Salehi Borujeni, K. Afshar, Time series analysis and short-term forecasting of solar irradiation, a new hybrid approach, Swarm Evol. Comput. 34 (2017) 75–88.
[59] V. Cerqueira, L. Torgo, M. Oliveira, B. Pfahringer, Dynamic and heterogeneous ensembles for time series forecasting, in: 2017 IEEE International Conference on Data Science and Advanced Analytics, DSAA, 2017, pp. 242–251.
[60] K. Bui, J. Cho, H. Yi, Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues, Appl. Intell. 52 (2021) 2763–2774.
[61] W. Jiang, J. Luo, Graph neural network for traffic forecasting: A survey, Expert Syst. Appl. 207 (2022) 117921.
[62] M. Jin, H. Koh, Q. Wen, D. Zambon, C. Alippi, G. Webb, I. King, S. Pan, A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection, 2023, arXiv:2307.03759.