A k-NN-Based Benchmark for Lag Selection in Time Series Forecasting

Carlos Fernández-Llebrez Acedo[1⋆[0009−0002−0280−7441]], Pablo Moreno Moreu[1[0009−0007−8099−5464]], Pablo Reina-Jiménez[1[0009−0008−8770−3767]], and José C. Riquelme[1,2[0000−0002−8243−2186]]

[1] Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, Av. Reina Mercedes S/N, 41012 Sevilla, Spain
{cfernandezllebrez,pmoreno7,prjimenez,riquelme}@us.es
[2] SCORE Lab, I3US Institute, Universidad de Sevilla, Sevilla, Spain

Abstract. The application of classical feature selection methods to time series forecasting creates a challenge in identifying which strategy effectively captures relevant dynamics in high-dimensional environments. This paper presents a modular framework for the systematic benchmarking and comparative analysis of feature selection strategies. By utilizing a k-Nearest Neighbors regressor as a diagnostic tool (chosen for its high sensitivity to feature space quality), the framework exposes the intrinsic strengths and weaknesses of various selection paradigms. A key contribution is the dynamic optimization of the k parameter and selection hyperparameters via Bayesian search for every candidate subset, ensuring a fair comparison where each method operates at a high potential. We validate the framework's discriminative power using a synthetic stress test with up to 100 noise variables with their respective lags and the ETTm1 and ETTm2 electricity benchmarks. Results on real datasets demonstrate that the framework effectively identifies relevant subsets, evaluating alternative paradigms to improve the baseline R^2 from -0.019 to 0.874 on ETTm1 and from 0.678 to 0.941 on ETTm2. Overall, the integrated selection mechanisms successfully identify relevant temporal patterns and improve computational efficiency.

Keywords: Feature Selection · Time Series Forecasting · k-Nearest Neighbors · Lag Selection · Benchmark Framework

1 Introduction

The rapid growth of sensor networks and IoT devices has led to the emergence of high-frequency, multivariate time series datasets. In these contexts, forecasting models face a vast number of lagged variables, many of which are redundant or encapsulate purely stochastic noise. This high-dimensional space triggers the "curse of dimensionality" [6], degrading the stability and computational efficiency of predictive algorithms. While many feature selection (FS) methods exist, their

⋆ Corresponding author.

performance is often masked by the internal regularization of complex models [17], hindering an objective comparison of their intrinsic ability to distill relevant information from temporal noise.

In this paper, we propose a modular framework for the systematic benchmarking and comparative analysis of FS strategies in time series forecasting. The approach is built on the premise that robust evaluation requires a sensitive diagnostic tool, thus, we apply the k-Nearest Neighbors (k-NN) algorithm [5] as an arbiter of feature quality. Because k-NN performance depends on the local geometry of the feature space, without internal weight optimization or structural pruning, it serves as an ideal baseline to expose the true effectiveness of various selection paradigms. In this setup, k-NN is not the final forecasting model, but a "stress test" for the selector: if a FS method fails to identify core non-linear dependencies or filter redundant lags, the performance drop is immediate and measurable.

The main contribution of this work is a diagnostic framework for assessing the intrinsic quality of feature selection strategies in time series forecasting. By using k-NN as a deliberately sensitive diagnostic regressor, the approach exposes whether a selected subset preserves the local structure required for forecasting, preventing complex models from masking poor feature selection through internal regularization or representation learning. This makes the framework particularly useful for analysing temporal redundancy and noise filtering prior to modeling. The joint optimization of the selector configuration and the k-NN hyperparameter ensures that each strategy is evaluated under favourable conditions, enabling a fair comparison across paradigms. Experiments on synthetic stress tests and ETT benchmarks show that the framework separates informative temporal lags from irrelevant variables, yielding subsets that stabilize performance under extreme noise and improve accuracy in industrial scenarios.

The remainder of this paper is structured as follows. Section 2 reviews related work on feature selection and time series forecasting frameworks. Section 3 presents the proposed diagnostic framework. Section 4 describes the experimental setup, including the synthetic stress tests and real-world benchmarks. Section 5 discusses the empirical results, and Section 6 concludes the paper and outlines future research directions.

2 Related Work

FS in machine learning is traditionally categorized into filters, wrappers, and embedded methods [6,9]. Filter methods, such as Pearson correlation or Mutual Information (MI) [10], evaluate the statistical relevance of features independently of the forecasting model. While computationally efficient, linear filters often fail to capture the complex, non-linear dynamics inherent in physical systems, a limitation compounded by the "temporal redundancy" of lagged variables. Consequently, high-dimensional benchmarks such as ETT [16] demand advanced selectors capable of isolating critical "needles" of information within a "haystack" of irrelevant temporal lags.

Several Python frameworks have been proposed to facilitate feature engineering, forecasting, and machine learning with time series. A prominent example is tsfresh [4], which automatically extracts a collection of statistical, spectral, and time-domain features from time series and applies statistical tests to identify relevant predictors for regression or classification tasks. Tsfresh is effective for automated feature engineering and for transforming raw temporal observations into tabular representations. However, its main goal is large-scale feature extraction, rather than the controlled comparison of alternative feature selection strategies over explicit lagged spaces in forecasting problems.

Other ecosystems such as sktime [11], Darts [8], and Kats [12] provide unified interfaces for time series forecasting and analytics. However, these general-purpose libraries focus on the construction and evaluation of forecasting pipelines, while the quality of the lagged feature space is usually treated as a preprocessing decision or assessed indirectly through final model performance. In contrast, our framework makes feature selection the central object of analysis, evaluating whether different strategies can isolate informative temporal lags from high-dimensional spaces affected by redundancy and noise.

Existing benchmarking studies often rely on complex architectures, such as LSTMs or Transformers, to assess the impact of FS. However, their internal mechanisms, including gating structures, attention heads, and representation learning, can partially compensate for poor feature selection [15,17]. Similarly, embedded methods such as LASSO [14] or Random Forest [3] are effective but tied to their own structural biases, such as linearity or hierarchical partitioning. Our work addresses this gap by using k-NN as a neutral, non-parametric diagnostic arbiter, providing a clearer view of the intrinsic value and information density of the selected features.

3 The Proposed Framework

The proposed framework is an automated, end-to-end pipeline tailored for high-dimensional temporal spaces. Tests like Granger causality and discrete-entropy methods (e.g., Information Gain) are excluded: the former is not naturally framed as a lag-wise tabular ranker, and the latter would require discretization of continuous series. MI is used instead to capture non-linear dependencies in continuous domains. The entire modular architecture operates through three interconnected stages, detailed in Figure 1: dynamic lag transformation, parameterized feature selection, and distance-based diagnostic validation.

3.1 Dynamic Lag Transformation

Given a multivariate time series $X \in \mathbb{R}^{T \times D}$, where T is the number of timestamps and D the number of variables, the framework generates a lagged matrix L. For a specified past history ph, each variable is expanded into a set of features $\{x_{d,t}, x_{d,t-1}, \ldots, x_{d,t-ph}\}$. This process yields a high-dimensional space where feature selection becomes a mandatory step for effective modeling.

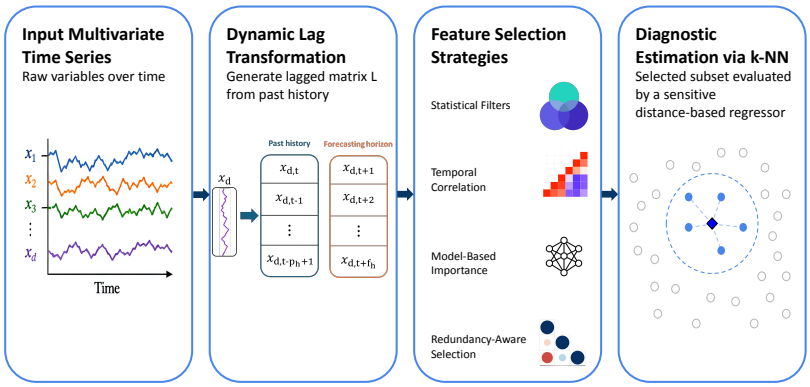


Fig. 1. Proposed Framework Architecture Pipeline.

3.2 Feature Selection Strategies

The framework implements a diverse array of selection strategies to handle different data natures:

- **Statistical Filters:** Using MI [10] to capture non-linear dependencies without assuming a specific functional form.
- **Temporal Correlation:** Cross-Correlation Function (CCF) analysis [2] to identify lead-lag relationships between predictors and the target.
- **Model-Based Importance:** Using Random Forest (RF) ensembles [3] to rank features based on mean impurity reduction, providing robustness against outliers.
- **Redundancy-Aware Selection:** Implementation of Correlation-based Feature Selection (CFS) [7] to prioritize subsets that are highly correlated with the target while penalizing inter-feature collinearity.

While these profiles represent the core architectural pillars, the performance and behavior of the complete taxonomy of implemented selectors are evaluated in Section 5.

3.3 Diagnostic Estimation via *k*-NN as an Arbiter Model

Once an optimal feature subset of cardinality n is isolated, the framework routes it to a non-parametric k-NN regressor [5] forecasting the target horizon fh, where k-NN acts as a diagnostic "arbiter" rather than a flexible forecasting engine. Because k-NN relies entirely on the localized geometry of the input space, lacking internal regularization or structural splits to bypass uninformative vectors, its distance metric collapses in high-dimensional noisy spaces. Consequently, any significant surge in k-NN's predictive power when transitioning from the full unpruned space to a restricted subset serves as a transparent "thermometer" of the selector's quality. Subsets that perform well under its strict locality constraints tend to remain informative when used with more flexible architectures.

3.4 Hybrid Hyperparameter Optimization

To ensure a rigorous comparative standard, the framework deploys a hybrid optimization strategy. For linear embedded models (*LASSO* [14] and *Elastic Net*), cross-validated variants (*LassoCV* / *ElasticNetCV*) trace optimal regularization paths (α) via coordinate descent, while for non-linear selectors, an Optuna [1] engine is integrated. Crucially, to prevent temporal data leakage and look-ahead bias, a decoupled two-stage pipeline is enforced: first, selector-specific hyperparameters are tuned using an internal `TimeSeriesSplit` with a baseline k-NN evaluator ($k = 5$) to isolate a definitive, invariant feature subset on the training partition; second, Optuna dynamically optimizes the diagnostic regressor's neighborhood parameter ($k \in [3, 50]$) over a subsequent 3-fold cross-validation split prior to final testing.

4 Experimental Setup

To construct the final feature matrix, a standard sliding window is applied to the multivariate time series. Formally, for a dataset with V exogenous variables and a single target y, a look-back window of ph historical lags generates a total feature space dimension $\mathcal{D}$ calculated as:

$$\mathcal{D} = ph \cdot (V + 1) + V \tag{1}$$

where $ph \cdot (V + 1)$ accounts for the historical lags of all variables (predictors and target), and the trailing V represents the exogenous variables at the current time step t. The current target y_t is excluded to prevent data leakage. To evaluate the discriminative power, architectural sensitivity, and structural robustness of the proposed diagnostic framework, we conducted systematic experiments across three complementary validation scenarios:

1. **Multi-Model Architectural Comparison:** This scenario establishes the baseline behavior of diverse predictive architectures, specifically k-NN, Histogram Gradient Boosting (HGB), and Multi-Layer Perceptrons (MLP), under various feature selection paradigms. To ensure full experimental control, this test utilizes the non-linear synthetic environment defined below, configured with a standard temporal window of 12 historical lags ($ph = 12$) and an immediate forecasting horizon ($fh = 0$), yielding a feature space of 1,442 total variables. By assessing how different models react to identical pruned subsets, this comparison empirically validates the strategic choice of k-NN as a highly sensitive, non-parametric diagnostic *arbiter* rather than a primary forecasting engine.
2. **Non-linear Synthetic Stress Test:** Extending the previous synthetic environment, this scenario evaluates the framework's capability to isolate non-linear dependencies under extreme dimensional expansion and temporal redundancy. Ten base variables are sampled from a uniform distribution $\mathcal{U}(-1, 1)$, defining the target as:

$$y_t = x_{1,t}^2 + \sin(2\pi x_{2,t}) + (x_{3,t} \cdot x_{4,t}) + \epsilon_t \tag{2}$$

where $\epsilon_t \sim \mathcal{N}(0, 0.1)$ represents Gaussian noise. To construct the search space, 100 uninformative uniform noise variables are appended. Crucially, to aggressively stress the selectors, the temporal sliding window is expanded to 50 historical lags ($ph = 50$) with a multi-step forecasting horizon of 3 steps ($fh = 3$), forcing the algorithms to discern the true non-linear drivers within a highly redundant 5,660-dimensional space.

3. **Real-World Benchmark (ETT):** To demonstrate practical generalizability, the framework is deployed on the standard Electricity Transformer Temperature datasets (ETTm1, ETTm2). These physical system series feature high-frequency (15-minute) sampling, complex operational seasonalities, and heavy temporal confounding, providing a rigorous environment for identifying informative lags.

Subset cardinality was fixed per scenario: $n = 6$ for the synthetic stress tests and $n = 5$ for the ETT benchmark (serving as an aggressive stress test). In all experiments, the data were split chronologically into an 80% train/validation partition and a 20% hold-out test partition. To ensure equity and prevent structural penalization across selectors, a 3-split *TimeSeriesSplit* co-optimizes the evaluator's hyperparameters alongside each selection method's internal parameters on the training partition, conducting the final evaluation exclusively on the test set. The framework is implemented as a modular Python library on top of Scikit-learn [13] and Optuna [1], with custom CFS modules; source code and configurations are publicly available[3].

5 Results and Discussion

This section reports the experimental findings across the three scenarios described in Section 4, followed by an ablation study on the contribution of each optimization component.

5.1 Multi-Model Architectural Comparison

Table 1 reports the experimental results across all selection methods and architectures. These results yield key methodological insights regarding architectural behavior under targeted feature compression:

1. **Architectural Sensitivity to Compressed Spaces:** The Histogram Gradient Boosting (HGB) model consistently achieves the best absolute performance, slightly outperforming the unpruned baseline under MI ($R^2 = 0.7995$). Noticeably, several selection strategies yield identical performance metrics across seeds when paired with HGB or k-NN. This is mathematically expected: the lack of row/column sub-sampling at this dataset scale makes the HGB tree-building process fully deterministic, while shared regularization principles in Lasso/Elastic Net and the absence of collinearity in

[3] `https://github.com/CFLlebrez/knn-lag-selection-benchmark`

Table 1. Comparative multimodal performance for each selection method across 3 random seed executions. The baseline configuration considers the full set of 1,442 features, derived from $V = 110$ exogenous variables and $ph = 12$ lags according to Eq. 1. Feature selection methods optimize the space to $n = 6$ variables.

Model	FS Method	R^2	RMSE	MAE
HGB	Baseline (No Selection)	0.7812	0.3912	0.3060
k-NN	Baseline (No Selection)	-0.0478 ± 0.0083	0.8560 ± 0.0034	0.7162 ± 0.0053
MLP	Baseline (No Selection)	-0.7628	1.1102	0.9006
HGB	MI	0.7995	0.3744	0.2929
k-NN	MI	0.1508 ± 0.0021	0.7706 ± 0.0010	0.6292 ± 0.0002
MLP	MI	0.5005 ± 0.0800	0.5898 ± 0.0471	0.4555 ± 0.0292
HGB	Random Forest	0.7955 ± 0.0043	0.3781 ± 0.0040	0.2875 ± 0.0049
k-NN	Random Forest	0.1810 ± 0.0400	0.7566 ± 0.0186	0.6178 ± 0.0231
MLP	Random Forest	0.4642 ± 0.0902	0.6106 ± 0.0521	0.4616 ± 0.0563
HGB	CCF	0.7071	0.4526	0.3602
k-NN	CCF	0.0738 ± 0.0022	0.8048 ± 0.0010	0.6653 ± 0.0020
MLP	CCF	0.3560 ± 0.0678	0.6704 ± 0.0357	0.5218 ± 0.0245
HGB	Elastic Net	0.7029	0.4558	0.3567
k-NN	Elastic Net	0.0540 ± 0.0045	0.8133 ± 0.0019	0.6687 ± 0.0029
MLP	Elastic Net	0.3307 ± 0.0278	0.6840 ± 0.0142	0.5422 ± 0.0025
HGB	Lasso	0.7029	0.4558	0.3567
k-NN	Lasso	0.0529 ± 0.0039	0.8138 ± 0.0017	0.6699 ± 0.0026
MLP	Lasso	0.3307 ± 0.0278	0.6840 ± 0.0142	0.5422 ± 0.0025
HGB	CFS Weka	0.6814	0.4720	0.3710
k-NN	CFS Weka	0.0932 ± 0.0046	0.7963 ± 0.0020	0.6482 ± 0.0028
MLP	CFS Weka	0.2901 ± 0.0389	0.7044 ± 0.0195	0.5563 ± 0.0199
HGB	Pearson	0.6814	0.4720	0.3710
k-NN	Pearson	0.0932 ± 0.0046	0.7963 ± 0.0020	0.6482 ± 0.0028
MLP	Pearson	0.3631 ± 0.1191	0.6655 ± 0.0610	0.5318 ± 0.0544
HGB	Sequential	0.6561	0.4904	0.3869
k-NN	Sequential	0.0259 ± 0.0047	0.8253 ± 0.0020	0.6785 ± 0.0007
MLP	Sequential	0.1945 ± 0.1161	0.7492 ± 0.0539	0.6034 ± 0.0408

Pearson/CFS Weka lead them to isolate identical feature subsets. In contrast, the MLP exhibits minor non-deterministic landscape variations ($\mathcal{O}(10^{-10})$) and degrades noticeably under linear selection strategies.

2. **Validation of k-NN as an Arbiter Model:** The k-NN regressor exposes the dataset's intrinsic complexity for distance-based metrics, with negative R^2 values in the full space ($R^2 = -0.0478$). This vulnerability to ambient noise, recurrent across subsequent tests, supports its role as arbiter: any subset that geometrically stabilizes k-NN signals selector robustness.

5.2 Synthetic Stress Test under Strong Noise

To evaluate the feature selection paradigms under extreme conditions, the framework was deployed on the non-linear synthetic environment of Section 4: the target manifold of Eq. 2 obscured by 100 uniform noise variables and expanded into

a redundant 5,660-dimensional space, devoid of standard autoregressive structures. In this scenario, the proposed k-NN diagnostic *arbiter* is subjected to a geometric saturation test. Table 2 summarizes the empirical performance metrics obtained across all feature selection methods over 3 independent runs. This regime is adversarial by design: no method is expected to reach positive $\mathbf{R^2}$; the experiment measures relative selector ordering, not absolute recovery.

Table 2. Synthetic Stress Test Performance. The unpruned baseline encompasses 5,660 features generated via Eq. 1 ($V = 110$ exogenous variables, $ph = 50$ temporal lags). All selection methods enforce a target compression scale of $n = 6$.

Method	N_{feat}	R^2 $(\mu \pm \sigma)$	**RMSE** $(\mu \pm \sigma)$	**MAE** $(\mu \pm \sigma)$	T_{sel} **(s)**
No Selection	5660	-0.0379 ± 0.0019	0.8578 ± 0.0008	0.7129 ± 0.0006	-
CCF	6	-0.0274 ± 0.0121	0.8534 ± 0.0050	0.7083 ± 0.0052	37.08
CFS Weka	6	-0.0305 ± 0.0082	0.8547 ± 0.0034	0.7094 ± 0.0040	52.54
Elastic Net	6	-0.0252 ± 0.0143	0.8525 ± 0.0060	0.7066 ± 0.0070	211.60
Lasso	6	-0.0247 ± 0.0151	0.8523 ± 0.0063	0.7063 ± 0.0076	34.61
MI	**6**	$\mathbf{-0.0121 \pm 0.0311}$	$\mathbf{0.8469 \pm 0.0130}$	$\mathbf{0.7038 \pm 0.0111}$	**20.76**
Pearson	6	-0.0236 ± 0.0162	0.8518 ± 0.0068	0.7062 ± 0.0075	1.87
Random Forest	6	-0.0191 ± 0.0212	0.8499 ± 0.0089	0.7053 ± 0.0084	115.49
Sequential	6	-0.0334 ± 0.0068	0.8559 ± 0.0028	0.7110 ± 0.0031	438.86

The experimental results yield two critical insights regarding the behavior of selectors under unstructured temporal noise:

1. **Geometric Collapse as a Diagnostic Indicator:** As evidenced in Table 2, the **No Selection** baseline highlights the impact of the curse of dimensionality on distance-based estimators, where feeding all 5660 unpruned features into the k-NN regressor completely dissolves the local neighborhood structure ($R^2 = -0.0379 \pm 0.0019$). Crucially, while the extreme noise-to-signal ratio drives the absolute R^2 metrics into negative values, every paradigm achieves a measurable relative improvement over the baseline while compressing the feature space by a factor of nearly $1 : 1000$ (enforcing $N_{feat} = 6$). Without complex internal representations or parameter architectures to mask poor inputs, the diagnostic arbiter reflects the quality of the pruned subspaces, validating its sensitivity under orthogonal, high-dimensional noise.
2. **Non-linear Discriminating Superiority:** Despite the hostile environment, differences emerge across the paradigms. The MI strategy achieves the least degradation ($R^2 = -0.0121 \pm 0.0311$) while maintaining a competitive computational footprint (20.76 s). This behavior confirms that information-theoretic criteria are inherently more resilient at detecting the non-monotonic, multi-variable interactions of the synthetic target than linear filters (Pearson) or sparse regularized linear models (Lasso and Elastic Net). Wrapper-based methods like Sequential exhibit a critical failure mode: they generate

the largest computational overhead (438.86 s) and the worst structural pruning performance ($R^2 = -0.0334 \pm 0.0068$), proving that greedy searches get easily trapped in local minima when surrounded by a dense lag distribution.

5.3 Benchmark Validation: ETTm1 and ETTm2

To validate the generalizability of the proposed framework in a real-world industrial setting, we utilized the Electricity Transformer Temperature (ETT) standard benchmarks. We evaluated the minutely-sampled variants (ETTm1 and ETTm2) configuring a forecast horizon of 24 steps and a look-back window of 96 historical lags ($ph = 96$). Since these physical systems track 6 exogenous predictors alongside the target oil temperature ($V = 6$), the structural formulation of Eq. 1 establishes a high-dimensional forecasting challenge consisting of exactly 678 potential temporal features per sample ($96 \times 7 + 6$). Table 3 presents the comparative performance metrics across all feature selection methods over 3 seeds, alongside the unpruned baseline configuration. The empirical evaluation on physical electrical infrastructure data provides three operational conclusions:

1. **Mitigation of Dimensional Noise:** The **No Selection** baseline results demonstrate a contrast in the intrinsic structural noise between both systems. In ETTm1, the model completely fails to yield predictive capabilities when unpruned due to high-dimensional saturation ($R^2 = -0.0192 \pm 0.0038$). Applying the framework to extract a subset of just 5 features triggers a generalization recovery, where CFS Weka isolates the optimal predictive manifold on ETTm1 ($R^2 = 0.8743 \pm 0.0000$), driving a 64.9% relative reduction in RMSE and a 70.6% reduction in MAE. Conversely, when evaluating the ETTm2 instance—which is more stable under unpruned conditions ($R^2 = 0.6777 \pm 0.0013$)—the Sequential wrapper manages to compress the feature space to 5 features that unlock an absolute gain of 0.26 in explanatory power, demonstrating that stable physical manifolds suffer heavily from temporal lag redundancy.
2. **Data-Driven Behavioral Divergence:** A key insight is that the optimal selector depends entirely on the dataset's latent dynamics. For ETTm1, fast statistical and information-theoretic filters dominate, with CFS Weka, CCF, and MI tightly grouping around $R^2 \sim 0.874$. In contrast, ETTm2 exhibits an inverse architectural preference where the wrapper-based Sequential search algorithm takes the leading spot ($R^2 = 0.9414 \pm 0.0005$), closely followed by sparse regularized models like Elastic Net (0.9201 ± 0.0001). This split reveals that while ETTm1 predictive features behave as distinct, univariable causal lags, ETTm2 contains multi-variable interactions that favor wrapper-based search over univariate correlation rankings.
3. **Computational Footprint and Latency Trade-offs:** Analyzing selection time (T_{sel}) reveals operational trade-offs for real-time edge deployment. Pearson proves to be a lightweight candidate, requiring roughly ~ 1.2 seconds to process the entire look-back horizon while delivering near-optimal

accuracy ($R^2 = 0.8697$ and 0.8930). In comparison, Random Forest introduces a computational bottleneck ($\sim$ 750 s) without providing any statistical edge. This validates the core utility of our diagnostic pipeline: it yields the structural insights necessary to skip computationally expensive black-box selectors in favor of lean, lightweight alternatives without compromising predictive accuracy.

Table 3. Benchmark Performance Results for ETTm1 and ETTm2 Datasets (Horizon=24, Lags=96).

Method	N_{feat}	R^2 ($\mu \pm \sigma$)	**RMSE** ($\mu \pm \sigma$)	**MAE** ($\mu \pm \sigma$)	T_{sel} **(s)**
ETTm1 Dataset (Real Industrial Benchmark)					
No Selection	678	-0.0192 ± 0.0038	3.4801 ± 0.0053	2.8556 ± 0.0032	-
CCF	5	0.8741 ± 0.0004	1.2228 ± 0.0019	0.8392 ± 0.0018	20.19
CFS Weka	5	$\mathbf{0.8743 \pm 0.0000}$	$\mathbf{1.2219 \pm 0.0002}$	$\mathbf{0.8385 \pm 0.0003}$	58.83
Elastic Net	5	0.8358 ± 0.0001	1.3966 ± 0.0006	1.0209 ± 0.0004	168.36
Lasso	5	0.8360 ± 0.0001	1.3961 ± 0.0004	1.0205 ± 0.0004	111.72
MI	5	0.8739 ± 0.0004	1.2242 ± 0.0019	0.8408 ± 0.0019	264.19
Pearson	5	0.8697 ± 0.0080	1.2436 ± 0.0376	0.8582 ± 0.0342	1.35
Random Forest	5	0.8711 ± 0.0056	1.2373 ± 0.0268	0.8528 ± 0.0249	792.87
Sequential	5	0.8624 ± 0.0016	1.2785 ± 0.0074	0.9049 ± 0.0062	120.76
ETTm2 Dataset (Real Industrial Benchmark)					
No Selection	678	0.6777 ± 0.0013	6.0652 ± 0.0125	4.6220 ± 0.0036	-
CCF	5	0.8911 ± 0.0018	3.5260 ± 0.0290	2.2230 ± 0.0193	18.21
CFS Weka	5	0.8936 ± 0.0018	3.4864 ± 0.0290	2.1959 ± 0.0182	51.47
Elastic Net	5	0.9201 ± 0.0001	3.0211 ± 0.0022	1.9556 ± 0.0012	141.93
Lasso	5	0.9199 ± 0.0002	3.0237 ± 0.0045	1.9569 ± 0.0023	109.81
MI	5	0.8922 ± 0.0025	3.5078 ± 0.0414	2.2103 ± 0.0272	206.16
Pearson	5	0.8930 ± 0.0024	3.4955 ± 0.0399	2.2014 ± 0.0254	1.13
Random Forest	5	0.8975 ± 0.0000	3.4211 ± 0.0000	2.1550 ± 0.0000	717.27
Sequential	5	$\mathbf{0.9414 \pm 0.0005}$	$\mathbf{2.5861 \pm 0.0114}$	$\mathbf{1.6894 \pm 0.0056}$	109.89

5.4 Optimization Impact Analysis

An ablation study was conducted to isolate the individual contributions of the framework's optimization components. The feature space was fixed using the subsets selected by the MI strategy, as a representative non-linear filter, given its consistent performance across both datasets without being tied to either dataset's optimal selector. Using this baseline subspace, we compared three configurations: *Vanilla* (default parameters, $k = 5$), Partial (Bayesian optimization restricted to the regressor's k), and the proposed Full Framework (joint tuning of both selector and regressor hyperparameters).

The results in Table 4 highlight a clear performance hierarchy. In the ETTm1 dataset, optimizing only the regressor (k) provides a moderate improvement, but

Table 4. Ablation study results: Comparing baseline (Vanilla), partial optimization (k-only), and proposed Hybrid Framework using the baseline subsets generated by MI.

Dataset	Strategy	$R^2 \uparrow$	RMSE $\downarrow$	MAE $\downarrow$	Best k	T_{opt} (s)
ETTm1	Vanilla	0.8491	1.3387	0.9402	5	-
	Partial (k-only)	0.8646	1.2682	0.8816	10	-
	Full Framework	**0.8742**	**1.2220**	**0.8386**	**37**	73.4
Synthetic	Vanilla	-0.0964	0.8815	0.7149	5	-
	Partial (k-only)	0.0138	0.8360	0.6943	31	-
	Full Framework	**0.0147**	**0.8356**	**0.6941**	**30**	11.1

the Full Framework yields the best results by shifting k to 37. This suggests that once the feature space is optimally pruned, the regressor can afford a higher degree of smoothing, leading to more robust generalization. In the Synthetic scenario, the "Vanilla" approach fails ($R^2 = -0.0964$) due to the high noise-to-signal ratio. While the Partial optimization manages to recover a positive predictive power, the Full Framework achieves the lowest error metrics. These findings confirm that the interaction between feature selection and hyperparameter tuning is non-trivial. The optimal k value depends strictly on the specific composition of the selected subset, justifying the need for a hybrid search space.

6 Conclusions and Future Work

We introduced a modular framework that uses a k-NN regressor as a diagnostic arbiter to expose the intrinsic ability of feature selectors to distill predictive signals from high-dimensional temporal noise. The experimental results lead to three primary conclusions. First, the hybrid optimization approach, which jointly tunes the selector's relevance threshold and the regressor's smoothing parameter, proved essential for stability. As shown in the ablation study, this synergy rescued predictive performance in extreme noise scenarios where standard "vanilla" configurations failed. Second, the framework demonstrated high discriminative power on industrial benchmarks (ETTm1 and ETTm2), achieving an R^2 of up to 0.874 on ETTm1 and 0.941 on ETTm2. Finally, no single selector dominates across datasets: filter-based methods lead on ETTm1, while wrapper-based and regularized linear methods perform best on ETTm2, highlighting the value of a benchmarking framework over committing to a single strategy.

Future work Future work will focus on expanding this validation. We aim to conduct a sensitivity analysis across varying subset cardinalities (n) to evaluate broader compression scales, integrate formal statistical tests to verify cross-paradigm performance margins, and stress the k-NN arbiter under non-stationary regimes and against state-of-the-art time series foundation models.

Acknowledgments. This research has been supported by grant PID2023-146037OB-C21 funded by MICIU/AEI/10.13039/501100011033. The authors would like to thank the Spanish Ministry of Universities for the FPU predoctoral fellowship (FPU23/03390).

References

1. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2623–2631 (2019). https://doi.org/10.1145/3292500.3330701
2. Box, G.E.P., Jenkins, G.M., Reinsel, G.C., Ljung, G.M.: Time Series Analysis: Forecasting and Control. John Wiley & Sons (2015)
3. Breiman, L.: Random forests. Machine Learning **45**(1), 5–32 (2001). https://doi.org/10.1023/A:1010933404324
4. Christ, M., Braun, N., Neuffer, J., Kempa-Liehr, A.W.: Time series feature extraction on basis of scalable hypothesis tests (tsfresh – a Python package). Neurocomputing **307**, 72–77 (2018). https://doi.org/10.1016/j.neucom.2018.03.067
5. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory **13**(1), 21–27 (1967). https://doi.org/10.1109/TIT.1967.1053964
6. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research **3**, 1157–1182 (2003). https://doi.org/10.1162/153244303322753616
7. Hall, M.A.: Correlation-based feature selection for discrete and numeric class machine learning. In: Proceedings of the Seventeenth International Conference on Machine Learning (ICML). pp. 359–366 (2000)
8. Herzen, J., Lässig, F., Piazzetta, S.G., Neuer, T., Tafti, L., Raille, G., Pottelbergh, V., et al.: Darts: User-friendly modern machine learning for time series. Journal of Machine Learning Research **23**(124), 1–6 (2022)
9. Jiménez-Navarro, M.J., Martínez-Ballesteros, M., Brito, I.S., Martínez-Álvarez, F., Asencio-Cortés, G.: Embedded feature selection for neural networks via learnable drop layer. Logic Journal of the IGPL **33**(5), jzae062 (12 2025). https://doi.org/10.1093/jigpal/jzae062
10. Kraskov, A., Stögbauer, H., Grassberger, P.: Estimating mutual information. Physical Review E **69**(6), 066138 (2004). https://doi.org/10.1103/PhysRevE.69.066138
11. Löning, M., Bagnall, A., Ganesh, S., Kazakov, V., Lines, J., Király, F.J.: sktime: A unified interface for machine learning with time series. In: Workshop on Systems for ML at NeurIPS (2019), https://arxiv.org/abs/1909.07872
12. Meta/Facebook Research: Kats: A generalizable framework to analyze time series data. https://facebookresearch.github.io/Kats/ (2021), accessed May 2026
13. Pedregosa, F., Varoquaux, G., Gramfort, A., et al.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
14. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological) **58**(1), 267–288 (1996). https://doi.org/10.1111/j.2517-6161.1996.tb02080.x
15. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 30 (2017)
16. Zhou, H.: ETDataset: Electricity transformer dataset. https://github.com/zhouhaoyi/ETDataset (2021), accessed May 2026
17. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.: Informer: Beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 11106–11115 (2021). https://doi.org/10.1609/aaai.v35i12.17325