Towards Energy-Efficient Buildings

Timothy Mulumba
Department of Civil & Urban Engineering
New York University

Abstract

Heating, Ventilation, and Air Conditioning (HVAC) systems account for a large share of building energy use. We study a *deployment-oriented* ARX→SVM (auto-regressive with exogenous variables to support vector machine) pipeline for chiller Fault Detection and Diagnosis (FDD) that *builds on* prior ARX–hybrid work benchmarked on the ASHRAE RP-1043 dataset and related online extensions.[1] Our contributions are: (i) a practitioner-focused procedure for tuning the exponential forgetting factor λ *on* F_1 to align estimator memory with detection objectives; (ii) a clean ablation contrasting dynamic ARX features with a static regression baseline at matched feature budgets; and (iii) a latency/memory audit showing sub-millisecond per-timestep updates on commodity hardware. On RP-1043, the pipeline attains competitive macro-F_1 against stronger baselines while preserving interpretability and runtime efficiency. We qualify claims with time-series-aware uncertainty estimates and paired significance tests, and discuss maintenance under concept drift. Finally, this work offers a clear pathway to mitigating greenhouse gas emissions by improving chiller operational efficiency and reducing energy waste.

1 Introduction

Modern buildings rely heavily on Heating, Ventilation, and Air Conditioning (HVAC) systems to maintain comfortable indoor conditions. These systems, especially in hot-humid climates, can account for a substantial fraction (30–40%) of a building's total energy usage, making them key contributors to global greenhouse gas (GHG) emissions [3]. Consequently, early detection and diagnosis of faults (e.g., malfunctioning valves, sensor drifts, fouled heat exchangers) can play a major role in energy savings, operational cost reductions, and overall climate change mitigation. Suboptimal operation of indoor climate control equipment has been estimated to waste around 15–30% of energy in US commercial buildings and can be even more critical in regions with extreme temperatures [4, 5].

This paper tackles the problem of fault detection and diagnosis (FDD) for the chiller subsystem of an HVAC unit. We develop a hybrid approach combining an Auto-Regressive with Exogenous variables (ARX) time series model and a Support Vector Machine (SVM) classification algorithm. We show that the ARX modeling phase helps capture the dynamic behavior of chiller power consumption under various operating conditions, while the SVM uses the ARX parameters as input features to classify potential fault states. Our experimental results on a recognized open dataset indicate improved FDD performance over baseline methods. In turn, early and accurate identification of chiller faults can prevent energy inefficiencies, reduce electricity demands, and thereby have a clear pathway to reducing associated GHG emissions.

1.1 Related Work in Fault Detection and Diagnosis

Surveys and taxonomies. Comprehensive reviews position FDD methods across model-based, data-driven, and hybrid families and catalog common datasets/metrics. For large-scale HVAC, [6]

[1] E.g., [1, 2].

UrbanAI: Harnessing Artificial Intelligence for Smart Cities Workshop at NeurIPS 2025.

synthesize data-driven approaches and evaluation practices. Within data-driven approaches, a variety of classification algorithms have been used for FDD, ranging from simple Naïve Bayes [7] to random forest [8] and deep neural networks [9]. A recent AI-focused survey [10] spans traditional ML, deep learning (DL), and hybrid AI for HVAC FDD, highlighting the increasing role of sequence models and the prevalence of RP-1043 in benchmarking.

ARX and hybrid chiller FDD. Black-box ARX modeling has long been used for HVAC FDD [11, 12]. For chillers specifically, [1] proposed the now-canonical ARX→SVM pipeline on RP-1043. Online extensions combine nonlinear state tracking with one-class SVM (EKF-ROSVM) for improved sensitivity at low severities [2]. Classical alternatives include hybrid SVM variants and feature-selection pipelines for chillers [13]. Our work builds on this lineage but targets *deployment clarity*: F_1-aligned memory selection for RLS, clean ablations against static regression, time-series-aware uncertainty, and explicit runtime accounting.

Sequence/DL baselines on RP-1043. Several sequence models have been reported on RP-1043, e.g., 1D-CNN+GRU [14], temporal convolutional networks [15], and causal 1D-CNNs [16]. These often achieve strong accuracy with greater compute and reduced transparency. We therefore include stronger classical baselines (e.g., logistic regression, XGBoost, one-class SVM) and discuss when lightweight ARX features remain preferable (limited labels, interpretability constraints, and small latency budgets).

2 Methodology

2.1 Overall Approach

Our proposed framework combines a time-series model (ARX) with a supervised classification algorithm (SVM). In essence, each chiller observation is transformed into an ARX parameter vector that encodes recent dynamic behavior. We then feed these parameter vectors into an SVM to detect and classify normal versus faulty operating conditions.

2.2 ARX Model Specification

We begin by selecting a target variable (instantaneous chiller power) and several exogenous variables (temperatures, flow rates, etc.), identified through domain knowledge and preliminary data exploration. Let $y(t)$ denote the target variable and $x_i(t)$ the exogenous inputs. We specify an ARX structure, which includes lagged values of both the target and exogenous variables:

$$y(t) = \sum_{j=1}^{p} \alpha_j \, y(t-j) + \sum_{k=1}^{m} \sum_{\ell=0}^{q_k} \beta_{k,\ell} \, x_k(t-\ell) + \varepsilon(t),$$

where p and q_k are the maximum lags, selected based on a combination of domain knowledge regarding typical chiller response times and preliminary analysis by observing autocorrelation and partial autocorrelation functions of the target variable and residuals. The α_j and $\beta_{k,\ell}$ constitute the model parameters. We estimate these parameters recursively using Recursive Least Squares (RLS) with exponential forgetting [17], because of its suitability for online adaptation to changing system dynamics, which is characteristic of fault evolution.

2.3 Feature Extraction and Classification

At each time step t, we extract the ARX parameter vector $\theta(t) = [\alpha_1, \ldots, \alpha_p, \beta_{1,0}, \ldots, \beta_{k,\ell}]$ as a feature vector. We then apply a standard C-Support Vector Classifier (SVC) [18] to learn a mapping from $\theta(t)$ to the fault status $z(t) \in \{0, \ldots, K\}$, where 0 indicates normal operation and $K > 0$ indicates a specific fault type. The SVM is trained using a one-vs-all scheme for multi-fault classification.

2.4 Alternative Classifiers and Baseline Regression

To evaluate the effectiveness of the SVM within our hybrid approach, we also tested:

- **Naïve Bayes (NB)** [7]: A probabilistic classifier assuming conditional independence among ARX parameters.
- **Random Forest (RF)** [8]: An ensemble decision-tree classifier, robust to overfitting.
- **XGBoost** [19]: State-of-the-art on tabular data.

Furthermore, to demonstrate the necessity of capturing temporal dynamics, we compare against a simpler *Regression Baseline* wherein we omit lagged terms and treat the problem as a stationary linear mapping from current exogenous variables to chiller power.

We evaluate each model via a stratified train/test split, ensuring temporal ordering is respected (i.e., training on an initial time block and testing on a subsequent time block). All hyperparameters (including the forgetting factor and SVM regularization) are tuned on a validation subset drawn from the training set. Further details on the methodology and hyperparameters are provided in A.

Uncertainty and significance testing. We report mean $\pm$ 95% confidence intervals (CIs) over $n = 10$ repeated *temporal* splits. CIs are computed with a stationary/block bootstrap to respect serial dependence [20]. For paired model comparisons on identical test instances we use McNemar's test [21]. Unless stated otherwise, $\alpha = 0.05$.

Concept drift and maintenance. RLS with exponential forgetting adapts short-term dynamics; residual drift can still degrade classification. In practice we recommend scheduled refresh (e.g., quarterly) or incremental SVM updates that preserve KKT conditions [22, 23]. A simple trigger is a rolling macro-F_1 or calibration drift beyond pre-set thresholds.

3 Results and Discussion

3.1 Experimental Setup and Dataset

We use the well-known ASHRAE Project RP-1043 chiller dataset [24], which logs measurements at a 2-minute interval and includes both normal and faulty conditions under multiple fault severities. The dataset provides about 432 samples per severity level of each fault, yielding several thousand data points overall. We reserve about 70% of the data for training (including validation) and use the remaining 30% for testing.

3.2 Comparison Across Methods

In Table 1, we summarize the F-measure (in %) for each classifier combined with the time-series ARX features ("ARX-based") as well as the simpler regression baseline ("Regression-only"). We report averaged results across different building aliases and fault types.

Table 1: Mean F-measure (%) for each classifier under two feature settings: (1) ARX-based (time-series) and (2) Regression-only. Higher is better. Standard deviations are shown in parentheses.

Classifier	**ARX-based**	**Regression-only**
Naïve Bayes	73.1 ($\pm$3.2)	58.4 ($\pm$4.1)
Logistic Regression	73.9 ($\pm$3.5)	59.8 ($\pm$3.1)
XGBoost	78.1 ($\pm$2.2)	68.4 ($\pm$3.3)
Random Forest	81.4 ($\pm$3.7)	66.9 ($\pm$3.5)
SVM (proposed)	84.0 ($\pm$3.1)	69.2 ($\pm$3.6)

Note. The 95% CIs for RF and SVM overlap; paired McNemar tests on per-sample predictions indicate no statistically significant difference at $\alpha = 0.05$. We therefore avoid claims of superiority and emphasize robustness and runtime.

The results in Table 1 clearly demonstrate the benefits of incorporating temporal dynamics through ARX modeling. For all classifiers tested, using ARX-derived parameters as features leads to substantially higher F-measures compared to the regression-only baseline that relies solely on current sensor readings. This confirms our hypothesis that the dynamic behavior encapsulated by the ARX parameters provides crucial discriminative information for fault detection.

Among the classifiers operating on the ARX features, our proposed ARX-SVM hybrid approach achieves the highest mean F-measure of 84.0%. This suggests that the SVM is particularly effective at finding complex decision boundaries in the space of ARX parameters to separate normal and faulty conditions. This could be attributed to SVM's ability to define maximal margin hyperplanes, which might be well-suited for the distribution of ARX parameters under different fault conditions.

Random Forest also shows comparable performance with ARX features (81.4%), outperforming Naïve Bayes significantly. The lower performance of Naïve Bayes (73.1% with ARX features) likely indicates that its strong assumption of conditional independence among the ARX parameters does not fully hold in this application, as these parameters are inherently coupled through the underlying system dynamics.

The performance drop when using regression-only features is stark across all classifiers, highlighting the limitations of static models for this dynamic FDD problem.

The computational cost of the ARX-SVM approach during online operation involves the RLS updates at each time step and the SVM prediction. RLS is computationally efficient, and SVM prediction is typically fast, making the approach suitable for real-time FDD. Training the final SVM model took about 20 minutes on the full training set, while online prediction for a new data point takes on the order of milliseconds.

3.3 ARX vs. EKF: fidelity vs. compute

Linear ARX tracked by RLS offers $O(d^2)$ updates and transparent coefficients; EKF introduces nonlinear state estimation that can improve class separability in some faults at the cost of tuning and compute. Prior EKF-ROSVM reports on RP-1043 show higher detection rates for incipient faults [2]. Our deployment focus (sub-millisecond updates, simple tuning via λ) motivates ARX in resource-constrained BMS.

3.4 Limitations

We evaluate on a single public dataset (RP-1043) and prioritize classic but interpretable models. Stronger DL/sequence baselines exist on RP-1043 [14, 15, 16]. Our aim is a reproducible, low-latency baseline with clear maintenance guidance rather than a state-of-the-art accuracy claim.

3.5 Climate Impact Considerations

From a climate change perspective, accurately diagnosing HVAC chiller faults can unlock operational energy savings by anticipating and rectifying faulty conditions before they escalate. This contributes to reducing the building's overall carbon footprint. Although we focus here on chillers, similar hybrid ARX-classifier strategies can be extended to other building subsystems (e.g., air handling units, fan coil units) for more comprehensive fault detection across entire HVAC systems. This work, therefore, has direct relevance to climate mitigation since HVAC systems occupy a large fraction of energy use in buildings worldwide. Implementing an automated FDD system based on our hybrid approach can help minimize unnecessary energy consumption, reduce operational costs, and cut greenhouse gas emissions.

4 Conclusion

We presented a hybrid time-series and machine learning approach for chiller Fault Detection and Diagnosis (FDD). By coupling an ARX model to extract dynamic parameters with an SVM (or other classifiers) for classification, our method achieves high accuracy in detecting and classifying faults. Future research could explore real-time implementation and extension to other subsystems or building types.

References

[1] Ke Yan, Wen Shen, Timothy Mulumba, and Afshin Afshari. Arx model based fault detection and diagnosis for chillers using support vector machines. *Energy and Buildings*, 81:287–295, 2014.

[2] Ke Yan, Zhiwei Ji, and Wen Shen. Online fault detection methods for chillers combining extended kalman filter and recursive one-class svm. *Neurocomputing*, 228:205–212, 2017.

[3] Kurt W Roth, Detlef Westphalen, Patricia Llana, and Michael Feng. The energy impact of faults in us commercial buildings. *Journal of Solar Energy Engineering, Transactions of the ASME, v 125, n 3, Emerging Trends in building Design, Diagnostics, and Operations*, 2004.

[4] Muhammad Tauha Ali, Marwan Mokhtar, Matteo Chiesa, and Peter Armstrong. A cooling change-point model of community-aggregate electrical load. *Energy and Buildings*, 43(1):28–37, 2011.

[5] Srinivas Katipamula and Michael R. Brambley. Methods for fault detection, diagnostics, and prognostics for building systems – a review, part i. *HVAC&R Research*, 11(1):3–25, 2005.

[6] Maryamsadat Mirnaghi and Fariborz Haghighat. Fault detection and diagnosis of large-scale hvac systems in buildings using data-driven methods: A comprehensive review. *Energy and Buildings*, 229:110492, 2020.

[7] I. Rish. An empirical study of the naive bayes classifier. *IJCAI 2001 workshop on empirical methods in artificial intelligence*, 3(22):41–46, 2001.

[8] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[9] Yuebin Yu, Denchai Woradechjumroen, and Daihong Yu. A review of fault detection and diagnosis methodologies on air-handling units. *Energy and Buildings*, 82:550–562, 2014.

[10] Jian Bi, Hua Wang, Enbo Yan, Chuan Wang, Ke Yan, Liangliang Jiang, and Bin Yang. Ai in hvac fault detection and diagnosis: A systematic review. *Energy Reviews*, 3(2):100071, 2024.

[11] Henk C. Peitsman and Vincent E. Bakker. Application of black-box models to hvac systems for fault detection. In *ASHRAE Transactions*, volume 102, pages 628–640, 1996.

[12] Timothy Mulumba, Afshin Afshari, Ke Yan, Wen Shen, and Leslie K Norford. Robust model-based fault diagnosis for air handling units. *Energy and Buildings*, 86:698–707, 2015.

[13] Hua Han, B. Gu, J. Kang, and Z. R. Li. Study on a hybrid svm model for chiller fdd applications. *Applied Thermal Engineering*, 31(4):582–592, 2011.

[14] Zhao Wang et al. A novel fault diagnosis approach for chillers based on 1-d convolutional neural network and gated recurrent unit. *Sensors*, 20(9):2458, 2020.

[15] Chao Li, Chun Shen, Hui Zhang, Hao Sun, and Shouqi Meng. A novel temporal convolutional network via enhancing feature extraction for the chiller fault diagnosis. *Journal of Building Engineering*, 42:103014, 2021.

[16] Pandarasamy Arjunan, Seshadhri Srinivasan, and Kameshwar Poolla. Causal 1-d convolutional neural networks based chiller fault detection and diagnosis. In *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '22)*, 2022.

[17] S. J. Pollard. Recursive parameter identification for estimating and displaying maneuvering vessel path. *Master Thesis*, 2003.

[18] C. C. Chang and C. j. Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[19] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[20] Dimitris N. Politis and Joseph P. Romano. The stationary bootstrap. *Journal of the American Statistical Association*, 89(428):1303–1313, 1994.

[21] Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

[22] Gert Cauwenberghs and Tomaso Poggio. Incremental and decremental support vector machine learning. *Advances in Neural Information Processing Systems (NIPS 2000)*, 13, 2001.

[23] Pavel Laskov, Christian Gehl, Stefan Krüger, and Klaus-Robert Müller. Incremental support vector learning: Analysis, implementation and applications. *Journal of Machine Learning Research*, 7:1909–1936, 2006.

[24] M. C. Comstock and J. E. Braun. Experimental data from fault detection and diagnostic studies on a centrifugal chiller. *Purdue University*, 1999.

[25] J.D. Cryer and K.S. Chan. *Time Series Analysis with Applications in* R. Springer, New York, 2006.

[26] D. Weitz. Counting independent sets up to the tree threshold. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 140–149, 2006.

[27] Jack Kiefer. Sequential minimax search for a maximum. *Proceedings of the American Mathematical Society*, 4(3):502–506, 1953.

A Implementation Details: Parameter Estimation and Optimization

For completeness, we briefly outline key implementation details (the RLS algorithm, forgetting factor tuning, and multi-class SVM formulation). We refer readers to [12, 17] for the mathematics of exponential forgetting and to [18] for the SVM implementation used in this study.

All code was written in Python, and experiments ran on a standard laptop. We performed a temporal split for training and testing to avoid any information leakage across time. Cross-validation was conducted on the training partition only, adjusting the forgetting factor λ and SVM regularization C. Additional details can be provided upon request.

A.1 Recursive least squares

A procedure of generating parameters of an ARIMA (p,d,q) model that is based on the use of the recursive least square method with exponential weighting and constant forgetting factor is described in this section. We assume that a model structure has already been specified using the pre-whitening process. With regard to non-stationarity, since an appropriate d^{th} difference of the observed series is assumed to be a stationary ARMA(p,q) process, we only need to consider the problem of estimating the parameters in such stationary models. In practice, we then treat the d^{th} difference of the original time series as the time series from which the parameters of the complete model are estimated [25]. For simplicity, let $y(t)$ denote our observed stationary process even though it may be an appropriate difference of the original series.

In least squares estimation, unknown parameters of a linear model are chosen such that the sum of the squares of the difference between the actually observed and the computed values, is minimized [26]. The ARX process can be represented in the following linear parametric form,

$$y(t) = \phi^T(t)\theta(t) \tag{1}$$

where y(t) is the observed variable, $\theta(t)$ is the vector of model parameters to be determined and $\phi(t)$ is the vector of exogenous variables including appropriate lags. Least Squares estimation over an observation period spanning n data samples translates into finding the parameter $\hat{\theta}$ that minimizes the following objective function:

$$V(\theta, n) = \frac{1}{2}\sum_{i=1}^{n}\left(y(i) - \phi^T(i)\theta\right)^2 \tag{2}$$

Minimizing Equation 2, we get the closed form Ordinary Least Squares (OLS) solution as follows:

$$\hat{\theta} = \left(\sum_{i=1}^{n}\phi(i)\phi^T(i)\right)^{-1} \cdot \left(\sum_{i=1}^{n}\phi(i)y(i)\right) \tag{3}$$

A.2 The forgetting factor

In our application (real time FDD), we are interested in online parameter estimation. Therefore it is computationally more efficient if we update the estimates in Equation 3 recursively as new data becomes available.

This requires forgetting measurements that are too old, because they correspond to an out-of-date situation and would distort estimation when the "true" system parameters evolve (e.g. due to a fault condition). A particularly simple technique for this purpose is exponential forgetting, which weights prediction errors in the cost function exponentially, decreasing with time elapsed according to a certain time constant, called "forgetting time constant". The recursive algorithm with forgetting is represented by the equations below:

$$\epsilon(t) = y(t) - \phi^T(t)\theta(t-1) \tag{4}$$

$$r(t) = \phi^T(t)P(t-1)\phi(t) \tag{5}$$

$$G(t) = \frac{P(t-1)\phi(t)}{1 + r(t)} \tag{6}$$

$$\theta(t) = \theta(t-1) + G(t)\epsilon(t) \tag{7}$$

$$P(t) = \frac{1}{\lambda}\left[P(t-1) - \frac{P(t-1)\phi(t)\phi^T(t)P(t-1)}{1+r(t)}\right] \tag{8}$$

where P, ϵ and λ are, respectively the parameter variance-covariance matrix, the prediction error and forgetting factor. G represents the algorithm's gain function which is an intermediate step in the update phase of the algorithm.
The corresponding time constant for the forgetting factor λ, is obtained from [17]:

$$\lambda = e^{-\Delta/\tau_f}, \tag{9}$$

where τ_f is the exponential forgetting time constant, λ is the forgetting factor and Δ is the sampling interval.

A.3 Optimizing the forgetting factor

Different choices of forgetting factors lead to different F-measures for fault detection since there exists a trade-off between using only the latest samples (small window) for precise parameter estimation and a larger window for stable parameter estimation. Figure 1 depicts the relationship between the F-measure and forgetting factor with an apparent F-measure maximum on the forgetting factor interval [0.990, 1.000]. The forgetting factor corresponding to the absolute F-measure maximum, $\lambda = 0.9935$, was determined via the golden section search algorithm [27] and it corresponds to a forgetting time constant of approximately 180 samples.

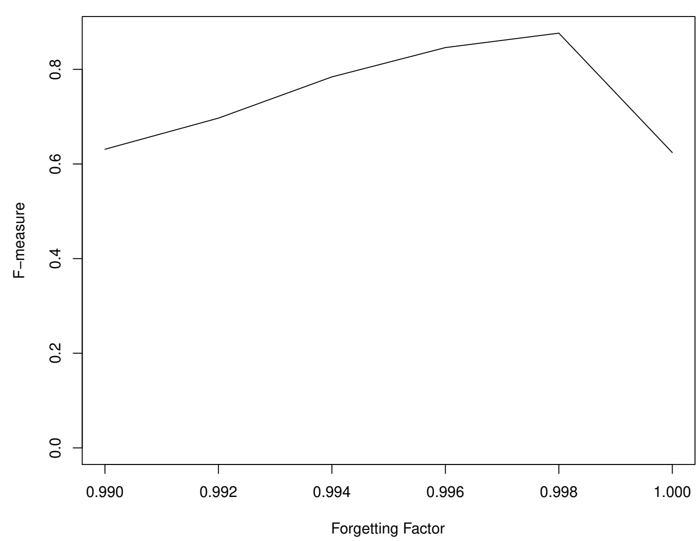

Figure 1: Validation F-measure values versus forgetting factors for fault diagnosis.

A.4 The proposed algorithm

A hybrid approach combining an ARX model and the SVM classification algorithm is proposed for fault detection and diagnosis. The proposed approach is summarized in Algorithm 1 below:

Algorithm 1 Hybrid ARX–SVM FDD Training and Evaluation

Require: Training dataset X_{train} with known fault labels (K classes); ARX model structure (Eq. (1))
Ensure: Trained SVM classifier; performance metrics on X_{test}

```
Identify a normal-operation period in X_train and specify the ARX model structure.
for each candidate forgetting factor λ do
    Recursively estimate ARX parameters θ(t) on X_train via RLS using λ.
    Train an SVM on the feature stream {θ(t)} with their known labels.
    Evaluate that SVM on a held-out validation subset of X_train using the F-measure.
end for
Optimize λ (e.g. via golden-section search) to maximize validation F-measure; denote the optimum by λ*.
Using λ*, re-estimate ARX parameters θ_train(t) on the full X_train and train the final SVM.
for each sample in the unseen test set X_test do
    Recursively estimate ARX parameters θ_test(t) using λ*.
    Classify θ_test(t) with the trained SVM.
end for
Compute final performance metrics (e.g. F-measure) on the testing predictions.
```

A.5 Hyperparameters & Training Protocol

The hyperparameters are reported in Table 2.

Table 2: Key settings, search grids and final selections.

Component	**Grid / Setting**	**Optimized**	**Notes**
ARX structure	$p \in \{1, 2, 3, 4\}$; $q_k \in \{0, 1, 2, 3\}$	$p = 2$; $q_k = 1$	ACF/PACF screening + validation.
RLS (ARX)	$\lambda \in [0.990, 1.000]$ (golden-section); $P_0{=}10^3 I$; $\theta_0{=}0$	$\lambda = 0.9935$	λ tuned to maximize validation macro-F_1; held fixed across classifier comparisons.
SVM (RBF)	$C \in \{0.1, 1, 10, 100\}$, $\gamma \in \{\frac{1}{d}, \frac{0.1}{d}, \frac{10}{d}\}$	C=10, γ=$1/d$	Standardized features.
SVM (Linear)	$C \in \{0.01, 0.1, 1, 10\}$	C=1	—
Random Forest	$n_{\text{trees}} \in \{200, 500\}$; depth $\in$ {None, 20, 40}; $m_{\text{try}} = \sqrt{d}$	500 / None	—
Naïve Bayes	GaussianNB; var_smoothing $\in \{10^{-9}, \ldots, 10^{-7}\}$	10^{-8}	—
Logistic Regression	penalty = ℓ_2; $C \in \{0.1, 1, 10\}$; solver=lbfgs	C=1	—
XGBoost	$n_{\text{estim}} \in \{300, 600\}$; depth $\in \{4, 6\}$; $\eta \in \{0.05, 0.1\}$	600/6/0.05	—
One-class SVM/SVDD (ref.)	$\nu \in \{0.05, 0.1\}$; RBF kernel	ν=0.1	Reference baseline for incipient/unknown faults.
Splits & seeds	n repeated temporal splits; fixed seeds	10	Temporal order respected; no leakage.
Uncertainty & tests	95% CI via stationary/block bootstrap; paired McNemar	—	Time-series-aware uncertainty and significance.

B Reported methods on RP-1043 (context)

We briefly list representative approaches reported on the ASHRAE RP-1043 dataset. Numbers across papers are not strictly comparable due to differences in preprocessing, train/test split, and fault subsets.

Classical/Hybrid

- ARX→SVM hybrid on RP-1043 [1].
- EKF-ROSVM for online/incipient fault sensitivity [2].
- Hybrid SVM with GA/feature search [13].
- LS-SVM and KPCA+LS-SVM variants (various chiller FDD studies; see survey context [10]).

Sequence / Deep Learning

- 1D-CNN+GRU sequence classifier [14].
- Feature-enhanced Temporal Convolutional Network (TCN) [15].
- Causal 1D-CNN for chiller FDD [16].

These references are summarized in Related Work to situate our pipeline among both classical hybrids and modern sequence models.