Benchmarking multivariate time series classification algorithms on open-source datasets for fault detection and diagnostics in HVAC systems

Mohammad Abdollah Fadel ABDOLLAH[1]*, Rossano SCOCCIA[1], Marcello APRILE[1]

[1]Politecnico di Milano, Department of Energy, Milan, Italy,
mohammadabdollah.abdollah@polimi.it

ABSTRACT

With the widespread adoption of building automation systems, alongside the progress in data analytics, sensing, and machine learning, the domain of data-driven fault detection and diagnostics (FDD) for building heating, ventilation, and air conditioning (HVAC) systems has attracted increasing interest. Numerous studies have tested various algorithms using diverse data sources, including simulations, laboratory testing, and real building environments. However, there is a notable gap in the literature regarding systematic benchmarking of these algorithms against each other using the same open-source datasets. In this study, we undertake a comprehensive benchmarking of different classification algorithms tailored for multivariate time series classification. We employ a publicly available data set provided by Berkeley Labs, which includes ground-truth data concerning the presence and absence of building faults. This dataset covers a wide spectrum of seasons and operational conditions, encompassing multiple building system types. It also includes detailed information on fault severity and data points indicative of measurements in building control systems, which are typically accessible to FDD algorithms. The data compilation leverages both simulation models and experimental test facilities. Our findings suggested that Canonical Interval Forest (CIF) and K-Nearest Neighbors (KNN) with Dynamic Time Warping (DTW) have the highest average performance over the datasets analyzed with 0.78 and 0.73 respectively. This is particularly notable given the lower computational resources required by these methods compared to deep learning-based classifiers.

1. INTRODUCTION

HVAC systems in buildings are often prone to defects which can result in suboptimal outcomes, such as increased energy consumption, elevated maintenance expenses, compromised comfort in terms of thermal conditions, and deteriorating air quality. These defects can arise from malfunctioning sensors, equipment breakdowns, or incorrect operations of the system. Research indicates that building system inefficiencies and inadequate control measures can lead to energy losses ranging from 15% to 30% [1]. Consequently, the implementation of Fault Detection and Diagnosis (FDD), or AFDD as it is sometimes called, is essential for the assurance of dependable system functioning and the conservation of energy. Fault detection is primarily concerned with recognizing any improper or unsatisfactory building operations, while fault diagnostics involves pinpointing the exact reasons for these operational failures [2]. In the U.S., within office spaces and institutions of higher learning, the application of FDD has been linked to median energy savings of about 10% per year, along with a simple payback period of two years [3]. This highlights the FDD systems' viability and appeal as an investment in the infrastructure domain.

Numerous investigations have demonstrated the effectiveness of supervised machine learning in identifying and diagnosing faults in heating, ventilation, and air conditioning (HVAC) systems [4,5]. These studies leverage supervised learning algorithms to decipher the intricate links between various monitoring parameters (like temperature, pressure, and flow rates) and the operational conditions (such as normal or faulty operations) [6, 7]. The resulting data-driven models vary in complexity, encompassing everything from simple linear to complex nonlinear equations, individual to collective models, and basic to advanced architectural designs [8, 9]. Significant progress has been noted, particularly in accurately identifying issues in critical HVAC components, including chillers [10,11] and air handling units (AHUs) [12].A fundamental assumption in supervised learning is the availability of labeled data for trustworthy predictive modeling. However, labeling data to accurately reflect the real operational status of systems can be an exhaustive and labor-intensive process. However, it appears no studies have yet been conducted to benchmark multivariate time series classification algorithms on datasets for FDD in HVAC systems. In this study, we used open-sourced datasets released by Lawrence Berkeley labs [13] to benchmark multivariate time series classification algorithms. In the first section, the different algorithms used will be briefly explained. The second section will demonstrate the datasets used and the data preparation process. The third section will showcase the results and discussion of it. Finally, the last section will be the conclusions.

2. BACKGROUND

In the classification of univariate time series, each example is defined by a pair (x, y), where x consists of m observations $(x_1, \dots, x_m)$ representing the time series, and y is a categorical outcome with a number of possible outcomes. Classification algorithms predict the probability distribution of y based on x. In the context of Multivariate Time Series Classification (MTSC), the data input is a collection of vectors within a d dimensional space denoted as $x = (x^1, \dots, x^d)$ with each x_k representing a sequence $(x_{1,k}, x_{2,k}, \dots, x_{m,k})$. We reference the j-th data point in the i-th sequence as $x_{i,j,k}$. MTSC presents unique challenges due to the possibility of relevant features stemming from the interplay between different dimensions, in addition to the autocorrelation within a single time series. Moreover, the substantial amount of data can hinder the identification of distinctive features. MTSC algorithms can be sorted into similar categories as univariate Time Series Classification (TSC) methods, based on their foundational approach, including methods based on distance metrics, shapelets, histograms, and neural networks. This section provides a brief explanation of the eight different algorithms employed in this study, categorized into five distinct types: distance-based, interval-based, deep learning-based, convolutional-based, and dictionary-based methods.

2.1 Distance based classifiers

Distance-based classifiers measure the similarity between time series using specialized distance functions tailored for time series data. Often referred to as elastic distances, these functions are designed to adjust for misalignments between series by allowing for shifts or modifications within the series. A highly favored method for Time Series Classification (TSC) involves the application of a 1-Nearest Neighbor classifier [14], integrated with a specialized distance function. This function is designed to correct for any misalignment in the series by enabling adjustments. The most widely used distance function for this task is Dynamic Time Warping (DTW).
The final computed distance using DTW is $DTW(m, n)$ after applying the recurrence relation throughout the matrix. In this study, we used the 1 NN method with DTW distance function. For short the method will be referred to as DTW.

2.2 Interval based classifiers

Approaches based on intervals examine specific segments of the complete series that are phase-dependent, deriving aggregate statistics from these subsections to facilitate classification. We used the Canonical Interval Forest (CIF) [15] in this study. CIF combines the capabilities of Time series Forest (TSF) [16] and catch22 [17]. While TSF traditionally uses simple summary statistics (mean, standard deviation, slope) for each interval, CIF incorporates a set of 22 more complex and descriptive features from the "catch22" toolkit. These features cover various aspects of the time series data, providing a richer set of descriptors. CIF employs a forest of decision trees to classify time series based on the extracted features. Each tree in the forest makes a decision based on a subset of features and intervals, and the final classification is determined by a majority vote across all trees in the forest. This ensemble approach helps improve accuracy and robustness.

2.3 Convolution based classifiers

Convolution involves using a subsequence to extract features from a time series. This process involves sliding the convolution across the series and computing the dot product at each position. This generates a new series, commonly referred to as an activation map or feature map, where higher values indicate a strong correlation with the convolution. ROCKET [18] derives two key features from the output feature maps: the highest value, often termed as a max pooling operation, and the ratio of positive values, also known as positive predictive value (PPV). For instance, considering the first element of the feature map, which is computed through a dot-product operation between $T_{1:3} * u = T_{1:3} \cdot u$ resulting in $0 + 0 + 3 = 3$ The max pooling method identifies the peak value from the feature map to be used as a feature, and in the given example, the PPV is calculated to be 8/11. Numerous random convolutions are created and integrated with these two features to form an enhanced training dataset. This enriched dataset is then applied to train a linear classifier. ROCKET also incorporates dilation, which effectively acts as a down sampling mechanism by creating intervals between data points. In this context, a convolution with a dilation factor of d is matched against data points that are spaced d steps apart to measure the separation. ARSENAL [19] an ensemble of ROCKET transformers using Ridge classifier [20] base classifier. Weights each classifier using the accuracy from the ridge cross-validation. Allows for generation of probability estimates at the expense of scalability compared to ROCKET.

2.4 Deep learning-based classifiers

Wide range of neural network-based architectures have been used for TSC purposes in the literature [21]. In this study we used three deep learning-based classifiers.

In their 2017 paper, Wang et al. introduced an advanced Residual Network (ResNet) [22]. This model is 11 layers, with the initial nine being convolutional layers followed by a Global Average Pooling (GAP) layer that processes the time series data across the temporal dimension. ResNets are distinguished by their use of shortcut residual connections that link the output of a residual block back to its input. This setup facilitates the direct flow of gradients during training, significantly mitigating the issue of vanishing gradients. The described network structure includes three such residual blocks, each capped by a GAP layer and culminating in a softmax classifier. The classifier is designed to have as many neurons as there are classes in the dataset. Within each block, the layers consist of three sequential convolutions, where the outputs are combined with the block's input and forwarded to the subsequent layer. Uniformly across the network, each convolution utilizes 64 filters, employs the ReLU activation function, and is preceded by a batch normalization step. The lengths of the filters in these convolutions vary, being 8, 5, and 3 for the first, second, and third convolutions, respectively.

The Multivariate Long Short-Term Memory Fully Convolutional Network (LSTMFCN) [23] merges LSTM and FCN technologies to enhance multivariate time series classification. This architecture leverages LSTMs to learn sequence dependencies and FCNs for feature extraction. Additional adaptations include squeeze-and-excitation blocks in the first two convolutional layers to adjust feature map interdependencies. The model comes in two variants, one with and one without an attention mechanism in the LSTM layers. However, studies across 35 datasets revealed minimal performance differences between the two. For simplicity and reproducibility, the version without the attention mechanism is preferred. In original tests, the number of LSTM cells was variable; in current applications, it is fixed at 64 for consistency across datasets [24].

InceptionTime is an ensemble of deep convolutional neural network models tailored for TSC [25]. This approach leverages the architecture of Inception-v4 by deploying multiple Inception modules within each network, where each module applies a variety of filters simultaneously to the input time series. This allows the network to capture and learn from a broad range of features at different scales. To enhance stability and performance, InceptionTime combines five such networks with randomly initialized weights, utilizing the ensemble's aggregate output for classification. This structure provides a robust and efficient way to handle the complexity and diversity of time series data in classification tasks.

2.5 Dictionary-based classifiers

Dictionary based approaches adapt the bag of words model commonly used in signal processing, computer vision and audio processing for time series classification [26]. These approaches use phase-independent subsequences by sliding a window over time series. However, rather than measuring the distance to a subsequence, as in shapelets, each window is transformed into a word, and the frequency of occurrence of repeating patterns is recorded. Dictionary based methods usually involve several steps:

1. Subseries Extraction: Each time series is divided into overlapping windows or subseries.
2. Discretization: Each window is transformed into a discrete-valued word. This involves normalizing the values in the window to have a uniform standard deviation, reducing the dimensionality using a truncated Fourier transform to retain only the most significant coefficients, and then converting these coefficients into symbols from a fixed-size alphabet.
3. Feature Vector Construction: A sparse feature vector is created from histograms of the word counts.
4. Classification: These feature vectors are then used with machine learning algorithms to classify the time series.

In this study, we use the MUSE method [27] which extends the WEASEL algorithm [28] to handle multivariate time series data, employing the Symbolic Fourier Approximation (SFA) for the discretization process. This method stands out by its ability to effectively transform real-valued measurements into discrete symbols, enabling sophisticated pattern recognition in complex datasets.

3. DATASETS DESCRIPTION AND PREPARATION

Three datasets from the LBNL fault detection and diagnostics datasets [13]. These datasets can be used to evaluate and benchmark the performance accuracy of FDD algorithms or tools. It contains operational data from simulation and laboratory experiments. In this study we used three of the eight datasets. The three

datasets used are generated using Modelica and Energyplus [29]. The simulations generated one year of data, each with a time step of one minute. Each dataset contains one file of fault free case and several files with different faults injected into the models. A brief description of the datasets and the faults imposed is given in the following subsection. The full description, data points definition and control sequences can be found in the documentation provided for each dataset in [13].

3.1 Dataset description

The first dataset is a simulated boiler plant serving a 12-story building with individual floors having a dedicated AHU serving five zones, and individual zones having a dedicated VAV terminal unit. Each terminal unit has a reheat coil that uses hot water produced by the plant. The plant consists of two parallel boilers and pumps that distribute the hot water to these reheat coils. Sensors and valves are also used to control water flow through the plant. The dataset contains 22 features both continuous and discrete signals. The faults imposed in the model are provided in Table 1.

Table 1: Input scenarios and fault imposed in the boiler plant model. Taken from [13]

Input scenarios			**Method of Fault Imposition**
Fault type		**Fault intensity**	
The hot water leaving temperature sensor of boiler 1	Sensor bias	-4°C, -2°C, 2°C, 4°C	Add bias to sensor output
The hot water leaving temperature sensor of the hot water loop		-4°C, -2°C, 2°C, 4°C	
The differential pressure sensor in the hot water loop		-20%, -10%, 10%, 20%	
Boiler 1 heat exchanger	Fouling	95%, 80%, 65%	Multiply intensity value by heat transfer coefficient
Controller PI for boiler supply temperature setpoint	Inappropriate tuning	-	Modify gain value of controllers

The second dataset is simulated chiller plant serves the same building as the boiler plant. The chiller plant serves the dedicated AHU on each floor with cold water. The plant consists of a primary loop with three chillers for producing chilled water, a secondary loop for delivering chilled water to the AHUs, and a condenser water loop with cooling towers for rejecting heat to the ambient. Sensors, pumps, and valves are used to control water flow through the plant. The dataset contains 77 features both continuous and discrete signals. The faults imposed in the model are provided in Table 2.

Table 2: Input scenarios and fault imposed in the chiller plant model. Taken from [13]

Input scenarios			**Method of Fault Imposition**
Fault type		**Fault intensity**	
The chilled water leaving temperature sensor of Chiller 1	Sensor bias	-2°C, -1°C, 1°C, 2°C	Add bias to sensor output
The condenser water leaving temperature sensor of Cooling tower 1			
The differential pressure sensor in the secondary chilled water loop		-20%, -10%, 10%, 20%	
The condenser water leaving the three-way valve	Leakage	25%, 50%, 75%	Increase the default minimum position setting
The condenser water leaving the three-way valve	Stuck	50%, 75%	Assign a fixed simulated controlled device position
Cooling tower 1 heat exchanger	Fouling	95%, 80%, 65%	Multiply intensity value by heat transfer coefficient

Controller PI for condenser loop supply temperature setpoint	Inappropriate tuning	-	Modify gain value of controllers

The third dataset is generated from a simulated system consisting of a single-duct air handling unit (SD-AHU) providing conditioned air to five VAV terminal units, each serving a single zone (four perimeter and one interior) on the middle floor of a three-story building. The SD-AHU has a chilled water-cooling coil, variable speed supply and return fans, and delivers air at a constant temperature and static pressure to the terminal units. Individual terminal units control the volume of air entering a zone and use hydronic reheat when necessary to satisfy the temperature setpoint in a zone. The dataset contains 30 features both continuous and discrete signals. The faults imposed in the model are provided in Table 3.

Table 3: Input scenarios and fault imposed in single duct AHU model. Taken from [13]

Input scenarios		**Method of Fault Imposition**
Fault type	**Fault intensity**	
Outdoor air temperature sensor bias	2°C, 4°C, -2°C, -4°C	Add bias to sensor output.
Supply air temperature sensor bias	2°C, 4°C, -2°C, -4°C	Add bias to sensor output.
Stuck outdoor air Damper	10%, 25%, 75%, 100% open	Automated override of outdoor air damper position to indicate that OA damper is stuck.
Leaking cooling coil valve	10%, 25%, 40%, 50%	Adjusted the minimum coil valve position value when the control signal is zero.
Stuck cooling coil valve	10%, 25%, 50%, 75%	Automated override of coil valve position to indicate that valve is stuck.

3.2 Datasets preparation

The procedure for the data preparation is the same for the three datasets. The data was first resampled from one-minute interval to ten minutes. This was done to reduce the computational cost after encountering multiple memory issues dealing with the data in the one-minute interval form. The resampling was done by the mean for the continuous variables and the mode for the discrete ones. Since the data was provided in a separate file for each fault state, the data was labeled, concatenated, and then normalized. All the TSC models need to have a specific format of the data of multi-index format, instances which is a consistent window of time and time points which is the timestep. We choose to prepare the data in a daily instance. The data was split into five folds each with an expanding number of instances for the training set while shifting the window of the testing set as demonstrated in Figure 1.

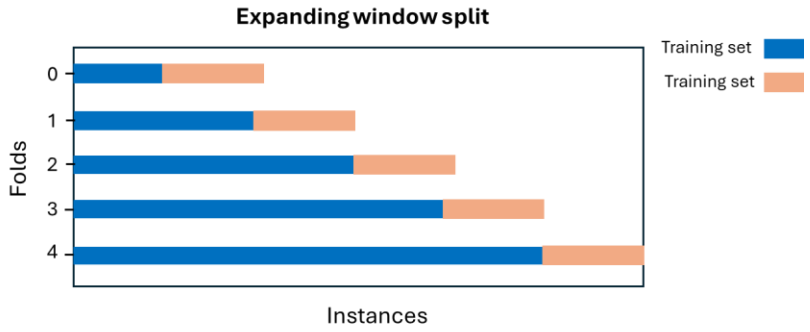

Figure 1: Demonstration of the expanding window method for cross validation

4. RESULTS

4.1 Performance evaluation

Table 4 provides the aggregated results from the five-fold cross-validation applied across three datasets. The results include accuracy, precision, recall and F1 score. Each metric is defined in function of True or False positives (TP, FP) and True or False negatives (TN, FN) as in Eq. 2 to 5 below. The CIF and 1nn DTW classifiers consistently

demonstrated high accuracy and F1 scores, suggesting their robustness in handling diverse operational conditions in HVAC systems. For the Boiler plant, 1nn DTW exhibited remarkably high accuracy at 0.97, indicating excellent alignment with the dataset's characteristics, which might be attributed to its dynamic time warping capability that efficiently handles time shifts and distortions in time series data. For the SHAHU dataset, most classifiers show moderate to good performance, with accuracy ranging mainly from 0.45 to 0.67. The CIF classifier and InceptionTime perform best in terms of balance across precision, recall, and F1 score. For the chiller plant dataset, all the classifiers showed better classification performance across most algorithms compared to the SD-AHU dataset, with accuracies predominantly above 0.58. Notably, the LSTMFCN (Long Short-Term Memory Fully Convolutional Network) shows very high performance metrics, indicating effective learning from this dataset's characteristics. While for the boiler plant dataset, all the classifiers showed highly variable performance across classifiers. CIF and 1nn DTW (1-Nearest Neighbor with Dynamic Time Warping) exhibit very high performance, with accuracies near or above 0.89, suggesting that these methods are particularly well-suited to the characteristics of the boiler plant data.

Table 4: Results for the three datasets

	SDAHU					**Chiller plant**					**Boiler plant**				
Classifier	Runtime (s)	Accuracy	Precision	Recall	F1 Score	Runtime (s)	Accuracy	Precision	Recall	F1 Score	Runtime (s)	Accuracy	Precision	Recall	F1 Score
CIF	3221	0.65	0.63	0.65	0.63	9106	0.84	0.84	0.84	0.83	6123	0.89	0.9	0.89	0.88
ResNet	30005	0.67	0.60	0.67	0.61	33485	0.86	0.87	0.86	0.86	29032	0.11	0.09	0.1	0.09
Arsenal	571	0.45	0.40	0.45	0.39	595	0.45	0.46	0.45	0.40	1711	0.59	0.59	0.59	0.58
InceptionTime	11761	0.67	0.63	0.67	0.64	13209	0.84	0.86	0.84	0.84	16339	0.06	0.006	0.06	0.008
TimeSeriesSVC	7403	0.66	0.60	0.66	0.62	10496	0.58	0.58	0.58	0.55	13035	0.18	0.15	0.18	0.14
1nn DTW	37	0.63	0.55	0.63	0.57	131	0.60	0.67	0.60	0.61	462	0.97	0.98	0.97	0.97
Rocket	3606	0.48	0.42	0.48	0.43	4083	0.46	0.49	0.46	0.45	6324	0.32	0.36	0.35	0.3
MUSE	693	0.63	0.51	0.63	0.55	1555	0.68	0.70	0.68	0.68	3248	0.62	0.74	0.62	0.63
LSTMFCNC	7185	0.66	0.61	0.66	0.62	13898	0.87	0.86	0.87	0.85	25873	0.088	0.042	0.088	0.044

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

$$F1\ \text{Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

4.2 Runtime analysis

Figure 2 illustrates the computational efficiency of each classifier. The 1nn DTW classifier not only performed well in accuracy but also in computational efficiency, with significantly lower runtime across all datasets. This aspect is particularly valuable in scenarios where rapid fault detection is crucial to preventing extended HVAC system downtimes. In contrast, deep learning models like ResNet and LSTMFCN, while providing decent accuracy, incurred much higher computational costs, which might limit their practicality in resource-constrained environments. CIF also shows longer runtimes, though not as extreme as ResNet, across all datasets. It strikes a balance between runtime and performance, as per the previous results table, which showed it as one of the top-performing classifiers.

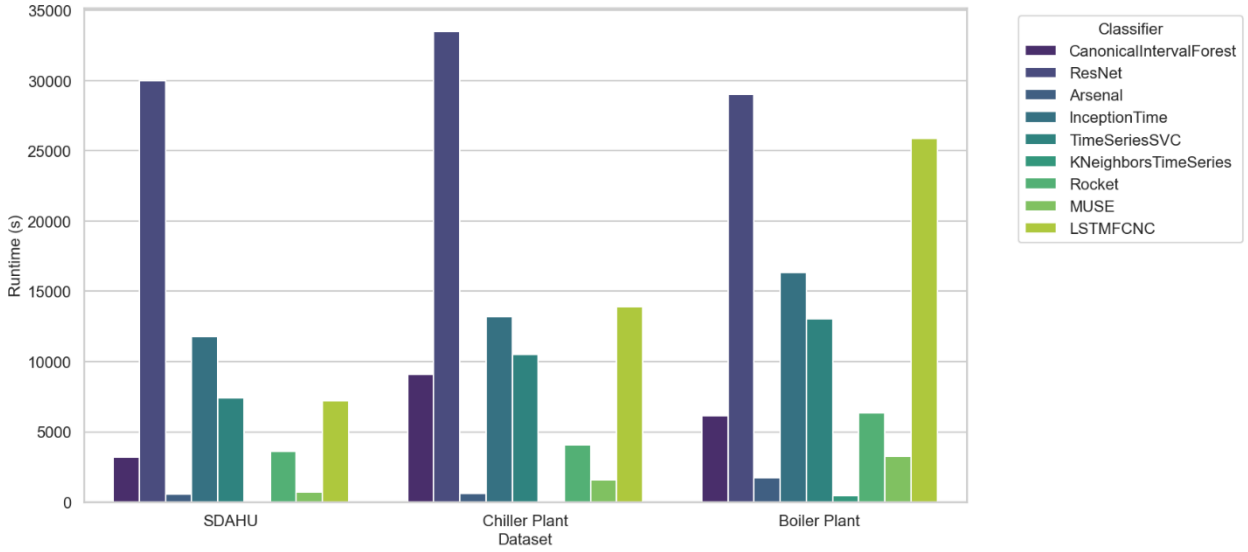

Figure 2: Runtime comparison across datasets

4.3 Comparative performance across datasets

As illustrated in Figure 3, the algorithms' performance metrics such as accuracy, precision, recall, and F1 score vary across the datasets. This variation could be attributed to the differing characteristics and distribution of features within each dataset. The Boiler plant dataset showed the largest discrepancy in algorithm performance, which suggests that feature distribution significantly influences the effectiveness of classification methods.

The accuracy box plot demonstrates that classifiers maintain moderate and consistent performance on the SDAHU dataset, as evidenced by a tight interquartile range (IQR). The Chiller Plant dataset reveals higher median accuracy but a broader spread, indicating variable performance. The Boiler Plant dataset presents the most significant challenge, with a wide range in accuracy and a notably lower median, suggesting diverse classifier efficacy on this dataset. Precision scores for the SDAHU dataset show narrow variance, denoting reliable positive prediction consistency among classifiers. The Chiller Plant dataset mirrors this reliability with a high median; however, the extended IQR and presence of outliers denote performance variability. Conversely, the Boiler Plant dataset's precision spread is considerably broader, with classifiers showing disparate levels of precision, likely reflective of complex feature interactions and fault presentations within this dataset. The recall metric for the SDAHU dataset maintains a narrow IQR, mirroring the accuracy trend, indicating consistent classifier sensitivity. Chiller Plant data showcases a commendable median recall, though outliers signal exceptional cases of underperformance. The recall for the Boiler Plant dataset is distributed across a broader spectrum, signaling inconsistency in classifiers' comprehensive detection of positive instances. The F1 score synthesis for the SD-AHU dataset aligns closely with precision and recall metrics, suggesting balanced classifier performance. The Chiller Plant dataset maintains high F1 scores but is not without exceptions, as outliers indicate notable deviations. The Boiler Plant dataset's wide F1 score dispersion and reduced median underscore the inherent difficulty in achieving precision-recall equilibrium across classifiers for this particular dataset.

To further understand the performance variations, statistical analyses were conducted on the results. Although CIF and KNN with DTW showed higher average metrics, the differences were not statistically significant (p-values > 0.05). This analysis suggests that while some classifiers may perform better on average, the differences might not be meaningful in practical applications without further optimization. The potential for improving classification results through advanced feature engineering is considerable. Adjusting data sampling rates and enhancing feature extraction techniques could lead to significant performance gains, as indicated by the initial discrepancies observed in classifiers' performances across the three datasets. Exploring these avenues could provide deeper insights and more robust models for fault detection in HVAC systems.

The extended analysis of classifier performance across multiple datasets highlights the importance of algorithm selection based on the specific characteristics of the data. While CIF and KNN with DTW are generally robust across various settings, their effectiveness can be enhanced through tailored feature engineering. This study's findings underline the need for comprehensive testing and optimization to deploy effective fault detection and diagnostics solutions in real-world scenarios.

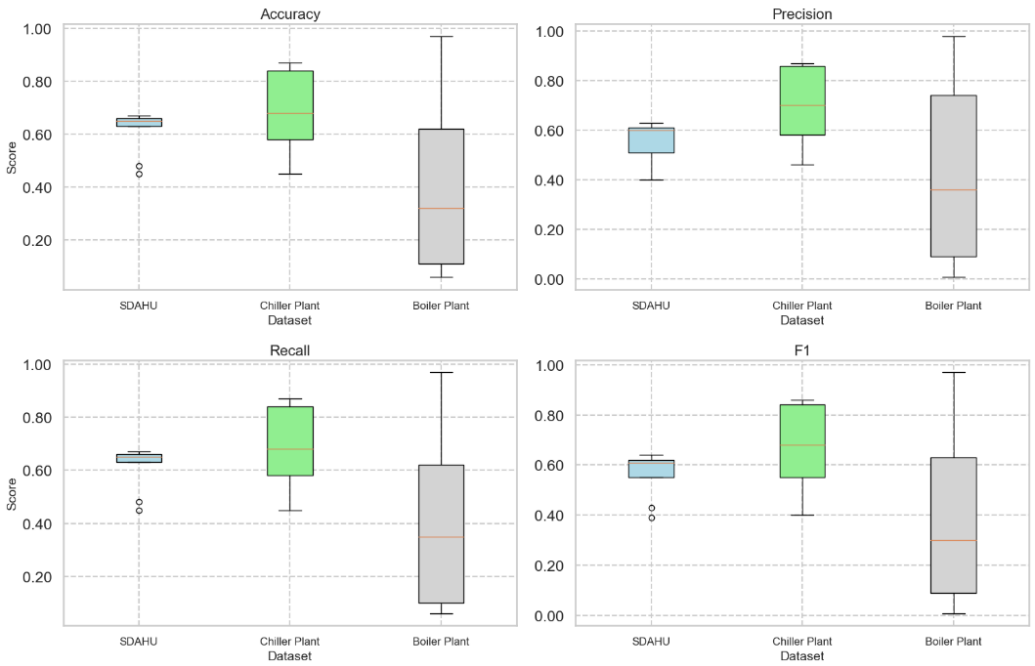

Figure 3: Box plot of performance metrics across datasets

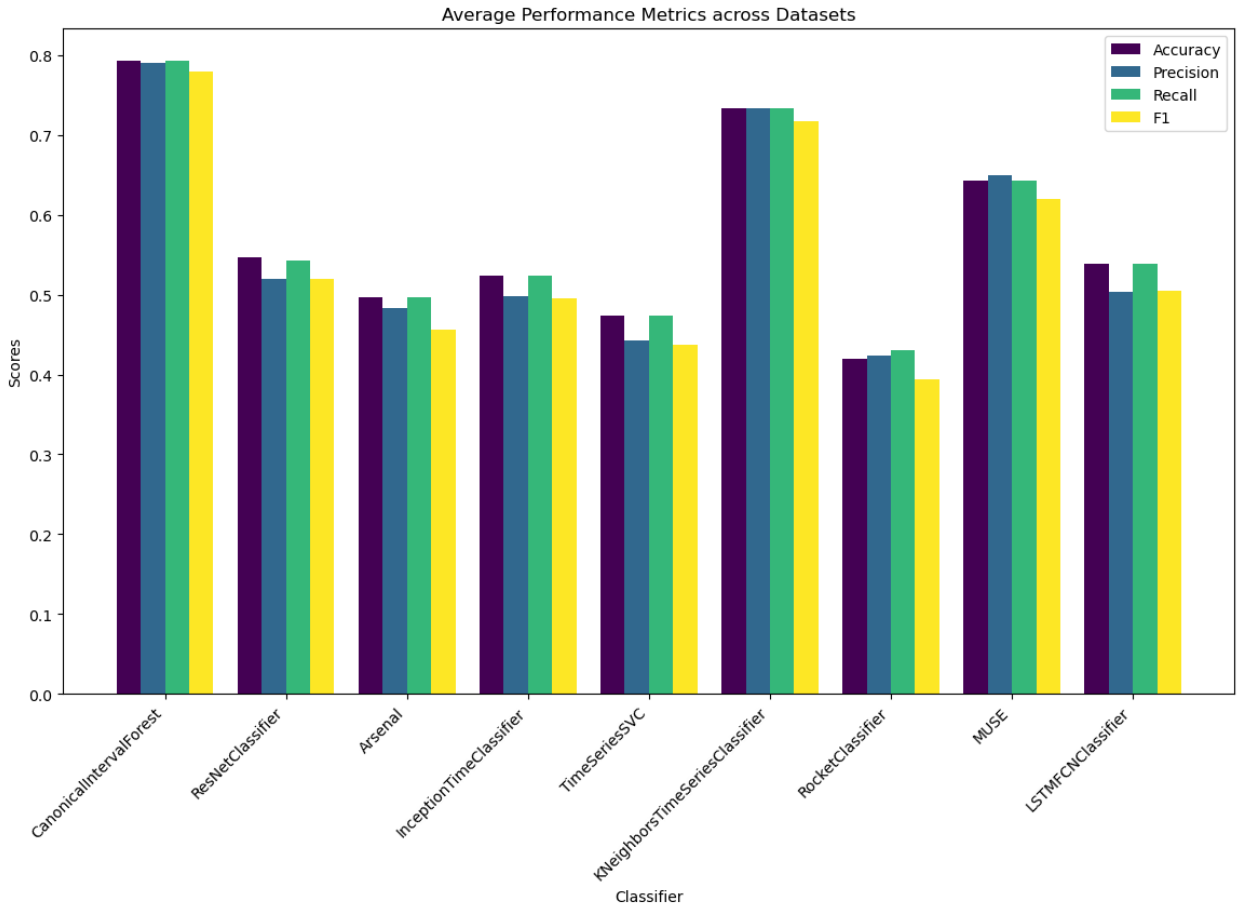

Figure 4: Average performance metrics across datasets

5. CONCLUSIONS

This study presents a benchmarking of multivariate time series classification algorithms using open-source datasets to address fault detection and diagnostics in HVAC systems. Our findings reveal that CIF and KNN with DTW exhibit the most promising performance, achieving average scores of 0.78 and 0.73, respectively. These results underscore the potential of using less computationally intensive methods over more complex deep learning approaches, especially in environments where computational resources are limited.

The analysis across different datasets highlights the variability in performance, which is largely influenced by the characteristics and distribution of features within each dataset. Notably, the boiler plant dataset posed significant challenges, affecting the algorithms' performance due to its feature distribution. This indicates a critical area for further research—optimizing feature engineering and data preprocessing to enhance model accuracy in complex scenarios.

Future studies should explore the effect of feature engineering on the performance of the algorithms. Moreover, more datasets of HVAC systems for FDD and more algorithms are required to generalize the results of this study.

By advancing the capabilities of FDD systems through such comprehensive benchmarks, we can significantly contribute to energy conservation and operational efficiency in building management systems, paving the way for smarter, more reliable HVAC operations.

REFERENCES

[1] S. Katipamula and M. R. Brambley, "Review article: Methods for fault detection, diagnostics, and prognostics for building systems—A review, part I," *HVAC and R Research*, vol. 11, no. 1, pp. 3–25, 2005, doi: 10.1080/10789669.2005.10391123.

[2] K. Cant and R. Evins, "Improved calibration of building models using approximate Bayesian calibration and neural networks," *J Build Perform Simul*, vol. 16, no. 3, pp. 291–307, May 2023, doi: 10.1080/19401493.2022.2137236.

[3] H. Kramer, G. Lin, C. Curtin, E. Crowe, and J. Granderson, "Building analytics and monitoring-based commissioning: industry practice, costs, and savings," *Energy Effic*, vol. 13, no. 3, pp. 537–549, Mar. 2020, doi: 10.1007/S12053-019-09790-2/TABLES/3.

[4] G. Li *et al.*, "Interpretation of convolutional neural network-based building HVAC fault diagnosis model using improved layer-wise relevance propagation," *Energy Build*, vol. 286, p. 112949, May 2023, doi: 10.1016/J.ENBUILD.2023.112949.

[5] V. Singh, J. Mathur, and A. Bhatia, "A comprehensive review: Fault detection, diagnostics, prognostics, and fault modeling in HVAC systems," *International Journal of Refrigeration*, vol. 144, pp. 283–295, Dec. 2022, doi: 10.1016/J.IJREFRIG.2022.08.017.

[6] Z. Du, X. Liang, S. Chen, X. Zhu, K. Chen, and X. Jin, "Knowledge-infused deep learning diagnosis model with self-assessment for smart management in HVAC systems," *Energy*, vol. 263, p. 125969, Jan. 2023, doi: 10.1016/J.ENERGY.2022.125969.

[7] L. Wang, J. Braun, and S. Dahal, "An evolving learning-based fault detection and diagnosis method: Case study for a passive chilled beam system," *Energy*, vol. 265, p. 126337, Feb. 2023, doi: 10.1016/J.ENERGY.2022.126337.

[8] Y. Zhao, T. Li, X. Zhang, and C. Zhang, "Artificial intelligence-based fault detection and diagnosis methods for building energy systems: Advantages, challenges and the future," *Renewable and Sustainable Energy Reviews*, vol. 109, pp. 85–101, Jul. 2019, doi: 10.1016/J.RSER.2019.04.021.

[9] Z. Chen *et al.*, "A review of data-driven fault detection and diagnostics for building HVAC systems," *Appl Energy*, vol. 339, p. 121030, Jun. 2023, doi: 10.1016/J.APENERGY.2023.121030.

[10] H. Han, X. Cui, Y. Fan, and H. Qing, "Least squares support vector machine (LS-SVM)-based chiller fault diagnosis using fault indicative features," *Appl Therm Eng*, vol. 154, pp. 540–547, May 2019, doi: 10.1016/J.APPLTHERMALENG.2019.03.111.

[11] Y. Gao, H. Han, Z. X. Ren, J. Q. Gao, S. X. Jiang, and Y. T. Yang, "Comprehensive study on sensitive parameters for chiller fault diagnosis," *Energy Build*, vol. 251, p. 111318, Nov. 2021, doi: 10.1016/J.ENBUILD.2021.111318.

[12] Y. Zhao, J. Wen, F. Xiao, X. Yang, and S. Wang, "Diagnostic Bayesian networks for diagnosing air handling units faults – part I: Faults in dampers, fans, filters and sensors," *Appl Therm Eng*, vol. 111, pp. 1272–1286, Jan. 2017, doi: 10.1016/J.APPLTHERMALENG.2015.09.121.

[13] J. Granderson *et al.*, "Lawrence Berkeley National Laboratory, LBNL FDD Data Sets. DOI: https://dx.doi.org/10.25984/1881324," 2022.
[14] A. P. Ruiz, M. Flynn, and A. Bagnall, "Benchmarking Multivariate Time Series Classification Algorithms," *Data Min Knowl Discov*, vol. 35, no. 2, pp. 401–449, Jul. 2020, doi: 10.1007/s10618-020-00727-3.
[15] M. Middlehurst, J. Large, and A. Bagnall, "The Canonical Interval Forest (CIF) Classifier for Time Series Classification," *Proceedings - 2020 IEEE International Conference on Big Data, Big Data 2020*, pp. 188–195, Dec. 2020, doi: 10.1109/BIGDATA50022.2020.9378424.
[16] H. Deng, G. Runger, E. Tuv, and M. Vladimir, "A Time Series Forest for Classification and Feature Extraction," *Inf Sci (N Y)*, vol. 239, pp. 142–153, Feb. 2013, doi: 10.1016/j.ins.2013.02.030.
[17] C. H. Lubba *et al.*, "catch22: CAnonical Time-series CHaracteristics," *Data Min Knowl Discov*, vol. 33, no. 6, pp. 1821–1852, Jan. 2019, doi: 10.1007/s10618-019-00647-x.
[18] A. Dempster, F. Petitjean, and G. I. Webb, "ROCKET: Exceptionally fast and accurate time series classification using random convolutional kernels," *Data Min Knowl Discov*, vol. 34, no. 5, pp. 1454–1495, Oct. 2019, doi: 10.1007/s10618-020-00701-z.
[19] M. Middlehurst, J. Large, M. Flynn, J. Lines, A. Bostrom, and A. Bagnall, "HIVE-COTE 2.0: a new meta ensemble for time series classification," *Mach Learn*, vol. 110, no. 11–12, pp. 3211–3243, Apr. 2021, doi: 10.1007/s10994-021-06057-9.
[20] C. Peng and Q. Cheng, "Discriminative Ridge Machine: A Classifier for High-Dimensional Data or Imbalanced Data," *IEEE Trans Neural Netw Learn Syst*, vol. 32, no. 6, pp. 2595–2609, Apr. 2019, doi: 10.1109/TNNLS.2020.3006877.
[21] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. A. Muller, "Deep learning for time series classification: a review," *Data Min Knowl Discov*, vol. 33, no. 4, pp. 917–963, Jul. 2019, doi: 10.1007/S10618-019-00619-1/FIGURES/16.
[22] X. Huang, C. Zhu, and W. Chen, "RestNet: Boosting Cross-Domain Few-Shot Segmentation with Residual Transformation Network," Aug. 2023, Accessed: Apr. 20, 2024. [Online]. Available: https://arxiv.org/abs/2308.13469v2
[23] F. Karim, S. Majumdar, H. Darabi, and S. Harford, "Multivariate LSTM-FCNs for Time Series Classification," *Neural Networks*, vol. 116, pp. 237–245, Jan. 2018, doi: 10.1016/j.neunet.2019.04.014.
[24] M. Abouelnaga, J. Vitay, and A. Farahani, "Multivariate Time Series Classification: A Deep Learning Approach," Jul. 2023, Accessed: Apr. 20, 2024. [Online]. Available: https://arxiv.org/abs/2307.02253v1
[25] H. I. Fawaz *et al.*, "InceptionTime: Finding AlexNet for Time Series Classification," *Data Min Knowl Discov*, vol. 34, no. 6, pp. 1936–1962, Sep. 2019, doi: 10.1007/s10618-020-00710-y.
[26] N. Tabassum, S. Menon, and A. Jastrzębska, "Time-series classification with SAFE: Simple and fast segmented word embedding-based neural time series classifier," *Inf Process Manag*, vol. 59, no. 5, p. 103044, Sep. 2022, doi: 10.1016/J.IPM.2022.103044.
[27] M. Middlehurst, P. Schäfer, and A. Bagnall, "Bake off redux: a review and experimental evaluation of recent time series classification algorithms," Apr. 2023, Accessed: Apr. 21, 2024. [Online]. Available: https://arxiv.org/abs/2304.13029v1
[28] P. Schäfer and U. Leser, "Fast and Accurate Time Series Classification with WEASEL," *International Conference on Information and Knowledge Management, Proceedings*, vol. Part F131841, pp. 637–646, Jan. 2017, doi: 10.1145/3132847.3132980.
[29] " . EnergyPlus™. Computer software. Version 00. September 30, 2017. https://www.osti.gov//servlets/purl/1395882."