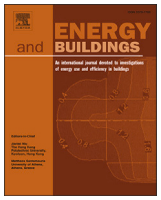
Energy & Buildings 295 (2023) 113327

Contents lists available at ScienceDirect

Energy & Buildings

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



Machine learning-based estimation of buildings' characteristics employing electrical and chilled water consumption data: Pipeline optimization

Farhang Raymand [a], Behzad Najafi [b,*], Alireza Haghighat Mamaghani [c], Amin Moazami [d,e], Fabio Rinaldi [b]

[a] Department of Environmental Science, Radboud Universtiy, Heyendaalseweg 135, 6525 AJ Nijmegen, Netherlands
[b] Dipartimento di Energia, Politecnico di Milano, Via Lambruschini 4, Milano 20156, Italy
[c] Department of Building, Civil and Environmental Engineering, Concordia University, 1455 de Maisonneuve Blvd. W., BE-351, Montreal, Quebec H3G 1M8, Canada
[d] Department of Ocean Operations and Civil Engineering, Faculty of Engineering, NTNU, Ålesund, Norway
[e] Department of Architectural Engineering, SINTEF Community, SINTEF AS, Børrestuveien 3, 0373 Oslo, Norway

ARTICLE INFO

Keywords:
Smart meter
Commercial buildings classification
Machine learning
Feature extraction
Feature selection
Pipeline optimization

ABSTRACT

Smart meter-driven remote auditing of buildings, as an alternative to the labor-intensive on-site visits, permits large-scale and rapid identification of buildings with low energy performance. The existing literature has mainly focused on electricity meters' data from a rather small set of buildings and efforts have often not been made to facilitate the models' physical interpretability. Accordingly, the present work focuses on the implementation and optimization of ML-based pipelines for building characterization (by use type (A), performance class (B), and operation group (C)) employing hourly electrical and chilled-water consumption data. Utilizing the Building Data Genome Project II dataset (with data from 1636 buildings), feature generation, feature selection, and pipeline optimization steps are performed for each pipeline. Results demonstrate that performing the latter two steps improves the model's accuracy (5.3%, 2.9%, and 3.9% for pipelines A, B, and C compared to a benchmark model), while notably reduces the number of utilized features (94.7%, 88.3%, 89.4%), enhancing the models' interpretability. Furthermore, adding features extracted from chilled-water consumption data boosts the accuracy (with respect to baseline) for the second subset by 12.4%, 13.5%, and 7.2%, while decreasing the feature count by 97.2%, 96.4%, and 96.5%, respectively.

1. Introduction

Building sector is regarded as a major energy-consumer with being responsible for about one third of the total global final energy use [1]. This energy consumption is divided almost equally between residential and non-residential (commercial) buildings. On the other hand, buildings account for approximately 25% of the global greenhouse gas emissions, which can make them a key player in the battle against climate change [2]. To realize the set targets in addressing climate change, building-related emissions have attempted to be controlled through a variety of strategies such as renovation and retrofitting. In this regard, the number of established initiatives, incentives, and regulations has been growing to reduce the energy demand of existing and new buildings on national and international levels.

From a more practical point of view, the significant electricity consumption of buildings negatively impacts the grid. It has been shown that residential and commercial buildings account for respectively 50% and 25% of total demand at peak hours [3]. Electrical grids face challenges in the form of higher variance in the peak load and overall demand, which in turn lead to congestion in the transmission network and increase in the energy prices. In order to meet the demand during peak times, operation of less energy-efficient power plants with higher carbon emissions becomes an unavoidable solution [4]. Therefore, many strides have been made in recent years to enhance the electrical grid and pertinent technologies by focusing on digital communication, data collection, and managerial shifts towards smarter decision-making approaches. These innovations are meant to usher a more intelligent use of electricity and increase the efficacy of power production and consumption [5]. Another outcome of the paradigm shift in grid management is a surge in smart meter deployment over recent years, which has made the hourly total building electricity consumption data widely available.

* Corresponding author.
E-mail address: behzad.najafi@polimi.it (B. Najafi).

https://doi.org/10.1016/j.enbuild.2023.113327
Received 7 March 2023; Received in revised form 8 June 2023; Accepted 26 June 2023
Available online 30 June 2023

Load profiles that can be retrieved from smart meter data [6] show the variations in the building usage, including the timing (temporal trend) and level of intensity (load ratio) of energy consumption. Employing these profiles, different studies can be conducted with focuses on portfolio analysis, retrofitting and/or commissioning. Building performance analysis and commissioning pave the way to notable energy-saving opportunities [7], reductions in emissions, and cuts in the operating costs of buildings [8]. In this context, a few works have been focused on employing machine learning-based analysis of smart meter data for disaggregating [9] the consumption of the buildings' HVAC systems [10,11], which is a key step in analyzing the performance of these units.

Another application of applying machine learning-based method to smart meter data is the estimation of energy performance of the buildings (remote auditing). Remote auditing of buildings permits rapid identification of buildings with lower performance. Implementing energy efficiency-related interventions in such buildings guarantees a higher impact in terms of the achieved overall energy saving. Furthermore, as these buildings have higher vulnerability to climate change and extreme climate events [12], improving their performance notably enhances the resilience of the urban energy systems [13]. In this context, Miller [14] proposed a machine learning-based methodology that utilized a rigorous feature generation procedure, employing smart meter data (Building Data Genome Project dataset [15]), to assess the buildings' use type, performance grade, and the corresponding operation group. Najafi et al. [16], considering the same data set and estimation objectives as in Miller's work [14], applied multiple feature selection procedures, which significantly decreased the number of required features while keeping the models as accurate or even improving their prediction capability [16]. However, in these works, only the electricity meter data was exploited, while as more and more smart meters are installed globally and building owners and utility companies try to better understand usage patterns, the availability of data from meters other than electricity has increased. Furthermore, most studies utilized a relatively small dataset of buildings' electrical meter data, while a larger dataset (Building Data Genome Project 2 [17] that also includes other meters) has recently become available. Noteworthy that none of the previous studies conducted in this area has attempted to optimize the employed machine learning-based estimation pipelines.

Motivated by the mentioned research gaps, in the present work, using Building Data Genome Project 2 dataset [17], machine learning-based pipelines are implemented and optimized while employing A) the features (proposed in [14]) generated from electrical consumption data and B) those extracted from a combination of both electrical and chilled water (CW) demand data. As a result, two different subsets of the dataset are considered: subset A includes all the buildings for which electrical consumption data is available; subset B instead comprises of the buildings with both electrical and CW consumption data. Accordingly, after performing the data cleaning step, subset A includes 1494 buildings, for which the feature generation procedure results in 374 features. Subset B instead involves 748 features generated from the electrical and CW consumption data of 374 buildings (as both consumption data is available for fewer buildings). While considering the building use type, performance class, and operation group as estimation targets and utilizing the above-mentioned subsets, the feature selection and algorithm optimization steps are implemented for each pipeline. Hence, the contributions of this paper, with respect to the related literature, can be summarized as:

- Performing building classification based on usage, performance class and operation strategy for a large dataset including over 1494 buildings with two years of hourly electricity consumption data
- Evaluation of different machine learning classifiers and feature selection algorithms for building classification
- Addition of chilled water meter data to electricity data, and assessing its impact on the performance achieved for different classification targets
- Extraction of the optimal pipeline (algorithm and the corresponding tuning parameters) for each target
- Investigating the physical phenomenon behind each added feature to facilitate physical interpretability of the pipelines

The dataset and classification objectives are introduced in section 2. The overall methodology as well as a description of the extracted features is then presented in section 3. Section 3 also discusses the utilized ML algorithms, the metrics used for model performance assessment, and the developed feature selection techniques. Section 4 represents the findings of the feature selection step and the corresponding discussions accompanied by an observation of physical interpretations of remaining chosen features. Lastly, section 5 provides several concluding remarks on the basis of the obtained results.

2. Case study

This work utilizes publicly available data from the Building Data Genome Project 2 [17]. The current version of the database consists of data collected from 1,636 non-residential buildings in hourly frequency for 2016 and 2017. This database is created utilizing data obtained from a variety of meters: electricity, steam, chilled water, hot water, water, irrigation, solar, and gas. The hourly format provides a suitable granularity while making analytical techniques based on weekly, monthly and annual trends possible. Each meter's dataset is accompanied by a metadata that includes the area of the building and its primary use type, and the corresponding local weather file [17]. Figs. 1 and 2 show the consumption profile for electricity and chilled water meters in a sample building from the dataset.

2.1. Prediction targets

This study explores three classification criteria: building principal use type, performance class, and operation group. The first two are the same as those taken into account in Miller's work [18], while a more recent study [16] altered the last criterion to produce more evenly distributed courses.

- **Principal Building Use:** Principal function for which the building was originally designed. Initially, the dataset had 13 types of primary space usage, which were combined into seven categories, namely, education, office, residential, public assembly, public services, parking/industrial, and others. For this classification task, features representative of in-class specificity (that belong to the group) generated from Jmotif package were put aside in order to evade data leakage issue.
- **Performance Class:** Performance class is determined on the basis of per-area normalized consumption. Buildings are divided into three categories of low, medium and high consumption performance class. For this target, all features that are directly resultant of consumption are removed.
- **General Operation Strategy:** Differentiates between buildings from different sites since buildings from the same site are most probably operated with the same strategies. In order to prevent data leakage in this classification task, features that are indicative of sensitivity to weather conditions are removed from the data.

Table 1 shows the types of features that were generated for the classification tasks, as well as a brief explanation for each type. A more thorough explanation of each type with examples is given in section 3.

Finally, it should be noted that in order to make sure a machine learning model can perform well on real world data, the dataset needs

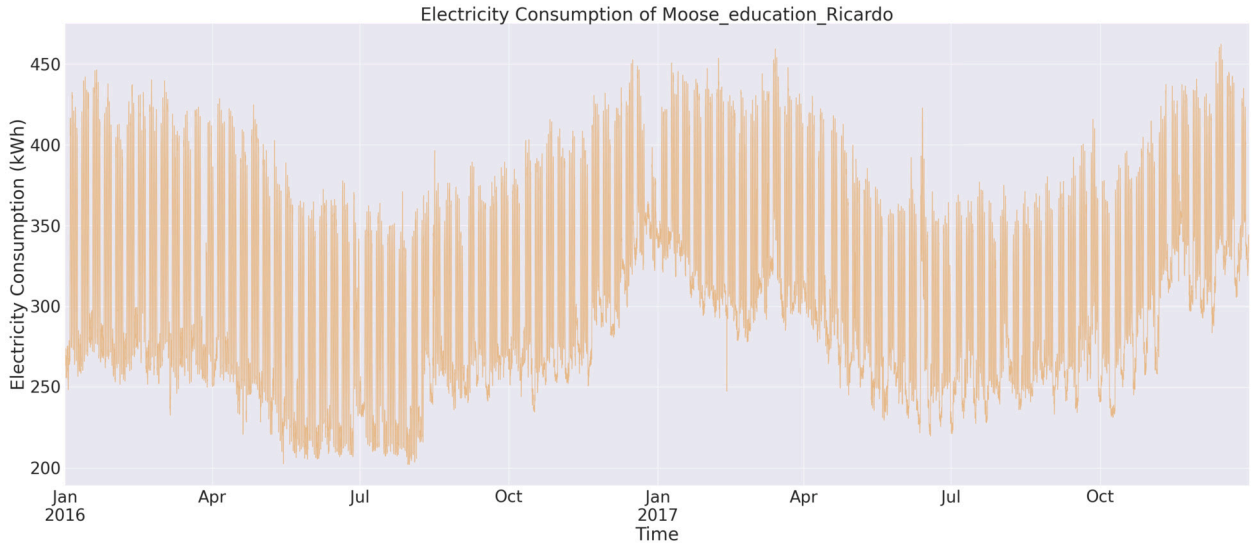


Fig. 1. An example of the electricity load profile for one of the buildings provided in the Genome Project 2 dataset.

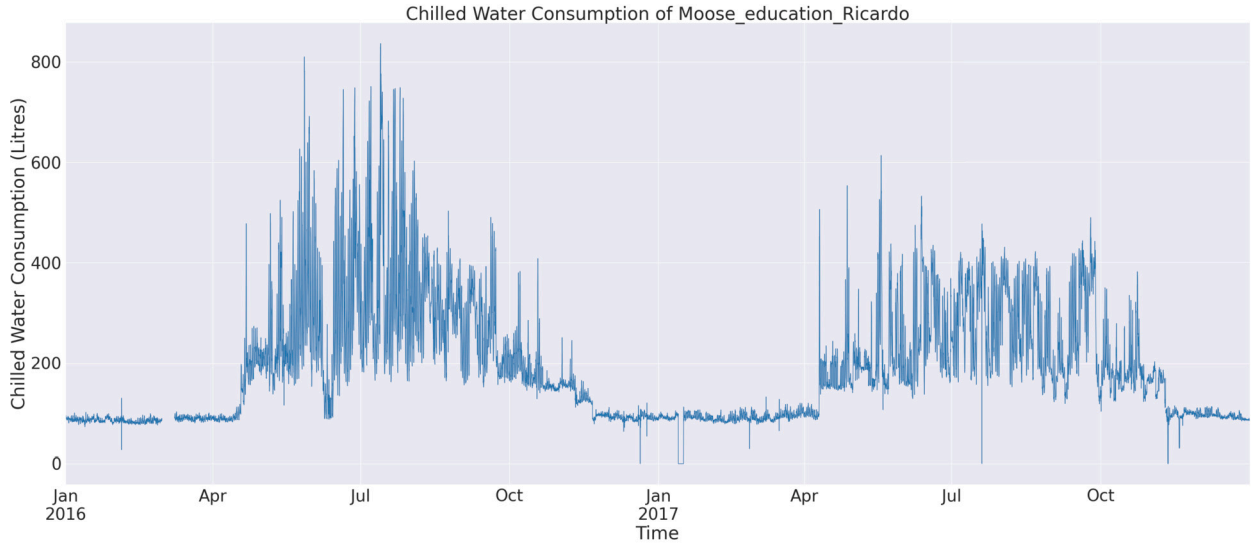


Fig. 2. An example of the chilled water load profile for one of the buildings provided in the Genome Project 2 dataset.

Table 1
Main types of extracted features used for building characterization.

Feature Category	Description
Statistics-based	Applying fundamental statistical functions to time series data, such as mean, median, minimum, maximum and standard deviation
Regression-based	Models developed for predicting, determining output parameters and attributes
Pattern-based	Extraction of recurring patterns on different time scales from the time series data

to be broken down into train, validation and test subsets. The training and validation subsets are used for training and assessing the model and tuning the hyper-parameters of the algorithms. The test subset is instead held out until final optimal pipelines have been identified and it is then used to assess their performance for unseen data. In this work, 20% of the data has been set aside as the test subset, while the remainder constitutes the training and validation subsets. A cross-validation procedure with 5 folds (which utilizes 20 percent of the remaining data as the training set in each fold) is then used to choose the training and validation sets iteratively in order to avoid over-fitting.

2.2. Prediction subsets

Two subsets are considered as follows:

- **Subset A**: All buildings for which electrical consumption data is available. This includes 1494 buildings for which the feature generation procedure resulted in 374 features.
- **Subset B**: Buildings with both electrical and chilled water consumption data. It involves 748 features generated from the electrical and chilled water consumption data of 374 buildings.

3. Overall methodology

The overall methodology followed in the present work is illustrated in Fig. 3 in a simplified manner, demonstrating the procedures taken, as well as the steps within each procedure. In the subsequent sections, a detailed explanation of each step is provided.

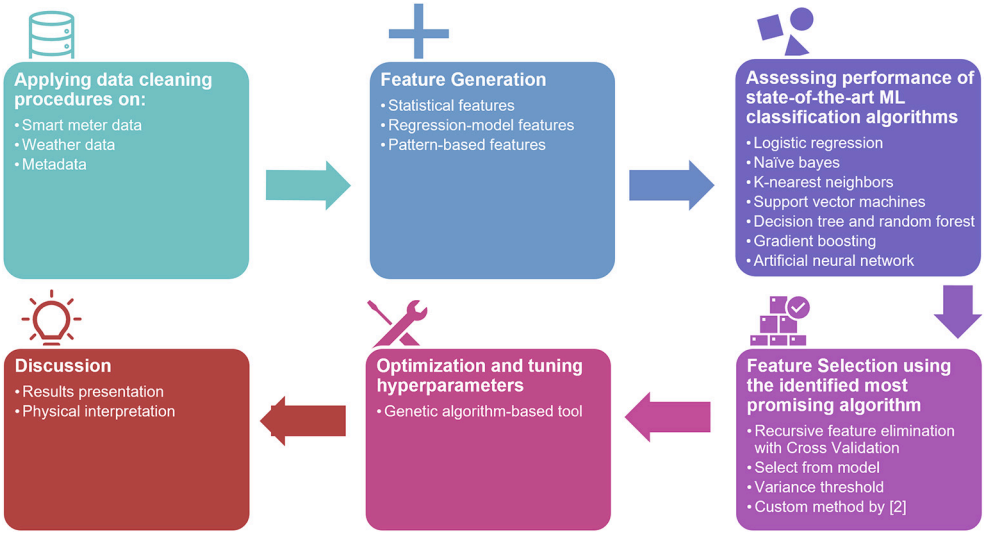


Fig. 3. Visual representation of the methodology of this work.

3.1. Data pre-processing

3.1.1. Meter data cleaning

For large datasets, it is commonly the case that data was gathered from a bevy of different sources. This can make handling the dataset more difficult as there could be many inconsistencies in the dataset regarding semantics, referencing systems, accuracy of measurements, frequency of measurements and availability of data from each source [19]. The building genome 2 dataset was gathered from 19 sites across the globe, making it critical to perform a thorough data cleaning and pre-processing step before using the data for feature generation. A few issues were observed in the data that need to be dealt with, namely:

– A group of buildings, of about 135, have mostly zero electricity consumption up to mid-march of 2016, then they start to show reasonable data points.

– Majority of buildings demonstrate false data encapsulated within many zero-consumption recordings. These happen at the beginning of recording period and are usually intertwined with the previous point.

– Due to faulty instruments, extraordinary incidents and/or simply errors in recording data, there are NAN (Not a Number) values in the dataset, which need to be corrected.

– Following the same logic, there are also sharp peaks and drops within the consumption profiles that should be trimmed away as they are simply outliers.

– In order to correctly train the models, there needs to be a minimum number of data points; therefore, the buildings which have less data points than this threshold value should be removed.

Besides these issues, we need to set a standard of quality for the data that we have in order to see which buildings we can be used and which ones are of sub-par quality. Thus, two parameters were defined: max-nans (maximum share of NAN values in the building data) and max-zero (maximum share of zeros in the building data). These two parameters were set to be 0.05 and 0.1 respectively. To illustrate the benefit of data cleaning, before- and after-cleaning plots for a sample building (id: Panther-education-Jerome), are given in Fig. 4.

This procedure was applied to the data from all meters. After the cleaning procedure, 1494 buildings remain for which electricity meter data is available. This applies to 397 buildings with chilled water, 248 with steam, 59 with hot water, 59 with gas, 28 with water, a single building with solar and no buildings with irrigation data.

Since ML models work in a promising way while being provided with a large group of samples, a meter of desire would be one that encompasses data from a large variety of buildings. Consequently, electricity and chilled water meter data were chosen to be studied, since the number of buildings for which both electricity and chilled water data is available is 374, much greater than that for other possible combinations.

3.1.2. Weather data cleaning

The only shortcoming of the Genome 2 dataset compared to its predecessor is the weather data, where there are fewer features and some of these included features have more than half of their data points as null.

3.1.3. Metadata cleaning

The metadata file contains a large array of characteristics for each building, among which the most important parameter that requires attention is the primary space usage. Initially, there are 13 groups of building usage types. This high-resolution breakdown of building usage types causes some usage groups to have very few samples compared to the more prominent groups (such as education and office building). This, in turn, would inflict damage on the models while performing the classification tasks. Therefore, usage types were narrowed down to education, office, lodging/residential, entertainment/public assembly, industrial/parking, and other. This regrouping process is demonstrated in Fig. 5.

3.2. Feature generation

3.2.1. Statistical features

In this sub-section, a brief description of the generated features that have been proposed by Miller [14] for building classification employing smart meter data is provided.

Basic Temporal Statistics: A large number of features can be extracted from time-series data according to basic statistical information such as mean, min, max, standard deviation, and variance (standard deviation squared). Mean and variance are computed using equations 1 and 2:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1}$$

$$\sigma^2 = \frac{\sum_{i=1}^{N}(x_i - \mu)^2}{n} \tag{2}$$

These statistical features are derived for different time ranges, such as daily, monthly, summer and winter (seasonal) and annual. Most of these features were meant to be extracted from the VISDOM package in R language, but due to incompatibility issues, the authors developed the same functionality in Python. A group of metrics are determined to look

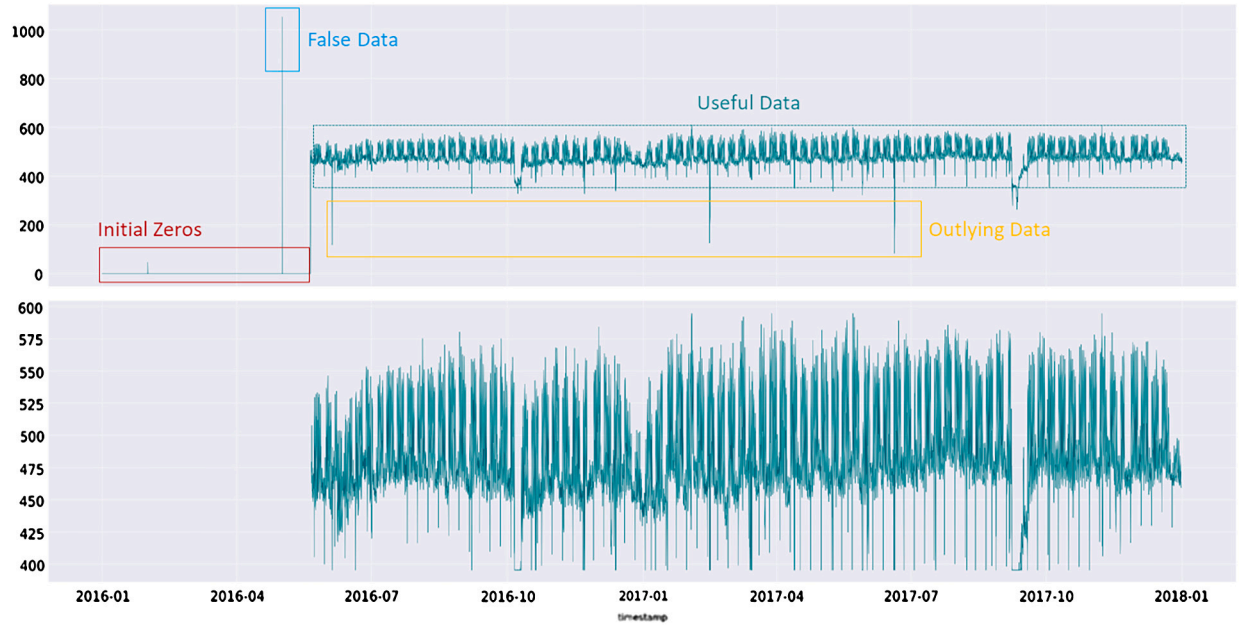


Fig. 4. Effect of Data Cleaning: Upper) Raw Data and Lower) Cleaned Data.

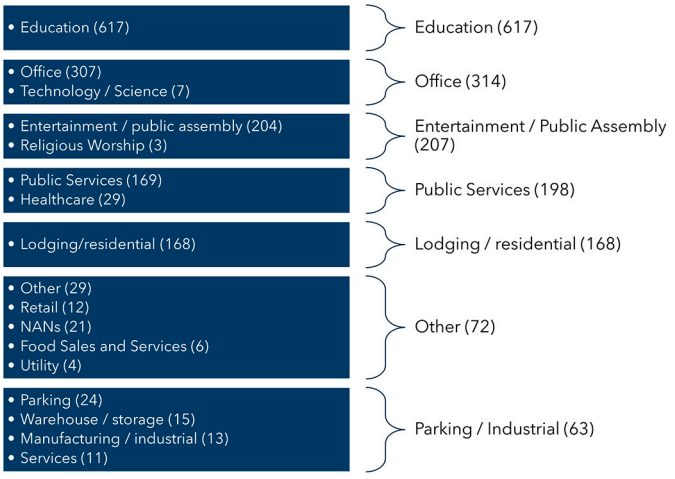


Fig. 5. Original Use types available in the dataset and their combinations that derive the use types in the current work.

into buildings' behavior at every hour. The majority of these features simply show the mean demand at each hour. Also, a few features are employed to find the hours of peak consumption in the 10% hottest days and the 10% coldest days of the year. Finally, a group of metrics are developed exclusively for January and August due to the greater cooling and heating demands in these months. [14]

Ratio Based Statistics: In these metrics, ratios are derived from different combinations of the indicators calculated in the previous section. The rationale behind this approach is that the features are being in a way normalized, which makes comparisons between buildings easier and more accurate. The most prominent of these features are (1) normalized consumption per area and (2) the daily load ratio, which is the ratio of minimum to maximum consumption on a daily basis. [14]

Spearman Rank Order Correlation Coefficient: The Spearman rank order coefficient, ranging between -1 and +1, reflects the degree of connection between a building's consumption and the weather. A cooling sensitive building is the one for which there is a strong positive association between consumption and temperature. Similarly, in case of substantially negative association of the latter parameters, the building is heating sensitive. [14]

3.2.2. *Regression-model features*

By using the output of performance prediction models, one can extract a building's quasi-physical behavior. Several common prediction models, mostly specific to electricity consumption, were used to create this subset of features.

Load-shape Features: In order to estimate consumption, identify anomalies, and assess the impact of demand response, the domain of electrical load prediction based on form and trends observed in electrical loads was established. Using the cooling and heating-degree days to normalize monthly consumption is the most common method in this category. Several methods have been developed, including neural networks, ARIMA models, and others. In contrast to more modern and sophisticated methods, here, simple methods have applied owing to their ease of use and simplicity. A regression model offers several measures that show how effectively a meter adheres to common assumptions for the production of temporal features. For instance, if measurements and projections agree well, the underlying behavior of the building's energetic systems is accurately documented. If not, a phenomenon that is not yet understood needs to be captured using a different model or feature. In this study, a modern, condensed load prediction technique is chosen to generate temporal features that determine whether the electrical measurement is merely a function of scheduling for the day of the week. The aforementioned model was created by Matthieu et al., and it was primarily used to evaluate electrical demand response [20] [14].

Change-Point Model Regression: A modeling approach that accounts for weather characteristics and its impact on the performance. It is possible to approximate how much energy is utilized for HVAC purposes by interpreting the outcomes of these models. This type of model is an offspring of PRISM method and has been widely utilized. This model is developed using daily consumption and outdoor dry-bulb air temperature in a multivariate, piece-wise manner. In the present work, the Open Meter Python library is used to develop these models [21] [14].

Trend Decomposition and Seasonality: Regardless of source of data collection, time series data tend to demonstrate similar behavior. As a consequence, we can apply the same feature extraction techniques employed in the social sciences or finance for the data at hand. The common aim between all these techniques is decomposing the data into several basic components which capture the data's true nature [22]. As an instance, building electricity data, collected through smart meters, usually exhibits cycles in a weekly time scale. The way buildings

are utilized by their inhabitants on a weekly basis has a relatively predictable pattern. A very common example is the way occupants come into work on weekdays at a certain hour and leave the office for home at a set time. Weekends are unoccupied periods in which there is practically no activity.

Another element that is typical of temporal data is trends. Increases or decreases in consumption over the long term that commonly do not adhere to any specific pattern are referred to as trends. Compared to seasonality, trends are typically caused by less predictable variables, and they frequently result from outside influences. Trends in building energy use appear as slow changes in consumption, taking place gradually from a few weeks to a few months. Changes in occupancy and user behavior, as well as HVAC system deterioration, are typically the causes of trends.

The seasonal-trend decomposition method is utilized to capture these aspects. The model works by first aggregating daily input data, and then subtracting the cooling and heating constituents derived from change point models in order to weather-normalize the data. This procedure is carried out to dampen the effect of weather conditions in trend decomposition. The seasonal, trend, and irregular components are extracted via the STL package originally developed for the R programming language [23] [14].

3.2.3. Pattern-based features

The goal of these features is to assign values to each building's consistency in usage over different time windows as well as analyzing whether or not certain building types exhibit behaviors that could be used to help predict corresponding metadata. Motifs and discords are the two most important concepts in temporal feature mining. A motif is a typical pattern that takes place with a regular frequency [24]. On the other hand, a discord is an unusual pattern within a dataset that identifies infrequent behavior. [25] [14]

Diurnal Pattern Extraction: The Dayfilter procedure uses 24 hour sliding window periods to extract motifs and discords from raw meter data. The Symbolic Aggregate Approximation (SAX) time-series data representation is used in this approach [26]. Time-series data are discretized by the SAX process, which changes temporal data into string type data. This methodology is utilized by numerous text mining and visualization methods. Diurnal pattern frequency, which measures the quantity of motifs obtained from a specific meter, is the major feature recovered by this technique in the current work. [14]

Pattern Specificity: SAX could be used to identify the patterns that best characterize each building use type. Such information is gathered with SAX-VSM process developed by Senin and Malinchik, originally developed for text mining. Pattern specificity is a measure of how well a building matches its peers of the same use type [27] [14].

Long-term Pattern Consistency: This concept is based on how unstable a building's electrical usage is over a long time period, such as a year. If a building experiences large changes in steady-state performance throughout the course of its data, it is referred to as volatile. This work quantifies the difference between these behaviors using a notion developed by Miller, known as breakout detection [28]. To process their time-series data, Twitter created a R programming tool. In a research by James et al., the specifics of this package are described [29].

3.3. Classification machine learning algorithms

Machine Learning is the collection of numerous algorithms used to enable a computer to find patterns within data [30]. Among different categories of machine learning problems, the present work falls under the supervised learning [31], in which the estimation target is available (data is labeled). Furthermore, as the estimation targets are categorical variables, within the supervised learning category, it is considered a classification problem, in which ML algorithms make a prediction based on the likeliness of newly input data to fall within one of the established categories [32].

Logistic Regression: A supervised ML algorithm used for classification problems is one of two types. Logistic regression employs a logistic function as shown in equation (3). Instead of outputs being either 0 or 1, logistic regression yields results within the range, having a probability as the output. A well-optimized implementation of logistic regression is included in Scikit-learn and supports multi-class classification tasks [33]. The presupposition that the dependent and independent variables have a linear relationship is the main drawback of logistic regression.

$$Logistic\,function = \frac{1}{1+e^{-}x} \tag{3}$$

Naive Bayes: Based on Bayes' Theorem, Naive Bayes evaluates the conditional probability based on prior knowledge and the simplifying assumption that each feature is independent from others. Bayes' theorem puts forward the following relationship for a class variable y and dependent feature vector x_1 through x_n:

$$P(y|x_1,...,x_n) = \frac{P(y)P(x_1,...,x_n|y)}{P(x_1,...,x_n)} \tag{4}$$

Despite the independence assumption being a strong one, naive Bayes classifiers are beneficial in a variety of situations, amongst them spam filtering and document classification. They only require small amounts of training data to estimate the required parameters. In comparison to more complex techniques, naive Bayes classifiers can be rather quick [34].

K-Nearest Neighbors (KNN): This approach works by locating a user-defined number of training datapoints that are physically closest to the new data point and then predicting the class or label based on those samples. The number of samples can be fixed (k-nearest neighbors) or can alter depending on the density of nearby points. The distance can be freely chosen. The most common choice is standard Euclidean distance. Due to the fact that neighbors based methods keep all of the training data, they are referred to as non-generalizing ML methods. Nearest neighbors works well in a variety of classification and regression problems, including recommendation systems. Since it is non-parametric, it frequently works in classification scenarios where the decision boundary is highly erratic [35].

Support Vector Machines (SVM): In this method, data points are taken to be n-dimensional vectors, and it is desired to determine whether one can separate such points with a (n-1)(n-1)-dimensional hyperplane. There exist many hyper-planes capable of classifying the data. A reasonable choice is one that results in the largest distance between the two classes [36]. SVM is an accurate algorithm with low certainty, therefore requiring cross validation to tune hyper-parameters. It is widely used in cancer research and handwriting classification [37].

Decision Trees and Random Forests (RF): Decision Trees are a non-parametric supervised learning method applied both for classification and regression tasks. By inferring straightforward decision rules resulting from the data features, the goal is to develop a model that predicts the value of a target. In this method, a tree is considered a piece-wise approximation [38], or more simply a flowchart, separating data points into two similar categories at each step from the tree's base or trunk to its extrema, also known as leaves, where the categories become more similar. Random forests use the concept of collective intelligence by building a group of decision trees independent from one another. Each decision tree is a simple predictor, but the outcomes are aggregated into one. This should be closer to the actual outcome in theory. Random forests have the drawback of being more challenging to interpret compared to a single tree. Additionally, they take longer to construct since the construction and evaluation of each tree in a random forest is performed independently [39]. Considering $T_i(x)$ a single tree constructed on the basis of a subset of input features [40] and the bootstrapped samples [41], the tree can be mathematically expressed as:

$$\hat{f}_{RF}^{C}(\mathbf{x}) = \frac{1}{C}\sum_{i=1}^{T} T_i(\mathbf{x}) \tag{5}$$

in which C is the tree count and x is the input variable in vector form [41].

In this study, a Random Forest classification algorithm is used as the benchmark model for classification tasks. It's performance is evaluated using average scores of a 5-fold cross validation as well as directly using a 20% training set to produce confusion matrices and other indicators. In order to make sure that the entire dataset is taken into account when assessing the effect of adding or removing features, cross validation scores are primarily used throughout the feature selection step.

Gradient Boosting: Simple model averages are the basis of common ensemble approaches such as random forests. The class of boosting methods is founded upon a different, constructive strategy. In this context, the term "boosting" refers to the sequential addition of new models to the ensemble. Taking into consideration the error of the entire ensemble that has been learned so far, a new weak, base-learner model is trained at each iteration. Gradient boosting machines, or GBMs, through iteration fit new models during the learning process to produce a more accurate estimation of the target variable. The main concept is to configure the new base-learners to be maximally correlated with the associated ensemble's loss function's negative gradient. The loss functions used can be chosen at will [42]. [43]

Artificial Neural Networks (ANN): These models mimic the learning process of the human brain, and are specially suited for non-linear problems, each neural network is composed of nodes, where given an input to the node, a function is applied to the input, resulting an output which is subsequently communicated to the next nodes via connecting links and through applied weights. Every ANN is composed of input, output and hidden layers, where the input and output layers are composed of the features and targets of the problem. The hidden layer can be structured with varying depth (layer count) and width (neuron count within layer). Neural Networks have major functionality in pattern recognition such as image processing applications. [44]

3.4. Feature selection

Feature selection is a strategy to select the most significant subset of features [45] for the development of a robust ML model. In this process, redundant features, those with high correlations amongst themselves, and those not contributing to the model's outcome improvement are removed, thus decreasing the computational cost of the model and increasing the efficiency. In general, four steps are taken in this process: (1) subset generation; (2) subset evaluation; (3) stopping criteria; and (4) validation. Subsets are chosen in step 1 based on the search strategy. In general, approach depends on the nature and method of the search. Step 2 is influenced by a number of evaluation factors, including distance, dependency, and consistency. Step 3 establishes the stopping criteria: once the error is smaller than the imposed or chosen value, the search must be finished. Step 4 validates the selected attributes using a variety of cutting-edge AI/ML algorithms [46].

One can divide the different types of feature selection into two main categories of wrapper and filter methods. These methods are most typically supervised and their performance is evaluated based on the results from an out-standing subset of data being fed into the model. Wrapper feature selection methods construct a large number of models using different subsets of input features, then select the features that produce the model with the highest performance score based on a particular metric. Although these methods can be computationally expensive, they are not concerned with the variable types. Wrapper methods compare several models using steps that add or remove predictors in order to find the ideal mixture that will maximize model performance [47].

Filter methods employ statistical techniques to evaluate the relationships among pairs of input and target variable, then these scores are used as grounds for selecting which input variables to use in the model. A third group of feature selection methods can be discussed that are known as intrinsic. This takes place automatically as part of the models training [48]. The feature selection algorithms used in this work are listed below [49]:

Table 2
Coefficients utilized in the Custom method for feature selection.

	Electricity Only	Electricity + CW
Use Type	RF Importance	RF Importance
Performance Class	Mutual Information	RF Importance
Operation Group	RF Importance	Mutual Information

Recursive Feature Elimination with Cross-Validation (RFECV): Selects features by continuously choosing more and more compact subsets of features, utilizing an external estimator that assigns weights to features. The importance of each feature is determined through any particular attribute after the estimator has first been trained on the initial, complete set of features [50]. The least crucial features are then removed from the list of features. Once the desired number of features to select has been reached, the procedure is iteratively replicated on the final set. To determine the ideal number of features, RFECV applies RFE in a cross-validation loop. [51]

Univariate Selection: A method for selecting the best feature subset based on univariate statistical tests. It can be viewed as a pre-processing step to an estimator. Approaches enclosed within this method include:

- *SelectKBest* keeps only a number of features with the highest scores, specified as k by the user;
- *SelectPercentile* keeps only a percentage of highest scoring features, specified by the user;
- Typical statistical tests such as *SelectFpr*, *SelectFdr* and *SelectFwe* respectively used for false positive rate, false discovery rate and family wise error;
- *GenericUnivariateSelect* chooses the highest-performing strategy combined with hyper-parameter search among the previously mentioned methods.

These objects take inputs in the form of a scoring function which can vary depending on the nature of the ML problem. These include: $f\_classif$, ANOVA F-values, $chi2$, etc.

Select from Model: A meta-transformer that can be used with any estimator that assigns each feature a certain amount of weight via an attribute or through a callable importance-getter after fitting. If the corresponding importance of the feature values falls below the specified threshold parameter, the features are removed as being unimportant. There are built-in arguments for finding a threshold using a string argument in addition to specifying the threshold numerically [16,52].

Variance Threshold: A simplistic strategy of feature selection. It works by removing all those features whose variance falls below a certain threshold. By default, it eliminates all zero-variance or constant value features. [49]

Custom Feature Selection Method proposed by [16]: In this method, first, mutual information coefficients are used to sort the features. Next, beginning with the features with the highest correlation, the loop adds a new feature to the set, only in the case that it leads to an improvement either in accuracy or the F1 score. The use of cross validation ensures that the selected feature is important for the entire dataset and not only for the selected portion where testing is being performed. Different correlation coefficients are used: RF feature importance, Pearson correlation, permutation importance and mutual information. The coefficients chosen for the final pipeline in each approach and for each target are brought in Table 2.

3.5. Pipeline optimization

The algorithm optimization step has been performed by employing the tree-based Pipeline Optimization Tool (TPOT) [53], which is an auto-ML tool that helps finding the optimal pipeline for an ML task [54].

It is built on the Scikit learn library and follows its API closely. TPOT uses a genetic search algorithm [55] to fine-tune hyper-parameters and model ensembles [56] by trying a pipeline, evaluating its performance, and randomly changing parts of the pipeline in search of better algorithms [53].

3.6. Model evaluation and metrics

3.6.1. Metrics

There are multiple metrics used to assess the performance of an ML model. Those used in this work are brought below with corresponding relationships.

Mean Absolute Error (MAE): A measure of error that is determined by averaging absolute errors. For n examples, for each value y and its prediction $\hat{y}$, MAE is defined as:

$$\frac{\sum_{i=1}^{D} |y_i - \hat{y}|}{n} \tag{6}$$

One downside of MAE, as observed in this work, is that if the true value to be predicted is zero, the error will become infinity which is non-ideal in feature selection and running the model [57].

Root square of Mean Squared Error (RMSE): Another important metric, defined as follows:

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2} \tag{7}$$

Accuracy: When computing accuracy in multi-class classification, accuracy is simply the fraction of correct classifications

$$Accuracy = \frac{CorrectClassifications}{AllClassifications} \tag{8}$$

Precision: Attempts to determine what percentage of positive identifications were accurate. The following is a definition of precision: [58]

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

Recall: Attempts to respond to the question: What percentage of actual positives were correctly identified? Mathematically, recall is defined as follows [58]

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

F1 Score: The F1 score is a harmonic mean of precision and recall, with the best and worst values corresponding to 1 and 0 respectively. Precision and recall both contribute equally in terms of percentage to the F1 score. F1 score is defined as:

$$F1 = \frac{2(Precision \times Recall)}{Precision + Recall} \tag{11}$$

In the multi-class and multi-label case, the above formula would be the average of the F1 score of each class with weighting depending on the average parameter. In this way, F1 score can be a measure of the models performance. [59]

Cross-Validation: A resampling technique that tests and trains a model on various iterations using various subsets of the data. Cross-validation is commonplace in prediction tasks, where one seeks to gauge how accurately a predictive model will function in real-world scenarios. In a prediction problem, a model is usually trained on a set of known datapoints (training dataset), and tested against an unknown subset of the data.

Averages: Since there are multiple classes for the model to predict, each class is assigned a separate score following the definitions explained previously. In the end however, only one result is desired from the entire model which is why three averaging methods are considered here. Macro Averaging is the most straightforward method, simply calculating the arithmetic mean of all classes with no assigned weights. Weighted average, as the name suggests, is calculated by assigning a weight to each class based on the number of its members. Finally, Micro Average is determined as follows:

$$\frac{TP}{TP + 0.5 * (FP + FN)} \tag{12}$$

Upon closer inspection, the reader can recognize that this is the same as accuracy, found for the entire database [60].

3.6.2. Correlation coefficients

Spearman Correlation: A non-parametric way to measure rank correlation is with Spearman's rank correlation coefficient. Unlike Pearson's correlation which searches for linear relationships, Spearman's correlation assesses how well a relationship between two variables can be described by a monotonic function (which may or may not be linear). Each variable must be a perfect monotone function of the other to have a perfect Spearman correlation of +1 or -1. If all n ranks are distinct integers, it can be computed using eq. (13) [61]:

$$r_s = 1 - \frac{6\sum d_i^2}{n(n^2 - 1)} \tag{13}$$

RF Feature Importance: After a Random Forest model is fitted on training data, it is possible to retrieve its calculated coefficients using the $feature\_importances\_$ attribute. Such coefficients are representative of how well a particular variable can predict the targets and allows the user to understand which features are most representative of each objective.

Feature importance is calculated using equation (14). Summation of the importances of the j-th nodes ni_j on which X_i is split, divided by all nodes' importances, and then averaged over all T trees in the forest yields the importance of input feature X_i for predicting Y [62].

$$Imp(X_i) = \frac{1}{T}\sum_{t \in allTrees}\frac{\sum_{j \in nodeSplitOnX_i} ni_j}{\sum_{k \in allNodes} ni_k} \tag{14}$$

4. Results and discussions

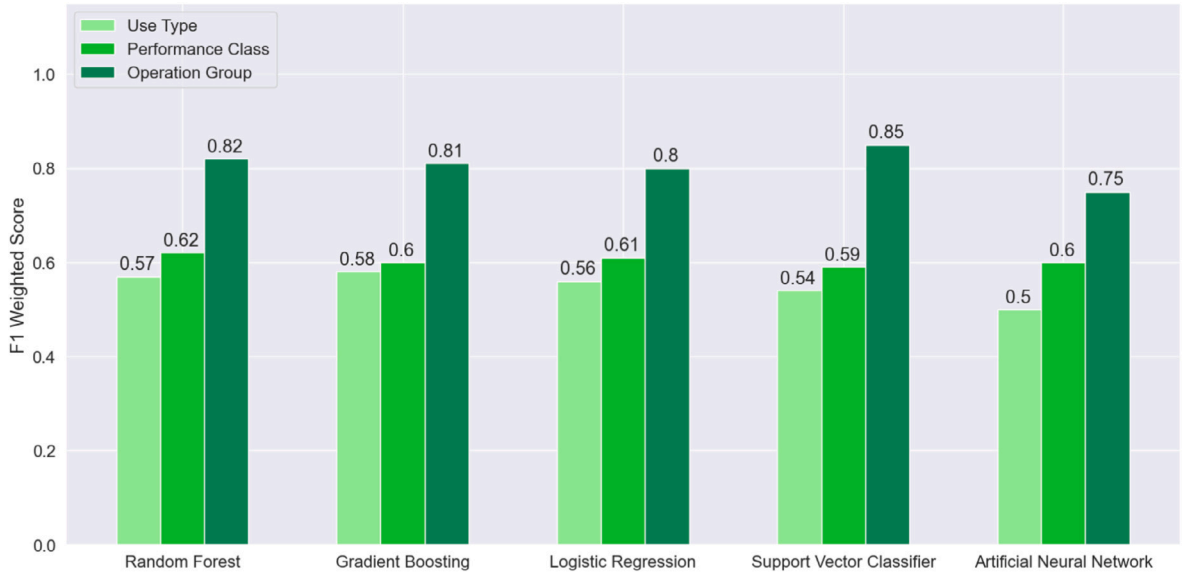
In this section, the considered subsets of the dataset (set of buildings with available electrical consumption data along with those for which both electrical and chilled water data are available) are first presented. Next, a comparison is made in order to identify the most promising state-of-the-art ML algorithm, for both subsets, to be utilized as the benchmark model. Subsequently, for each subset, feature selection results are presented and discussed. The most suitable pipeline, identified in the pipeline optimization step, is then presented and the corresponding performance is compared with the one offered by the benchmark algorithm. Finally, physical interpretations of the selected features are sought and discussed.

4.1. Considered subsets and the performance of state-of-the-art ML algorithms

As was previously pointed out, pipelines are implemented and optimized while employing A) the features generated from electrical consumption data and B) those extracted from a combination of both electrical and chilled water demand data. Accordingly, two different subsets of the dataset are considered: subset A includes all the buildings for which electrical consumption data is available; Subset B instead involves the buildings with both electrical and CW consumption data. After performing the data cleaning step, subset A includes 1494 buildings, for which the feature generation procedure results in 374 features. Subset B instead involves 748 features generated from the electrical and CW consumption data of 374 buildings (as both consumption data are available for fewer buildings).

As the first step, the performance of pipelines developed employing a selection of commonly utilized (state-of-the-art) ML algorithms (de-

(a) obtained performance of ML algorithms for subset A



(b) Obtained performance of ML algorithms for subset B

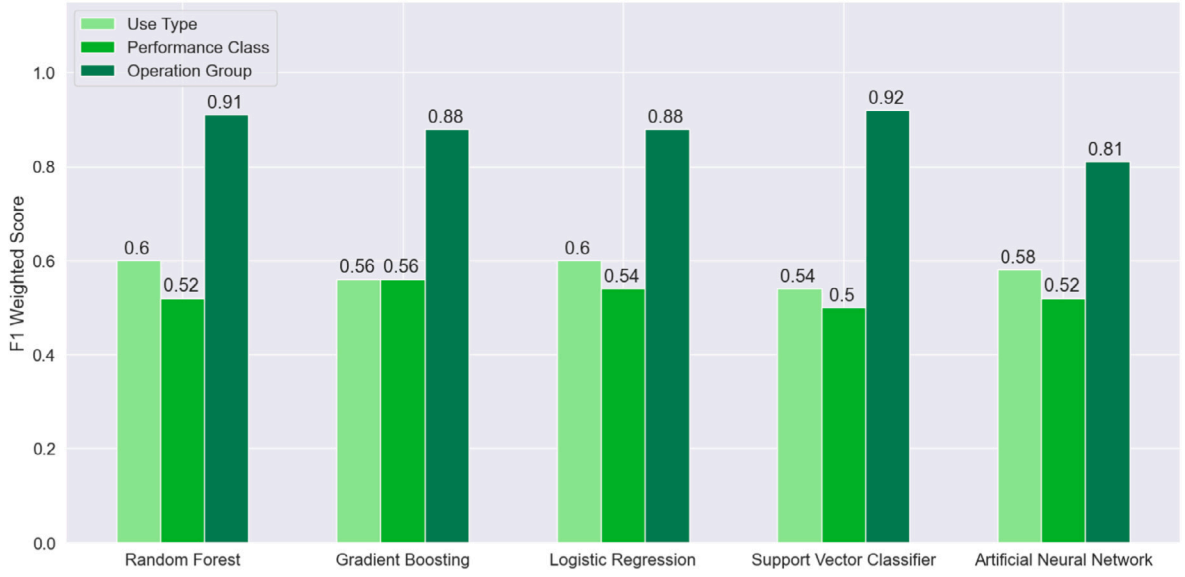


Fig. 6. Weighted F1 score of different ML classification methods for a) subset of data utilizing only electricity-based features (Subset A), and b) subset of data utilizing both Chilled water and electricity-based features (Subset B).

scribed in section 3) is investigated in order to identify the most promising classifier and obtain a benchmark accuracy. Among the initially considered set of classification algorithms, Naive Bayes and K-Nearest classifiers performed poorly compared to the rest of the classifiers and were thus not brought in subsequent figures. Accordingly, the accuracy and F1-weighted score (both of which are determined employing a 5-fold cross-validation procedure) obtained employing Random Forest (RF), Gradient Boosting (GB), Logistic Regression (LR), Support Vector Classifier (SVC) and Artificial Neural Networks (ANN), while providing all extracted features for subsets A and B, are assessed. The obtained results reported in Fig. 6, demonstrate that the Random Forest (RF) classifier performs consistently well across all targets and over both subsets. Therefore, this classifier is chosen as the benchmark algorithm to create a baseline model for both subsets. In addition, the RF classifier is also employed as the model in the implemented feature selection procedures.

4.2. Pipelines with features generated from only electrical consumption data (subset A)

4.2.1. Feature selection and algorithm optimization results

While considering subset A and employing the features generated from the electrical consumption data, different feature selection procedures (using RF as the model) are implemented and the resulting performance, in terms of accuracy and F1 score along with the number of selected features, is compared. As illustrated in Fig. 7, for building use classification, the custom method proposed by Najafi et al. [16] achieves an elevated accuracy and F1 score while reducing the required features by 95%. For the performance class and operation group classification instead, the recursive feature elimination (RFE) method offers a slightly better performance compared to the custom method, but it chooses a higher number of features (leading to an increase in the model's complexity and computational cost). Accordingly, the custom method [16] has been chosen as the most suitable feature selection al-

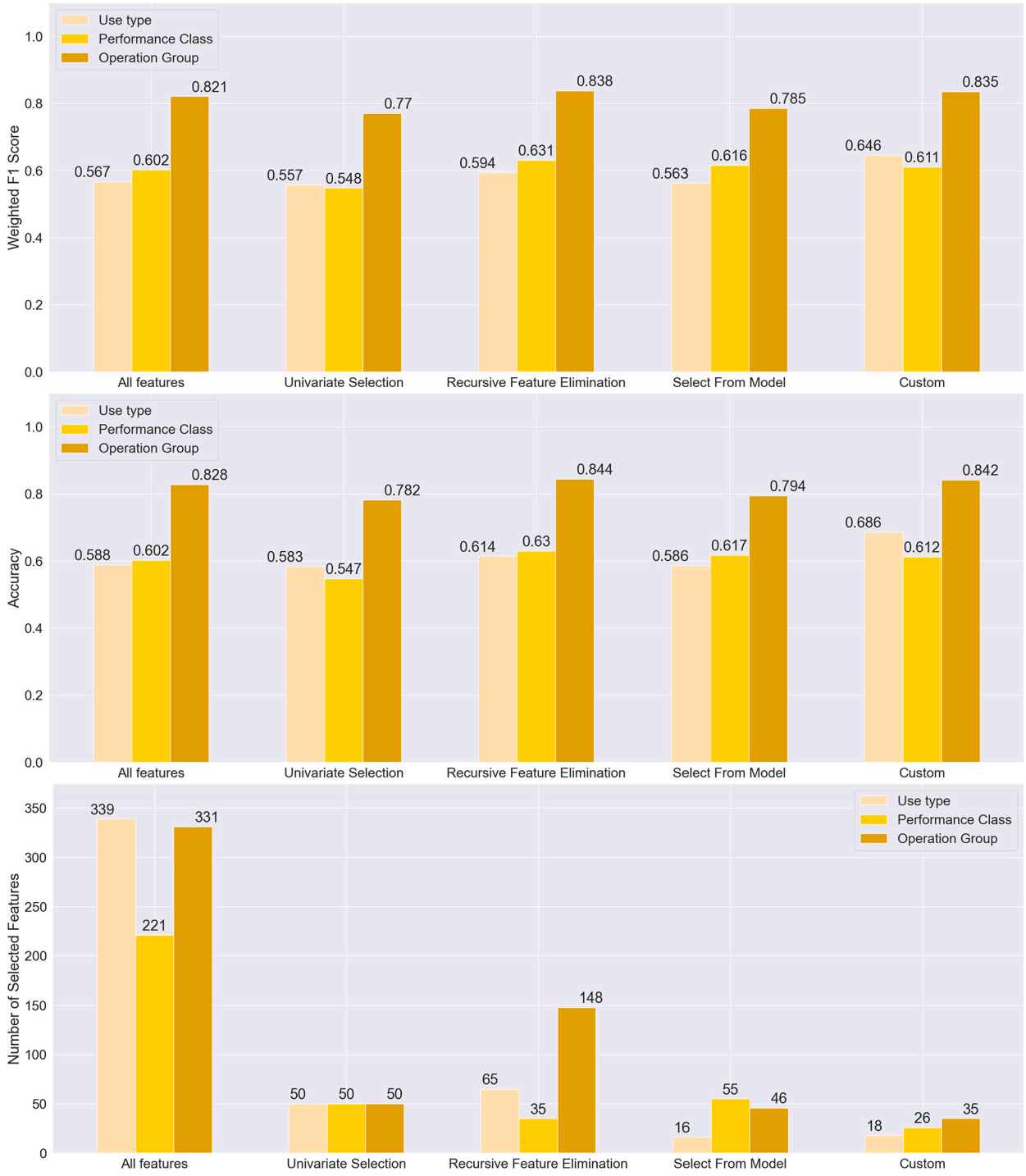


Fig. 7. Comparison of feature selection methods and classification targets.

gorithm for all classification targets and the resulting selected features are considered in the next step.

It is thus observed that performing the selected custom feature selection method [16], compared to the initial pipeline with all features, notably reduces the number of employed features (from 339 to 18 for the use type classification, from 221 to 26 for performance class estimation, and from 331 to 35 for estimating the operation group), while it even marginally improves the achieved accuracy (and weighted F1 score) for all of the considered targets (as also reported in Table 3). The resulting notable reduction not only decreases the model's complexity (dimensionality) and computational cost, but it also facilitates the physical interpretation of the selected feature (discussed in the next sub-section). It is noteworthy that the employed custom feature selection method [16] utilizes Mutual Information Coefficients for use type and operating group classification, while applying Random Forest Importance coefficients for performance class estimation.

In order to assess the performance of the identified pipeline for an unseen dataset (which has not been used in the feature selection procedure), the corresponding estimation performance over the test set has also been investigated. Table 3 represents the obtained accuracy over both the validation (determined using 5-fold cross-validation over the training set) and the test sets. It is observed that the RF model fed with selected features not only slightly improves the validation accuracy (and weighted F1 score) but also does not result in any significant

Table 3
Validation (Val.) and Test Weighted F1 Score as well as Accuracy for Pipelines studied in Subset A.

Target	Random forest-based pipeline with Electricity-based features				Random forest-based pipeline with selected electricity-based features				Optimal pipeline with selected electricity-based features			
	F1 Score		Accuracy		F1 Score		Accuracy		F1 Score		Accuracy	
	Val.	Test	Val.	Test	Val.	Test	Val.	Test	Val.	Test	Val.	Test
Use Case	0.567	0.666	0.588	0.652	0.597	0.644	0.617	0.632	0.597	0.644	0.617	0.632
Performance Class	0.602	0.604	0.602	0.605	0.611	0.616	0.612	0.615	0.620	0.622	0.621	0.622
Operation Group	0.821	0.780	0.828	0.773	0.835	0.800	0.842	0.793	0.852	0.842	0.860	0.819

decrement in the obtained performance over the test set (it has resulted in a marginal reduction for the use case estimation while it has even slightly improved the accuracy obtained for the performance class and operation group estimation). Accordingly, it is demonstrated that performing the feature selection procedure has not resulted in over-fitting to the training data and the selected features have been demonstrated to be a promising subset also for previously unseen (in the feature selection process) data.

In the next step, while employing the selected features, the pipeline optimization step is performed using a GA-based tool [63], in which (for each estimation target) the most promising pre-processing step(s), ML algorithm, and the corresponding tuning parameters that result in the highest performance (validation weighted F1 score determined using 5-fold cross-validation over the training set) are identified. The performance of the identified optimal pipeline for each estimation target over the test set is also assessed. As demonstrated in Table 3, the previously utilized RF algorithm is still selected as the most promising algorithm for use type estimation. For performance class and operation group targets instead, other pipelines resulting in higher validation weighted F1 scores are identified. The details of these pipelines, both of which employ Gradient Boosting classifier with different tuning parameters and pre-processing steps, are reported in Appendix B (that includes Tables B.1, B.2, and B.3 for the pipelines, in which use type, performance class, and operation group are considered as the estimation targets respectively). Finally, these two identified optimal pipelines (as represented in Table 3) are also shown to be able to achieve higher performance (compared to the RF model with selected features) over the test set, which demonstrates that the algorithm optimization procedure has not resulted in over-fitting to the training dataset.

4.2.2. Physical interpretation

As was previously pointed out, a feature selection procedure that lowers the number of utilized features facilitates the physical interpretation of the obtained results. Accordingly, for each considered classification target, the contribution of each feature to the achieved accuracy is reported. Next, physical/logical reasoning that can explain the influence of each parameter (for each classification target) is provided.

Use Type: Table 4 presents the relative improvement and achieved F1 score and accuracy metrics, with the addition of each feature (among those chosen in the feature selection process suggested by [16]), in the order of selection. As is evident from Table 4, the majority of features relevant to classification based of the use type, belong to the STL and Visdom packages, which represent the seasonality and trend decomposition as well as a range of statistical operations over variable time-frames, respectively.

The majority of the features chosen for this classification target pertain to the magnitude and trends of consumption within the buildings. Evidently, industrial buildings would have a larger consumption compared to parkings, classrooms, and offices. On the other hand, it makes sense to utilize the trends in weekly, monthly, and seasonal windows to classify buildings. For instance, education buildings have much lower attendance during summer, which results in a seasonal decrease in their consumption, while office buildings experience this in a much shorter duration of holidays. Some of the features with the largest contribution to the achieved accuracy will be provided below together with a brief description of their definition as well as the logic behind their influence on the classification peformance.

consumpstats_maxHOD indicates the hour of the day at which the mean consumption of a building is at its peak. Buildings with different uses tend to have different periods of peak loads. As an example, office buildings typically experience peak load in the morning, while for schools it might fall on lunch break or class change time. Meanwhile, for residential buildings, the peak demand generally takes place in the evening when residents are using the facilities.

consumpstats_kw90: Represents the 90th percentile of each building's consumption, demonstrating sustained high consumption. End-consumers such as air conditioners could contribute to a higher sustained load in residential buildings, while industrial buildings typically include types of equipment with a notable consumption.

stlweeklypattern_thur_mean, *stlweeklypattern_sat_mean*, *stlweeklypattern_sun_mean*, *stlweeklypattern_fri_mean*: These features are derived from the stl package, and are good indicators of occupancy behavior within each building type. For instance, offices tend to be occupied during the weekdays, while residential buildings are mostly occupied during weekends and weeknights.

stats_monthlySlope_8: Indicates the variation between consumption in September and August. With the start of the academic year in September, this feature is very useful in differentiating academic buildings from other ones. Moreover, the start of the academic year induces changes in the consumption patterns within other types of buildings, which are captured by this and other features.

stats_min_day_pct: indicates the percentage of days, in which the temperature was lower than that of the day with minimum consumption. Such a feature is able to capture the efficacy of a building's HVAC system, therefore possibly be able to differentiate between an office and an entertainment venue. On the other hand, the variations pertaining to the requirements of temperature settings (e.g. commonly utilized setpoints) within different types of buildings.

Performance Class: Classification accuracy is fairly promising for buildings with either high or low consumption levels, whereas the ones in the intermediate class are often mis-classified. This behavior is due to the very similar distribution of values for the features for this target. Table 5 depicts the effect of adding each feature on boosting the accuracy and F1 score. Features that are directly related to consumption were excluded from this classification to prevent data leakage. Overall, the majority of features that were selected are representative of load diversity in the form of ratios and patterns. The majority of features were derived from Visdom and Jmotif packages, with the former providing general overview-level data and statistics and the latter providing information about the similarity and deviations of each building from others with the same functionality. The impact of this feature on the accuracy of buildings' classification by performance class can be attributed to the fact that industrial buildings' consumption is mostly high and relatively uniform from the latter perspective. Similar to the previous case, the most significant features are provided below:

weekdays_meanvs95_std: Represents the standard deviation of the ratio of 95th percentile of consumption against the mean. It captures the volatility of consumption during the working days and is useful in distinguishing offices from other buildings where consumption may vary heavily.

Table 4
Relative improvement and achieved Weighted F1 score and Accuracy, following the addition of each feature for Building Use Type - Subset A (Buildings with only electricity meter data).

Feature ID	Meter type	Feature type	Feature details	Time window	Stats operator	F1 Score		Accuracy	
						Relative improvement	Value till this feature	Relative improvement	Value till this feature
stlweeklypattern_sun_mean	El	STL	weekly pattern	Sunday	mean	18.74	18.74	26.11	26.11
consumpstats_maxHOD	El	Visdom	hour of day @ max cons.	–	–	16.75	35.49	8.45	34.56
consumpstats_kw90	El	Visdom	90th percentile of cons.	–	–	4.11	39.60	3.85	38.41
stlweeklypattern_thur_mean	El	STL	weekly pattern	Thursday	mean	4.22	43.82	5.44	43.85
stlweeklypattern_sat_mean	El	STL	weekly pattern	Saturday	mean	2.55	46.37	3.43	47.28
consumpstats_kw_total_2016_norm	El	Visdom	Annual normalized cons.	Annual	total	1.87	48.23	2.18	49.46
stlweeklypattern_tue_mean	El	STL	weekly pattern	Tuesday	mean	0.70	48.93	1.09	50.54
stats_monthlySlope_8	El	Visdom	Monthly ratio	Sep v Aug		2.11	51.04	1.92	52.47
stlweeklypattern_fri_mean	El	STL	weekly pattern	Friday	mean	0.18	51.22	0.67	53.14
consumpstats_kw_mean_annual	El	Visdom	Annual normalized cons.	Annual	mean	0.81	52.04	1.17	54.31
stltrend_mar_mean	El	STL	Trend	march	mean	1.14	53.18	0.84	55.15
weekend_meanvs95_std	El	Visdom	mean vs 95th percentile	weekend	std	0.45	53.63	0.50	55.65
seasonal_Jan_n2d	El	Visdom	night v day ratio	january	mean	0.03	53.66	0.25	55.90
consumpstats_t10kw	El	Visdom	cons. @ 10 percent cold days	entire data	mean	0.87	54.53	1.34	57.24
stats_min_day_pct	El	Visdom	percent time when temp is lower than @min cons.	entire data		1.69	56.22	1.09	58.33
stltrend_apr_mean	El	STL	Trend	april	mean	0.77	56.99	0.84	59.16
all_meanvs95_std	El	Visdom	mean vs 95th percentile	entire data	std	0.18	57.17	0.17	59.33
consumpstats_kw_Sep_norm	El	Visdom	Normalized monthly cons.	september	total	0.10	57.28	0.00	59.33
weekdays_minvs95_mean	El	Visdom	min vs 95th percentile	weekdays	mean	0.09	57.37	0.25	59.58
seasonal_Aug_n2d	El	Visdom	night v day ratio	august	mean	0.47	57.84	0.50	60.08
seasonal_Nov_n2d	El	Visdom	night v day ratio	novemeber	mean	0.52	58.35	0.17	60.25
stats_min_day_tout	El	Visdom	outside temp. @ min cons.	–	mean	-0.27	58.09	0.08	60.33
consumpstats_kw_Aug_norm	El	Visdom	Normalized monthly cons.	august	total	0.33	58.42	0.08	60.42
dayfilterfreq_3_2h_min	El	Dayfilter	Symbol count = 3	2 hr	min	0.51	58.93	0.33	60.75
consumpstats_daily_kw_min_var	El	Visdom	minimum daily cons.	daily	var	-0.01	58.92	0.17	60.92
dayfilterfreq_9_8h_min	El	Dayfilter	Symbol count = 9	8 hr	min	0.62	59.54	0.50	61.42
stlreminder_jun_mean	El	STL	Remainder	june	mean	0.13	59.67	0.08	61.51
weekdays_meanvsmax_min	El	Visdom	mean vs max cons.	weekdays	min	0.00	59.67	0.17	61.67

meanvs95_std: Same as the previous feature but for all days. It can distinguish between buildings with low (office buildings) and high (industrial or public buildings) variance in consumption.

loadshape_mape_interval_daytime: This feature calculates the mean absolute percentage error between the consumption profile of a single building and that of the average of all buildings, thus helping in identifying buildings with unique load profiles, whether due to deviations in activities from the norm, or due to malfunctions in the energy systems.

all_meanvsmax_std: A high consumption would result in a lessening effect on the variations in the usage. This feature, which evaluates the variance in the minimum daily consumption, is able to predict low/intermediate consuming buildings due to the fact that they would have higher variances in their consumption. The selection of (multiple) breakout features can be explained in the same way.

Operation Group: Regarding the last classification criterion, as presented in Table 6, features are selected across multiple packages, such as Day filter, Visdom, STL, and EE meter, all contributing information that is helpful in distinguishing buildings from different climates and operation strategies. The most influential features are *El_dayfilterfreq_3_2h_min* and *minId*. The former is a pattern-based feature. In fact, about half of the features selected (for this classification target) are pattern-based, which is attributable to these features' ability to differentiate between operating strategies over different time windows. The latter indicates the date with the least electrical consumption of the building. Buildings that have the same date for their minima are more likely to be from the same sites and these minima happening due to holidays, plant intervention, and/or weather-related events (Which would typically take place across all buildings with similar operating strategies). Other impacting features include:

stats_monthlySlope_11 represents the ratio of consumption in December to November. This reveals the changes to building consumption due to a combination of holidays and beginning of winter. It should be noted that other monthly slopes are also selected in this feature selection process, such as *El_stats_monthlySlope_2*, *stats_monthlySlope_8* and *stats_monthlySlope_12*. These features represent the changes in consumption in the beginning of spring, autumn and new year, which are all parameters that are likely to group buildings consumption profiles based on the management policies implemented in them.

consumpstats_daily_kw_var: This feature represents the variance of daily electricity consumption. Buildings with standard schedules, efficient HVAC systems and optimized control systems tend to have lower variance in consumption, as opposed to those with less sophisticated equipment and strategies. Therefore, such a feature could be useful in differentiating between various operation strategies.

Thanks to the procedures implemented in the present study, the achieved performance for all targets is in the same range as the ones reported by the pioneering study in this field [14] while considering a dataset with significantly higher heterogeneity (in terms of building use type and operating site). It should also be pointed out that the maximum of classification accuracy that can be achieved for these pipelines is not limited by the type of the employed algorithm or feature engineering processes, but rather by the notably challenging nature of the problem (estimating buildings' use type, performance class, and operation group while being only provided with smart meter data and not other information about the building).

4.3. Pipelines with features extracted from electrical and chilled water consumption data (subset B)

In the present section, the performance of the ML-based pipelines while being fed with the features extracted from electrical and chilled water consumption data (over subset B which includes the buildings for which both sets of consumption data are available) are represented. In order to create a baseline for subset B, the performance of the benchmark model (i.e., RF) over this subset while being provided with all the features that are only extracted from the electrical consumption is assessed. Next, the accuracy offered by the RF model while being provided with all features extracted from both electrical and CW is determined. Table 7 reports the obtained performances, which reveals that adding the features extracted from the chilled water consumption data, while attempting to estimate the use case and operating group, increases the classification performance of the RF algorithm (fed by all features). On the other hand, it marginally worsens the accuracy achieved for performance class estimation. The latter observation can be attributed to the negative impact of the noise created by adding unnecessary features (extracted from CW data) that is then evaded by performing the feature selection procedure (reported in the next sub-section). This is a further proof that increasing the sheer amount of data is not necessarily helpful in obtaining better solutions and requires proper data handling procedures.

4.3.1. Feature selection and algorithm optimization results

Fig. 8 illustrates the achieved performance indices of different feature selection procedures (while employing RF algorithms) along with the resulting number of selected features while employing the features extracted from both electrical and CW data (using subset B). Similar to the case of subset A, it is shown that the custom feature selection method (proposed by [16]), results in the highest (validation) metrics for use type and performance class estimation. For the operation group instead, it offers a slightly lower performance compared to the RFE method though it requires a lower number of features. Thus, the custom method [16] (similar to the case of subset A) is chosen as the most suitable feature selection procedure.

As demonstrated in Fig. 8, applying the chosen feature selection substantially reduces the number of employed features (from 678 to 18 for the use type estimation, from 441 to 16 for performance class classification, and from 574 to 20 for estimating the operation group). An interesting observation to be noted is that, as the features extracted from CW provide additional information that was not offered by those generated from electrical data, the final number of features selected from the features extracted from both types of consumption is lower than those chosen from the features that were selected only from electrical data.

Furthermore, as reported in Table 7, performing the feature selection, while leading to a significant reduction in the dimensionality (complexity) of the model, also causes a slight improvement in the achieved validation accuracy (identified using 5-fold cross validation over the training set) for all of the considered classification targets. As previously mentioned, it specifically compensates the previously observed reduction in the accuracy achieved for performance class estimation (which was created owing to the noise of unnecessary features extracted from the CW data). The determined performance of the pipeline with selected features over the test set (reported in Table 7) shows an improvement compared to the one achieved by the pipeline provided with all features, confirming that the over-fitting issue has been evaded.

Finally, performing the algorithm optimization step, while employing the chosen set of features, results in identifying ML-based pipelines that offer a slightly higher validation accuracy. Taking into account the improvement that is achieved also for the test set (represented in Table 7), it is demonstrated that these pipelines offer higher performance also for previously unseen data. Appendix C (that includes Tables C.1, C.2, and C.3 for the pipelines, in which use type, performance class, and operation group are considered as the estimation targets respectively) reports the pre-processing step and the identified optimal tuning parameters of the RF algorithm that are used in these pipelines.

4.3.2. Physical interpretation

For this subset, the focus is on determining which, if any, features are added to the feature selection process that are relevant to Chilled Water consumption. These features are denoted with a CW in the beginning of their names.

Table 5
Relative improvement and achieved Weighted F1 score and Accuracy, following the addition of each feature for Building Performance Class - Subset A (Buildings with only electricity meter data).

Feature ID	Meter type	Feature type	Feature details	Time window	Stats operator	F1 Score		Accuracy	
						Relative improvement	Value till this feature	Relative improvement	Value till this feature
weekdays_meanvs95_std	El	Visdom	mean vs 95th percentile	Weekdays	Std	45.55	45.55	45.77	45.77
all_meanvsmax_std	El	Visdom	mean vs max	Entire Data	Std	2.39	47.93	2.34	48.12
weekdays_meanvsmax_mean	El	Visdom	mean vs max	Weekdays	Mean	3.91	51.84	4.02	52.13
all_meanvs95_std	El	Visdom	mean vs 95th percentile	Entire Data	Std	0.24	52.09	0.08	52.22
all_meanvsmax_mean	El	Visdom	mean vs max	Entire Data	Mean	1.18	53.27	1.17	53.39
weekdays_meanvs95_mean	El	Visdom	mean vs 95th percentile	Weekdays	Mean	0.01	53.28	0.00	53.39
all_meanvs95_mean	El	Visdom	mean vs 95th percentile	Entire Data	Mean	0.31	53.59	0.33	53.72
consumpstats_Jan_mn2mx	El	Visdom	min vs max	January	–	2.33	55.93	2.34	56.07
jmotiftemporal_24_8_8_std	El	Jmotif	SAX alphabet = 8 , PAA = 21	Daily	Std	2.87	58.80	2.93	59.00
weekdays_meanvsmax_min	El	Visdom	mean vs max	Weekdays	Min	0.41	59.20	0.33	59.33
jmotif_inclasssim_168_8_21	El	Jmotif	SAX alphabet = 8 , PAA = 21	Weekly	–	0.69	59.89	0.67	60.00
jmotif_inclasssim_24_6_6	El	Jmotif	SAX alphabet = 6 , PAA = 6	Daily	–	0.06	59.95	0.08	60.08
stats_n2d	El	Visdom	Night vs day consumption	Entire Data	–	1.66	61.62	1.67	61.76
loadshape_mape_interval_daytime	El	Loadshape	MAPE Interval	Daytime	–	0.10	61.72	0.25	62.01
jmotiftemporal_168_8_14_std	El	Jmotif	SAX alphabet = 8 , PAA = 14	Weekly	Std	0.20	61.92	0.17	62.18
jmotiftemporal_24_12_12_min	El	Jmotif	SAX alphabet = 12 , PAA = 12	Daily	Min	0.96	62.88	0.75	62.93
jmotiftemporal_168_8_14_max	El	Jmotif	SAX alphabet = 8 , PAA = 14	Weekly	Max	0.16	63.04	0.08	63.01
stltrend_jul_mean	El	STL	Trend	July	Mean	0.32	63.36	0.25	63.26
jmotiftemporal_168_8_14_min	El	Jmotif	SAX alphabet = 8 , PAA = 14	Weekly	Min	0.00	63.36	0.08	63.35
stlreminder_sep_mean	El	STL	Remainder	September	Mean	-0.17	63.19	0.00	63.35
weekend_minvsmax_max	El	Visdom	min vs max	Weekend	Max	1.13	64.32	1.00	64.35

Table 6
Relative improvement and achieved Weighted F1 score and Accuracy, following the addition of each feature for Building Operating Group - Subset A (Buildings with only electricity meter data).

Feature ID	Meter type	Feature type	Feature details	Time window	Stats operator	F1 Score		Accuracy	
						Relative improvement	Value till this feature	Relative improvement	Value till this feature
dayfilterfreq_3_2h_min	El	Day filter	symbol count: 3	2 hr	Min	29.55	29.55	30.79	30.79
minId	El	Visdom Statistics	date of min cons.	Entire Data	Min	35.63	65.18	34.64	65.44
dayfilterfreq_9_8h_min	El	Day filter	symbol count: 9	8 hr	Min	0.35	65.53	0.33	65.77
loadshape_corr_interval_daytime	El	Loadshape	correlation	Daytime	–	0.70	66.23	0.67	66.44
stats_monthlySlope_12	El	Visdom Statistics	Monthly ratio	Jan vs Dec	–	1.09	67.33	1.51	67.95
seasonal_quarterlySlope_1	El	Visdom Statistics	Quarterly ratio	2nd Quarter vs 1st	–	-0.08	67.25	0.17	68.12
dayfilterfreq_7_8h_min	El	Day filter	symbol count: 7	8 hr	Min	0.37	67.62	0.33	68.45
stats_monthlySlope_3	El	Visdom Statistics	Monthly ratio	Apr vs Mar	–	0.79	68.41	0.67	69.12
dayfilterfreq_5_8h_min	El	Day filter	symbol count: 5	8 hr	Min	0.81	69.22	0.84	69.96
dayfilterfreq_5_4h_min	El	Day filter	symbol count: 5	4 hr	Min	0.50	69.72	0.42	70.38
jmotif_inclasssim_24_12_12	El	Jmotif	SAX alphabet = 12,PAA = 12	Daily	–	0.45	70.17	0.50	70.88
consumpstats_daily_kw_var	El	Visdom Statistics	daily consumption	Entire Data	Var	1.71	71.89	2.01	72.89
weekend_minvs95_max	El	Visdom Statistics	min vs 95th percentile	Weekends	Max	0.46	72.35	0.42	73.31
stats_monthlySlope_11	El	Visdom Statistics	Monthly ratio	Dec vs Nov	–	2.38	74.73	2.26	75.56
stats_monthlySlope_5	El	Visdom Statistics	Monthly ratio	Jun vs May	–	0.84	75.57	0.84	76.40
stats_monthlySlope_2	El	Visdom Statistics	Monthly ratio	March vs Feb	–	1.24	76.81	1.09	77.49
stlreminder_jun_mean	El	STL	remainder	June	Mean	0.32	77.13	0.33	77.82
stlreminder_sep_mean	El	STL	remainder	September	Mean	-0.14	76.98	0.08	77.91
consumpstats_kw_mean_annual_2016	El	Visdom Statistics	consumption	Annual	Mean	0.40	77.39	0.25	78.16
dayfilterfreq_7_2h_mean	El	Day filter	symbol count: 7	2hr	Mean	0.64	78.03	0.67	78.83
stlreminder_oct_mean	El	STL	remainder	October	Mean	-0.05	77.99	0.00	78.83
stlreminder_mar_mean	El	STL	remainder	March	Mean	0.52	78.51	0.50	79.33
weekdays_minvs95_std	El	Visdom Statistics	min vs 95th percentile	weekdays	Std	0.27	78.77	0.33	79.67
stats_monthlySlope_8	El	Visdom Statistics	Monthly ratio	Sept vs Aug	–	0.36	79.14	0.33	80.00
stats_nv2dv	El	Visdom Statistics	consumption	Night vs day	Var	1.19	80.32	1.17	81.17
stlreminder_feb_mean	El	STL	remainder	February	Mean	0.07	80.40	0.08	81.26
stlreminder_nov_mean	El	STL	remainder	November	Mean	0.29	80.69	0.25	81.51
weekdays_meanvsmax_std	El	Visdom Statistics	mean vs max	weekdays	Std	0.05	80.73	0.08	81.59
eemeter_cooling_max	El	ee Meter	cooling	Entire Data	Max	0.68	81.41	0.59	82.18
stltrend_oct_mean	El	STL	trend	October	Mean	-0.03	81.38	0.08	82.26
eemeter_heating_max	El	ee Meter	heating	Entire Data	Max	0.63	82.01	0.50	82.76
jmotiftemporal_24_8_8_max	El	Jmotif	SAX alphabet = 8 , PAA = 8	Daily	Max	0.28	82.29	0.33	83.10
eemeter_coolslope	El	ee Meter	cooling line slope	Entire Data	–	0.15	82.44	0.08	83.18
eemeter_heatslope	El	ee Meter	heating line slope	Entire Data	–	0.20	82.64	0.08	83.26
consumpstats_daily_kw_max_var	El	Visdom Statistics	max daily cons.	Entire Data	Var	0.87	83.51	0.92	84.18

Table 7
Validation (Val.) and Test Weighted F1 Score as well as Accuracy for Pipelines studied in Subset B.

Target	Random Forest-based pipeline with Electricity-based features				Random Forest-based pipeline with all available features				Random forest-based pipeline with selected features				Optimal pipeline with selected features			
	F1 Score		Accuracy		F1 Score		Accuracy		F1 Score		Accuracy		F1 Score		Accuracy	
	Val.	Test	Val.	Test	Val.	Test	Val.	Test	Val.	Test	Val.	Test	Val.	Test	Val.	Test
Use case	0.568	0.630	0.619	0.573	0.600	0.650	0.649	0.587	0.646	0.653	0.686	0.613	0.661	0.666	0.696	0.627
Performance Class	0.539	0.591	0.548	0.587	0.524	0.517	0.535	0.507	0.606	0.590	0.612	0.587	0.618	0.604	0.622	0.600
Operation Group	0.857	0.877	0.870	0.867	0.913	0.909	0.923	0.907	0.921	0.909	0.930	0.907	0.924	0.926	0.933	0.920

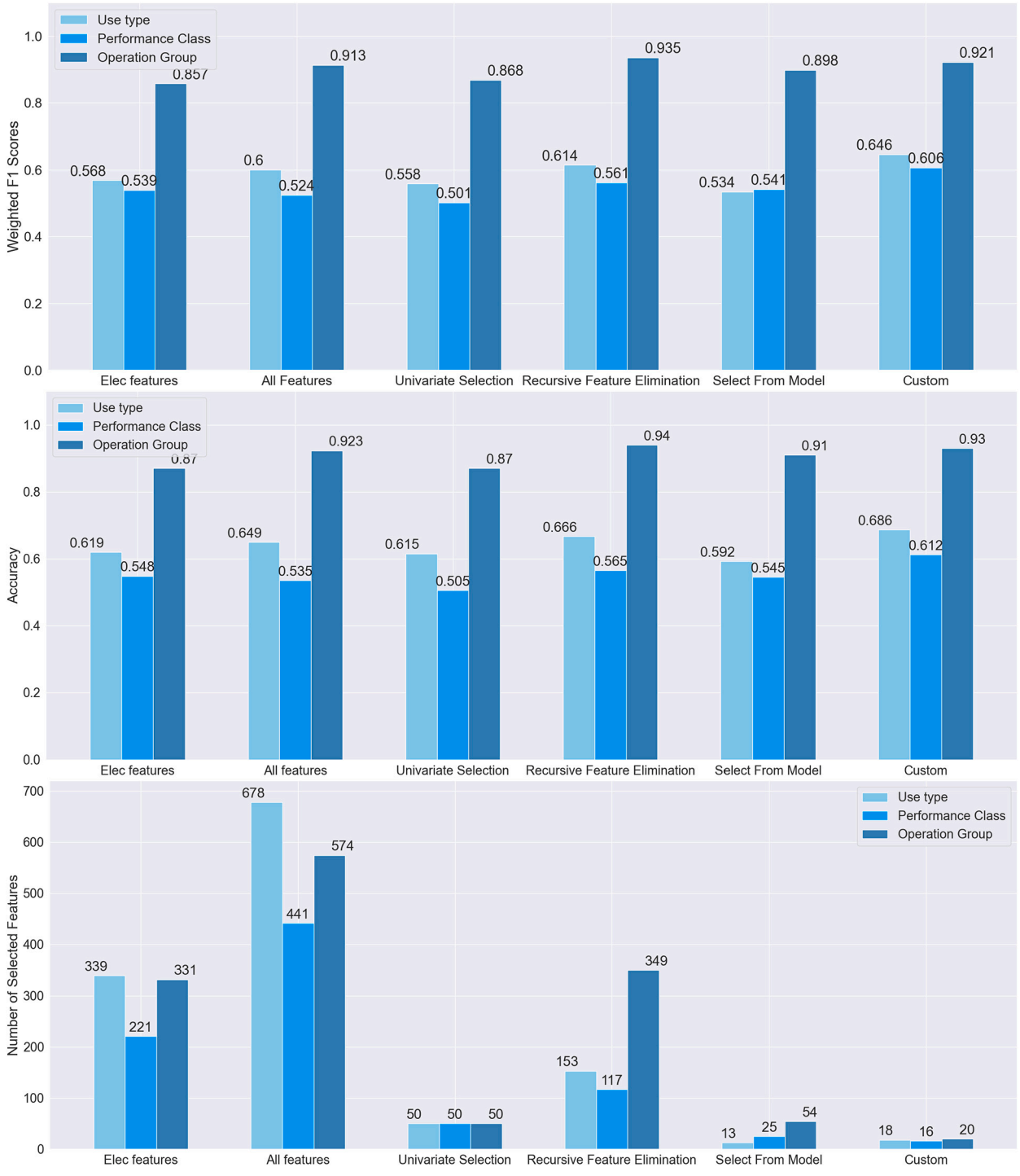


Fig. 8. Comparison of feature selection methods and classification targets.

Table 8
Relative improvement and achieved Weighted F1 score and Accuracy, following the addition of each feature for Building Use Case - Subset B (Buildings with both electricity and chilled water meter data).

Feature ID	Meter type	Feature type	Feature details	Time window	Stats operator	F1 Score		Accuracy	
						Relative improvement	Value till this feature	Relative improvement	Value till this feature
stlweeklypattern_sun_mean	El	STL	Weekly pattern	Sundays	Mean	39.28	39.28	50.85	50.85
stlweeklypattern_fri_mean	CW	STL	Weekly pattern	Fridays	Mean	7.86	47.14	0.32	51.16
stlweeklypattern_sat_mean	CW	STL	Weekly pattern	Saturdays	Mean	0.68	47.82	0.66	51.83
stlweeklypattern_tue_mean	El	STL	Weekly pattern	Tuesdays	Mean	2.80	50.62	3.01	54.83
stlweeklypattern_thur_mean	El	STL	Weekly pattern	Thursdays	Mean	0.14	50.76	0.66	55.49
stlweeklypattern_fri_mean	El	STL	Weekly pattern	Fridays	Mean	0.37	51.13	0.33	55.82
consumpstats_t10kw	El	Visdom	Mean temp @10th percentile cons.	Entire data	Total	1.28	52.41	1.68	57.50
breakouts_max_30_1_3	El	Breakouts	Breakout size = 30, penalization = 1	Entire data	Max	0.48	52.89	0.34	57.84
weekend_meanvsmax_mean	El	Visdom	mean vs max	Weekends	Mean	3.49	56.38	4.02	61.86
weekend_meanvs95_mean	El	Visdom	Mean vs 95th percentile	Weekends	Mean	0.99	57.37	0.32	62.18
weekdays_meanvsmax_mean	El	Visdom	mean vs max	Weekdays	Mean	1.09	58.46	1.03	63.21
consumpstats_kw_mean_annual_2016	El	Visdom	Annual consumption	Annual	Mean	3.38	61.84	2.33	65.54
stats_monthlySlope_1	El	Visdom	February vs January	Monthly	Total	-0.10	61.74	0.01	65.55
dayfilterfreq_5_6h_min	El	Dayfilter	Symbol count = 5	6 hr	Min	1.16	62.90	0.99	66.54
all_meanvs95_mean	El	Visdom	Mean vs 95th percentile	Entire data	Mean	0.51	63.42	0.34	66.88
stlreminder_sep_mean	El	STL	Remainder	September	Mean	1.71	65.13	1.67	68.55
stltrend_oct_mean	El	STL	Trend	October	Mean	0.26	65.40	0.32	68.87
dayfilterfreq_7_4h_max	CW	Dayfilter	Symbol count = 7	4 hr	Max	0.17	65.57	0.35	69.22

Table 9
Relative improvement and achieved Weighted F1 score and Accuracy, following the addition of each feature for Building Performance Class - Subset B (Buildings with both electricity and chilled water meter data).

Feature ID	Meter type	Feature type	Feature details	Time window	Stats operator	F1 Score		Accuracy	
						Relative improvement	Value till this feature	Relative improvement	Value till this feature
weekdays_meanvs95_std	El	Visdom	mean vs 95th percentile	Weekdays	Std	38.26	38.26	39.12	39.12
weekend_minvs95_std	El	Visdom	min vs 95th percentile	Weekends	Std	9.12	47.38	8.71	47.83
weekend_meanvsmax_mean	El	Visdom	mean vs max	Weekends	Mean	0.89	48.27	1.32	49.15
jmotiftemporal_24_8_8_min	El	Jmotif	SAX alphabet = 8,PAA = 8	Daily	Min	1.22	49.49	1.34	50.49
jmotiftemporal_24_8_8_mean	El	Jmotif	SAX alphabet = 8,PAA = 8	Daily	Mean	0.95	50.44	0.35	50.84
weekdays_minvsmax_min	El	Visdom	min vs max	Weekdays	Min	3.45	53.89	3.67	54.50
stltrend_dec_mean	El	STL	December	December	Mean	2.07	55.97	2.00	56.50
weekdays_minvs95_min	El	Visdom	min vs 95th percentile	Weekdays	Min	1.43	57.40	0.99	57.50
stlreminder_nov_mean	El	STL	November	November	Mean	-0.16	57.24	0.01	57.51
jmotiftemporal_168_6_14_std	El	Jmotif	SAX alphabet = 8,PAA = 8	Weekly	Std	0.79	58.03	0.96	58.46
stats_min_day_pct	El	Visdom	percent time when temp is lower than @min cons.	Entire Data	Pct	1.36	59.40	1.35	59.82
weekend_meanvs95_std	CW	Visdom	mean vs 95th percentile	Weekends	Std	0.30	59.70	0.37	60.19
jmotiftemporal_168_8_14_mean	El	Jmotif	SAX alphabet = 8,PAA = 8	Weekly	Mean	0.49	60.19	0.31	60.50
all_meanvs95_min	El	Visdom	mean vs 95th percentile	Entire Data	Min	-0.37	59.81	0.00	60.50
stltrend_mar_mean	El	STL	March	March	Mean	0.97	60.79	0.68	61.19
dayfilterfreq_9_4h_std	El	Dayfilter	Symbol count = 9	4 hr	Std	-0.21	60.57	0.00	61.19

Table 10
Relative improvement and achieved Weighted F1 score and Accuracy, following the addition of each feature for Building Operating Group - Subset B (Buildings with both electricity and chilled water meter data).

Feature ID	Meter type	Feature type	Feature details	Time window	Stats operator	F1 Score		Accuracy	
						Relative improvement	Value till this feature	Relative improvement	Value till this feature
minId	El	Visdom	date @ min cons.	Entire Data	-	78.61	78.61	78.28	78.28
dayfilterfreq_7_4h_min	El	Dayfilter	Symbol count = 7	4 hr	Min	7.2	85.81	6.68	84.96
dayfilterfreq_5_6h_min	El	Dayfilter	Symbol count = 5	6 hr	Min	0.58	86.39	0.67	85.63
normalizedcons_std	CW	Visdom	Normalized cons.	Entire Data	Std	-1.54	84.85	0.31	85.94
consumpstats_maxHOD	CW	Visdom	Hour of day @ max cons.	Entire Data	-	-0.08	84.77	0.01	85.95
consumpstats_mean	CW	Visdom	Total mean cons.	Entire Data	Mean	-0.04	84.73	0.01	85.96
jmotif_inclassim_168_8_21	El	Jmotif	SAX alphabet = 8,PAA = 21	Weekly	-	-0.18	84.55	0.01	85.97
hourlystats_HOD_mean_06	CW	Visdom	mean consumption of hour	6th hour of day	Mean	0.36	84.91	0.33	86.3
all_minvsmax_max	CW	Visdom	min vs max cons.	Entire Data	Max	1.63	86.54	1.33	87.63
stats_monthlySlope_3	CW	Visdom	Ratio of cons. in April vs March	Monthly	Total	0.53	87.07	0.32	87.95
hourlystats_HOD_mean_04	CW	Visdom	mean consumption of hour	4th hour of day	Mean	0.39	87.46	0.68	88.63
all_meanvs95_max	CW	Visdom	mean vs 95th percentile	Entire Data	Max	0.33	87.79	0.33	88.96
consumpstats_daily_kw_var	CW	Visdom	daily consumption	Daily	Var	0.51	88.3	0.34	89.3
stats_monthlySlope_9	CW	Visdom	Ratio of cons. In October vs September	Monthly	Total	0.03	88.33	0.34	89.64
dayfilterfreq_7_2h_std	CW	Dayfilter	Symbol count = 7	2 hr	Std	1.51	89.84	1.34	90.98
dayfilterfreq_9_2h_std	El	Dayfilter	Symbol count = 9	2 hr	Std	0.21	90.05	0.01	90.99
stlreminder_oct_mean	El	STL	Remainder	October	Mean	1.34	91.39	1.33	92.32
breakouts_max_30_1_3	CW	Breakouts	Penalization = 1	Monthly	-	-0.03	91.36	0.01	92.33
jmotif_inclassim_168_6_21	El	Jmotif	SAX alphabet = 21,PAA = 6	Weekly	-	0.3	91.66	0.33	92.66
dayfilterfreq_7_4h_mean	El	Dayfilter	Symbol count = 7	4 hr	Mean	0.46	92.12	0.33	92.99

Use Type: Table 8 demonstrates the relative improvement and the overall achieved F1 score and accuracy with the addition of each feature. About half of all features selected are STL features, most of which are weekly patterns for different days of the week. Overall, the features selected after addition of chilled water data, are quite similar to those selected for only electricity-based features, with the advantage of less features being necessary due to the addition of chilled water based features. In this context important features include:

stlweeklypattern_fri_mean: represents the mean trend of chilled water consumption on Fridays. this feature is a promising indicator of end-of-the week cooling loads and helps differentiate buildings such as public environments and malls which experience a surge of visitors on Fridays. *stlweeklypattern_sat_mean*: similar to the previous feature, this feature utilizes the STL library to determine mean consumption of chilled water on Saturdays. This could be beneficial in the case of distinguishing residential buildings, which have higher cooling loads in the weekends, compared to other types of buildings.

dayfiltereq_7_4h_max: the maximum weekly chilled-water consumption based on 4-hour long windows. As the peak cooling load demand is different between each building, this feature could be taken advantage of for the classification of buildings based on functionality.

Performance Class: For this target, only one feature was selected from chilled water features, while the rest are mostly derived from Visdom and Jmotif packages, similarly to patterns observed for the subset with only electricity-based features. Table 9 demonstrates the improvement (in the F1 score and accuracy) that is achieved by adding of each feature along with the corresponding overall obtained value.
weekend_meanvs95_min is an indicator of the variance of the ratio of 95th percentile of chilled water consumption to the mean during weekends. Buildings with higher performance, tend to have more sophisticated HVAC systems, leading to a less overall variation.

Operation Group: The highest increase in the achieved accuracy, by adding the chilled water based features, is observed for this target; therefore, it was expected to observe more features from this group being selected. This holds true as more than half of selected features are indeed based on chilled water consumption. Table 10 shows the improvement that is obtained in the F1 score and accuracy with the addition of each feature along with the corresponding overall achieved value. *all_minvsmax_max* determines the highest disparity between daily minimum and maximum consumption for each building *all_meanvs95_max* calculates the same but for the ratio between mean consumption and the 95th percentile of consumption. The higher these numbers are, the more discrepancy in weather in the corresponding site exists, which separates the minimum and maximum consumption in a single day. On the other hand, as the number reaches zero, it could either mean that there is no cooling load to begin with or that the load is constant during the entire day (which helps differentiating between different sites).
stats_monthlySlope_3 and *CW_stats_monthlySlope_9* show the ratio of the cooling loads in April to May and October to September, respectively. These are suitable variables to determine the slope of the cooling load being disabled and enabled, respectively.

normalizedcons_std: represents the normalized standard deviation of chilled water consumption of buildings. Normalization helps enable the comparison of buildings with different sizes or cooling loads. Buildings with the same operation strategy tend to have similar variance in this regard.

consumpstats_max_HOD: represents the hour of the day at which chilled water consumption is maximum, thus representing the hour of the day most typically associated with the peak demand. Buildings from the same zone and operation strategy, tend to have similar outside temperature and occupancy patterns, thus this is a useful feature to distinguish between different operation strategies. hourlystats_HOD_mean_06 and hourlystats_HOD_mean_04 are also selected for this classification target, following the same logic.

5. Conclusion

In the present work, following a number of feature generation methodologies proposed in the literature, a pipeline development and optimization procedure was implemented for smart meter-based estimation of buildings' use type, performance class, and operation strategy. Using the Building Data Genome 2 dataset, after performing preprocessing and extensive feature generation step, feature selection (to reduce model's dimensionality) and algorithm optimization procedures were performed. In the first part of the work, the pipeline was implemented on a subset of 1494 buildings from the dataset, which includes electricity consumption's recordings. Obtained results confirmed the efficacy of the developed pipeline, leading to 2.9-5.3% improvements in accuracy for the three classification targets, while reducing the number of employed features by more than 88%. The second subset of buildings, for which both chilled water meter data and electricity consumption were available, demonstrated similar trends with even higher gains in the model's accuracy (between 7.2% and 13.5%) after employing the developed ML-based pipelines. It was thus demonstrated that the addition of chilled water consumption-based features yields significant improvements in model's accuracy (before and after the feature selection process). Moreover, as could be expected, pipeline optimization brings about (even if marginal) improvements in the achieved accuracy of each pipeline.

Furthermore, the impact of adding each feature on the overall model's accuracy was investigated. This analysis facilitated uncovering the mechanisms through which the ML model predicts the target variables and permitted the corresponding physical/logical interpretation. It was demonstrated that, to differentiate between different buildings' use cases, features based on the maximum consumption's hourly and weekly patterns for Thursday through Sunday are of the highest importance since they contain schedule-centric information. In the case of classification based on performance class, features containing information about how closely a building's consumption resembles that of its own past or others with similar functions (such information is represented by the features from the Jmotif package), were proved to be the most influential ones. On the other hand, regarding the operation group's classification, it was shown that about half of the selected features are pattern-based, which can be attributed to these features' ability to differentiate between operating strategies over different time windows. Another key finding of present work was that the addition of secondary meter data, e.g. chilled water here, enables the model to work with fewer features, as these secondary meter features provide the pipelines with additional information.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The utilized dataset is open access.

Appendix A. Online repository of the implemented procedures

The utilized processed dataset, the obtained optimal sets of features (for each considered target), the implemented feature selection procedures, along with the optimal pipelines are provided in an online repository (Link).

Appendix B. Identified optimal ML-based pipelines for subset A

Table B.1
Characteristics of determined optimal pipeline for Subset A - Use case classification target.

Optimal Pipeline Step	Arguments	Definitions	Values
Step 1 (Only Step): RandomForestClassifier	class_weight	Weight associated with classes in the form {class_label:weight}	balanced
	n_estimators	The number of trees in the forest	150

Table B.2
Characteristics of determined optimal pipeline for Subset A - Performance class classification target.

Optimal Pipeline Step	Arguments	Definitions	Values
Step 1 (Only Step): RandomForestClassifier	bootstrap	Whether bootstrap samples are used when building trees	True
	criterion	The function to measure the quality of a split	entropy
	max_features	The number of features to consider when looking for the best split	0.2
	min_samples_leaf	The minimum number of samples required to be at a leaf node	3
	min_sample_split	The minimum number of samples required to split an internal node	9
	n_estimator	The number of trees in the forest	100

Table B.3
Characteristics of determined optimal pipeline for Subset A - Operating group classification target.

Optimal Pipeline Step	Arguments	Definitions	Values
Step 1: FunctionTransformer	–	–	–
Step 2: OneHotEncoder	minimum_fraction	Minimum fraction of numerical features to get boolean values	0.15
	sparse	Will return sparse matrix if set True else will return an array	False
	threshold	Thresholds numerical features to get boolean values	10
Step 3: FunctionTransformer	–	–	–
Step 4: GradientBoosting Classifier	learning_rate	Shrinks the contribution of each tree by this factor	0.1
	max_depth	The maximum depth of the individual regression estimators	6
	max_features	The number of features to consider when looking for the best split	0.1
	min_samples_leaf	The minimum number of samples required to be at a leaf node	17
	min_samples_split	The minimum number of samples required to split a node	13
	n_estimators	The number of boosting stages to perform	100
	subsample	The fraction of samples to be used for fitting the individual base learners	0.95

Appendix C. Identified Optimal ML-based Pipelines for subset B

Table C.1
Characteristics of determined optimal pipeline for Subset B - Use case classification target.

Optimal Pipeline Step	Arguments	Definitions	Values
Step 1: MaxAbsScaler	–	–	–
Step 2: OneHotEncoder	minimum_fraction	Minimum fraction of numerical features to get boolean values	0.15
	sparse	Will return sparse matrix if set True else will return an array	False
	threshold	Thresholds numerical features to get boolean values	10
Step 3: RandomForest Classifier	bootstrap	Whether bootstrap samples are used when building trees	False
	criterion	The function to measure the quality of a split	gini
	max_features	The number of features to consider when looking for the best split	0.3
	min_samples_leaf	The minimum number of samples required to be at a leaf node	1
	min_samples_split	The minimum number of samples required to split an internal node	18
	n_estimators	The number of trees in the forest	100

Table C.2
Characteristics of determined optimal pipeline for Subset B - Performance class classification target.

Optimal Pipeline Step	Arguments	Definitions	Values
Step 1: FastICA	tol	The tolerance at which the un-mixing is considered to have converged	0.55
Step 2: SelectFromModel estimator = ExtraTrees Classifier	criterion	The function to measure the quality of a split	gini
	max_features	The number of features to consider when looking for the best split	0.95
	n_estimators	The number of trees in the forest	100
	threshold	The threshold value to use for feature selection	0.95
Step 3: RandomForestClassifier	bootstrap	Whether bootstrap samples are used when building trees	False
	criterion	The function to measure the quality of a split	entropy
	max_features	The number of features to consider when looking for the best split	0.35
	min_samples_leaf	The minimum number of samples required to be at a leaf node	4
	min_sample_split	The minimum number of samples required to split an internal node	5
	n_estimator	The number of trees in the forest	100

Table C.3
Characteristics of determined optimal pipeline for Subset B - Operating group classification target.

Optimal Pipeline Step	Arguments	Definitions	Values
Step 1: FunctionTransformer	–	–	–
Step 2: RandomForestClassifier	class_weight	Weights associated with classes in the form {class_label:weight}	balanced

References

[1] IEA, Tracking buildings 2021, iea, paris https://www.iea.org/reports/tracking-buildings-2021, 2021.
[2] S. Guo, D. Yan, S. Hu, J. An, Global comparison of building energy use data within the context of climate change, Energy Build. 226 (2020) 110362.
[3] P. Wattles, Ercot demand response overview & status report, in: AMIT-DSWG Workshop, AMI's Next Frontier: Demand Response, 2011.
[4] I. Fakhari, P. Behinfar, F. Raymand, A. Azad, P. Ahmadi, E. Houshfar, et al., 4e analysis and tri-objective optimization of a triple-pressure combined cycle power plant with combustion chamber steam injection to control nox emission, J. Therm. Anal. Calorim. 145 (3) (2021) 1317–1333.
[5] K.S. Cetin, Z. O'Neill, Smart meters and smart devices in buildings: a review of recent progress and influence on electricity use and peak demand, Curr. Sustain./Renew. Energy Rep. 4 (1) (2017) 1–7.
[6] S. Zhan, Z. Liu, A. Chong, D. Yan, Building categorization revisited: a clustering-based approach to using smart meter data for building energy benchmarking, Appl. Energy 269 (2020) 114920.
[7] D.J. F., K.B.S. H., N. B., H.M. A., R. F., Application of Machine Learning in Occupant and Indoor Environment Behavior Modeling: Sensors, Methods, and Algorithms, Springer, ISBN 978-3-030-72322-4, 2022.
[8] J. Granderson, S. Touzani, C. Custodio, M.D. Sohn, D. Jump, S. Fernandes, Accuracy of automated measurement and verification (m&v) techniques for energy savings in commercial buildings, Appl. Energy 173 (2016) 296–308.
[9] B. Najafi, S. Moaveninejad, F. Rinaldi, Data analytics for energy disaggregation: methods and applications, in: Big Data Application in Power Systems, Elsevier, 2018, pp. 377–408.
[10] B. Najafi, L. Di Narzo, F. Rinaldi, R. Arghandeh, Machine learning based disaggregation of air-conditioning loads using smart meter data, IET Gener. Transm. Distrib. 14 (21) (2020) 4755–4762.
[11] I. Rahman, M. Kuzlu, S. Rahman, Power disaggregation of combined hvac loads using supervised machine learning algorithms, Energy Build. 172 (2018) 57–66.
[12] A. Moazami, V.M. Nik, S. Carlucci, S. Geving, Impacts of future weather data typology on building energy performance – investigating long-term patterns of climate change and extreme weather conditions, Appl. Energy 238 (2019) 696–720, https://doi.org/10.1016/j.apenergy.2019.01.085, https://www.sciencedirect.com/science/article/pii/S0306261919300868.
[13] A. Perera, T. Hong, Vulnerability and resilience of urban energy ecosystems to extreme climate events: a systematic review and perspectives, Renew. Sustain. Energy Rev. 173 (2023) 113038, https://doi.org/10.1016/j.rser.2022.113038, https://www.sciencedirect.com/science/article/pii/S1364032122009194.
[14] C. Miller, What's in the box?! Towards explainable machine learning applied to non-residential building smart meter classification, Energy Build. 199 (2019) 523–536.
[15] C. Miller, F. Meggers, The building data genome project: an open, public data set from non-residential building electrical meters, Energy Proc. 122 (2017) 439–444.
[16] B. Najafi, M. Depalo, F. Rinaldi, R. Arghandeh, Building characterization through smart meter data analytics: determination of the most influential temporal and importance-in-prediction based features, Energy Build. 234 (2021) 110671.
[17] C. Miller, A. Kathirgamanathan, B. Picchetti, P. Arjunan, J.Y. Park, Z. Nagy, et al., The building data genome project 2, energy meter data from the ashrae great energy predictor iii competition, Sci. Data 7 (1) (2020) 1–13.
[18] C. Miller, Screening meter data: Characterization of temporal energy data from large groups of non-residential buildings, Ph.D. thesis, ETH, Zurich, 2016, https://doi.org/10.3929/ethz-a-010811999.
[19] J. Han, M. Kamber, J. Pei, Data Mining Trends and Research Frontiers, Data Min, 2012, pp. 585–631.
[20] J.L. Mathieu, P.N. Price, S. Kiliccote, M.A. Piette, Quantifying changes in building electricity use, with application to demand response, IEEE Trans. Smart Grid 2 (3) (2011) 507–518.
[21] Eemeter openeemeter, https://github.com/openeemeter/eemeter, 2018.
[22] T. Mitsa, Temporal Data Mining, Chapman and Hall/CRC, 2010.
[23] S.t.l. package on cran, https://cran.r-project.org/web/packages/stlplus/index.html.
[24] P. Patel, E. Keogh, J. Lin, S. Lonardi, Mining motifs in massive time series databases, in: 2002 IEEE International Conference on Data Mining, 2002. Proceedings, IEEE, 2002, pp. 370–377.
[25] E. Keogh, J. Lin, A. Fu, Hot sax: efficiently finding the most unusual time series subsequence, in: Fifth IEEE International Conference on Data Mining (ICDM'05), Ieee, 2005, p. 8.
[26] J. Lin, E. Keogh, S. Lonardi, B. Chiu, A symbolic representation of time series, with implications for streaming algorithms, in: Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, 2003, pp. 2–11.
[27] P. Senin, S. Malinchik, Sax-vsm: interpretable time series classification using sax and vector space model, in: 2013 IEEE 13th International Conference on Data Mining, IEEE, 2013, pp. 1175–1180.
[28] C. Miller, A. Schlueter, Forensically discovering simulation feedback knowledge from a campus energy information system, in: SpringSim (SimAUD), 2015, pp. 136–143.
[29] N.A. James, A. Kejariwal, D.S. Matteson, Leveraging cloud data to mitigate user experience from 'breaking bad', in: 2016 IEEE International Conference on Big Data (Big Data), IEEE, 2016, pp. 3499–3508.
[30] H. Belyadi, A. Haghighat, Machine Learning Guide for Oil and Gas Using Python, vol. 10, Elsevier, 2021.
[31] Top machine learning algorithms for classification, https://towardsdatascience.com/top-machine-learning-algorithms-for-classification-2197870ff501.
[32] Classification algorithms, https://monkeylearn.com/blog/classification-algorithms/.
[33] S. Raschka, Python Machine Learning, Packt Publishing Ltd, 2015.
[34] Naive bayes classifiers on scikit learn, https://scikit-learn.org/stable/modules/naive_bayes.html.
[35] K-neighbors methods on scikit learn, https://scikit-learn.org/stable/modules/neighbors.html.
[36] Scikit-learn, Support vector regressor scikit-learn.org/stable/modules/generated/sklearn.svm.SVR, 2020.

[37] R. Rezazadegan, M. Sharifzadeh, Applications of artificial intelligence and big data in industry 4.0 technologies, in: Industry 40 Vision for Energy and Materials: Enabling Technologies and Case Studies, 2022, pp. 121–158.
[38] decision trees, https://scikit-learn.org/stable/modules/tree.html.
[39] Random forest and gradient boosting, differences, https://towardsdatascience.com/decision-trees-random-forests-and-gradient-boosting-whats-the-difference-ae435cbb67ad.
[40] M. Manivannan, B. Najafi, F. Rinaldi, Machine learning-based short-term prediction of air-conditioning load through smart meter analytics, Energies 10 (11) (2017).
[41] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[42] A. Natekin, A. Knoll, Gradient boosting machines, a tutorial, Front. Neurorobot. 7 (2013) 21.
[43] E.W. Huang, W.J. Lee, S.S. Singh, P. Kumar, C.Y. Lee, T.N. Lam, et al., Machine-learning and high-throughput studies for high-entropy materials, Mater. Sci. Eng., R Rep. 147 (2022) 100645.
[44] E. Grossi, M. Buscema, Introduction to artificial neural networks, Eur. J. Gastroenterol. Hepatol. 19 (12) (2007) 1046–1054.
[45] B. Najafi, K. Ardam, A. Hanušovský, F. Rinaldi, L.P.M. Colombo, Machine learning based models for pressure drop estimation of two-phase adiabatic air-water flow in micro-finned tubes: determination of the most promising dimensionless feature set, Chem. Eng. Res. Des. 167 (2021) 252–267.
[46] H. Malik, N. Fatema, A. Iqbal, Intelligent Data-Analytics for Condition Monitoring: Smart Grid Applications, Academic Press, 2021.
[47] M. Kuhn, K. Johnson, et al., Applied Predictive Modeling, vol. 26, Springer, 2013.
[48] Feature selection for categorical data, https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/.
[49] Feature selection on scikit learn, https://scikit-learn.org/stable/modules/feature_selection.html#rfe.
[50] B. Najafi, P. Bonomi, A. Casalegno, F. Rinaldi, A. Baricci, Rapid fault diagnosis of PEM fuel cells through optimal electrochemical impedance spectroscopy tests, Energies 13 (14) (2020) 3643.
[51] Recursive feature elimination, https://scikit-learn.org/stable/modules/feature_selection.html#rfe.
[52] X. Wu, W. Zheng, M. Pu, J. Chen, D. Mu, Invalid bug reports complicate the software aging situation, Softw. Qual. J. 28 (1) (2020) 195–220.
[53] R.S. Olson, R.J. Urbanowicz, P.C. Andrews, N.A. Lavender, L.C. Kidd, J.H. Moore, Automating biomedical data science through tree-based pipeline optimization, in: European Conference on the Applications of Evolutionary Computation, Springer, 2016, pp. 123–137.
[54] I.A. Campodonico Avendano, F. Dadras Javan, B. Najafi, A. Moazami, F. Rinaldi, Assessing the impact of employing machine learning-based baseline load prediction pipelines with sliding-window training scheme on offered flexibility estimation for different building categories, Energy Build. 294 (2023) 113217, https://doi.org/10.1016/j.enbuild.2023.113217, https://www.sciencedirect.com/science/article/pii/S0378778823004474.
[55] B. Najafi, H. Najafi, M. Idalik, Computational fluid dynamics investigation and multi-objective optimization of an engine air-cooling system using genetic algorithm, Proc. Inst. Mech. Eng., Part C, J. Mech. Eng. Sci. 225 (6) (2011) 1389–1398.
[56] K. Ardam, B. Najafi, A. Lucchini, F. Rinaldi, L.P.M. Colombo, Machine learning based pressure drop estimation of evaporating r134a flow in micro-fin tubes: Investigation of the optimal dimensionless feature set, Int. J. Refrig. 131 (2021) 20–32, www.scopus.com.
[57] Mean absolute error, https://developers.google.com/machine-learning/glossary?hl=en#MSE.
[58] Classification: Precision and recall, https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall.
[59] Wikipedia, Precision and recall, wikipedia.org/wiki/Precision_and_recall, 2020.
[60] Averages of f1 and accuracy, https://towardsdatascience.com/micro-macro-weighted-averages-of-f1-score-clearly-explained-b603420b292f.
[61] Wikipedia, Spearman's rank correlation coefficient, en.wikipedia.org/wiki/Spearman's_rank_correlation_coefficient, 2020.
[62] Towards Data Science, The mathematics of decision trees, random forest and feature importance in scikit-learn and spark, www.towardsdatascience.com, 2018.
[63] R.S. Olson, N. Bartley, R.J. Urbanowicz, J.H. Moore, Evaluation of a tree-based pipeline optimization tool for automating data science, in: Proceedings of the Genetic and Evolutionary Computation Conference 2016, ACM, 2016, pp. 485–492.