The generalizability of pre-processing techniques on the accuracy and fairness of data-driven building models: a case study

Ying Sun[1], Benjamin C. M. Fung[2], Fariborz Haghighat*[1]
[1]Energy and Environment Group, Department of Building, Civil and Environmental Engineering Concordia, University, Montreal, Canada
[2]School of Information Studies, McGill University, Montreal, Canada

Official publication: https://doi.org/10.1016/j.enbuild.2022.112204

Abstract

In recent years, massive data collected from buildings made development and application of data-driven building models is a hot research topic. Due to the variation of data volume in different conditions, existing data-driven building models (DDBMs) would present distinct accuracy for different users or periods. This may create further fairness problems. To solve these issues, balancing training dataset between different conditions using pre-processing techniques could help. In this study, a sequentially balanced sampling (SBS) technique is proposed. Its generalizability to improve fairness and preserve accuracy of DDBMs is compared with four existing pre-processing techniques—random sampling (RS), sequential sampling (SS), reversed preferential sampling (RPS), and sequential preferential sampling (SPS). Totally, 4960 cases are carried out to apply these pre-processing techniques to process training dataset before developing 4 types of classifiers for one-week ahead lighting status prediction of 155 lights in 16 apartments through a year. Note that the collected data show 5 distribution modes.

The newly proposed SBS shows comparable performance to RPS. They significantly improve predictive accuracy for minority classes but decrease the accuracy for majority classes. On the other hand, SS and SPS show a slight accuracy improvement for minority classes with an acceptable price of accuracy decrease on majority classes. In terms of fairness improvement, SBS, RS, and RPS could effectively increase the recall rate. However, RS and RPS show more negative effect on accuracy rate and specificity rate. The results of this study provide guidance for researchers to select proper pre-processing techniques to improve the preferred predictive performance under different data distribution.

Keywords: Fairness; Generalizability; Accuracy; Data-driven Model; Building

Corresponding Author*: Fariborz.Haghighat@Concordia.ca

1. Introduction

1.1. Background

In recent years, buildings and building construction sectors have become data rich, due to the rapidly popularity of Internet of Things (IoT) and building management systems (BMS) [1]. This means that plenty of information (such as indoor environment parameters, occupancy-related data, energy consumption, and equipment and device operational data, etc.) has been dynamically collected from buildings [2]. These data could be used as a basis to develop data-driven building models (DDBMs) to predict indoor air temperature [3], energy consumption [4–6], thermal comfort [7], occupancy status/numbers [8–10], indoor air quality [11], fire hazard [12] or HVAC system performance [13,14].

Predictive accuracy is an important performance criteria for DDBMs to reflect the similarity of predicted values to measured/simulated values [15,16]. Existing studies on DDBMs mainly focus on improving these models' predictive accuracy, to ensure that the predictive result could reasonably represent indoor environment parameters, energy consumption patterns, or device operational status. For instance, González-Vidal et al. [17] proposed a feature selection structure to improve the mean absolute error (MAE) by 42.28% and root mean square error (RMSE) by 36.62% for energy prediction regression models. Kallio et al. [18] compared the accuracy of four machine learning techniques for predicting indoor CO_2 concentration in terms of MAE and proposed the possible application of DDBMs for proactive indoor environment control.

However, a DDBM with high overall predictive accuracy could not ensure a good predictive performance under all different scenarios/operation situations [19]. In reality, different volume of training dataset under various conditions may result in better predictive accuracy under majority conditions and poor accuracy under other conditions [20]. For instance, HVAC system operational data are mostly collected in normal conditions. Few are measured during faulty scenarios. As a result, even if a model trained on these data works well to predict the normal HVAC operation status, the problem remains that it wrongly predicts faulty scenarios as normal [1].

The different predictive performance under different conditions may further leads to fairness problems [21]. For instance, if an energy consumption predictor works much better for users who provide more data than others, it would be unfair for other users who use the predictor with poor performance.

Three types of fairness concepts have been introduced into the building application [21]. *Type I:* The predicted output is independent of the protected attribute(s), whose values are not willing to be disclosed. For building models, occupancy-related data could be the protected attribute(s). This is because these data may be denied as inputs by users with the concern of avoiding to reveal occupants' individual location and behavior [22–24]. *Type II:* Some performance measures (e.g., accuracy) are equal across classes/conditions defined by the protected attribute(s). For instance, when developing energy predictive models for users from different occupations, the predictive accuracy is expected to be similar for them no matter what energy consumption habits caused by their work pattern. In this example, users' occupation is the protected attribute. *Type III*: Predictive outcomes should be independent of the predictive probability score of different classes/conditions defined by the protected attribute(s). For example, if a data-driven predictor predicts the same probability score of approving house loan to people coming from different races (protected attribute), their loan application result should be the same.

Achieving *Type II* fairness in the building application could ensure uniform predictive performance under different situations. To improve *Type II* fairness, we previously proposed three types of data pre-processing techniques — sequential sampling (SS), reversed preferential sampling (RPS), and sequential preferential sampling (SPS)—to sample a balanced training dataset [21]. However, these techniques were only applied to data collected from one apartment. Their generalizability (i.e., the extent to which the study results could be applied to other situations [25]) to other apartments/buildings could not be indicated. Besides, among these techniques, SS shows the advantage of capturing the most recent pattern from the original training dataset. It could be helpful to preserve the predictive accuracy for situations where the data pattern changes with time. However, the data collection time required by SS to produce a balanced training dataset could be too long if the collected data is extremely imbalanced. Therefore, it would be interesting to propose a pre-processing technique that maintains the recent information from original training dataset and produces a balanced training dataset at the first time of implementation.

1.2. Literature review

The predictive performance of DDBMs depends on the representativeness of training dataset, such as proper selected features [26], data quality [27], and balanced data [28]. Imbalanced training dataset may result in perfect predictive result on majority classes that are represented by most samples but poor performance on minority conditions. Approaches to deal with imbalanced training data could be commonly divided into two categories: 1) Data pre-processing methods. Note that data pre-processing methods usually manipulate data by specific tasks, such as data cleaning, data scaling, data transformation, data reduction, data partitioning, and data augmentation, etc., before it is used [29]. In this paper, data pre-processing methods refer to produce a balanced training dataset; 2) Algorithm-based methods, or also called cost-sensitive learning. This kind of methods assigns different misclassification costs to data from different classes, and thus, forces the classifier concentrate on minority classes or classes that desire higher predictive accuracy.

To produce a balanced training dataset, the fundamental of reviewed data pre-processing methods are mostly in line with stratified sampling that divides the data into homogeneous conditions, and then, samples data for each condition using probability sampling methods, such as random sampling or clustering [30]. To be more specific, when separating data into different conditions, discrimination would occur which means that some conditions may contain more data than others. To balance data among these conditions, data pre-processing methods will oversample for minority conditions and/or undersample for majority conditions.

Data pre-processing methods could be used to produce a balanced training dataset for fault detection and diagnosis (FDD). For instance, Fan et al. [31] applied an oversampling technique, called SMOTE, to oversample faulty samples before training the support vector machine (SVM) model for chiller fault diagnosis. The re-balanced training dataset improved diagnostic accuracy. However, if the oversampling size is larger than 100% of original data, the large volume of synthetic samples would increase the classification uncertainty. Similarly, Zhou et al. [32] applied SMOTE to balance data between normal status and faulty status of a variable refrigerant flow system. The balanced dataset could increase the fault detection accuracy by more than 43%. However, SMOTE may suffer the risk of changing data distribution and overfitting [33]. Yan et al. [34] increased data in faulty conditions among the training dataset by inserting confidently predicted samples by data-driven classifiers. It results in over 80% diagnostic accuracy for air handling units (AHUs) in summer and 89% in winter. Note that to apply this method, the inserted samples should be accurately predicted to represent the actual scene. Besides, data augmentation methods, such as generative model-based methods, could be used to enrich the training dataset by creating samples for conditions that initially do not have adequate respective data [29]. For instance, generative adversarial network (GAN) have been widely used for FDD of devices in buildings [1,35,36]. However, these techniques could be hard to train because of the problems of non-convergence and diminished gradient [37].

Cost-sensitive learning algorithms are also used in building engineering to improve classification accuracy of minority classes. For instance, Li et al. [38] set different user-defined misclassification costs for false negative instances and false positive instances in two-class classification problems, such as rolling-up shading prediction and rolling-down shading behavior classification. The cost matrix was integrated into the objective function to minimize the total expected cost when training classification models, such as RF, SVM, and DT. The results showed that the proposed cost-sensitive learning algorithm improved the classification accuracy of minority conditions, i.e., behavior (such as rolling-up shading) occurred situations in their study. Tang et al. [39] proposed a cost-sensitive extremely randomized trees algorithm for wind turbine generator fault detection. The proposed algorithm considered different misclassification costs between missed detection (false negative) instances and false alarm (false positive) instances when calculating misclassification cost gain for branch nodes and leaf nodes of extreme decision trees. Through adding higher misclassification cost for missed detection instances, it significantly decreased the missing detection rates compared to traditional randomized trees. Furthermore, AdaBoost models that assign higher weights for misclassified points are commonly used cost-sensitive models in FDD [40,41] and energy prediction [42].

Although cost-sensitive learning algorithms enable DDBMs to improve predictive accuracy of minority classes, proper definition and integration of misclassification costs could be challenging as it requires expertise in the target problem and data-driven model framework [43]. In contrast, data-preprocessing methods are usually easier to apply. When applied to new tasks/scenarios, they require simple modifications on the sampling size. Besides, their processed data could be directly used by various models without requiring for objective function modification. Furthermore, Weiss et al. [44] found that data-preprocessing methods perform better on smaller dataset, while cost-sensitive learning outperforms sampling methods if the training dataset contains more than 10,000 samples.

Moreover, it is worth to mention that even if above reviewed approaches are widely used for imbalanced dataset, they have not been used to solve fairness problems in the area of building engineering application. Besides, existing studies usually apply the studied approaches to a specific building/system. Thus, it would be a challenge to evaluate their generalizability. For instance, most existing data-driven building models were analyzed in a case study of a single building, so their predictive performance could not be guaranteed for other buildings [45]. In fact, the generalizability of data-driven models is hard to summarize [46], as their predictive performance relies on many factors, such as input features, the scope of training dataset, and hyperparameters, etc.

Similarly, widely applied fairness-improvement techniques can be grouped into: 1) Pre-processing: preprocesses training data to remove discrimination before the training phase; 2) In-processing: adds fairness-related constraints or penalties to the model's optimization objective during the training phase [47]; and 3) Post-processing: modifies a classifier's predictive results to achieve fairness [48].

As pre-processing techniques are usually easy to implement, they have been used to solve *Type II* fairness problems. Their function is to re-balance the training dataset under different conditions defined by the protected attribute and predictive output. For instance, Kamiran and Calders [49] suggested uniform sampling and preferential sampling techniques to achieve fairness among classification problems with a binary protected attribute, such as income prediction and crime prediction with selecting gender as the protected attribute. In the previous study [21], we compared three proposed pre-processing techniques with uniform sampling and preferential sampling in terms of their ability to obtain similar lighting status prediction accuracy under different conditions defined by the protected attribute (i.e., motion status). These techniques could improve *Type II* fairness while preserving accuracy. However, this study relies on data gathered from only one residential unit. Thus, further research is needed to demonstrate the generalizability of these approaches based on training dataset with different structures. Note that the generalizability of these pre-processing techniques should be easier to evaluate than that of data-driven models given there are less hyperparameters.

1.3. Objective and contributions

This study is aimed at 1) proposing a pre-processing technique (sequentially balanced sampling (SBS)) that maintains the recent pattern from the original training dataset and produces a balanced training dataset at the first time of implementation; 2) evaluating the generalizability of this proposed technique and three previously proposed pre-processing techniques — SS, RPS, and SPS — to improve predictive performance similarity between different conditions (*Type II* fairness). These techniques are compared with RS that randomly samples data for each condition and reference cases that does not apply any pre-processing technique. To do so, a case study is conducted to apply these techniques to process training data for two-class classification problems (lighting status prediction of 155 lights in 16 apartments) with a binary protected attribute (motion status). The data could be classified into 5 modes based on the distribution on the protected attribute and target output. Then, 4 types of classifiers are developed based on the processed data. The predictive performance is evaluated in terms of accuracy measures and fairness measures. Accordingly, the generalizability of a pre-processing technique is assessed by its ability to improve any accuracy/fairness measure no matter what mode the training dataset is distributed in and/or what types of data-driven models are developed.

Therefore, the contributions of this work include: 1) propose the SBS technique to address the shortcoming of SS that fails to sample a balanced training dataset at the beginning stage of implementation; 2) verify the generalizability of the newly proposed technique and three previously proposed pre-processing techniques through a case study, and thus, provide guidance on selecting proper pre-processing techniques for imbalanced training dataset under different data distribution modes based on the preferred predictive performance (accuracy and/or fairness). The findings for these pre-processing techniques could be easily applied to other problems in building and indoor environment, if their data distribution could be classified as one data mode in this paper and their target performance is achieved by one of the pre-processing techniques.

The paper is organized as follows: Section 2 introduces the pre-processing techniques. Section 3 uses a case study to evaluate their generalizability. The accuracy and fairness measures used to evaluate predictive performance are described in Section 4. In Section 5, results are analyzed in terms of accuracy measures and fairness measures. Further discussion on results is presented in Section 6. Finally, Section 7 provides the conclusion for this study.

2. Methodology

The newly proposed sequentially balanced sampling (SBS) technique, three previously proposed pre-processing techniques (namely SS, RPS, and SPS), as well as random sampling (RS) will be introduced in this section. Their purpose is processing the original candidate training dataset ($X_{candidate}$) to generate a designed training dataset ($X_{designed}$) that distributes evenly among conditions defined by the protected attribute and output labels.

For two-class classification problems with a binary protected attribute, there would be four conditions, i.e., *PP, PN, NP,* and *NN*, as listed in Table 1. For example, in lighting status (ON/OFF) prediction considering motion status (ON/OFF) as the protected attribute, *PP* means the condition that lighting status is ON and motion status is ON; *PN* refers to the period during which lighting status is OFF and motion status is ON; *NP* is the situation that lighting status is ON and motion status is OFF; *NN* means that lighting status and motion status are OFF. In addition, the number of samples in these conditions are represented by |*PP*|*,* |*PN*|*,* |*NP*|*,* and |*NN*|, respectively. To eliminate bias among these conditions, the expected number of data points for each condition in $X_{designed}$ is calculated using Equation 1.

Table 1: PP, PN, NP and NN defined by the protected attribute and output labels

		Y	
		Positive	Negative
S	Positive	PP	PN
	Negative	NP	NN

Note *S* is the protected attribute, and *Y* is the class label of the training point.

$$|PP|_{design} = |PN|_{design} = |NP|_{design} = |NN|_{design} = 0.25 * \left| X_{designed} \right| \quad (1)$$

where $|PP|_{design}$, $|PN|_{design}$, $|NP|_{design}$, and $|NN|_{design}$ is the expected number of data points in *PP, PN, NP,* and *NN* of $X_{designed}$, respectively; $\left| X_{designed} \right|$ is the size of $X_{designed}$.

For multi-class classification problems with multi-class protected attributes, such as a *i*-class prediction problem with a *j*-class protected attribute, there would be *i*j* conditions (see Table 2) and the expected number of data points in each condition is $\frac{1}{i*j}|X_{designed}|$.

Table 2: Conditions defined by a *i*-class prediction problem with a *j*-class protected attribute

	Y

		Y_1	Y_2	…	Y_i
S	S_1	S_1Y_1	S_1Y_2	…	S_1Y_i
	S_2	S_2Y_1	S_2Y_2	…	S_2Y_i
	…	…		…	…
	S_j	S_jY_1	S_jY_2	…	S_jY_i

Note $SjYi$ is the condition in which data's protected attribute is Sj and output label is Yi.

In the following subsections, detailed procedure of these techniques is mainly explained for two-class classification problems with a binary protected attribute, while additional explain for multi-class classification problems with multi-class protected attributes is provided when it is necessary. In general, the procedure could be simply modified for multi-class classification problems with multi-class protected attributes through considering more conditions in each step.

2.1. Sequentially balanced sampling (SBS)

Sequentially balanced sampling (SBS) is aimed at getting a balanced training dataset at the first time of processing $X_{candidate}$ with preserving the latest information from $X_{candidate}$. Its procedure is as follows:

Step 1. Partitioning samples in $X_{candidate}$ into *PP, PN, NP,* and *NN*.

Step 2. List the data in each condition in descending order of collection time.

Step 3. Sample the most recently collected $0.25*|X_{designed}|$ data points from each condition. If the number of points in one condition is less than $0.25*|X_{designed}|$, it would duplicate the most recently collected data until the data in that condition reaches $0.25*|X_{designed}|$.

These steps are summarized in Figure 1. Detailed coding algorithms for SBS and other techniques are presented in the supplementary information.

In addition, for a i-class prediction problem with a j-class protected attribute, Step 1 would divide samples into $i*j$ conditions, while Step 3 would sample $\frac{1}{i*j}|X_{designed}|$ recently collected data for each condition.

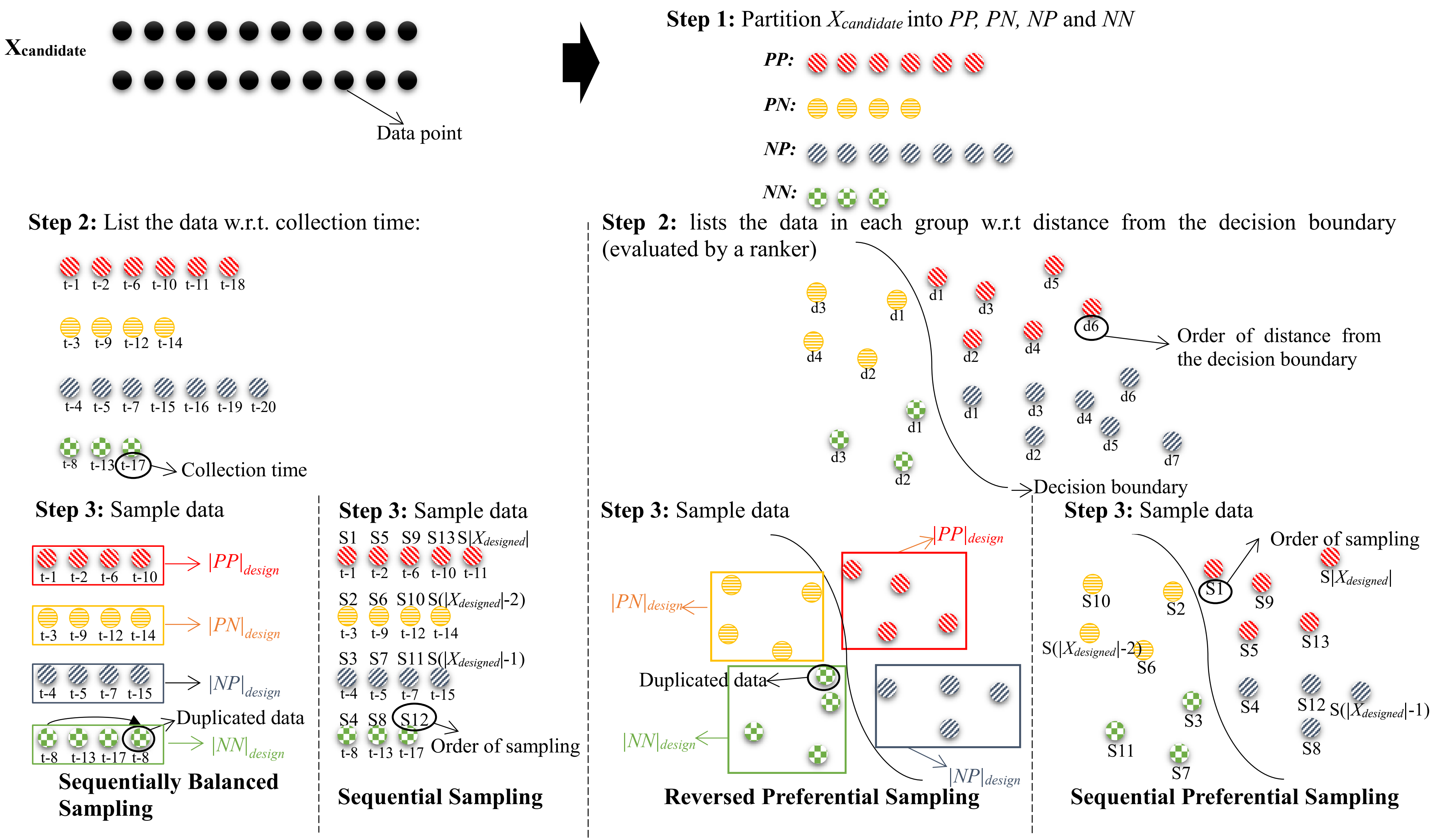

Xcandidate
Data point
Step 1: Partition $X_{candidate}$ into PP, PN, NP and NN
PP:
PN:
NP:
NN:
Step 2: List the data w.r.t. collection time:
t-1 t-2 t-6 t-10 t-11 t-18
t-3 t-9 t-12 t-14
t-4 t-5 t-7 t-15 t-16 t-19 t-20
t-8 t-13 t-17
Collection time
Step 2: lists the data in each group w.r.t distance from the decision boundary (evaluated by a ranker)
d1 d2 d3 d4 d5 d6
Order of distance from the decision boundary
Decision boundary
Step 3: Sample data
$|PP|_{design}$
$|PN|_{design}$
$|NP|_{design}$
$|NN|_{design}$
Duplicated data
Sequentially Balanced Sampling
S1 S5 S9 S13 $S|X_{designed}|$
S2 S6 S10 $S(|X_{designed}|-2)$
S3 S7 S11 $S(|X_{designed}|-1)$
S4 S8 S12
Order of sampling
Sequential Sampling
Reversed Preferential Sampling
S1 S2 S3 S4 S5 S6 S7 S8 S9 S10 S11 S12 S13
$S|X_{designed}|$
$S(|X_{designed}|-2)$
$S(|X_{designed}|-1)$
Sequential Preferential Sampling

Figure 1: Procedure of SBS, SS, RPS, and SPS in a two-class classification problem with a binary protected attribute

2.2. Sequential sampling (SS)

Sequential sampling (SS) also captures the latest information from $X_{candidate}$, but it will make $X_{designed}$ balance as the time of updating $X_{candidate}$ increases and the number of data points in each condition of $X_{candidate}$ are more than the expected number.

As shown in Figure 1, its first two steps are the same as SBS, the difference is in Step 3: SS samples a data point from each condition to $X_{designed}$ each time by ascending order, until reaching $|X_{designed}|$. To be more specific, it samples the first point from *PP, PN, NP,* and *NN* in turns at the first time. Then, it gets the second data point and next round the third point. After each round, the sampling will move to the next data. If all data points in one condition are sampled but this condition in $X_{designed}$ still does not get $0.25*|X_{designed}|$ points, SS will continue sample data from the next condition. In other words, when newly observed data comes to update $X_{candidate}$, it will catch this data to $X_{designed}$, and then, delete one old data from the majority condition.

2.3. Reversed preferential sampling (RPS)

Reversed preferential sampling (RPS) was suggested following the hypothesis that duplicating data close to the decision boundary (a hypersurface separating the dataset into two classes) in minority conditions will help distinguish the decision boundary, while removing data furthest from the decision boundary in majority conditions could avoid making large changes to the decision boundary. Its procedure is explained as below.

Step 1. The same as SBS.

Step 2. List the data in each condition in ascending order of distance from the decision boundary. For conditions with *Y*=Positive, data's distance from the decision boundary is represented by its possibility of classifying as positive (*p_positive*), while evaluated by the probability of negative (*p_negative*) when *Y*=Negative. In this study, *p_positive* and *p_negative* are calculated by a ranker, in which a probabilistic classifier, such as logistic regression (LR) or Naïve Bayes (NB), is trained using $X_{candidate}$. Detailed algorithm for the ranker is coded in the supplementary information. For multi-class classification problems, data is listed by the ascending order of correct prediction probability. For instance, in condition S_jY_i, data is listed by the ascending order of the possibility of predicting a data as class Y_i.

Step 3. If the number of actual samples in one condition is above $0.25*|X_{designed}|$, slice the first $0.25*|X_{designed}|$ training points to $X_{designed}$. Otherwise, duplicate points in that condition by ascending order until reaching $0.25*|X_{designed}|$ and sample them to $X_{designed}$.

2.4. Sequential preferential sampling (SPS)

Sequential preferential sampling (SPS) gradually gets a balanced training dataset, while maintaining the most representative data for distinguishing the decision boundary. Therefore, its difference from the RPS is in Step 3: SPS iteratively samples a training point each time from the four conditions in $X_{candidate}$ to $X_{designed}$, in ascending order of distance from the decision boundary. Like SS, it will get a balanced training dataset as the times of updating $X_{candidate}$ increases.

2.5. Random Sampling (RS)

Random sampling (RS) randomly samples the expected number of data points from each condition in $X_{candidate}$ to $X_{designed}$. For instance, in two-class classification problems with a binary protected attribute, when the actual points in one condition are more than $0.25*|X_{designed}|$, randomly select $0.25*|X_{designed}|$ training points from that condition to $X_{designed}$. Otherwise, randomly duplicate points in that condition until $0.25*|X_{designed}|$ is reached, and then sample these data to $X_{designed}$.

3. Case study for generalizability investigation

To investigate the generalizability of the proposed pre-processing techniques, a case study is designed to extend their application to data collected from 16 apartments. These data could be classified into 5 modes based on their distribution on the output labels (ON/OFF lighting status) and protected attribute labels (ON/OFF motion status). Detailed description of the collected data and study cases is shown in Section 3.1 and Section 3.2, respectively.

3.1.Data description

The data used is collected from 16 apartments in a residential building located in Lyon, France. These 16 apartments present two types of lay-out: Lay-out Type I (Apt #1 to Apt #8) and Lay-out Type II (Apt #9 to Apt #16), as shown in Figure 2. Motion status and lighting status are collected from these apartments by presence sensors and lighting sensors, respectively. A detailed description of installed sensors is presented in Table 3. Besides, weather information, such as altitude of the sun, global horizontal illuminance, diffuse horizontal illuminance, global horizontal irradiance, and diffuse horizontal irradiance, etc., is collected from a local weather station.

The data was collected for the year 2016 with one-minute time intervals and processed to 5-minute intervals. Further analysis were carried out for missing data and outliers [24,50]. The remaining samples for each light in each apartment are presented in Figure 3.

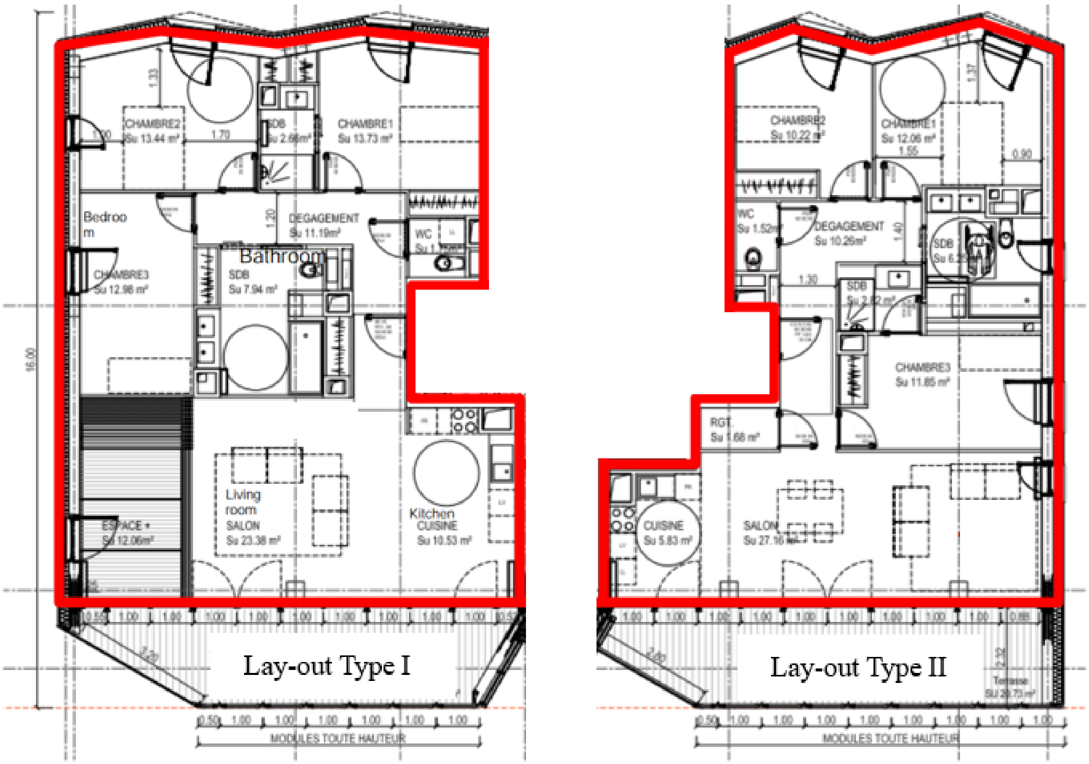


Figure 2: Lay-out of studied apartments

Table 3: Description of sensors

Sensor	**Company name**	**Type**	**Accuracy**	**No. of sensors in Lay-out Type I**	**No. of sensors in Lay-out Type II**
Presence Detector	Theben	PlanoCentro A-KNX	-(detection area 64 m^2 if seated)	14	14
Lighting Sensor	ABB	KNX Energy Module: EM/S 3.16.3	±2/3/6%	14	13

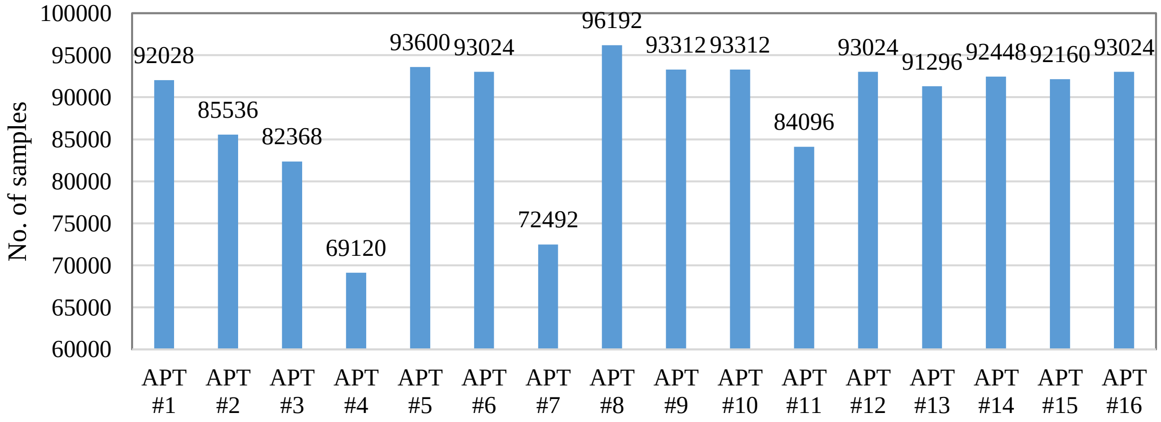


Figure 3: Number of samples for each apartment

Except the motion status data collected from presence sensors, one attribute called 'Motion Status_total' is added to each apartment to represent the overall motion status in the corresponding apartment. It is recorded as ON if at least one presence sensor in that apartment detected the ON signal. Besides, if a light is in one status (such as turn OFF) all the time, its corresponding lighting status attribute would be excluded from the collected dataset. As a result, data collected from 155 lights remain in the training dataset and work as data-driven models' outputs.

In this investigation, 'Motion Status_total' is the protected attribute, while lighting status the classifiers' outputs. The ratios of conditions *PP, PN, NP,* and *NN* for different lights in these 16 apartments are presented in Figure 4. Based on the distribution of *PP, PN, NP* and *NN*, the dataset could be separated into 5 modes. A description of each mode is in Table 4.

Table 4: Description of data modes

Mode No.	**Description**	**Output of Data**	**Number of Lights**
1	The light is OFF most of the time (>90%) and 'Motion Status_total' is OFF 60-70%. **Detailed distribution:** *NN* contains most samples (55-70%), followed by *PN*(20-40%), *PP*(0-8%) and *NP*(0-8%).	**APT #1:** Light_1, Light_4, Light_7, Light_8, Light_9, Light_10, Light_11, Light_12. **APT #2:** Light_2, Light_3, Light_4, Light_5, Light_6, Light_7, Light_8, Light_9, Light_10, Light_11, Light_12. **APT #4:** Light_1, Light_2, Light_3, Light_4, Light_5, Light_6, Light_7, Light_8, Light_9, Light_10, Light_11, Light_12. **APT #6:** Light_1, Light_3, Light_4, Light_5, Light_6, Light_7, Light_8, Light_9. **APT #7:** Light_1, Light_2, Light_3, Light_4. **APT #8:** Light_1, Light_2, Light_3, Light_4, Light_5, Light_6, Light_7, Light_8, Light_9, Light_10, Light_11, Light_12, Light_13. **APT #11:** Light_1, Light_2, Light_3, Light_5, Light_6, Light_7, Light_8. **APT #12:** Light_1, Light_2, Light_3, Light_4, Light_5, Light_7, Light_8 Light_9, Light_10, Light_11. **APT #14:** Light_1, Light_2, Light_3, Light_4, Light_5, Light_6, Light_7, Light_8, Light_9, Light_10. **APT #16:** Light_1, Light_2, Light_3, Light_4, Light_5, Light_6, Light_7, Light_8, Light_9.	**92**
2	The light is OFF ~90% of the time, and 'Motion Status_total' is OFF ~70% of the time.	**APT #1:** Light_2, Light_3, Light_5, Light_6. **APT #2:** Light_1. **APT #6:** Light_2.	**6**
3	The light status and 'Motion Status_total' are OFF most of the time (>90%).	**APT #3:** Light_1, Light_2, Light_3, Light_4, Light_5, Light_6, Light_7, Light_8. **APT #10:** Light_1, Light_2, Light_3, Light_4.	**12**
4	The light is turned OFF most of the time (>90%), and the 'Motion Status_total' is evenly distributed with ~50% 'OFF' labels.	**APT #5:** Light_1, Light_2, Light_4, Light_5, Light_6, Light_7, Light_8, Light_9, Light_10, Light_11, Light_12. **APT #9:** Light_1, Light_2, Light_3, Light_4, Light_5, Light_6, Light_7, Light_8, Light_9, Light_10, Light_11. **APT #13:** Light_1, Light_2, Light_3, Light_4, Light_5, Light_6, Light_7, Light_8, Light_9, Light_10. **APT #15:** Light_1, Light_2, Light_3, Light_4, Light_5, Light_6, Light_7, Light_8, Light_9, Light_10.	**42**
5	The 'OFF' light status is <50%.	**APT #5:** Light_3. **APT #11:** Light_4. **APT #12:** Light_6.	**3**

		PP	PN	NP	NN
APT #1	Light_1	0.10%	31.46%	0.00%	68.44%
	Light_2	8.19%	23.37%	2.14%	66.30%
	Light_3	9.09%	22.47%	2.80%	65.64%
	Light_4	0.29%	31.27%	0.05%	68.39%
	Light_5	10.44%	21.12%	2.69%	65.75%
	Light_6	10.86%	20.70%	3.20%	65.24%
	Light_7	1.01%	30.55%	0.11%	68.33%
	Light_8	0.84%	30.72%	0.15%	68.29%
	Light_9	0.57%	30.99%	0.03%	68.41%
	Light_10	0.18%	31.38%	0.02%	68.42%
	Light_11	0.31%	31.25%	0.02%	68.42%
	Light_12	0.32%	31.24%	0.02%	68.42%
APT #2	Light_1	9.00%	24.47%	10.25%	56.28%
	Light_2	3.28%	30.20%	0.32%	66.20%
	Light_3	2.47%	31.00%	0.50%	66.02%
	Light_4	1.81%	31.67%	0.43%	66.09%
	Light_5	1.09%	32.39%	0.03%	66.50%
	Light_6	2.18%	31.30%	0.07%	66.46%
	Light_7	1.49%	31.99%	0.05%	66.48%
	Light_8	0.81%	32.67%	0.09%	66.44%
	Light_9	2.34%	31.14%	0.09%	66.43%
	Light_10	0.87%	32.61%	0.06%	66.46%
	Light_11	1.76%	31.72%	0.18%	66.35%
	Light_12	2.93%	30.55%	0.23%	66.29%
APT #3	Light_1	1.67%	8.33%	1.42%	88.57%
	Light_2	3.12%	6.89%	0.85%	89.15%
	Light_3	1.80%	8.21%	0.25%	89.75%
	Light_4	4.52%	5.48%	0.72%	89.27%
	Light_5	5.02%	4.98%	0.86%	89.13%
	Light_6	1.83%	8.18%	0.29%	89.70%
	Light_7	0.77%	9.24%	0.23%	89.76%
	Light_8	0.40%	9.61%	0.16%	89.84%
APT #4	Light_1	3.40%	32.73%	0.51%	63.37%
	Light_2	4.60%	31.52%	0.57%	63.30%
	Light_3	3.09%	33.03%	0.30%	63.58%
	Light_4	0.36%	35.76%	0.03%	63.85%
	Light_5	4.93%	31.19%	1.97%	61.90%
	Light_6	2.73%	33.40%	2.83%	61.05%
	Light_7	0.23%	35.89%	0.01%	63.87%
	Light_8	0.89%	35.23%	0.05%	63.83%
	Light_9	5.15%	30.97%	0.23%	63.64%
	Light_10	1.68%	34.45%	0.16%	63.72%
	Light_11	2.80%	33.32%	0.08%	63.80%
	Light_12	3.76%	32.36%	0.14%	63.74%
APT #5	Light_1	5.15%	45.33%	0.06%	49.46%
	Light_2	9.79%	40.69%	0.17%	49.34%
	Light_3	32.15%	18.34%	30.23%	19.29%
	Light_4	5.90%	44.58%	0.23%	49.29%
	Light_5	3.84%	46.64%	0.27%	49.25%
	Light_6	14.99%	35.49%	0.48%	49.03%
	Light_7	1.27%	49.22%	0.59%	48.93%
	Light_8	0.85%	49.63%	0.04%	49.48%
	Light_9	9.83%	40.65%	0.02%	49.50%
	Light_10	4.27%	46.21%	0.18%	49.33%
	Light_11	4.42%	46.06%	0.01%	49.51%
	Light_12	6.43%	44.06%	0.02%	49.49%
APT #6	Light_1	4.81%	22.92%	0.19%	72.08%
	Light_2	8.51%	19.23%	0.66%	71.61%
	Light_3	2.04%	25.69%	0.07%	72.20%
	Light_4	6.00%	21.74%	0.35%	71.92%
	Light_5	1.18%	26.55%	0.06%	72.21%
	Light_6	1.26%	26.48%	0.15%	72.12%
	Light_7	1.50%	26.24%	0.13%	72.14%
	Light_8	1.47%	26.26%	0.09%	72.17%
	Light_9	2.03%	25.71%	0.10%	72.17%
APT #7	Light_1	0.54%	31.62%	0.36%	67.49%
	Light_2	1.50%	30.65%	0.40%	67.45%
	Light_3	2.88%	29.27%	0.19%	67.66%
	Light_4	0.48%	31.67%	0.22%	67.63%
APT #8	Light_1	3.32%	25.84%	1.28%	69.56%
	Light_2	3.20%	25.96%	1.54%	69.30%
	Light_3	3.98%	25.18%	2.81%	68.03%
	Light_4	3.12%	26.04%	1.70%	69.14%
	Light_5	3.44%	25.72%	1.17%	69.67%
	Light_6	4.02%	25.14%	1.62%	69.23%
	Light_7	2.30%	26.86%	0.85%	69.99%
	Light_8	1.93%	27.23%	0.45%	70.39%
	Light_9	0.80%	28.36%	0.35%	70.49%
	Light_10	2.98%	26.18%	0.92%	69.92%
	Light_11	2.29%	26.87%	0.87%	69.97%
	Light_12	2.17%	26.99%	0.82%	70.03%
	Light_13	3.34%	25.82%	3.11%	67.74%
APT #9	Light_1	4.92%	43.64%	0.56%	50.87%
	Light_2	3.82%	44.74%	0.10%	51.34%
	Light_3	5.90%	42.67%	0.39%	51.04%
	Light_4	7.61%	40.95%	1.42%	50.02%
	Light_5	5.31%	43.26%	0.49%	50.95%
	Light_6	3.36%	45.20%	0.39%	51.04%
	Light_7	8.07%	40.50%	0.51%	50.93%
	Light_8	6.34%	42.23%	0.38%	51.06%
	Light_9	5.27%	43.30%	1.27%	50.17%
	Light_10	4.59%	43.97%	0.28%	51.16%
	Light_11	3.08%	45.48%	0.27%	51.16%
APT #10	Light_1	1.94%	3.62%	0.27%	94.17%
	Light_2	1.36%	4.20%	0.13%	94.32%
	Light_3	2.23%	3.33%	0.20%	94.25%
	Light_4	1.64%	3.92%	0.23%	94.21%
APT #11	Light_1	2.33%	30.47%	1.06%	66.14%
	Light_2	3.67%	29.13%	0.11%	67.09%
	Light_3	4.00%	28.80%	0.37%	66.82%
	Light_4	32.23%	0.58%	66.24%	0.96%
	Light_5	3.07%	29.73%	0.08%	67.12%
	Light_6	1.26%	31.54%	0.25%	66.94%
	Light_7	1.30%	31.51%	0.04%	67.16%
	Light_8	0.20%	32.60%	0.10%	67.09%
APT #12	Light_1	1.69%	39.37%	0.13%	58.81%
	Light_2	4.08%	36.98%	0.28%	58.67%
	Light_3	3.20%	37.86%	1.44%	57.51%
	Light_4	5.09%	35.97%	1.95%	57.00%
	Light_5	0.36%	40.69%	0.09%	58.85%
	Light_6	22.27%	18.78%	28.14%	30.81%
	Light_7	1.09%	39.97%	0.26%	58.69%
	Light_8	6.57%	34.48%	0.84%	58.11%
	Light_9	8.05%	33.01%	1.14%	57.80%
	Light_10	3.20%	37.86%	0.47%	58.47%
	Light_11	2.13%	38.93%	0.15%	58.80%
APT#13	Light_1	1.20%	53.43%	0.04%	45.33%
	Light_2	4.43%	50.20%	0.41%	44.96%
	Light_3	3.22%	51.42%	0.28%	45.09%
	Light_4	7.72%	46.91%	0.91%	44.45%
	Light_5	2.34%	52.30%	0.07%	45.30%
	Light_6	1.57%	53.06%	0.04%	45.33%
	Light_7	4.25%	50.38%	0.10%	45.27%
	Light_8	3.26%	51.37%	0.11%	45.26%
	Light_9	3.64%	51.00%	0.10%	45.27%
	Light_10	1.14%	53.49%	0.09%	45.27%
APT#14	Light_1	3.76%	32.27%	1.14%	62.83%
	Light_2	2.83%	33.20%	0.09%	63.88%
	Light_3	0.77%	35.26%	0.08%	63.89%
	Light_4	1.61%	34.42%	0.18%	63.79%
	Light_5	2.17%	33.87%	0.27%	63.70%
	Light_6	1.80%	34.23%	0.28%	63.69%
	Light_7	6.81%	29.22%	3.32%	60.64%
	Light_8	3.75%	32.28%	1.16%	62.81%
	Light_9	3.26%	32.77%	1.20%	62.77%
	Light_10	1.26%	34.77%	0.10%	63.87%
APT#15	Light_1	5.89%	44.84%	0.23%	49.05%
	Light_2	3.33%	47.39%	0.33%	48.95%
	Light_3	2.20%	48.53%	0.08%	49.20%
	Light_4	4.41%	46.32%	0.16%	49.11%
	Light_5	3.50%	47.23%	0.58%	48.70%
	Light_6	7.70%	43.03%	0.49%	48.79%
	Light_7	2.29%	48.43%	0.37%	48.90%
	Light_8	1.07%	49.65%	0.02%	49.25%
	Light_9	8.10%	42.62%	0.19%	49.08%
	Light_10	2.39%	48.34%	0.05%	49.22%
APT#16	Light_1	0.74%	35.89%	0.06%	63.31%
	Light_2	3.20%	33.42%	1.08%	62.29%
	Light_3	1.56%	35.07%	0.08%	63.29%
	Light_4	2.47%	34.16%	0.44%	62.94%
	Light_5	2.95%	33.67%	1.16%	62.21%
	Light_6	2.75%	33.87%	0.74%	62.64%
	Light_7	1.90%	34.73%	0.06%	63.31%
	Light_8	1.63%	35.00%	0.23%	63.14%
	Light_9	1.08%	35.55%	0.09%	63.28%

Figure 4: Ratio of PP, PN, NP, and NN among 16 apartments

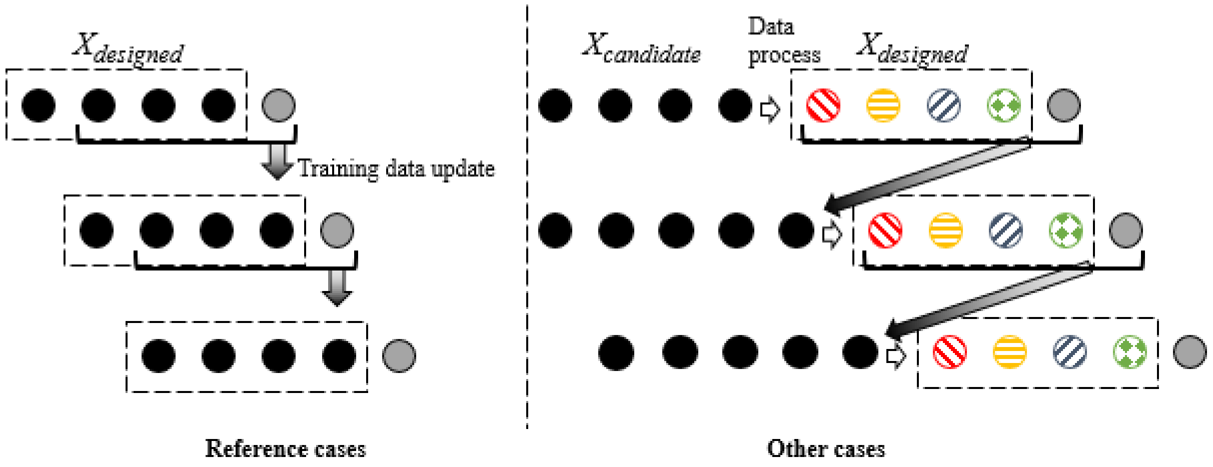
3.2.Study cases

Eight kinds of cases are designed and named by the utilized pre-processing techniques, see Table 5. The suffix (LR or NB) for RPS and SPS indicates the ranker (LR or NB) utilized in the corresponding pre-processing technique. Note that these case studies are applied to 16 apartments with totally 155 lightings. For each kind of cases, 4 types of classifiers (i.e., SVM, ANN, LR, NB) are utilized. Therefore, a total of 4960 cases are investigated (the combination of 8 types of cases, 4 types of classifiers and 155 types of outputs (8*4*155=4960)). In other words, there are 2944 (8*4*92) cases for Mode 1, 192 (8*4*6) cases for Mode 2, 384 (8*4*12) cases for Mode 3, 1344 (8*4*42) cases for Mode 4, and 96 (8*4*3) cases for Mode 5, respectively.

Table 5: Description of study cases

Case Type	Pre-processing techniques	Inputs	Classifiers	Output
Reference Case		Hour of The Day, Day of The Week, Time of The Day, Altitude of The Sun, Global Horizontal Illuminance, Diffuse Horizontal Illuminance, Global Horizontal Irradiance, Diffuse Horizontal Irradiance, Motion Status	SVM, ANN, LR, NB	Lighting status (ON/OFF)
RS	Random Sampling			
SS	Sequential Sampling			
SBS	Sequentially Balanced Sampling			
RPS_LR	Reversed Preferential Sampling with using Logistic Regression as a ranker			
RPS_NB	Reversed Preferential Sampling with using Naïve Bayes as a ranker			
SPS_LR	Sequential Preferential Sampling using Logistic Regression as a ranker			
SPS_NB	Sequential Preferential Sampling with using Naïve Bayes as a ranker			

Training and validation procedure of these cases are present in Figure 5. For reference cases, classifiers are trained by previous four weeks' data to predict next week's lighting status. The training data is updated once a week by the newly observed data. The procedure for other cases is: Firstly, the $X_{candidate}$ is processed by the corresponding pre-processing technique to produce the $X_{designed}$ that contains four weeks' data. Next, $X_{designed}$ is used to train classifiers that will be used to predict one-week ahead lighting status. Then, the newly observed next week's data and $X_{designed}$ are updated as $X_{candidate}$ for next time's prediction.



Note: ● is the measured data, ○ is the predicted data, ⊘ ⊜ ⊘ ⊕ is the processed data by pre-processing techniques.

Figure 5: Training and validation procedure

To test if these pre-processing techniques show consistent effect on different classifiers' predictive performance, four types of commonly used classifiers (SVM, ANN, LR, and NB) with different mathematical nature are developed. Among these classifiers, SVM predicts the class label by maximizing the margin between different categories [51]; ANN predicts the output by trained activation functions of neurons in hidden layers and output layer; LR predicts the possibility of an object belonging to a positive class using a logistic function [52]; and NB conducts a prediction by applying Bayes' algorithm with the 'naive' assumption of conditional independence between attributes given the class label value [53]. NB classifiers are generally classified as three types: Gaussian NB, Multinomial NB, and Bernoulli NB. This study uses Gaussian NB.

As the predictive performance of classifiers is affected by their hyperparameters [54], the hyperparameters of classifiers in reference case are optimized by a RandomizedSearchCV function in Python [55]. It optimizes hyperparameters by randomly selecting a chosen number of hyperparametric pairs from a given domain and testing only those. The search space of hypermeters for each kind of classifier is listed in Table 6. Moreover, this case study is aimed at investigating the effect of pre-processing techniques on predictive results instead of the effect of classifiers' hyperparameters, thus, the same kind of classifiers in other cases use the same hyperparameters as in reference cases.

Table 6: Search space for hyperparameter optimization

Classifier	Hyperparameter	Description	Search space
SVM	C	Regularization parameter. The strength of the regularization is inversely proportional to C.	loguniform(1e0, 1e3)
ANN	solver	The solver for weight optimization	‘adam’, ‘lbfgs’
	hidden layer sizes	The i-th element represents the number of neurons in the i-th hidden layer.	(5,2), (3,3), (5,5), (4,4)
LR	solver	Algorithm to use in the optimization problem.	'newton-cg', 'lbfgs', 'liblinear'
	C	Regularization parameter. The strength of the regularization is inversely proportional to C.	loguniform(1e0, 1e3)
	Class weight	Weights associated with classes	‘balanced’, None
NB	var_smoothing	Portion of the largest variance of all features that is added to variances for calculation stability.	logspace(0,-9, num=100)

All simulations are performed by Python 3.7 on a laptop with Intel Core i7-7700HQ CPU @2.80GHz and 8GB of RAM.

4. Performance evaluation criteria

This section will introduce accuracy measures for evaluating the difference between predicted values and measured values, and fairness measures that could be helpful to indicate *Type II* fairness achievement through rating the difference of predictive performance between conditions defined by the protected attribute.

4.1.Accuracy measures

In this study, accuracy (Equation 2), recall (Equation 3), and specificity (Equations 4) are selected to evaluate the predictive accuracy for the studied two-class classification problems. Accuracy is the overall predictive accuracy, which means the rate of accurate predicted samples to the entire scope of samples. Recall stipulates the true positive rate, which is the rate of accurate prediction when *Y*=Positive. In other words, it reflects the predictive accuracy of *PP* and *NP* conditions. Specificity (also called true negative rate) is the portion of accurate prediction when *Y*=Negative. Thus, it shows the predictive accuracy for validation data in conditions *PN* and *NN*.

$$\text{Accuracy } = P\left[\hat{Y} = y \,\middle|\, Y = y\right] \tag{2}$$

$$\text{Recall } = P\left[\hat{Y} = \text{Positive} \,\middle|\, Y = \text{Positive}\right] \tag{3}$$

$$\text{Specificity } = P\left[\hat{Y} = \text{Negative} \,\middle|\, Y = \text{Negative}\right] \tag{4}$$

where $\hat{Y}$ means the predicted label, $\hat{Y} \in [\text{Negative}, \text{Positive}]$

4.2.Fairness measures

The accuracy measures under different conditions defined by the protected attribute (denoted by S) are defined as the group conditional accuracy measures, i.e., c-Accuracy (Equation 5), c-Recall (Equation 6) and c-Specificity (Equation 7), etc. On the one hand, when S = Positive, its conditional accuracy measures are called 1-Accuracy, 1-Recall and 1-Specificity. Note that 1-Accuracy reflects the overall predictive accuracy of *PP* and *PN*, while 1-Recall is the predictive accuracy of *PP* and 1-Specificity shows the accuracy of *PN*. On the other hand, for *S* = Negative, its conditional accuracy measures are called 0-Accuracy, 0-Recall and 0-Specificity. 0-Accuracy is the predictive accuracy of *NP* and *NN*, while 0-Recall is the accuracy of *NP* and 0-Specificity presents the predictive accuracy of *NN*.

$$\mathrm{c-Accuracy} = P\left[\hat{Y} = y \,\middle|\, Y = y, S = s\right] \tag{5}$$

$$\mathrm{c-Recall} = P\left[\hat{Y} = \mathrm{Positive} \,\middle|\, Y = \mathrm{Positive}, S = s\right] \tag{6}$$

$$c - Specificity = P\left[\hat{Y} = Negative \,\middle|\, Y = Negative, S = s\right] \tag{7}$$

where c shows the group conditional accuracy measures; $\mathrm{c} \in [0, 1]$.

To quantify the performance similarity between S = Positive and S = Negative, accuracy rate (Equation 8), recall rate (Equation 9) and specificity rate (Equation 10) are selected as fairness measures. Furthermore, to determine whether *Type II* fairness exists (i.e., predictive performance between different conditions is similar enough), “80 percent rule” [56] could be utilized. This rule illustrated that the predictive result is fair when the selected fairness measure is higher than 80%.

$$\text{Accuracy rate } = \frac{\min(1 - \mathrm{Accuracy}, 0 - \mathrm{Accuracy})}{\max(1 - \mathrm{Accuracy}, 0 - \mathrm{Accuracy})} \tag{8}$$

$$\text{Recall rate } = \frac{\min(1 - \mathrm{Recall}, 0 - \mathrm{Recall})}{\max(1 - \mathrm{Recall}, 0 - \mathrm{Recall})} \tag{9}$$

$$\text{Specificity rate } = \frac{\min(1 - \mathrm{Specificity}, 0 - \mathrm{Specificity})}{\max(1 - \mathrm{Specificity}, 0 - \mathrm{Specificity})} \tag{10}$$

5. Results

The predictive results of study cases are summarized in terms of accuracy measures (Section 5.1) and fairness measures (Section 5.2).

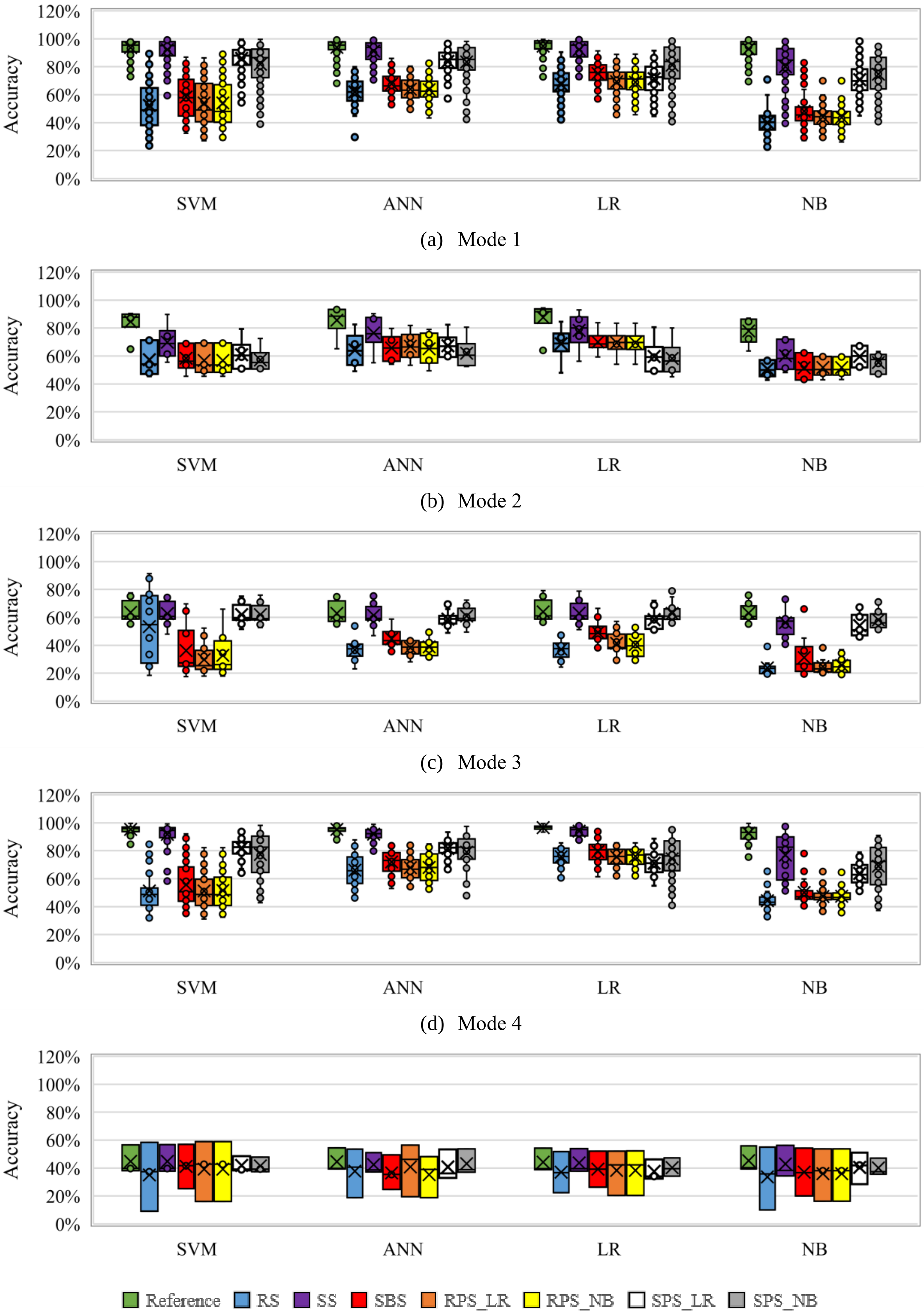
5.1. Results: accuracy

The overall accuracy (y axis) of different classifiers (x axis) trained based on data processed by different pre-processing techniques (legend), for lights under different modes (subfigure title) are summarized in Figure 6. It shows that reference cases only present a slight predictive accuracy variation between different classifiers in the same mode. However, when using the same classifier, reference cases under Mode 4 usually present the highest accuracy (higher than 90% mostly), followed by Mode 1 (higher than 85% mostly), Mode 2 (65% - 95%), Mode 3 (55% - 80%), and

Mode 5 (40% - 55%). This indicates that even if lights classified in Mode 1 to Mode 4 show the same lighting status pattern (i.e. turned OFF most of the time), their predictive accuracy would be affected by the motion status distribution. The more evenly the motion status is distributed, the higher the lighting status predictive accuracy. Furthermore, the accuracy of all cases in Mode 5 is quite low and not acceptable. It might be because of the irrelevance between inputs features and the output. However, it is still analyzed to show the general effect of pre-processing techniques on the predictive output.

For most cases, the RS shows the most harmful influence on the accuracy, followed by RPS. For instance, in Figure 6(a), RS results in the worst accuracy than other pre-processing techniques. RPS decreases the accuracy to be lower than 80% for most lights in Mode 1, Mode 2 and Mode 4, while the accuracy is dropped to be less than 50% in Mode 3. The newly proposed SBS shows slightly better accuracy than RPS, but its reduction effect on the overall accuracy is more significant than SS and SPS. Besides, the effect of ranker on the predictive accuracy is ignorable for RPS, while using LR as the ranker for SPS shows higher accuracy than using NB under Mode 2 and lower accuracy than NB under Mode 3.

In Mode 5, RS significantly decreases the overall accuracy for Light_4 of APT #11 from ~40% to ~ 20% when using ANN and LR and ~10% when using SVM and NB. RPS decreases the accuracy for Light_4 of APT #11 to be lower than ~20%, while SBS could maintain the accuracy at ~25%. On the other hand, the effect of RS, RPS, and SBS on Light_3 of APT #5 and Light_6 of APT #12 is slight and usually depends on classifiers. Furthermore, SS and SPS usually slightly decrease the overall predictive accuracy for Mode 5.



(a) Mode 1

(b) Mode 2

(c) Mode 3

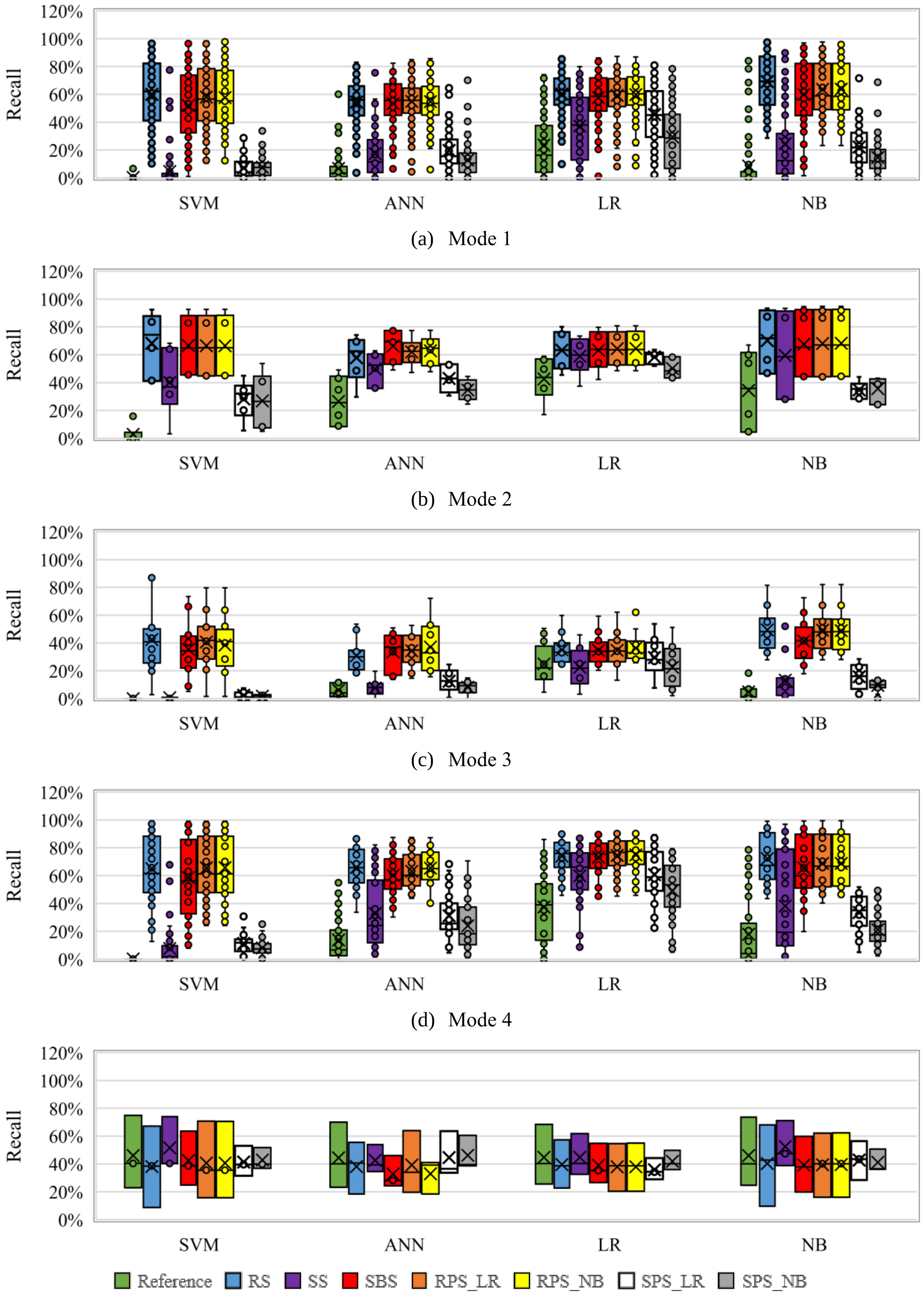
(d) Mode 4

(e) Mode 5

Figure 6: Accuracy for different modes

Besides, the effect of pre-processing techniques and classifiers on the recall are presented in Figure 7. For Mode 1 to Mode 4, the recall is zero for most of reference cases when using SVM. However, the recall of reference cases could be increased to ~5% - ~71% for Mode 1, ~17% - ~56% for Mode 2, ~5% - ~47% for Mode 3, and ~5% - ~86% for Mode 4, by using LR. Moreover, when using the same classifier for lights in Mode 1 to Mode 4, cases using RS, RPS, or SBS show significantly higher recall than cases using SS or SPS. This result is contrary to the overall accuracy. It is because that lights in Mode 1 to Mode 4 are turned OFF most of the time in $X_{candidate}$ and the validation dataset, thus, their overall accuracy is in line with the specificity (see Figure 8). When using these pre-processing techniques to process $X_{candidate}$, data with 'ON' lighting status would be increased in $X_{designed}$ and data with 'OFF' lighting status would be decreased. As a result, the recall would be increased, while specificity and accuracy would be decreased. Among these pre-processing techniques, RPS has the most powerful ability to process a balanced dataset while sampling the most representative data for distinguishing 'ON' class label, thus, it presents the highest recall improvement ability. Besides, SBS could get a balanced $X_{designed}$ at the first time of implementation and capture the people's most recent lighting usage habits, thus, it shows comparable recall improvement ability than RPS, especially in Mode 2. In addition, selecting different kinds of ranker would not affect the recall improvement ability of RPS. However, using LR as the ranker in SPS show higher recall than using NB.

However, in Mode 5, all pre-processing techniques, especially RS, decrease the recall for Light_3 in APT #5 and Light_4 in APT #11, because these two lights are mostly turned ON in $X_{candidate}$ and applying pre-processing techniques would decrease this ratio. By contrast, these pre-processing techniques increase the recall for Light_6 of APT #12. Moreover, SS presents the highest recall for Light_6 of APT #12, when utilizing SVM or NB.



(a) Mode 1

(b) Mode 2

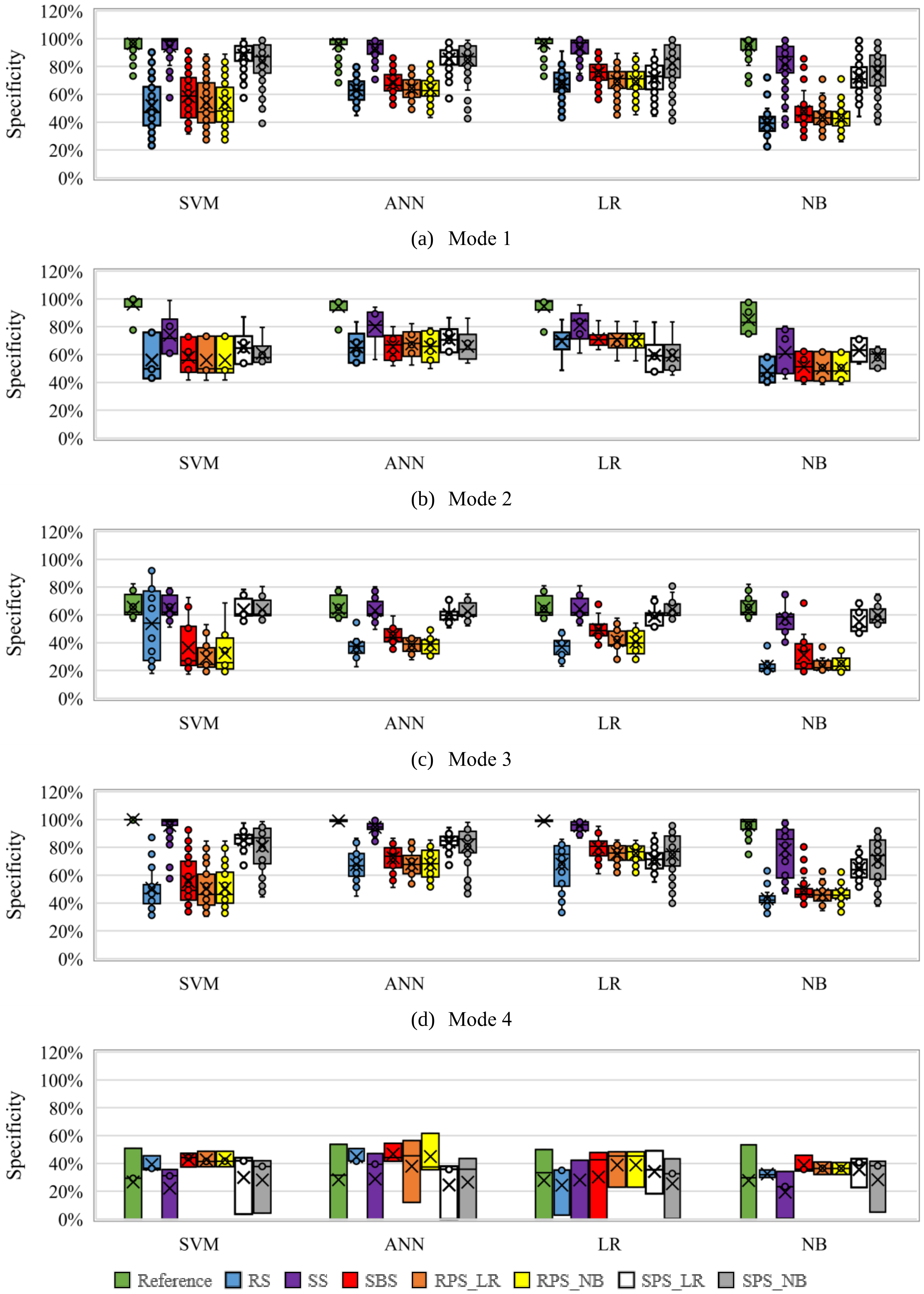
(c) Mode 3

(d) Mode 4

(e) Mode 5

Figure 7: Recall for different modes

As illustrated in previous paragraphs, for all cases in Mode 1 to Mode 4, the overall accuracy is in line with the specificity. Therefore, detailed description and analysis of specificity for these modes will not be presented. However, the situation is different for Mode 5 (see Figure 8(e)). Although SBS and RPS increase the mean specificity, they show a specificity reduction for Light_6 of APT #12, whose status was evenly distributed among ON/OFF classes. Besides, the effect of RS is not consistent for different classifiers. Furthermore, the SS slightly decreases the mean specificity for Mode 5, because of the least specificity increasing ability for Light_3 of APT #5 and Light_4 of APT #11.



(a) Mode 1

(b) Mode 2

(c) Mode 3

(d) Mode 4

(e) Mode 5

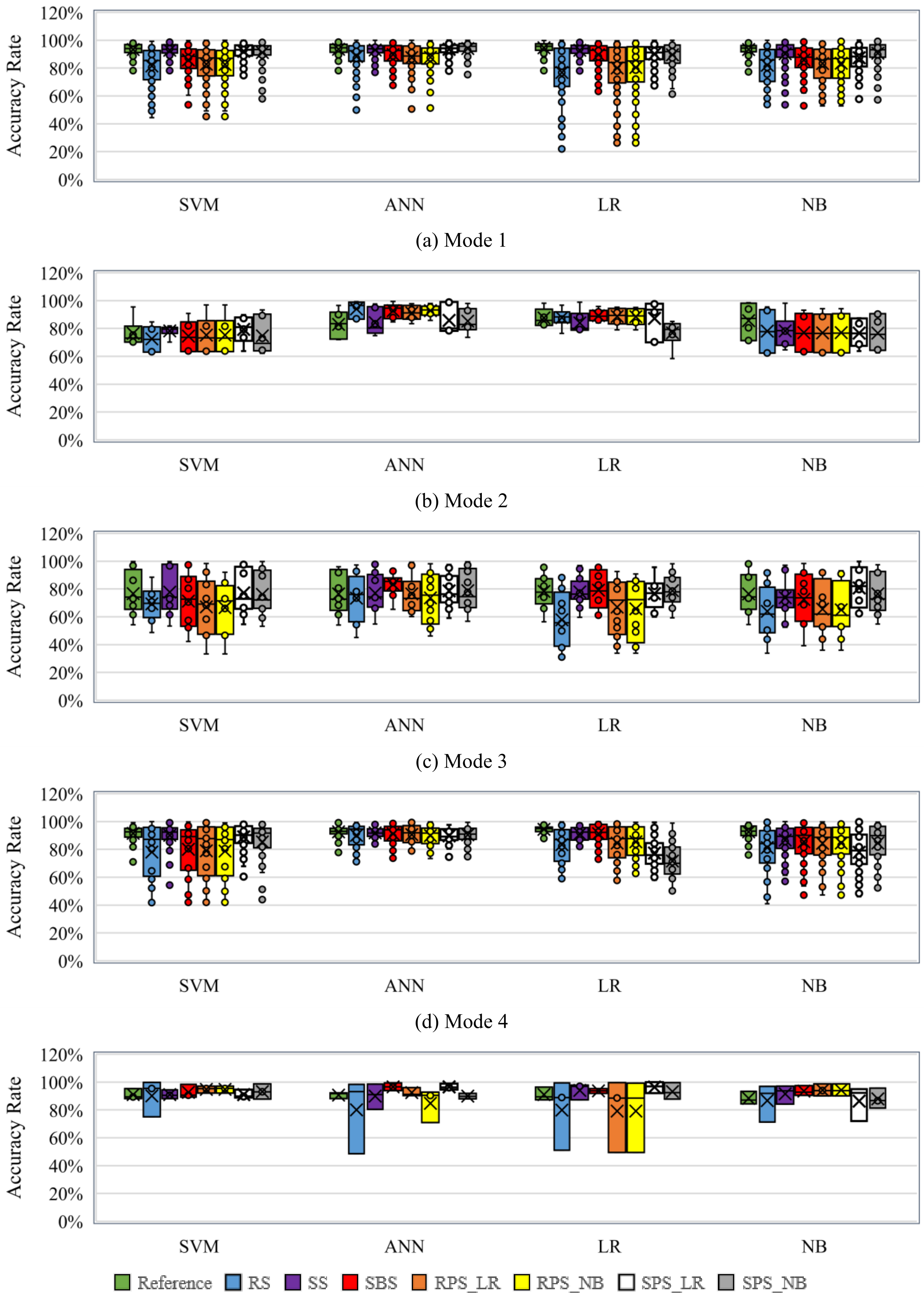
Figure 8: Specificity for different modes

5.2. Results: fairness

The accuracy rate for different modes is present in Figure 9. Reference cases in Mode 1, Mode 4 and Mode 5 could ensure the accuracy rate to be higher than 80%. RS or RPS often results in the lowest accuracy rate compared to other pre-processing techniques in Mode 1, Mode 3 and Mode 4. For instance, in Figure 9(a), RS and RPS decrease the mean accuracy rate from 93% (reference case) to 82% when using SVM, and their lowest accuracy rate even drops to 45%. However, their effect on the accuracy rate of Mode 2 various among classifiers, while RS decreases the accuracy rate of Mode 5 but RPS increases it. More importantly, SBS results in higher accuracy rate for most lighting series than RPS. The effect of SS is comparable with SPS. For all modes, both slightly change the accuracy rate compared to reference cases. The effect of rankers on the accuracy rate is neglectable for both RPS and SPS. Moreover, most cases in Mode 5 present a higher than 80% accuracy rate.

The recall rate (see Figure 10) for Mode 1, Mode 3 and Mode 4 is almost zero in most reference cases, while the recall rate of reference cases varies between 0% and 80% for Mode 2. Applying RS, RPS and SBS could effectively improve the recall rate. For example, RPS increases the mean recall rate to be higher than 80% in Mode 1, Mode2 and Mode 4, and higher than 60% in Mode 3. In general, the recall rate improvement ability of SBS is ~7% lower than RPS in Mode 1, Mode 3 and 4 when using SVM, ANN, or LR, while it is almost equal to RPS when using NB or in Mode 2. RS also shows better recall rate than SBS in Mode 1 and Mode 4. In addition, SPS results in better recall rate than SS for Mode 1, Mode 3 and Mode 4. The influence of ranker on the recall rate is not significant for RPS, while using LR as the ranker in SPS usually presents higher recall rate than NB. Furthermore, for Mode 5, the recall rate of most cases is similar to their corresponding accuracy rate. RS presents the lowest mean recall rate in Mode 5.

The specificity rate (see Figure 11) is similar to the accuracy rate for cases under Mode 1 and Mode 3. However, for Mode 2 and Mode 4, the specificity rate of reference cases is higher than their corresponding accuracy rate. This indicates that the poor recall rate shows negative effect on the accuracy rate. Furthermore, RS show lower specificity rate than SBS and RPS in Mode 1, Mode 3, and Mode 4. Moreover, for Mode 5 (see Figure 11(e)), the lowest specificity rate in each box represents the results for Light_4 of APT #11, which is turned ON most of the time. Its 0% specificity rate is caused by the 0% 1-Specificity and 0-Specificity; thus, it does not affect the accuracy rate.



(a) Mode 1

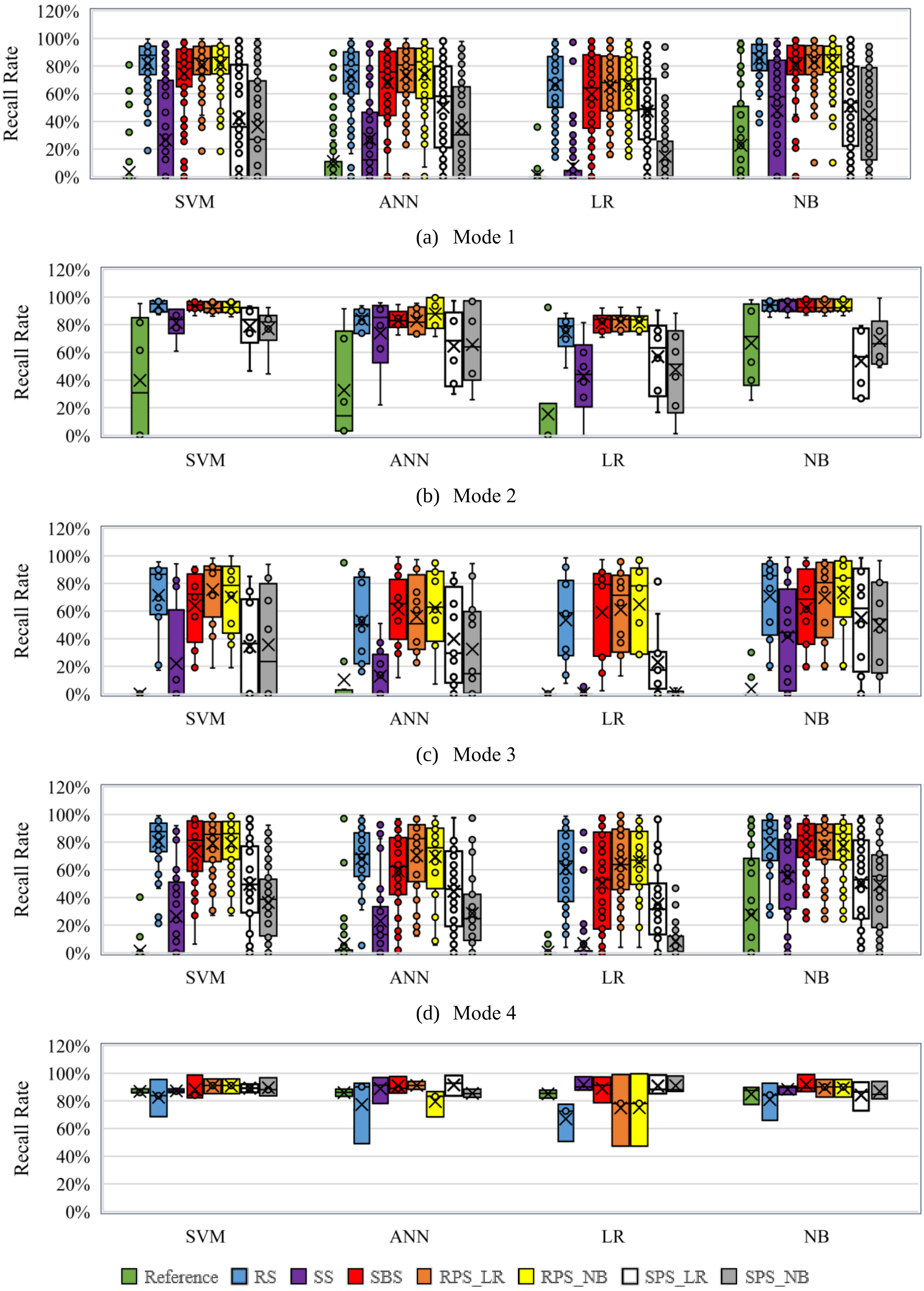
(b) Mode 2

(c) Mode 3

(d) Mode 4

(e) Mode 5

Figure 9: Accuracy rate for different modes



(a) Mode 1

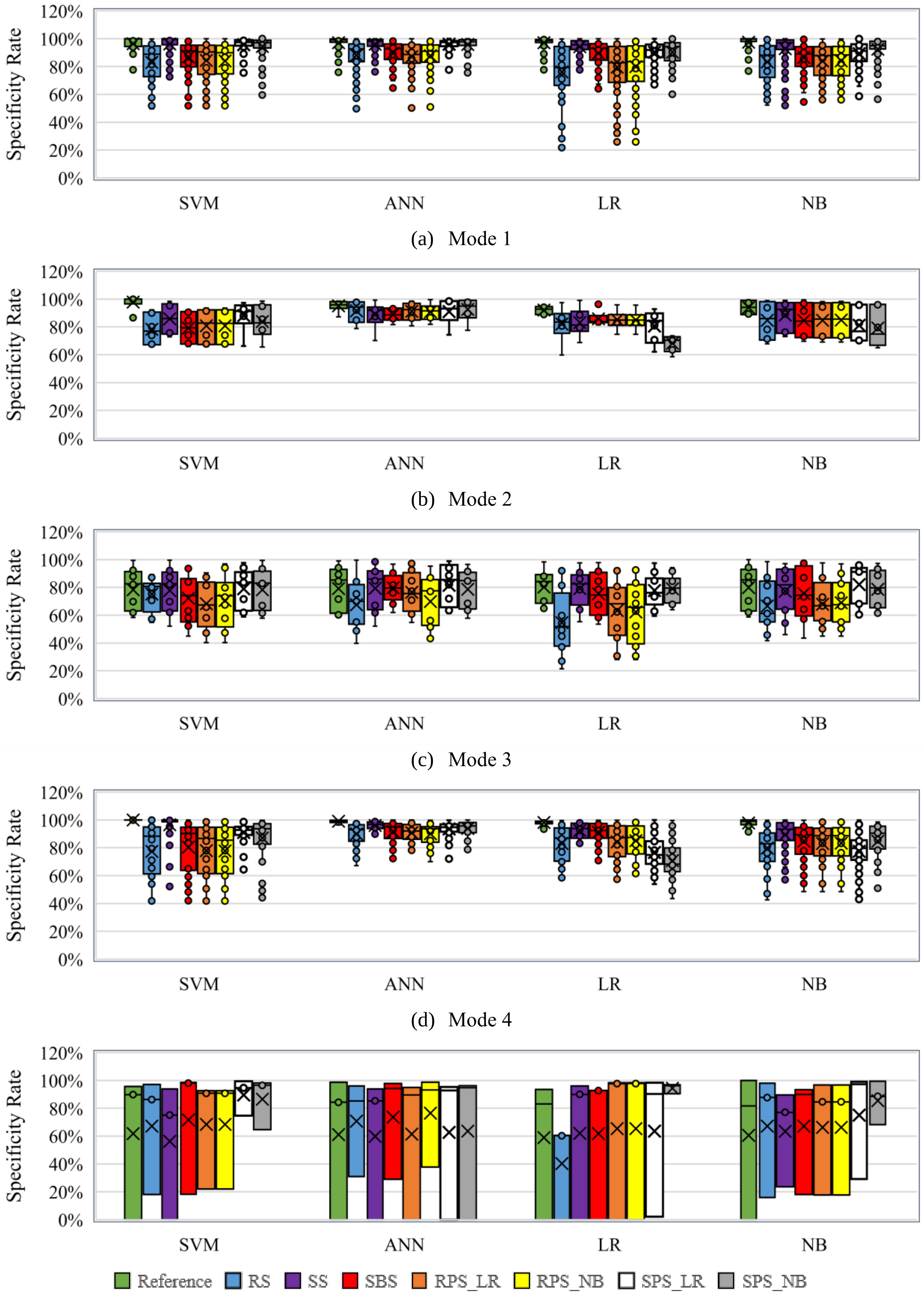
(b) Mode 2

(c) Mode 3

(d) Mode 4

(e) Mode 5

Figure 10: Recall rate for different modes



(a) Mode 1

(b) Mode 2

(c) Mode 3

(d) Mode 4

(e) Mode 5

Figure 11: Specificity rate for different modes

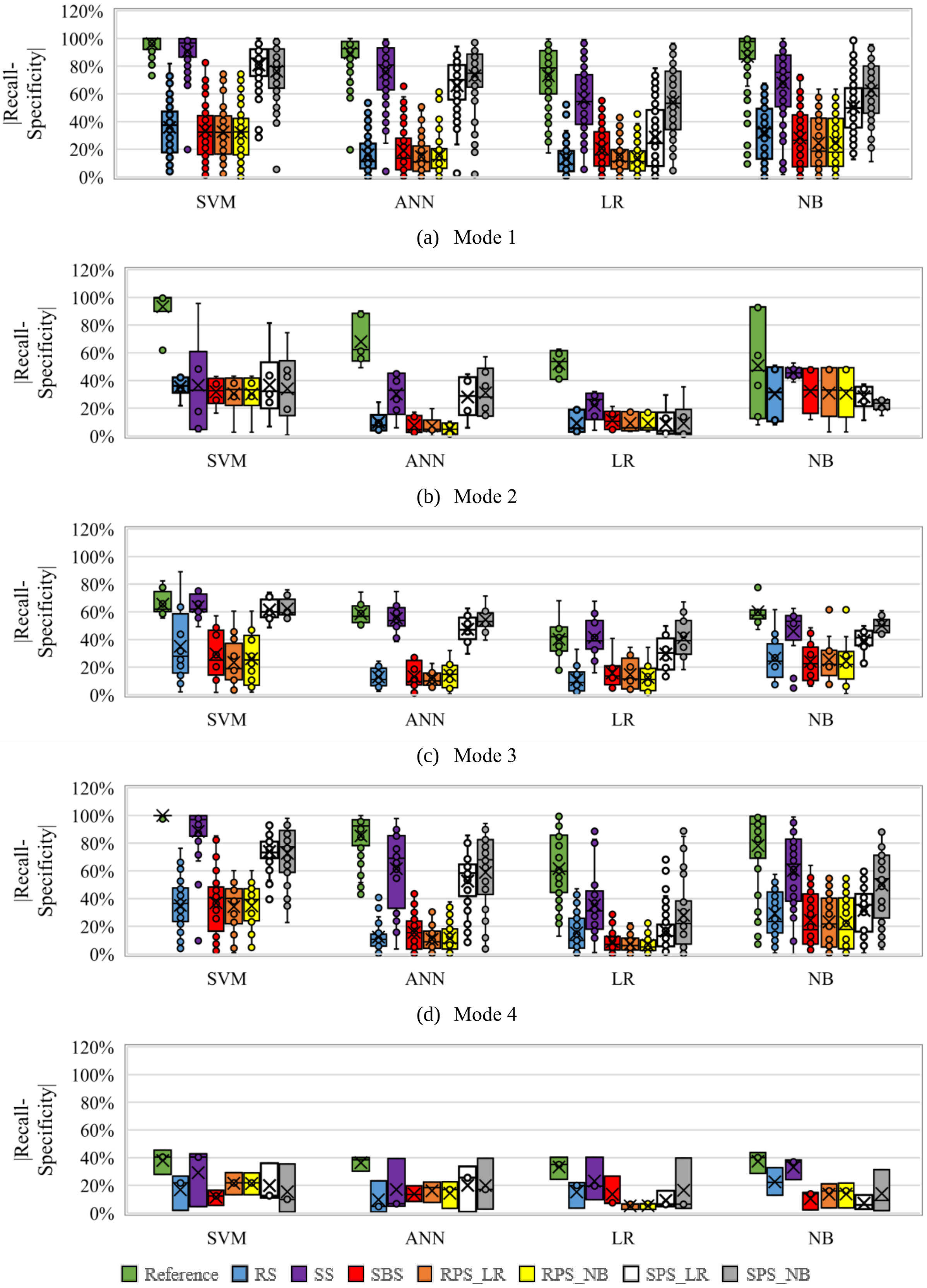
6. Discussion

Through analyzing results presented in Section 5.1, imbalanced training dataset would result in higher predictive accuracy for majority classes and lower for minority classes. For instance, lights in Mode 1 were turned OFF most of time in $X_{candidate}$, which means that lighting status 'OFF' is the majority class and 'ON' is the minority class. Reference cases for these lights usually present a higher than 90% specificity but almost 0% recall. To increase the predictive accuracy of the minority class, RPS would be the first choice. If a relatively simple algorithm is another requirement, RS and SBS could be good choices as they do not contain a ranker which contributes more hyperparameters to the pre-processing technique. The great predictive performance increasing ability of RS, RPS and SBS for minority class is because they produce a balanced training dataset at the first time of implementation. This finding reveals their potential application to fault detection, which requires higher accuracy for minority conditions.

In some cases, SBS even shows comparable result than RPS. This is because occupancy habits, such as lighting usage and occupancy pattern, are seasonally changed [57]. Therefore, in these cases, newly collected data would be more representative than old ones. As a result, $X_{designed}$ of SBS would be similar as of RPS.

One the other hand, when utilizing SS or SPS, predictive accuracy for minority classes maybe increased gradually. This is because the amount of data in minority classes is increased as time goes on (discussed in our previous study [21]).

Furthermore, in some building and indoor environment cases, predictive performance for majority classes and minority classes all matters. Therefore, decreasing their difference also need to be considered. Absolute difference between recall and specificity are summarized in Figure 12. Reference cases present the highest absolute difference compared to other cases. It further indicates that imbalanced training dataset results in perfect predictive performance for majority classes but poor performance for minority classes. To narrow the absolute difference between recall and specificity, RS, SBS and RPS could be selected. In most cases, SBS shows comparable results with RPS. For instance, when using SVM in Mode 1, these two pre-processing techniques could decrease the mean absolute difference from ~96% (reference cases) to ~32%. However, SS shows the least difference decline. Therefore, it is not suitable for studies that pursue a model with uniform predictive performance for all classes.



(a) Mode 1

(b) Mode 2

(c) Mode 3

(d) Mode 4

(e) Mode 5

Figure 12: Absolute difference between recall and specificity for different Modes

One problem in this study is the poor predictive accuracy for lights in Mode 5. It is resulted from the non-representative input features. From Figure 13, for Mode 5, all input features are not significantly related to the light status (output), while there are some strong relationships between input features and lights in Mode 1 to Mode 4. Note that in this study, input features are kept as the same for all cases to make the pre-processing techniques or classifiers as the control variables. However, in real-world application, proper features should be selected for specific problems. Another interesting finding from Figure 13 is that lighting status could be highly dependent on motion status in some cases, such as in Mode 3. In these cases, suppressing motion status from input features may destroy the predictive performance. This means that excluding protected attribute could not ensure the achievement of *Type I* fairness: The predictive result is independent of the protected attribute. Moreover, detailed correlation matrix values are present in the supplementary information.

Furthermore, this paper reveals that training dataset, in which the protected attribute is distributed imbalance, could results in worse fairness rate, when the protected attribute is related to the output. For instance, in Mode 1 and Mode 4, the 'Motion Status_total' is almost evenly distributed among ON/OFF labels, while it is turned OFF most of time in Mode 3 and 5. Thus, from Figure 9, reference cases in Mode 1 and Mode 4 present higher accuracy rate than reference cases in Mode 2 and Mode 3.

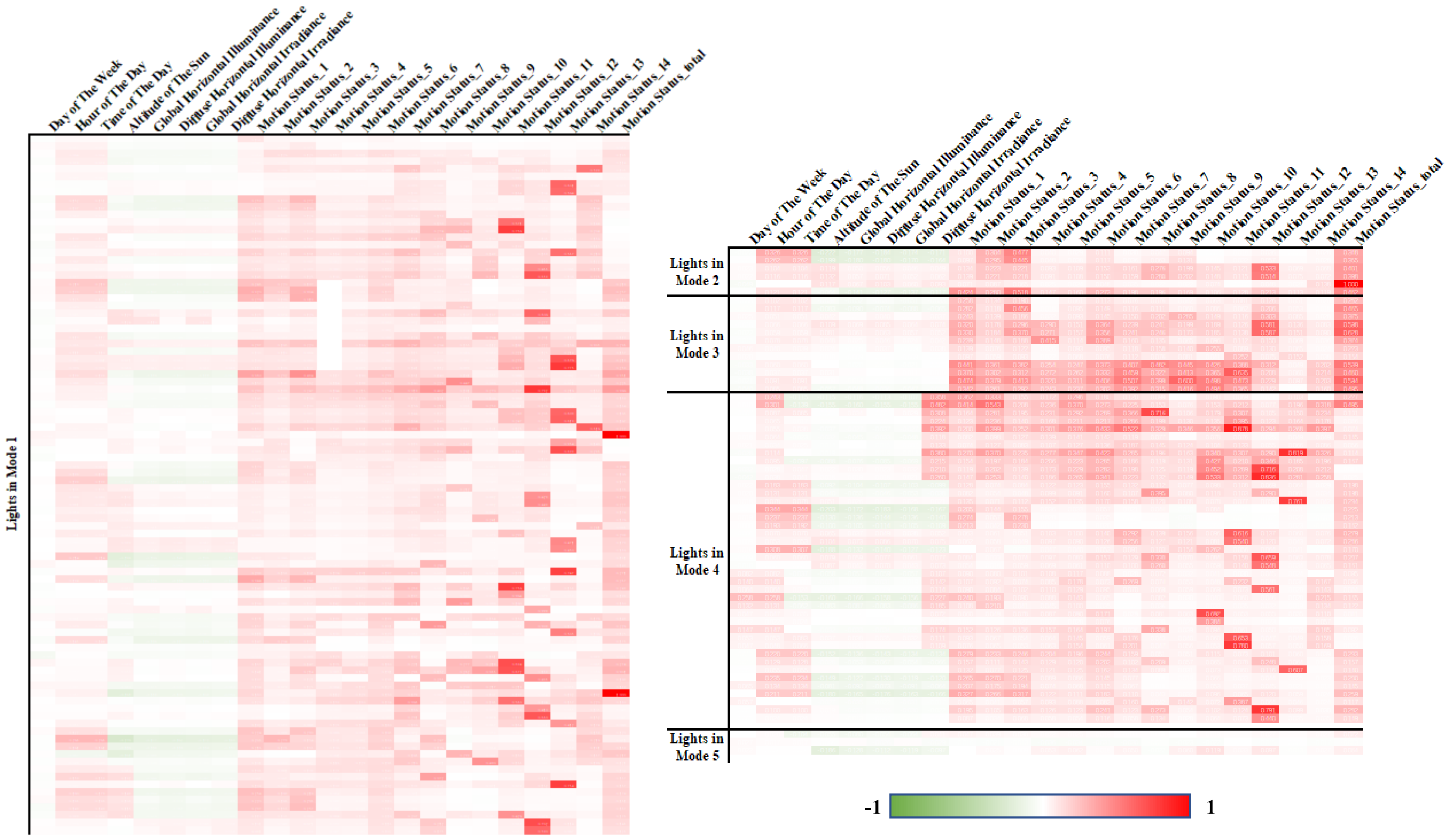


Figure 13: Correlation matrix of input features and outputs

7. Conclusion

In this study, SBS was proposed to process $X_{candidate}$ to a balanced dataset that maintains the most recent information. Then, generalizability of four pre-processing techniques—SS, SBS, RPS, and SPS—were studied and compared with RS through a case study that applied them to process 5 modes of $X_{candidate}$ before predicting the lighting status in 16 apartments. Totally, 4,960 cases were investigated. The following conclusions are draw from the case study:

(1) Imbalanced training dataset not only results in poor predictive accuracy for minority classes, but also bad fairness rates.

(2) In terms of the effect on predictive accuracy, SBS, RS, and RPS shows the most significant accuracy improvement ability for minority classes, but RS shows the most harmful influence on accuracy of majority classes. When $X_{candidate}$ is quite imbalanced, its negative effect on the overall predictive accuracy could be unacceptable. Besides, the newly proposed SBS shows comparable effect on predictive accuracy as RPS, but it is simpler as it does not require a ranker. On the other hand, SS and SPS shows a slightly accuracy improvement for minority classes with an acceptable price of accuracy decrease on majority classes. Rankers in pre-processing techniques could affect the predictive performance; however, no consistent pattern has been found.

(3) From the aspect of fairness improvement, all pre-processing techniques, especially SBS, RS, and RPS, could effectively increase the recall rate. However, RS would result in the greatest decrease of specificity rate for Mode 1, Mode 3, and Mode 4. SS and SPS could remain the specificity rate for most cases in Mode 1 and Mode 4 to be higher than 80%. Moreover, RPS or RS often results in the lowest accuracy rate in Mode 1, Mode 3 and Mode 4.

(4) The newly proposed SBS could show similar effects on accuracy and *Type II* fairness as RPS, when patterns in training dataset are changed with time.

Overall, this study verifies the generalizability of the proposed pre-processing techniques to improve *Type II* fairness while preserving accuracy. Researchers are recommended to select the proper pre-processing techniques based on their research objective and training data distribution. However, this study presents some limitations: (1) The input features were the same for all cases, which results in poor predictive accuracy for cases whose output is not highly related to these features. (2) The sample size for Mode 5 is too small. (3) The influence of relationship between time change and data pattern on the result of SBS technique is not quantified. (4) Pre-processing methods are only tested on lighting status prediction, while fault detection could be a potential application field. Future studies focusing on solving these drawbacks could be interesting.

Acknowledgments

The authors would like to express their gratitude to Concordia University for the support through the Concordia Research Chair in Energy & Environment.

Abbreviations

ANN	Artificial Neural Network
AHUs	Air Handling Units
BMS	Building Management Systems
DDBMs	Data-Driven Buildings Models
DT	Decision Tree
FDD	Fault Detection and Diagnosis
GAN	Generative Adversarial Network
HVAC	Heating Ventilation and Air-Conditioning
IoT	Internet of Things
KNN	k-Nearest-Neighbor
LR	Linear Regression
MLP	Multi-Layer Perceptron
MPC	Model Predictive Controller
NB	Naïve Bayes
RF	Random Forest
RPS	Reversed Preferential Sampling
RS	Random Sampling
SBS	Sequentially Balanced Sampling
SMOTE	Synthetic Minority Oversampling Technique
SPS	Sequential Preferential Sampling
SS	Sequential Sampling
SVM	Support Vector Machine

Nomenclature

D	Unprotected attributes

MAE	Mean Absolute Error
$\boldsymbol{NN}$	The condition with Negative protected attribute and Negative actual class label
$\lvert\boldsymbol{NN}\rvert$	The number of data points in *NN* of $X_{candidate}$
$\lvert\boldsymbol{NN}\rvert_{\boldsymbol{design}}$	The expected number of data points in *NN* of $X_{designed}$
$\boldsymbol{NP}$	The condition with Negative protected attribute and Positive actual class label
$\lvert\boldsymbol{NP}\rvert$	The number of data points in *NP* of $X_{candidate}$
$\lvert\boldsymbol{NP}\rvert_{\boldsymbol{design}}$	The expected number of data points in *NP* of $X_{designed}$
$\boldsymbol{PN}$	The condition with Positive protected attribute and Negative actual class label
$\lvert\boldsymbol{PN}\rvert$	The number of data points in *PN* of $X_{candidate}$
$\lvert\boldsymbol{PN}\rvert_{\boldsymbol{design}}$	The expected number of data points in *PN* of $X_{designed}$
$\boldsymbol{PP}$	The condition with Positive protected attribute and Positive actual class label
$\lvert\boldsymbol{PP}\rvert$	The number of data points in *PP* of $X_{candidate}$
$\lvert\boldsymbol{PP}\rvert_{\boldsymbol{design}}$	The expected number of data points in *PP* of $X_{designed}$
p_negative	The possibility of classifying a data point as negative
p_positive	The possibility of classifying a data point as positive
RMSE	Root Mean Square Error
$\boldsymbol{S}$	Protected attributes
$\boldsymbol{X_{candidate}}$	Candidate training dataset
$\boldsymbol{X_{designed}}$	Designed training dataset
$\lvert\boldsymbol{X_{designed}}\rvert$	The number of data points in $X_{designed}$
$\boldsymbol{Y}$	Class label of the training point
$\boldsymbol{\hat{Y}}$	Predicted class label

Reference:

[1] B. Li, F. Cheng, H. Cai, X. Zhang, W. Cai, A semi-supervised approach to fault detection and diagnosis for building HVAC systems based on the modified generative adversarial network, Energy Build. 246 (2021) 111044. https://doi.org/10.1016/j.enbuild.2021.111044.

[2] D. Han, J. Lim, Smart home energy management system using IEEE 802.15.4 and zigbee, IEEE Trans. Consum. Electron. 56 (2010) 1403–1410. https://doi.org/10.1109/TCE.2010.5606276.

[3] Y. Sun, M.M. Joybari, K. Panchabikesan, A. Moreau, M. Robichaud, F. Haghighat, Heating demand and indoor air temperature prediction in a residential building using physical and statistical models: a comparative study, IOP Conf. Ser. Mater. Sci. Eng. 609 (2019) 072022. https://doi.org/10.1088/1757-899X/609/7/072022.

[4] S. Naji, A. Keivani, S. Shamshirband, U.J. Alengaram, M.Z. Jumaat, Z. Mansor, M. Lee, Estimating building energy consumption using extreme learning machine method, Energy. 97 (2016) 506–516. https://doi.org/10.1016/j.energy.2015.11.037.

[5] E. Mocanu, P.H. Nguyen, M. Gibescu, W.L. Kling, Deep learning for estimating building energy consumption, Sustain. Energy Grids Netw. 6 (2016) 91–99. https://doi.org/10.1016/j.segan.2016.02.005.

[6] B. Talebi, F. Haghighat, P.A. Mirzaei, Simplified model to predict the thermal demand profile of districts, Energy Build. 145 (2017) 213–225. https://doi.org/10.1016/j.enbuild.2017.03.062.

[7] F. Ascione, N. Bianco, C. De Stasio, G.M. Mauro, G.P. Vanoli, Artificial neural networks to predict energy performance and retrofit scenarios for any member of a building category: A novel approach, Energy. 118 (2017) 999–1017.

[8] Y. Peng, A. Rysanek, Z. Nagy, A. Schlüter, Using machine learning techniques for occupancy-prediction-based cooling control in office buildings, Appl. Energy. 211 (2018) 1343–1358.

[9] D. Gariba, B. Pipaliya, Modelling human behaviour in smart home energy management systems via machine learning techniques, in: IEEE, 2016: pp. 53–58.

[10] Y. Zhou, J. Chen, Z.J. Yu, J. Li, G. Huang, F. Haghighat, G. Zhang, A novel model based on multi-grained cascade forests with wavelet denoising for indoor occupancy estimation, Build. Environ. 167 (2020) 106461. https://doi.org/10.1016/j.buildenv.2019.106461.

[11] W. Wei, O. Ramalho, L. Malingre, S. Sivanantham, J.C. Little, C. Mandin, Machine learning and statistical models for predicting indoor air quality, Indoor Air. 29 (2019) 704–726. https://doi.org/10.1111/ina.12580.

[12] W.C. Tam, E.Y. Fu, R. Peacock, P. Reneke, J. Wang, J. Li, T. Cleary, Generating Synthetic Sensor Data to Facilitate Machine Learning Paradigm for Prediction of Building Fire Hazard, Fire Technol. (2020). https://doi.org/10.1007/s10694-020-01022-9.

[13] Y. Wang, W. Li, Z. Zhang, J. Shi, J. Chen, Performance evaluation and prediction for electric vehicle heat pump using machine learning method, Appl. Therm. Eng. (2019) 113901.

[14] M.S. Mirnaghi, F. Haghighat, Fault detection and diagnosis of large-scale HVAC systems in buildings using data-driven methods: A comprehensive review, Energy Build. 229 (2020) 110492. https://doi.org/10.1016/j.enbuild.2020.110492.

[15] R. Boghetti, F. Fantozzi, J.H. Kämpf, G. Salvadori, Understanding the performance gap: a machine learning approach on residential buildings in Turin, Italy, J. Phys. Conf. Ser. 1343 (2019) 012042. https://doi.org/10.1088/1742-6596/1343/1/012042.

[16] A. Figueiredo, J. Kämpf, R. Vicente, R. Oliveira, T. Silva, Comparison between monitored and simulated data using evolutionary algorithms: Reducing the performance gap in dynamic building simulation, J. Build. Eng. 17 (2018) 96–106. https://doi.org/10.1016/j.jobe.2018.02.003.

[17] A. González-Vidal, F. Jiménez, A.F. Gómez-Skarmeta, A methodology for energy multivariate time series forecasting in smart buildings based on feature selection, Energy Build. 196 (2019) 71–82. https://doi.org/10.1016/j.enbuild.2019.05.021.

[18] J. Kallio, J. Tervonen, P. Räsänen, R. Mäkynen, J. Koivusaari, J. Peltola, Forecasting office indoor CO2 concentration using machine learning with a one-year dataset, Build. Environ. 187 (2021) 107409. https://doi.org/10.1016/j.buildenv.2020.107409.

[19] Y. Pan, L. Zhang, Data-driven estimation of building energy consumption with multi-source heterogeneous data, Appl. Energy. 268 (2020) 114965. https://doi.org/10.1016/j.apenergy.2020.114965.

[20] C. Zhang, J. Li, Y. Zhao, T. Li, Q. Chen, X. Zhang, W. Qiu, Problem of data imbalance in building energy load prediction: Concept, influence, and solution, Appl. Energy. 297 (2021) 117139.

[21] Y. Sun, F. Haghighat, B.C. Fung, Trade-off Between Accuracy and Fairness of Data-driven Building and Indoor Environment Models: A Comparative Study of Pre-processing Methods, Energy. (2021) 122273. https://doi.org/10.1016/j.energy.2021.122273.

[22] X. Wang, P. Tague, Non-invasive user tracking via passive sensing: Privacy risks of time-series occupancy measurement, in: Proc. 2014 Workshop Artif. Intell. Secur. Workshop, 2014: pp. 113–124.

[23] R. Jia, R. Dong, S.S. Sastry, C.J. Sapnos, Privacy-Enhanced Architecture for Occupancy-Based HVAC Control, in: 2017 ACMIEEE 8th Int. Conf. Cyber-Phys. Syst. ICCPS, 2017: pp. 177–186.

[24] J. Li, K. Panchabikesan, Z. Yu, F. Haghighat, M.E. Mankibi, D. Corgier, Systematic data mining-based framework to discover potential energy waste patterns in residential buildings, Energy Build. 199 (2019) 562–578. https://doi.org/10.1016/j.enbuild.2019.07.032.

[25] R. Bermingham, Interpreting research evidence, (2020). https://post.parliament.uk/interpreting-research-evidence/ (accessed November 25, 2021).

[26] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.

[27] T. Ekström, S. Burke, M. Wiktorsson, S. Hassanie, L.-E. Harderup, J. Arfvidsson, Evaluating the impact of data quality on the accuracy of the predicted energy performance for a fixed building design using probabilistic energy performance simulations and uncertainty analysis, Energy Build. 249 (2021) 111205.

[28] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, A. Galstyan, A survey on bias and fairness in machine learning, ArXiv Prepr. ArXiv190809635. (2019).

[29] C. Fan, M. Chen, X. Wang, J. Wang, B. Huang, A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data, Front. Energy Res. 9 (2021). https://www.frontiersin.org/article/10.3389/fenrg.2021.652801 (accessed April 29, 2022).

[30] V.L. Parsons, Stratified Sampling, in: Wiley StatsRef Stat. Ref. Online, John Wiley & Sons, Ltd, 2017: pp. 1–11. https://doi.org/10.1002/9781118445112.stat05999.pub2.

[31] Y. Fan, X. Cui, H. Han, H. Lu, Chiller fault diagnosis with field sensors using the technology of imbalanced data, Appl. Therm. Eng. 159 (2019) 113933. https://doi.org/10.1016/j.applthermaleng.2019.113933.

[32] Z. Zhou, H. Chen, G. Li, H. Zhong, M. Zhang, J. Wu, Data-driven fault diagnosis for residential variable refrigerant flow system on imbalanced data environments, Int. J. Refrig. 125 (2021) 34–43. https://doi.org/10.1016/j.ijrefrig.2021.01.009.

[33] Q. Wen, L. Sun, F. Yang, X. Song, J. Gao, X. Wang, H. Xu, Time Series Data Augmentation for Deep Learning: A Survey, Proc. Thirtieth Int. Jt. Conf. Artif. Intell. (2021) 4653–4660. https://doi.org/10.24963/ijcai.2021/631.

[34] K. Yan, C. Zhong, Z. Ji, J. Huang, Semi-supervised learning for early detection and diagnosis of various air handling unit faults, Energy Build. 181 (2018) 75–83. https://doi.org/10.1016/j.enbuild.2018.10.016.

[35] K. Yan, J. Huang, W. Shen, Z. Ji, Unsupervised learning for fault detection and diagnosis of air handling units, Energy Build. 210 (2020) 109689.

[36] K. Yan, A. Chong, Y. Mo, Generative adversarial network for fault detection diagnosis of chillers, Build. Environ. 172 (2020) 106698. https://doi.org/10.1016/j.buildenv.2020.106698.

[37] Disadvantages of GANs || Am I real or a Trained Model to write?, OpenGenus IQ Comput. Expert. Leg. (2020). https://iq.opengenus.org/disadvantages-of-gans/ (accessed November 4, 2021).

[38] Z. Li, H. Zhu, Y. Ding, X. Xu, B. Weng, Establishment of a personalized occupant behavior identification model for occupant-centric buildings by considering cost sensitivity, Energy Build. 225 (2020) 110300. https://doi.org/10.1016/j.enbuild.2020.110300.

[39] M. Tang, Y. Chen, H. Wu, Q. Zhao, W. Long, V.S. Sheng, J. Yi, Cost-Sensitive Extremely Randomized Trees Algorithm for Online Fault Detection of Wind Turbine Generators, Front. Energy Res. 9 (2021). https://www.frontiersin.org/article/10.3389/fenrg.2021.686616 (accessed March 25, 2022).

[40] K. Li, G. Zhou, J. Zhai, F. Li, M. Shao, Improved PSO_AdaBoost Ensemble Algorithm for Imbalanced Data, Sensors. 19 (2019) 1476. https://doi.org/10.3390/s19061476.

[41] Z. Qu, H. Liu, Z. Wang, J. Xu, P. Zhang, H. Zeng, A combined genetic optimization with AdaBoost ensemble model for anomaly detection in buildings electricity consumption, Energy Build. 248 (2021) 111193. https://doi.org/10.1016/j.enbuild.2021.111193.

[42] T. Pinto, I. Praça, Z. Vale, J. Silva, Ensemble learning for electricity consumption forecasting in office buildings, Neurocomputing. 423 (2021) 747–755. https://doi.org/10.1016/j.neucom.2020.02.124.

[43] A. Fernández, S. García, M. Galar, R.C. Prati, B. Krawczyk, F. Herrera, Learning from imbalanced data sets, Springer, 2018.

[44] G.M. Weiss, K. McCarthy, B. Zabar, Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs?, Dmin. 7 (2007) 24.

[45] C. Miller, More Buildings Make More Generalizable Models—Benchmarking Prediction Methods on Open Electrical Meter Data, Mach. Learn. Knowl. Extr. 1 (2019) 974–993. https://doi.org/10.3390/make1030056.

[46] L. Zhang, J. Wen, A systematic feature selection procedure for short-term data-driven building energy forecasting model development, Energy Build. 183 (2019) 428–442. https://doi.org/10.1016/j.enbuild.2018.11.010.

[47] M.B. Zafar, I. Valera, M. Gomez Rodriguez, K.P. Gummadi, Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment, in: Proc. 26th Int. Conf. World Wide Web, 2017: pp. 1171–1180.

[48] Z. Zhong, A Tutorial on Fairness in Machine Learning, Medium. (2020). https://towardsdatascience.com/a-tutorial-on-fairness-in-machine-learning-3ff8ba1040cb (accessed January 25, 2021).

[49] F. Kamiran, T. Calders, Data preprocessing techniques for classification without discrimination, Knowl. Inf. Syst. 33 (2012) 1–33. https://doi.org/10.1007/s10115-011-0463-8.

[50] K. Panchabikesan, F. Haghighat, M.E. Mankibi, Data driven occupancy information for energy simulation and energy use assessment in residential buildings, Energy. 218 (2021) 119539. https://doi.org/10.1016/j.energy.2020.119539.

[51] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (1995) 273–297. https://doi.org/10.1007/BF00994018.

[52] S.H. Walker, D.B. Duncan, Estimation of the Probability of an Event as a Function of Several Independent Variables, Biometrika. 54 (1967) 167–179. https://doi.org/10.2307/2333860.

[53] 1.9. Naive Bayes — scikit-learn 0.23.2 documentation, (n.d.). https://scikit-learn.org/stable/modules/naive_bayes.html (accessed September 15, 2020).

[54] M. Claesen, B. De Moor, Hyperparameter search in machine learning, ArXiv Prepr. ArXiv150202127. (2015).

[55] sklearn.model_selection.RandomizedSearchCV, Scikit-Learn. (n.d.). https://scikit-learn/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html (accessed November 5, 2021).

[56] D. Biddle, Adverse Impact and Test Validation: A Practitioner's Guide to Valid and Defensible Employment Testing, 2 edition, Routledge, Aldershot, Hampshire, England : Burlington, VT, 2006.

[57] B. Yang, F. Haghighat, B.C.M. Fung, K. Panchabikesan, Season-Based Occupancy Prediction in Residential Buildings Using Machine Learning Models, E-Prime - Adv. Electr. Eng. Electron. Energy. 1 (2021) 100003. https://doi.org/10.1016/j.prime.2021.100003.