Improved Dequantization and Normalization Methods for Tabular Data Pre-Processing in Smart Buildings

Hari Prasanna Das and Costas J. Spanos
{hpdas,spanos}@berkeley.edu
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, CA, USA

ABSTRACT

Ubiquitous deployment of IoT sensors marks a defining characteristic of smart buildings, for they constitute the source of data on building operation, diagnosis, and maintenance. For machine learning applications in buildings, often the sensor data is augmented with several other artificial variables or metadata corresponding to building components including the occupants. Above datasets are usually organized in the form of a table with rows and columns, and inherently comprise a mix of continuous and discrete (nominal, ordinal) features/columns, thus are called tabular datasets. A vast majority of smart building datasets are tabular in nature. Machine learning algorithms, especially deep neural networks are generally designed as smooth function approximators, and hence are difficult to train optimally with tabular data without appropriate pre-processing. In this work, we analyze the challenges faced by conventional methods for tabular data pre-processing, and propose the use of two improved data transformation methods, namely variational dequantization (for discrete features), and mode-specific normalization (for continuous features). We show improved thermal preference classification performance for two key thermal comfort datasets with the proposed pre-processing. Since the methods are designed in a generalizable way to work for any tabular dataset, we envision them to be an integral part of machine learning algorithm development pipeline for a plethora of smart building applications.

CCS CONCEPTS

• **Computing methodologies → Machine learning**.

KEYWORDS

Tabular Data, Continuous and Discrete Features, Data Pre-Processing, Thermal Comfort, Classification

ACM Reference Format:
Hari Prasanna Das and Costas J. Spanos. 2022. Improved Dequantization and Normalization Methods for Tabular Data Pre-Processing in Smart Buildings. In *The 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '22), November 8–10, 2022, Boston, MA, USA.* , 9 pages. https://doi.org/10.1145/xxxxxxx.xxxxxxx

BuildSys '22, November 8–10, 2022, Boston, MA, USA

ACM ISBN XXXX.
https://doi.org/10.1145/xxxxxxx.xxxxxxx

1 INTRODUCTION

Recent years have witnessed an exponential growth in machine learning implementation in smart buildings. At the core of machine learning is data: its continuous availability, intelligent processing, efficient handling and storage. Smart buildings are equipped with an array of Internet-of-Things (IoT) devices that ensure the availability of rich data. The data is then fed to machine learning algorithms after appropriate curation and pre-processing to perform some task that achieves an objective, be it enhancing energy efficiency or improving occupant thermal comfort and productivity. For intelligent machine learning model design, it is crucial that the data pre-processing is done properly to handle the diverse data collected in buildings in an unified manner. Motivated by the above requirement, in this work we focus on some improved data transformation methods for one of the key types of data commonly found in smart buildings, namely tabular data.

Tabular data is defined as data that is structured into rows, and columns of information. Each row contains the same number of cells (although some of these cells may be empty), which is considered as a single data sample. Each column in tabular data represent a variable, or a property or a feature of the system to which the dataset corresponds to. The columns in tabular datasets can be continuous, which refers to variables whose values come from the real number set, and can be uncountably infinite, or discrete, which refers to variables that are categorical and can have countably limited number of values. Another kind of structured data is graphical data that encodes the relationship between multiple entities either in a directed or undirected way. Graph structures are useful for certain types of problems, such as network optimization and recommender systems. Some examples for the unstructured type of data include images that are organized in terms of pixels, and textual data that is organized as sequences of characters with no particular pre-defined storage model. As we will see in the next paragraphs, a large number of smart building datasets are tabular in nature, which is why we focus this work to design pre-processing methods specifically for them. Nevertheless, the proposed methods can also be used for other data types with minor modifications.

The data obtained in smart buildings can be broadly divided into four classes [26]: occupant data, facility data, enterprise data, and distributed energy resources (DER) data. Occupant data refers to the data collected from occupants pertaining to their occupancy, thermal comfort preferences, energy usage, etc. For instance, to ensure occupants are thermally comfortable in buildings, there is an array of research [3, 19, 20, 24, 30] focusing on understanding which parameters affect the thermal preference of an individual or a group, and design physics-based or machine learning based predictors to predict them. The data collected from occupants and their immediate environment include environmental variables [18] such as standard effective temperature, air temperature, relative humidity, and air velocity, occupant specific variables [10, 19] such as clothing level, metabolic rate, and in some cases, physiological signals such as heart rate and temperatures at different key body points. All of the above readings can be taken as instantaneous readings for several subjects, or by performing a field experiment with a set of subjects over a period of time. In both the cases, the data is organized into a tabular form, with the above features as columns and each row representing data at a time stamp for an occupant. Some of the above features are continuous and some discrete. Thermal comfort is a key example of smart building components that prevalently have tabular data. Other occupant data, such as the CO_2 concentration of the return air (used to measure occupancy in buildings [40]), infrared radiation changes using PIR sensors (used to reflect the movement information of objects, and hence detect both occupancy and presence [29]), and energy resource consumption data (used to monitor the usage and encourage energy-efficient behavior by providing incentives [15]), are also organized in the form of tables and hence classified as tabular data.

The second class of data in smart buildings is facility data. This corresponds to the data obtained primarily from and for the various mechanical systems present in the building. The data collected might be used to optimize the operation of different systems such as the Heating, Ventilation, and Air Conditioning (HVAC), or to diagnose faults in the system for predictive maintenance. For example, for monitoring and opportunistically optimizing HVAC system, the energy consumption, temperature and humidity in different zones [13] in a building are collected. For diagnosing faults in the system, parameters such as flow-rate for water systems, actuator statuses (e.g., valve, pump) [17] etc. are collected. All the above datasets are tabular in nature since they are readings that are coming as a stream with a particular frequency from sensors fitted in various appliances.

The third class is enterprise data, which includes data from software systems governing a smart building. For example, data streams from digital twins of a building might contain synthetic measurements of building parameters [12]. The fourth class is DER data, which comprises of data corresponding to renewable energy (mostly solar) generation and consumption measurements [21], occupant/building energy consumption schedule and patterns throughout the day [38], and data corresponding to demand response programs [31]. All of the above datasets are tabular in nature. In retrospect, we realize that a significant number of datasets collected and utilized by machine learning algorithms in smart buildings are tabular in nature and demand specialized methods for pre-processing.

Data pre-processing is a vital step in the machine learning based task design and implementation process since inconsistencies among the diverse features in a dataset can cause any algorithm to be suboptimal. Mainly focusing on the data transformation part in the data pre-processing pipeline, steps such as normalization are necessary to scale the features to common limits (e.g. min-max normalization), and also to model them to follow known distribution (e.g. standard/gaussian normalization). At the same time, dequantization of discrete features is also necessary for models to learn the data distribution efficiently. Based on our study, we observed that most of the prior works treat continuous and discrete features alike. The most common data transformation step in existing works are standard or min-max normalization. In the best case, some works convert the discrete features to one-hot vectors. Since many of the machine learning models are smooth function approximators, and are also sensitive to the modes present in the data, above pre-processing methods do not offer the flexibility for the machine learning models to learn the data distribution optimally.

In this work, we focus on the above challenges for tabular data, and propose the use of two novel data transformation (the other steps in data pre-processing that precede data transformation, such as data cleaning are kept the same) methods, namely mode-based normalization for continuous features, and uniform and variational dequantization for discrete features. Dequantization refers to adding noise to the discrete variables before they are fed to the machine learning models. By considering thermal comfort dataset as representative tabular dataset for smart buildings, we show that our proposed methods achieve significant improvement in thermal comfort prediction performance as compared to the state-of-the-art model. Needless to say, the proposed methods, being designed in a generic manner for tabular datasets, extend seamlessly for use by other smart building tabular datasets. To the best of our knowledge, we are the first to propose and conduct an extensive study into the data pre-processing methods for the most commonly found data in smart buildings, i.e. tabular data.

2 RELATED WORKS

Since we focus on the data transformation step in the whole data processing pipeline, we discuss and compare our proposed methods with data transformation methods used in the previous works. For continuous features, standard/gaussian or min-max normalization have been the gold standard in previous works. For instance, authors in [32] use standard normalization or $z-$normalization and apply it to the subjective response data to scale it uniformly and to better determine the overall response trends. In [39], standard normalization is used for metedata normalization in design of a dynamic multi-task thermal comfort prediction model. Min-max normalization has also been used in [35] to normalize the data for use in K-nearest neighbor based thermal model. Another work that focuses on study of HVAC control strategies using personal thermal comfort and sensitivity models [11] uses min-max normalization to scale the thermal comfort readings. Authors in [37] use min-max normalization on occupant behavior data to study the influence the same on building energy consumption. There are also some manually engineered ways for normalization as done in [4], where authors perform normalization of skin temperature (continuous feature) by specifically designing a factor that indicates the unclothed/exposed body surface area. They also show that normalization improves the stratification of thermal classes. In our work, we state the shortcomings of the above methods (Sec 3.2.1) for continuous features, and propose the use of a novel method, namely, mode-based normalization (Section 3.2). The above method, originally proposed in [36], is used to generate synthetic samples for tabular datasets among other possible applications.

When it comes to transformation for discrete features, not much special attention has been given to dequantize them before feeding them into machine learning models such as neural networks that are designed to approximate a smooth function with desirable accuracy provided sufficient neurons are used. At the best, one-hot encoding has been used to encode categorical variables. For example, Wang et al. [34] study the thermal comfort models designed using ASHRAE database [18], and state that one-hot encoding is commonly used to encode categorical variables such as building type. Authors in [16] also perform one-hot encoding of the categorical features during data pre-processing. Similar is the case for works on data-driven optimization of building designs [28], modeling of energy demand response in buildings [1], etc. This does not only result in high-dimensional data when the categorical variables have many levels, it also gives rise to multiple more variables that are discrete in themselves. To the best of our knowledge based on extensive literature search, there are no existing works that focus on using dequantization methods for discrete feature transformation for machine learning applications in smart buildings. We propose two methods for dequantizing discrete features, namely uniform and variational dequantization [9]. We discuss the ways and cases where the proposed methods can be used, and implement them for a real-life smart building dataset to test for their strength.

3 METHODOLOGY

In this section, we describe the proposed pre-processing steps for tabular data. We first provide a brief introduction of generative flow models, and affine flow as a type of normalizing flow models, since both of them are essential components of the pre-processing steps that follow later in the section.

3.1 Preliminary

3.1.1 Normalizing Flow Models. Generative flow models are invertible mapping between the data space which has an unknown and complex probability distribution, and a latent space which is a known simple distribution, mostly taken as the standard gaussian $\mathcal{N}(0, \mathbf{I})$. They are trained using maximum-likelihood estimation, usually with unsupervised data, except when explicitly engineered to include some conditions, e.g. [6]. Below, we briefly cover the formulation of flow models.

Let $\mathbf{X}$ be a high-dimensional random vector with unknown true distribution $\mathcal{P}(\mathbf{X})$. The following formulation is directly applicable to continous data, and with some pre-processing steps such as dequantization [9, 27, 33] to discrete data. Let $\mathbf{Z}$ be the latent variable with a known standard distribution $\mathcal{P}(\mathbf{Z})$, such as a standard multivariate gaussian. Using an i.i.d. dataset $\mathcal{D}$, the target is to model $\mathcal{P}(\mathbf{X})$ with parameters $\boldsymbol{\theta}$. A flow, $\mathbf{f}_{\boldsymbol{\theta}}$ is defined to be an invertible transformation that maps observed data $\mathbf{X}$ to the latent variable $\mathbf{Z}$. A flow is invertible, so the inverse function $\mathcal{T}$ maps $\mathbf{Z}$ to $\mathbf{X}$, i.e.

$$\mathbf{Z} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{X}) = \mathcal{T}^{-1}(\mathbf{X}) \text{ and } \mathbf{X} = \mathcal{T}(\mathbf{Z}) = \mathbf{f}_{\boldsymbol{\theta}}^{-1}(\mathbf{Z}) \tag{1}$$

The log-likelihood can be expressed as,

$$\mathcal{P}_{\theta}(\mathbf{X}) = \mathcal{P}(\mathbf{Z}) \left| \det \left(\frac{\partial \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{X})}{\partial \mathbf{X}^T} \right) \right| \tag{2}$$

$$\log \mathcal{P}_{\theta}(\mathbf{X}) = \log \mathcal{P}(\mathbf{Z}) + \log \left| \det \left(\frac{\partial \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{X})}{\partial \mathbf{X}^T} \right) \right| \tag{3}$$

where $\frac{\partial \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{X})}{\partial \mathbf{X}^T}$ is the Jacobian of $\mathbf{f}_{\boldsymbol{\theta}}$ at $\mathbf{X}$. The invertible nature of a flow allows it to be capable of being composed of other flows of compatible dimensions. In practice, flows are constructed by composing a series of component flows. Let the flow $\mathbf{f}_{\boldsymbol{\theta}}$ be composed of K component flows, i.e. $\mathbf{f}_{\boldsymbol{\theta}} = \mathbf{f}_{\theta_K} \circ \mathbf{f}_{\theta_{K-1}} \circ \cdots \circ \mathbf{f}_{\theta_1}$. Then the log-likelihood of the composed flow is,

$$\log \mathcal{P}_{\theta}(\mathbf{X}) = \log \mathcal{P}(\mathbf{Z}) + \log \left| \det \left(\frac{\partial (\mathbf{f}_{\theta_K} \circ \mathbf{f}_{\theta_{K-1}} \circ \cdots \circ \mathbf{f}_{\theta_1}(\mathbf{X}))}{\partial \mathbf{X}^T} \right) \right| \tag{4}$$

which follows from the fact that $\det(A \cdot B) = \det(A) \cdot \det(B)$. The reverse path, from $\mathbf{Z}$ to $\mathbf{X}$ can be written as a composition of inverse flows, $\mathbf{X} = \mathbf{f}_{\boldsymbol{\theta}}^{-1}(\mathbf{Z}) = \mathbf{f}_{\theta_1}^{-1} \circ \mathbf{f}_{\theta_2}^{-1} \circ \cdots \circ \mathbf{f}_{\theta_K}^{-1}(\mathbf{Z})$. Confirming with above properties, different types of flows can be constructed [2, 5, 7, 8, 14].

3.1.2 Affine Flow. The design of flow models has to satisfy the criteria of invertibility and computable jacobian, all the while being expressive enough to capture the data distribution. One of the designs of such flow models is affine flow [8]. In this, the feature vector $\mathbf{X} \in \mathbb{R}^p$ is partitioned into two parts, $\mathbf{X}_1 \in \mathbb{R}^r$, and $\mathbf{X}_2 \in \mathbb{R}^{p-r}$, where $0 < r < p$. An affine flow $\mathbf{f}_{\boldsymbol{\theta}}$ mapping $\mathbf{X}$ to $\mathbf{Z}$ is designed in the following way:

$$\begin{aligned} \mathbf{Z}_1 &= \mathbf{X}_1 \\ \mathbf{Z}_2 &= \mathbf{X}_2 . \exp(NN_{\theta}(\mathbf{X}_1)) + NN_{\delta}(\mathbf{X}_1) \\ \mathbf{Z} &= [\mathbf{Z}_1, \mathbf{Z}_2] \end{aligned}$$

where NN_{θ} and NN_{δ} are two neural networks parametrized by θ and δ respectively. In the above formulation, $\mathbf{Z}$ can be viewed as an affine function of $\mathbf{X}$. Affine flow is one of the commonly used flow models because of its simplicity. We will use affine flows in one of the pre-processing methods we propose to use for tabular datasets.

3.2 Data Pre-Processing

Data pre-processing involves a series of steps, such as data cleaning to get rid of or replace missing and/or noisy data, data transformation, dimensionality reduction (if needed) etc. We particularly focus on the data transformation part, keeping the other steps same as others existing in the literature. Let us assume the dataset in hand is represented by $\mathbf{X} \in \mathbb{R}^{n \times p}$, which means we have n samples, and p features. The p features are be a mix of both continuous and discrete/categorical columns. Let us represent the continuous feature vectors by $X_1^c, X_2^c, \cdots, X_{\alpha}^c$, and the discrete feature vectors by $X_1^d, X_2^d, \cdots, X_{\beta}^d$. Note here that $\alpha + \beta = p$, and each of the above feature vectors have the dimension of $n \times 1$. A continuous feature comprises of values from a continuous domain (e.g., $\mathbb{R}$). A discrete feature takes a value from a discrete set and can either be nominal or ordinal. The number of possible values for each discrete feature can vary among the set of discrete features. Both the continuous and discrete features must be processed in specialized ways for it to be compatible for machine learning (especially neural network) models. Therefore, we propose two data pre-processing methods towards the above goal: mode-specific normalization for continuous features, and variational dequantization for discrete features. An illustration of above pre-processing is shown in Fig. 1.

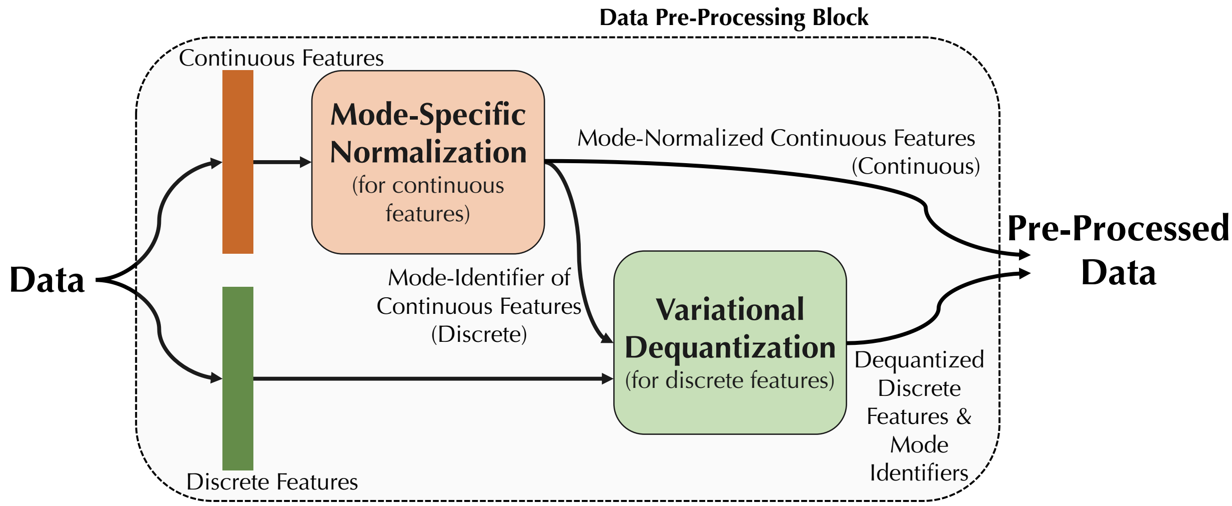

Figure 1: Illustration of the proposed data-preprocessing method.

3.2.1 Mode-specific Normalization. Continuous features in tabular data are usually non-Gaussian and applying transformations that has been used in prior works, such as standard or min-max normalization will lead to vanishing gradient problem [36]. Detecting the modes present in the data and using their parameters to normalize the data helps in handling features with complex distributions, a process referred to as mode-specific normalization [36]. In mode-specific normalization, unlike conventional min-max or standard normalization, we first detect a mode of the feature distribution from which a particular data sample is highly probable to have come from, and then normalize it with the mean and standard deviation of that particular mode. Post normalization, each feature vector is transformed into two feature vectors, one corresponding to the mode-normalized values which is continuous in nature, and another to the identifier of the mode which was selected for normalization which is discrete in nature. The steps of this process are as follows:

(1) A variational gaussian mixture model (VGM) [23] is trained to estimate the number of possible modes for continuous features $X_1^c, X_2^c, \cdots, X_\alpha^c$. For illustration, let us assume for i^{th} continuous feature X_i^c, m number of modes were found. For j^{th} data sample (i.e. j^{th} row of the dataset), the probability of occurence of the value x_{ij}^c in feature X_i^c is,

$$\mathbb{P}_{X_i^c}(x_{ij}^c) = \sum_{k=1}^{m} \eta_k \mathcal{N}(x_{ij}^c; \mu_k, \phi_k)$$

where, η_k, μ_k, ϕ_k are the weight, the mean and the standard deviation of mode k.

(2) To choose a mode to normalize data x_{ij}^c, we compare the probability of that value coming from each of the possible modes, i.e. mode k^* is chosen for normalization as per,

$$k^* = \arg\max_{k=1}^{m} \eta_k \mathcal{N}(x_{ij}^c; \mu_k, \phi_k)$$

(3) Finally, the normalized output and identifier are:

$$\begin{aligned} \text{Mode-normalized value} &= \frac{x_{ij}^c - \mu_{k*}}{4\phi_{k*}} \\ \text{Mode Identifier} &= k^* \end{aligned}$$

We represent the feature vector with mode-normalized values (which is a continuous feature) for X_i^c as X_i^{cc}, and the feature vector with corresponding mode-identifiers (which is a discrete feature) as X_i^{cd}. Effectively, X_i^c is transformed into X_i^{cc} and X_i^{cd}.

3.2.2 Variational Dequantization. For dequantization, generated noise is added to discrete feature values. The distribution from which the noise is extracted brings in the novelty among the dequantization methods. We use two methods for dequantization, namely uniform dequantization, and variational dequantization [9]. In uniform dequantization, noise from a compatible uniform distribution is added to the discrete features, whereas, in variational dequantization, the amount of noise that has to be added is dependent on the original data distribution. At this stage, we dequantize the original discrete features that were present in the dataset $(X_1^d, X_2^d, \cdots, X_\beta^d)$, along with the hybrid discrete features that were created as part of the mode-based normalization process before $(X_1^{cd}, X_2^{cd}, \cdots, X_\alpha^{cd})$, let us denote the union of both the above sets of discrete features as $\tilde{X}^d$.

For dequantization, we add noise $\mathbf{u}$ to the feature set $\tilde{X}^d$, i.e.

$$\tilde{X}^d_{dequantized} = \tilde{X}^d + \mathbf{u}$$

In uniform dequantization, $\mathbf{u}$ is sampled from an uniform distribution $[0,1]^{\alpha+\beta}$. As can be observed, the noise added does not have any relation with the data to which it gets added. On the other hand, in variational dequantization, $\mathbf{u}$ comes from a variational posterior distribution $q(\mathbf{u} \mid \tilde{X}^d)$. Variational dequantization is powerful as compared to uniform dequantization because the noise added is dependent on the data, hence producing a smooth processed data distribution that is easier for the downstream machine learning

Table 1: List of continuous and discrete features for the datasets used in the experiment

Dataset	Continuous Features	Discrete Features
Comfort Database	Standard Effective Temperature, air temperature, relative humidity, air velocity	Clothing level, metabolic rate
Wearables Dataset	Temperature, humidity, wind velocity, physiological parameters: temperature at wrist, ankle, and pant, heart rate	Vote time (morning (7am-12pm), afternoon (12pm-5pm), evening (5pm-10pm), night (10pm-7am)), location during vote (indoors/outdoors)

model to learn. We model the posterior distribution as a conditional generative flow as $\mathbf{u} = q_{\tilde{x}^d}(\epsilon)$, where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ is gaussian noise. The conditional flow model is jointly trained with the downstream neural network model being trained on the pre-processed data.

We model the conditional flow with coupling transformations as has been proposed in [9]. The coupling transformations (F) are designed to follow the cumulative density function (CDF) of mixture of M logistic distributions, represented by LMCDF, i.e.

$$F_{LMCDF}(y; \pi, \mu, \mathbf{s}) = \sum_{i=1}^{m} \pi_i \sigma((y - \mu_i)\exp(-s_i))$$

where, $\pi, \mu, \mathbf{s} \in \mathbb{R}^{\dim(y)}$ are the parameters of logistic mixture distribution corresponding to mixture weight, component means, and component scales, respectively, and $\sigma(.)$ denotes the sigmoid function. The input noise vector ϵ is partitioned into two parts, $\epsilon = [\epsilon_1, \epsilon_2]$, as is done for affine flow models explained in Sec. 3.1.2. The dequantization noise $\mathbf{u}$ is formulated as,

$$\begin{aligned} \mathbf{y} &= NN_\theta(\tilde{x}^d) \\ \pi, \mu, \mathbf{s} &= NN_\delta([\epsilon_1, \mathbf{y}]) \\ \mathbf{u_1} = \epsilon_1, \mathbf{u_2} &= F_{LMCDF}((\epsilon_2; \pi, \mu, \mathbf{s}) \\ \mathbf{u} &= \sigma([\mathbf{u_1}, \mathbf{u_2}]) \end{aligned}$$

where, $NN(\theta)$ and $NN(\delta)$ are neural networks parametrized by θ and δ respectively. We stack multiple such layers in a cascaded manner to generate the dequantization noise $\mathbf{u}$.

An important observation to have here is that in variational dequantization, the networks generating noise are trained in tandem with the downstream model that gets fed with the pre-processed data. Additionally, variational dequantization is designed using neural networks as noise generators. Hence, above method should be used when the downstream model used is a neural network itself that trains using stochastic gradient descent, which essentially holds true for all the deep learning applications in buildings. In cases where the downstream model is not a neural network, uniform dequantization can be a good choice for discrete data transformation.

After the above preprocessing steps, the original data $\mathbf{X}$ becomes,

$$\begin{aligned} \mathbf{X} = X_1^{cc} \oplus \cdots \oplus X_\alpha^{cc} \oplus X_{1,dequantized}^{cd} \oplus \cdots \oplus X_{\alpha,dequantized}^{cd} \oplus \\ \oplus X_{1,dequantized}^{d} \oplus \cdots \oplus X_{\beta,dequantized}^{d} \end{aligned}$$

which is then used for downstream tasks such as forecasting, prediction, segmentation or synthetic data generation.

4 EXPERIMENTS

In this section, provide the features metadata of datasets we use, and then share the experimental settings and results.

4.1 Datasets

As a representative of tabular datasets available in smart buildings, we choose two publicly available thermal comfort datasets (from right-here-right-now readings as well as personal thermal comfort field experiments) for testing the proposed pre-processing method.

4.1.1 Comfort Database/ASHRAE Global Thermal Comfort Database II. The ASHRAE Global Thermal Comfort Database II [18], or as we will call "comfort database"in rest of the paper, is one of the large and mostly used dataset when it comes to designing and testing thermal comfort algorithms, as well as to study the thermal comfort distribution across building types, geographies etc. It is built up off the data from thermal comfort studies conducted around the world in the last two decades from the time the paper was published. It provides thermal comfort measurements, as well as the preference label. We picked six of the most significant variables for data-driven thermal comfort in line with previous researches using this dataset [25]. Specifically, the features chosen are Standard Effective Temperature (SET), clothing level, metabolic rate, air temperature, relative humidity, air velocity. The characteristic type (continuous/discrete of these features is given in Table 1. Post data cleaning to get rid of missing values/ NaNs, the total number of data samples remaining was 56148. The distribution of data samples in the three thermal preference classes was "Prefer cooler": 17794, "Prefer no change": 28195, "Prefer cooler": 10159.

4.1.2 Wearables Dataset. We refer wearables dataset to the data collected from personal thermal comfort experiment using wearable sensors by Liu et al. [19]. The authors conducted an experiment to collect physiological signals (e.g., skin temperature at various parts of the body, heart rate) of 14 subjects (6 female and 8 male adults) and environmental parameters (e.g., air temperature, relative humidity) for 2–4 weeks (at least 20 h per day). The subjects also took an online survey on a daily basis, where they reported their thermal sensation (on a scale of -3 to +3), thermal preference (Warmer, Cooler, No Change), and position (indoor/outdoor) among other parameters. The authors have performed feature engineering to obtain the mean, standard deviation and gradient of physiological features over last 5 mins, 15 mins, and 60 mins of the vote time, which we use in our work. We ranked the features in the dataset as per the amount of missing values/NaNs existing in them, and got rid of those with large number of missing values. After data cleaning, we had approximately 210 samples available per subject. We also converted the vote time variable to a categorical variable as per the following mapping: "Morning"(7am to 12pm), "Afternoon"(12pm-5pm), "Evening"(5pm-10pm), "Night"(10pm to 7am). The distribution of continuous and discrete features that we use for experimentation using this dataset is given in Table 1. The

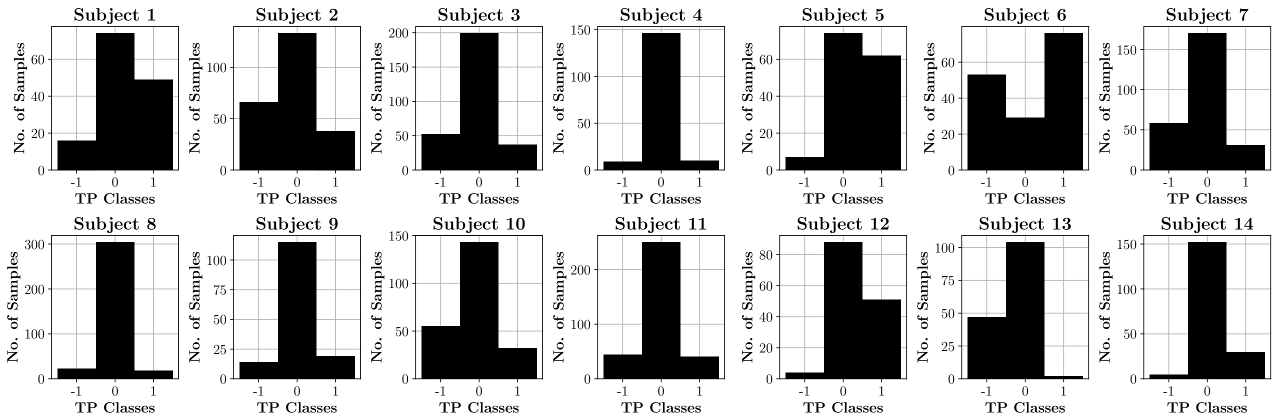

Figure 2: Subject wise distribution of data samples in each of the thermal preference classes. Here, "-1"represents "Prefer cooler"class, "0"represents "Prefer no change"class, and "1"represents "Prefer warmer"class.

subject wise distribution of data samples in each of the thermal preference classes is shown in Fig 2. As it can be observed, the dataset for every subject is highly class-imbalanced with the "Prefer no change"class being the most frequent class.

4.2 Experimental Settings

4.2.1 Testing Procedure: For comfort database, we designed classifiers to classify the thermal preference classes. For wearables dataset, we designed personal thermal comfort models (specific to each subject) to classify their individual thermal preference. As per standard practice [19], for each classifier, we conducted 5-fold cross validation repeated 20 times to estimate the average predictive performance. We report the classification accuracy. Since the datasets are highly class-imbalanced, accuracy alone is not a correct representative of classification performance. So, along with accuracy, we report the cross-validated macro F-1 score [22].

4.2.2 Machine Learning Models and Data Pre-Processing: We experimented with a number of machine learning models ranging from kernel based and tree based methods, to neural networks. Specifically, we use Linear Discriminant Analysis (LDA), K-Nearest Neighbors (KNN), Gaussian Naive-Bayes (GNB), Extra Trees, Random Forest, and feed-forward neural networks. We use standard normalization for continuous features, and one-hot encoding for discrete features as the baseline pre-processing methods. We also test our proposed pre-processing methods: mode-based normalization for continuous features, and uniform/variational dequantization for discrete features along with the neural network models. For variational dequantization, we use 4 layers of flow models with each layer having small feed-forward neural networks representing the NN as mentioned in Sec. 3.2.2. For wearables dataset, we only report results from random forest (since it is considered as the state-of-the-art model for thermal preference prediction), and neural network models for a better presentation of the results across multiple subjects. We run the neural network models in a NVIDIA A100 GPU, and use Adam optimizer with a learning rate of $1e-4$.

4.3 Results

The classification metrics: accuracy and F-1 scores for different machine learning models combined with different data pre-processing methods for comfort database is given in Table 2. Among the kernel and tree-based methods, it can be observed that random forest performs the best in terms of accuracy and F-1 score among other

Table 2: Thermal preference classification performance for comfort database using various machine learning models and data pre-processing methods.

Data Pre-processing Method	Machine Learning Models	Accuracy (%)	F-1 Score (%)
Standard normalization for continuous features and One-hot encoding for discrete features (Conventional Method)	Linear Discriminant Analysis (LDA)	53.8 ± 0.4	38.9 ± 0.5
	K-Nearest Neighbors	52.8 ± 0.4	46.7 ± 0.5
	Gaussian Naive-Bayes	52.5 ± 0.4	43.1 ± 0.5
	Extra Trees	57.1 ± 0.5	50.1 ± 0.5
	Random Forest	57.2 ± 0.5	50.1 ± 0.5
	Neural Network	59.7 ± 0.7	53.4 ± 0.8
Mode-based normalization for continuous features and uniform dequantization for discrete features (Our Work)	Neural Network	61.3 ± 0.6	57.5 ± 0.6
Mode-based normalization for continuous features and variational dequantization for discrete features (Our Work)	Neural Network	63.6 ± 0.6	59.9 ± 0.4

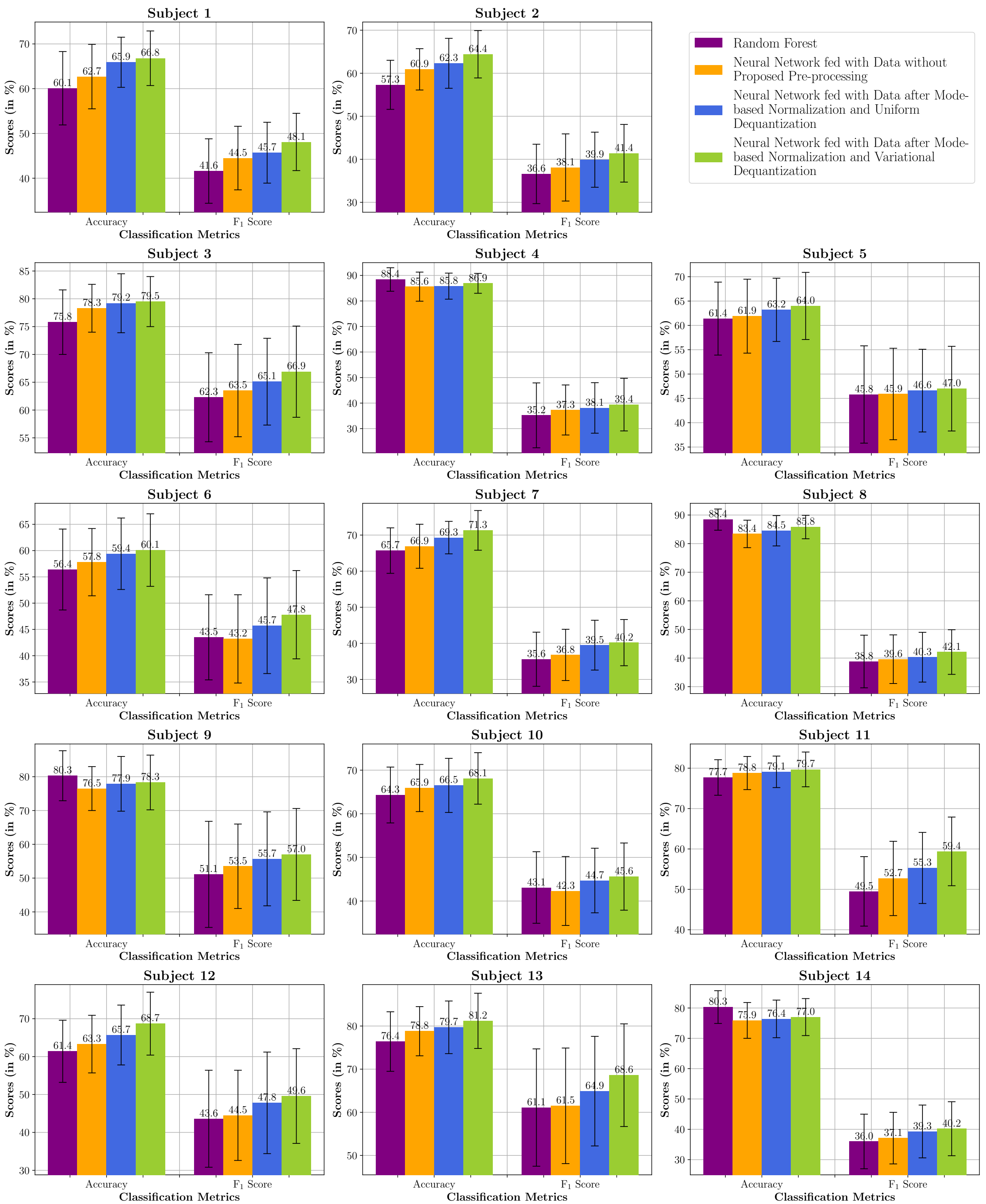

Figure 3: Personal thermal preference classification performance for various machine learning models and data pre-processing methods. Since the datasets for each subject is class-imbalanced, we report both the accuracy and F-1 scores.

models. With a feed-forward neural network, which comes with better expressivity potential, while keeping the data preprocessing method the same, we see a 4.37% relative improvement in accuracy, and a 6.59% relative improvement in F-1 score as compared to the random forest results. With our proposed pre-processing methods, mode-based normalization for continuous features, and uniform dequantization for discrete features along with the same neural network model, we see a relative performance improvement of 7.17% in accuracy and a significant 14.77% improvement in F-1 score over random forest. It is to be expected because effectively by dequantizing and normalizing, we are smoothing the distribution for the continuous neural network models to learn. In the above combination, if we replace uniform dequantization with variational dequantization, we observe a relative improvement of 11.19% in accuracy, and a 19.56% improvement in F-1 score over random forest. This improvement in scores is indicative of the potential of the proposed data transformation methods for tabular data.

In the case of wearables dataset, we designed personal thermal comfort predictors using the above machine learning models. The accuracy and F-1 scores for various models for each subject is given in Fig 3. Across all subjects, the average relative improvement over random forest in accuracy was 0.72%, and in F-1 score was 2.79% for a neural network model with standard normalization for continuous features, and one-hot encoding for discrete features. When we implemented our proposed mode-based normalization, and uniform dequantization, the average relative improvement over random forest increased to 2.71% in accuracy and 7.33% in F-1 score. Finally, with mode-based normalization and variational dequantization with a neural network model, we observed the highest average relative improvement over random forest: 4.51% in accuracy and 11.22% in F-1 score. It can be observed that the improvement in F-1 score with our proposed methods is significant as compared to that in accuracy. It can be attributed to better encoding of the minority classes, an added benefit for imbalanced datasets commonly found in smart buildings. An important observation to note is that for subjects 4,8,9, and 14, the classification accuracy degrades with the implementation of neural networks and proposed pre-processing methods. One of the reasoning for the the same can be the extreme class-imbalance found in thermal preference classes for those subjects as observed in Fig 2. The ratio between sum of all the minority classes and the single majority class for these subjects is as high as 1:7. However, the F-1 score always improves with implementation of proposed pre-processing methods. Since our methods are specifically designed for neural networks and not random forest models, a fair separation and ablation study of machine learning models and the pre-processing method to understand the contribution of each towards the improvement/degradation is difficult in this particular case. However, keeping the neural network model fixed, when we implement standard normalization, mode-based normalization + uniform dequantization, and mode-based normalization + variational dequantization, the classification scores increases in that order across all of the subjects. This proves that the combination of the above proposed methods is beneficial for tabular data preprocessing in smart buildings. The choice of transformation method to be used depends on the particular application, and the machine learning models that are planned to be implemented (Sec 3.2.2).

5 CONCLUSION AND FUTURE WORK

In this research, we proposed the use of several novel data transformation methods for use in tabular data pre-processing, namely mode-specific normalization (for continuous features), and uniform and variational dequantization (for discrete features). We conducted experimental analysis of thermal comfort prediction models (both group-based and personal thermal comfort) with the above data pre-processing methods, and showed significant improvement in classification accuracy and F-1 score as compared to state-of-the-art results. In Sections 3.2.1, and 3.2.2, we also summarized the scenarios when the above methods can be used. Since the methods proposed are generalizable for any tabular data, they can be seamlessly used for any smart building tabular dataset, and can aid in efficient machine learning system design.

A line of future work include the study of performance improvement by using the proposed pre-processing methods in several smart building and energy system machine learning tasks, e.g. timeseries based energy use forecasting, demand response, occupancy detection using CO_2 concentration data in return air, HVAC control using reinforcement learning, etc. As stated in this work, the methods can be used with certain modifications for other structured data such as graphical data, and unstructured data such as images. Hence, another line of research can be to study the required modifications, and implementations for use cases in smart buildings and energy systems involving such datasets, e.g. power transmission in grids organized as graphs, and satellite imagery for buildings.

REFERENCES

[1] Ioannis Antonopoulos, Valentin Robu, Benoit Couraud, and David Flynn. 2021. Data-driven modelling of energy demand response behaviour based on a large-scale residential trial. *Energy and AI* 4 (2021), 100071.

[2] Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. 2018. Invertible residual networks. *arXiv preprint arXiv:1811.00995* (2018).

[3] Tanaya Chaudhuri, Yeng Chai Soh, Hua Li, and Lihua Xie. 2017. Machine learning based prediction of thermal comfort in buildings of equatorial Singapore. In *2017 IEEE International Conference on Smart Grid and Smart Cities (ICSGSC)*. IEEE, 72–77.

[4] Tanaya Chaudhuri, Deqing Zhai, Yeng Chai Soh, Hua Li, and Lihua Xie. 2018. Thermal comfort prediction using normalized skin temperature in a uniform built environment. *Energy and Buildings* 159 (2018), 426–440.

[5] Ricky TQ Chen, Jens Behrmann, David Duvenaud, and Jörn-Henrik Jacobsen. 2019. Residual Flows for Invertible Generative Modeling. *arXiv preprint arXiv:1906.02735* (2019).

[6] Hari Prasanna Das, Ryan Tran, Japjot Singh, Xiangyu Yue, Geoff Tison, Alberto Sangiovanni-Vincentelli, and Costas J Spanos. 2021. Conditional Synthetic Data Generation for Robust Machine Learning Applications with Limited Pandemic Data. In *arXiv preprint arXiv:2109.06486*.

[7] Laurent Dinh, David Krueger, and Yoshua Bengio. 2014. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516* (2014).

[8] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. 2016. Density estimation using Real NVP. *CoRR* abs/1605.08803 (2016). arXiv:1605.08803

[9] Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. 2019. Flow++: Improving Flow-Based Generative Models with Variational Dequantization and Architecture Design. *arXiv preprint arXiv:1902.00275* (2019).

[10] Prageeth Jayathissa, Matias Quintana, Mahmoud Abdelrahman, and Clayton Miller. 2020. Humans-as-a-sensor for buildings—intensive longitudinal indoor comfort models. *Buildings* 10, 10 (2020), 174.

[11] Wooyoung Jung and Farrokh Jazizadeh. 2019. Comparative assessment of HVAC control strategies using personal thermal comfort and sensitivity models. *Building and Environment* 158 (2019), 104–119.

[12] Siavash H Khajavi, Naser Hossein Motlagh, Alireza Jaribion, Liss C Werner, and Jan Holmström. 2019. Digital twin: vision, benefits, boundaries, and creation for buildings. *IEEE access* 7 (2019), 147406–147419.

[13] Arash Khalilnejad, Roger H French, and Alexis R Abramson. 2020. Data-driven evaluation of HVAC operation and savings in commercial buildings. *Applied Energy* 278 (2020), 115505.

[14] Durk P Kingma and Prafulla Dhariwal. 2018. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*. 10215–10224.

[15] Ioannis C Konstantakopoulos, Hari Prasanna Das, Andrew R Barkan, Shiying He, Tanya Veeravalli, Huihan Liu, Aummul Baneen Manasawala, Yu-Wen Lin, and Costas J Spanos. 2019. Design, benchmarking and explainability analysis of a game-theoretic framework towards energy efficiency in smart infrastructure. *arXiv preprint arXiv:1910.07899* (2019).

[16] Tobias Kramer, Veronica Garcia-Hansen, Sara Omrani Vahid M Nik, and Dong Chen. 2021. A Machine Learning approach to enhance indoor thermal comfort in a changing climate. In *Journal of Physics: Conference Series*, Vol. 2042. IOP Publishing, 012070.

[17] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. 2019. MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks. In *International conference on artificial neural networks*. Springer, 703–716.

[18] Veronika Földváry Ličina, Toby Cheung, Hui Zhang, Richard De Dear, Thomas Parkinson, Edward Arens, Chungyoon Chun, Stefano Schiavon, Maohui Luo, Gail Brager, et al. 2018. Development of the ASHRAE global thermal comfort database II. *Building and Environment* 142 (2018), 502–512.

[19] Shichao Liu, Stefano Schiavon, Hari Prasanna Das, Ming Jin, and Costas J Spanos. 2019. Personal thermal comfort models with wearable sensors. *Building and Environment* (2019), 106281. https://doi.org/10.1016/j.buildenv.2019.106281

[20] Maohui Luo, Zhe Wang, Kevin Ke, Bin Cao, Yongchao Zhai, and Xiang Zhou. 2018. Human metabolic rate and thermal comfort in buildings: The problem and challenge. *Building and Environment* 131 (2018), 44–52.

[21] Rasmus Luthander, Annica M Nilsson, Joakim Widén, and Magnus Åberg. 2019. Graphical analysis of photovoltaic generation and load matching in buildings: A novel way of studying self-consumption and self-sufficiency. *Applied Energy* 250 (2019), 748–759.

[22] Aditya Mishra. 2018. Metrics to Evaluate your Machine Learning Algorithm. https://tinyurl.com/clf-metrics

[23] Nikolaos Nasios and Adrian G Bors. 2006. Variational learning for Gaussian mixture models. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 36, 4 (2006), 849–862.

[24] Jack Ngarambe, Geun Young Yun, and Mat Santamouris. 2020. The use of artificial intelligence (AI) methods in the prediction of thermal comfort in buildings: Energy implications of AI-based thermal comfort controls. *Energy and Buildings* 211 (2020), 109807.

[25] Matias Quintana, Stefano Schiavon, Kwok Wai Tham, and Clayton Miller. 2020. Balancing thermal comfort datasets: We GAN, but should we?. In *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 120–129.

[26] Navigant Research Report. [n. d.]. Intelligent Building Technologies for Sustainability.

[27] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. 2017. PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications. *CoRR* abs/1701.05517 (2017). arXiv:1701.05517 http://arxiv.org/abs/1701.05517

[28] Andrew Sonta, Thomas R Dougherty, and Rishee K Jain. 2021. Data-driven optimization of building layouts for energy efficiency. *Energy and Buildings* 238 (2021), 110815.

[29] Kailai Sun, Qianchuan Zhao, and Jianhong Zou. 2020. A review of building occupancy measurement systems. *Energy and Buildings* 216 (2020), 109965.

[30] Mohammad Taleghani, Martin Tenpierik, Stanley Kurvers, and Andy Van Den Dobbelsteen. 2013. A review into thermal comfort in buildings. *Renewable and Sustainable Energy Reviews* 26 (2013), 201–215.

[31] Rui Tang and Shengwei Wang. 2019. Model predictive control for thermal energy storage and thermal comfort optimization of building demand response in smart grids. *Applied Energy* 242 (2019), 873–882.

[32] Ahmet Uğursal and Charles H Culp. 2013. The effect of temperature, metabolic rate and dynamic localized airflow on thermal comfort. *Applied energy* 111 (2013), 64–73.

[33] Benigno Uria, Iain Murray, and Hugo Larochelle. 2013. RNADE: The real-valued neural autoregressive density-estimator. In *Advances in Neural Information Processing Systems*. 2175–2183.

[34] Zhe Wang, Hui Zhang, Yingdong He, Maohui Luo, Ziwei Li, Tianzhen Hong, and Borong Lin. 2020. Revisiting individual and group differences in thermal comfort based on ASHRAE database. *Energy and Buildings* 219 (2020), 110017.

[35] Lei Xiong and Ye Yao. 2021. Study on an adaptive thermal comfort model with K-nearest-neighbors (KNN) algorithm. *Building and Environment* 202 (2021), 108026.

[36] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. Modeling tabular data using conditional gan. *Advances in Neural Information Processing Systems* 32 (2019).

[37] Zhun Yu, Benjamin CM Fung, Fariborz Haghighat, Hiroshi Yoshino, and Edward Morofsky. 2011. A systematic procedure to study the influence of occupant behavior on building energy consumption. *Energy and buildings* 43, 6 (2011), 1409–1417.

[38] Jie Zhao, Bertrand Lasternas, Khee Poh Lam, Ray Yun, and Vivian Loftness. 2014. Occupant behavior and schedule modeling for building energy simulation through office appliance power consumption data mining. *Energy and Buildings* 82 (2014), 341–355.

[39] Zimu Zheng, Yimin Dai, and Dan Wang. 2019. DUET: Towards a portable thermal comfort model. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 51–60.

[40] MS Zuraimi, A Pantazaras, KA Chaturvedi, JJ Yang, KW Tham, and SE Lee. 2017. Predicting occupancy counts using physical and statistical Co2-based modeling methodologies. *Building and Environment* 123 (2017), 517–528.