7th International Project and Construction Management Conference (IPCMC2022)
Yildiz Technical University, Faculty of Civil Engineering, Department of Civil Engineering, İstanbul, Turkey

Handling Imbalanced Data in Building Energy Benchmarking

Said Bolluk, Senem Seyis
Özyeğin University, Department of Civil Engineering, İstanbul, Turkey
said.bolluk@ozu.edu.tr, senem.seyis@ozyegin.edu.tr

Abstract

The energy demand of the building stocks reached a critical value by almost covering 30% of the world's overall energy consumption. Therefore, the importance of building energy benchmarking gains importance. Building energy benchmarking might suffer from imbalanced datasets as a classification model. Using an existing dataset to observe the effectiveness of several machine learning models in building energy benchmarking in Seattle, several classification algorithms are first utilized to predict the buildings' energy benchmark label over an imbalanced dataset. Cost-sensitive loss functions and several decision thresholds are then introduced to the artificial neural network (ANN) models, and their effectiveness in handling imbalanced data is examined. Finally, several classification evaluation metrics are analyzed. The results display that the cost-sensitive approach with an empirical decision threshold slightly improved the loss of the ANN models and classification of the minority class. This study shows that the cost-sensitivity concept in ANN is a promising study field, and significant improvements in loss reduction can be achieved through empirical hyperparameter testing. Another prominent result is that machine learning could render building energy benchmarking possible at the urban scale as their results are repeatable for different scopes and regions. Therefore, this study underlines the importance of data-driven methods in urban building energy performance assessment.

Keywords: *Automation in Construction, Building Energy Benchmarking, Cost-Sensitive Learning, Machine Learning.*

1. Introduction

Building energy performance assessment emerges as a necessity considering the massive greenhouse gas (GHG) emissions caused by the built environment (Hong et al., 2020). Owing to population growth and, consequently, the expansion of cities, the energy demand for building stocks reached a critical threshold by almost covering 30% of the world's overall energy consumption (Ritchie, 2018). This brings the following questions: Is it possible to correctly forecast the consumption behavior of the built environment in urban areas, and if so, how to conduct the energy estimation of the urban building stocks with various characteristics? Even though the answer is ambiguous, one thing is certain that models with insights into the urban building stock's energy consumption patterns can potentiate sustainable environment practices over retrofit and urban planning actions. Here, one of the essential actors is the building energy benchmarking models. Building energy benchmarking is a process that evaluates the internal and external performance of a building and whether the building complies with the relevant regulations (Ohlsson & Olofsson, 2021). This forms a classification task for the modelers with the features of the building's energy-related parameters and the target variable, showing the building's energy benchmark label. Classification models in machine learning suffer from imbalanced datasets. While training such models, the imbalanced distribution of the class labels might induce misclassification. This is due to the update of the model parameters occurring inevitably according to specific classes with numerical superiority (Khan et al., 2015). Despite the rapid advancements in the deep learning domain, the class imbalance problem, which is frequent in real-world data collection scenarios, remains challenging for most classification tasks. In the literature, the research on the class imbalance problem is investigated from two central angles: Data Level Approach and Algorithm Level Approach (Hou et al., 2021). In data-level approaches, balanced datasets are acquired via either oversampling the minority classes or undersampling the majority classes. On the other hand, algorithm-level approaches operate in the training phase and try to amplify the effect of minority class representation in the optimization process. Loss functions are critical components of every optimization problem, as they define an objective on which the model's performance is evaluated. Loss functions measure the overall distance, or error, between the estimated and actual value of a specific input set. Based on this evaluation, internal updates of the model weights are affected when performing backpropagation in artificial neural network (ANN) models. Depending on the scenario, different loss functions are used for developing a well-defined objective. Therefore, there is no ground truth in concatenating loss functions to algorithms in machine learning. The most important factors to be considered when deriving a loss function are the selected model, the optimization approach with the derivatives' property, and the dataset's characteristics. One of the most popular loss functions used in machine learning is cross-entropy loss, a metric used to evaluate classifier performance. However, as mentioned above, the cross-entropy loss might be paralyzed when handling imbalanced datasets. As an algorithm-level approach to dealing with an imbalanced dataset, cost-sensitive loss functions might be an alternative. The cost-sensitive approach minds the distribution of the class labels in a dataset and calculates the error of the model's prediction regarding those distributions (Takase et al., 2018). With this approach, cost-sensitive models can classify the datasets through a balanced training process. For example, Focal Loss, which is a modified version of the cross-entropy loss, down-weights the examples that are easy to train (majority classes) and focuses on learning complex examples (instances from minority classes) with tunable controlling parameters (hyperparameters) (Lin et al., 2017).

Using an existing dataset, this study analyzes the relationship between the building-specific parameters and the energy efficiency and performs the followings: (1) Data preprocessing is applied to remove the unnecessary features and outliers. (2) Several classification algorithms are then performed to predict the buildings' energy benchmark label over imbalanced data. (3) Cost-sensitive loss functions with several decision thresholds are introduced to the ANN models, and their effectiveness in handling imbalanced data is examined. (4) Finally, several classification evaluation metrics are analyzed to determine the optimal classifier.

2. Methodology

2.1 Data Preprocessing

A building energy benchmarking dataset was obtained from an open-source platform by the city of Seattle (2020 Building Energy Benchmarking, n.d.). The recordings are from 2020, with 3628 specific buildings and 42 energy-related features. Although there are similar datasets in the same platform between 2015 and 2019, the 2020 dataset reserves the richest content with more buildings and features. In the data imputation part, mean and median values of the relevant features were used to deal with the missing values of the numerical features, including the largest property use type and the site energy use. The categorical variables were then encoded into binary expressions since the classification algorithms could only work with numerical values. In this sense, the target variable, concluding whether or not a building is compliant for energy benchmarking, is transformed into one (1) for compliant and zero (0) for not compliant. The next topic was the distribution of the classes. In the dataset, 97.2% of the instances were compliant, while 2.8% were not compliant. This indicates that the dataset is highly imbalanced. Such imbalanced distributions might cause undertraining of the model for the minority class, zeros, since the model's parameters are destined to be updated according to the majority class, ones. To lessen this issue's effect, 60% of the classes with compliance statutes were removed from the model. The updated distributions were 93% and 7% for the compliant and not compliant classes. Even though the numbers are still imbalanced, decreasing the number of the dataset might induce additional problems. Plus, this study aims to observe the power of different machine learning models in handling imbalanced data. Therefore, the authors assumed this distribution well enough to perform the following classification task. Finally, the processed dataset was divided into the training and test sets as 25% of the data for testing purposes.

2.2 Model Development

The classification models examined in this study are as follows: Support Vector Machine (SVM), Multinomial Naive Bayes, Logistic Regression, K-Nearest Neighbors, Decision Tree, Random Forest, and Multilayer Perceptron (MLP). These classifiers have various settings and parameters that may lead to utterly different outputs. Therefore, automating the process and obtaining the best hyperparameter settings for each classifier was necessary. To do this, the grid search procedure was utilized. A grid search is a framework that analyzes the power of the machine learning models according to their hyperparameters enabling statistical calculations. It performs cross-validation over the distinct parameters and provides the users with the optimal settings. The grid search mechanisms were established for each model according to their characteristics, and the classification process was initiated. The analysis

was performed in Python using the following libraries: ***Scikit-learn*** for the classical machine learning models and ***PyTorch*** for the MLP models.

The first model was the SVM Classifier. SVM models create a hyperplane that discriminates different classes in the best manner (Shalev-Shwartz & Ben-David, 2014). This separation is achieved using a loss function that maximizes the distance between each class's hyperplane and extreme data points. The regularization parameters and kernel types were the hyperparameters evaluated in the grid search. Here, the regularization parameters, ranging between 0.01-1.0, form the minimum threshold for the algorithm to make an accurate prediction, while the kernel types (e.g., linear or polynomial) determine the numerical approach that the algorithm will follow. The second model was the Multinomial Naïve Bayes Classifier. The Bayes decision theory is based on observing the prior distributions of classes to derive posterior distributions. It is a very naïve assumption that the data dataset has a particular distribution and the features are statistically independent (Alpaydın, 2020). In the case of this study, neither the dataset had a particular distribution nor the features are not correlated. Therefore, the Multinomial Naïve Bayes model was employed since it can handle multinomially distributed data. Next, the Logistic Regression model was created and tested in the grid search. Logistic Regression models calculate the probability of occurrence of classes using the sigmoid function. The output for each class is a probability ratio. The model updates its parameters using the gradient of the calculated loss calculated between the output and the actual probabilities. The tested hyperparameters of the Logistic Regression were the regularization parameters, ranging from 0.01-0.6, and the solver parameters used in result optimization. The third model was the K-Neighbors Classifier. The grid search observed the following hyperparameters: the number of neighborhoods and the measurement tools for the weight and metric. The class label of a data point is determined according to the nearest points' label with a certain number of neighbors set in this classification model. The uniform or distance weights identify the method used in calculating the impact of the nearest data points on the observed data. Finally, the metrics, such as Euclidean, Manhattan, and Minkowski, indicate the methods used in calculating the distance between data points. The fourth model was the Decision Tree Classifier. Decision Trees are non-parametric models that evaluate the dataset in a hierarchical order. This means that the model sets some initial weights over the first observed data points, increases the number of weights, and updates their values as it progresses on more data points. Each iteration generates decision boundaries for the model (Hastie et al., 2009). The hyperparameters tuned in grid search were the criterion that calculates the loss of each split, the splitter strategy according to the calculated loss, and the class weights that balance the learning process. The Random Forest Classifier was the following algorithm analyzed through the grid search. The parameters observed in the mechanism were the number of estimators and the criterion types for data splitting. The number of the estimator, ranging between 10-100, is the number of trees (decision trees) created by the model to split the data points into clusters and measure the aggregated accuracy. The criterion parameters, such as the Gini Index and entropy, are used to detect the impurity level of the dataset (Hastie et al., 2009).

The final model was the MLP Classifier. MLP is the perceptron algorithm having multiple hidden layers. This is also called Artificial Neural Network (ANN) model. MLP functions differently than most other machine learning models. In MLP, features are evaluated as single sets (input layers) rather than the dataset as a whole (Figure 1). This enables MLP to update model weights in each iteration according to the loss calculated over the utilization of each input layer. This process is called online learning (Alpaydın, 2020). The input layer either consists of single instance or an instance set. In machine learning literature, the size of the

input layer, which is the number of instances the model evaluates at each iteration, is called batch size. For a classification task in machine learning, let us say a set of x represents the inputs of a single instance where the batch size is one, the set of w represents the corresponding weights, and the n represents the total number of data points, then the probability of each output y is calculated using Equation 1:

$$y_i = \sum_{j=1}^{n} w_{ij} x_j + w_{i0} \tag{1}$$

The weighted sums are then placed in activation functions that transform the relationship into a non-linear form. Following this path, the final process is to put the activated sums into the final activation function, which will return a value specific to the aim of the analysis. For example, if the analysis is a regression, the final activation layer should return a linear value, whereas if it is classification, the final outputs might be a probability value for each class. After calculating the possible outcomes in forward-pass calculations, the MLP model determines the loss between the outcomes and actual values and updates the model parameters using Stochastic Gradient Descent. This is called backpropagation (Dong & Yang, 2018) (Jeewajee & Kaelbling, 2020). This update lasts until achieving a certain amount of convergence. In the grid search structure, several hyperparameters of the MLP model were tested to create a baseline model. The hidden layer size and the number of units in each hidden layer determine the complexity of the MLP model. In this study, considering the computational complexity, the authors decided to develop a simple layer setting with nine hidden units. This is the mean of the input (16) and output sizes (2). However, an empirical analysis should be done with the advanced test setups to decide the optimal layer settings.

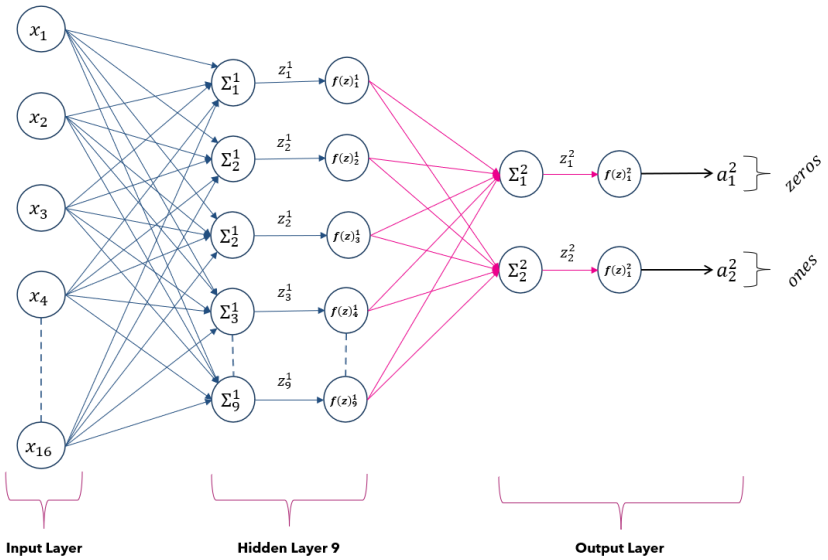

Figure 1. Multilayer Perceptron Architecture

Once creating a baseline architecture for the MLP, the aim was to increase the accuracy of the classification of the imbalanced dataset through a more advanced hyperparameter tuning. This

tuning includes the following hyperparameters: the activation function, learning rate, weight initialization, momentum, and loss function. The activations functions were the sigmoid and tanh functions. Learning rate α is the parameter ranging between 0-1.0 that determines the scope of the update along with the gradient of the loss (Equation 2):

$$\Delta w_i = \alpha \times \frac{\partial Loss}{\partial w_i} \tag{2}$$

To initialize the weights of the MLP, three different initializations were tested. The first was the random initialization that creates weights according to the Gaussian distribution with zero mean and one standard deviation. The second one is Xavier initialization which produces a uniform distribution with an exponential term based on the number of inputs (He et al., 2015). The last one is the He Initialization generating weights according to the Gaussian distribution, where the mean and standard deviation are both the square root of the input numbers (He et al., 2015). Weight generalization in ANN is a crucial topic since initial weights might have an enormous effect on the results. For example, a proper initialization might yield faster convergence. On the contrary, randomly initializing the weight in deep networks might cause a vanishing gradient problem (Marion et al., 2022). When this happens, the model's loss stops improving, and the weights get stuck at certain values. Therefore, these three methods in weight initialization were tested in this study. Another hyperparameter tested in multiple scenarios was momentum. When the weights update for each iteration varies significantly from the others, the change in the model's loss exhibits fluctuant patterns. Momentum helps prevent this fluctuation and provides a stable convergence. The momentum value is usually taken between 0.5-1.0 (Alpaydın, 2020).

Finally, and most importantly, different loss functions were evaluated to understand the cost-sensitivity concept in ANN. The most common loss function for classification is the Cross-Entropy (CE) Loss. Assume that n is the number of classes, which is two in our case, r is the occurrence probability of the target (actual) label, and y is the predicted label. Then, the binary CE Loss for each iteration is calculated using Equation 3:

$$CE\ Loss = -\sum_{i=1}^{k=2} r_i \times \log y_i \tag{3}$$

CE Loss provides an efficient learning process for classification problems. However, when the data is imbalanced, the model cannot update its parameters according to the minority class, zeros. Even though the accuracy might be high, the cost raised by the misclassification of the minority class would cause some significant problems. For example, the effect of misclassification can be more evident in cancer detection (Qader et al., n.d.) or fraud detection in the banking sector (de La Bourdonnaye & Daniel, 2022). There are some cost-sensitive approaches in the literature accordingly. One of the most famous cost-sensitive loss functions is called Focal Loss (Lin et al., 2017). Focal Loss is created through an empirical process, and its assumption is based on a random constant updating the loss, called gamma γ (Equation 4):

$$\text{Focal Loss} = (1 - e^{-CE\ Loss})^{\gamma} \times CE\ Loss \tag{4}$$

To improve both the CE Loss and Focal Loss, two distinct class ratios were derived from the class distribution of the dataset: *one-ratio* (0.933) and *zero-ratio* (0.067). These ratios were then used in different loss functions. According to the efficiency of the tested functions, the

authors chose to continue with only one loss function: Cost-Sensitive Exponential CE Loss (Equation 5):

$$\text{Cost-Sensitive Exponential CE Loss (CSCE)} = \begin{cases} CE\ Loss^{(1+zero_{ratio})}, & r = 1 \\ CE\ Loss^{(1+one_{ratio})}, & r = 0 \end{cases} \quad (5)$$

It should be remembered that the update of the ANN models is performed over the gradient of the loss. r is the occurrence probability of each class when the target label is known. For example, when the target is one, r_1 = 1 and r_0 = 0. However, if r_0 is taken zero when the target label is one, it is impossible to update the weights of the hidden units (activation and pre-activation units) that generate the predictions for the zero class. If the zeros are notably less in number than the ones, the weights of the hidden units cannot be updated and thus the model cannot predict zeros correctly. Therefore, this study determined a fixed occurrence probability of 0.1 for zero instances when the target label is one. Equation 5 introduces the class ratios in the loss function as exponential terms, thus the gradient term would have two constants with two class ratios. In this case, on the assumption that z is the pre-activated output and y is the model's prediction when the target label r is zero, the multiplier term aims to increase the loss where the exponential term becomes one_{ratio} and has a decreasing effect on the weight update since the base CE Loss is a between zero and one (Equation 6). However, as the class distributions are getting more imbalanced, the one_{ratio} increases. The term $CE^{one_{ratio}}$ then cannot drastically decrease the overall loss and update but balances the loss increment caused by the multiplier. Thus, it might provide a more balanced learning process. The inverse effect of this assumption applies to the cases when the target label is one with smaller loss amplification due to a smaller class ratio, $zero_{ratio}$. In brief, the proposed loss function enabled the weights of the hidden units for zeros to be constantly updated even when the target is one and provided enhanced updates when the target is zero.

$$\frac{\partial CSCE}{\partial w} = (1 + one_{ratio}) \times CE^{one_{ratio}} \frac{\partial CE}{\partial a} \times \frac{\partial a}{\partial z} \times \frac{\partial z}{\partial w} \quad (6)$$

2.3 Classification Evaluation Metrics

Evaluating the reliability of classification models with imbalanced data requires intense attention that is far beyond merely examining the accuracy of the results. This is because the models' behavior might result in favor of the majority classes. In such cases, as in our case, the class label of the truly and falsely predicted labels might be more important than the absolute accuracy of the model. Therefore, one should consult multiple evaluation metrics to comprehend the reliability of the classification. To that end, the authors read the classification results over the following metrics: Accuracy, F1 Score, and Brier Score Loss (Table 1). Lastly, confusion matrices were introduced as visual tools to summarize the classification, showing the truly predicted labels in the diagonals.

Table 1. Classification Evaluation Metrics

Evaluation Metric	**Formula**
Accuracy	$\frac{1}{n_{samples}} \times \sum_{i=1}^{n_{samples}} \begin{cases} 1, & y_i = r_i \\ 0, & y_i \neq r_i \end{cases}$
F1 Score	$\frac{TP}{TP + \frac{1}{2}(FP + FN)}$
Brier Score Loss (Mean-Square Difference)	$p_i = \Pr(y_i = r_i)$ $\frac{1}{n_{samples}} \times \sum_{i=1}^{n_{samples}} (r_i - p_i)^2$

3. Results

In the first part of the analysis, six classical machine learning models were initiated with their optimal hyperparameters, and their performance in classifying the imbalanced dataset was observed. For each classification metric, including the Accuracy, F1 Score, and Brier Score, the Random Forest Classifier outperformed the other eight classifiers (Table 2). However, these metrics cannot be merely sufficient to interpret the misclassification of different classes. Therefore, the confusion matrices should also be examined.

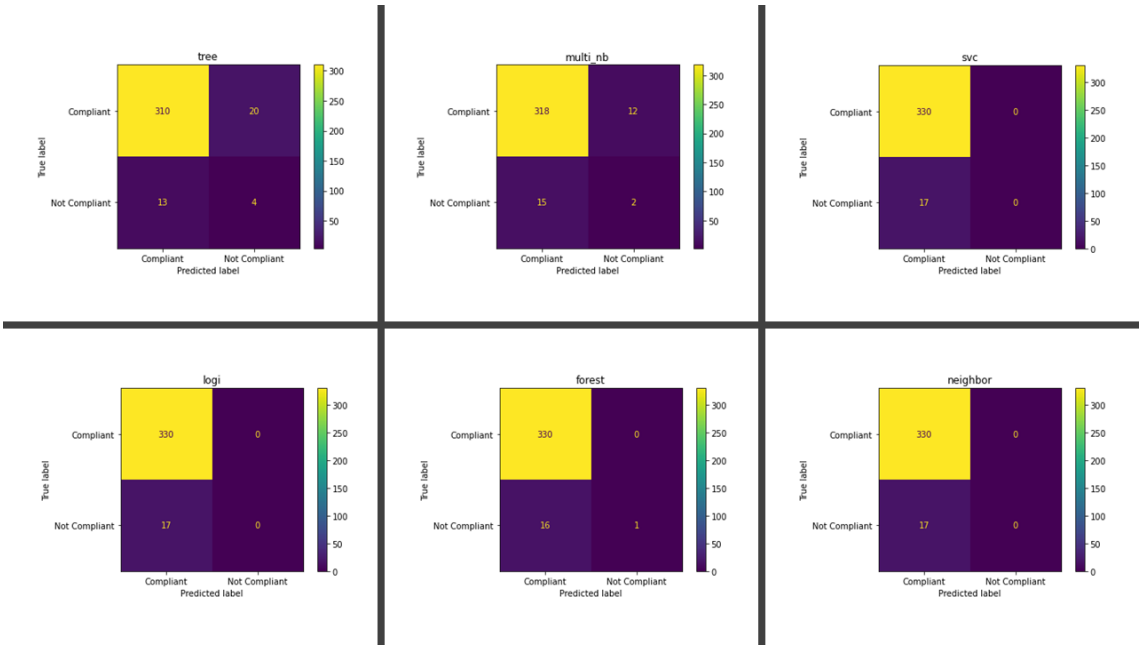

Figure 2. Confusion matrices of the classical models

Even though the classical models exhibited satisfactory results in terms of standard evaluation metrics, it is seen that most models poorly predicted the labels of the minority class, zeros (Figure 2). These models falsely predicted most zeros since the models considered a specific distribution (e.g., Gaussian) and made estimations over the distribution parameters, where the assumed distribution was not perfectly Gaussian for each feature distribution. Plus, it was almost impossible to manipulate their framework in calculating the final loss. This is very natural because these models were trained based on the weights of the ones as they occurred almost fourteen times more than the zeros in the analysis. However, there are outliers for the classical models: The Decision Tree and Multinomial Naïve Bayes. These models performed slightly balanced in predicting different class labels. For example, the Decision Tree classifier faintly learned the weights in favor of the minority class, zeros, as it predicted four (4) true negatives (Figure 2). Nevertheless, it only predicted 310 true positives, significantly affecting the model's overall accuracy (Table 2). On the other hand, the MLP models were not the optimal ones in terms of standard evaluation metrics (Table 2). However, considering the number of true positives and true negatives, MLP models provided the most balanced distribution between classes (Figure 3). This is because the ANN structure enables modelers to conduct extensive hyperparameter tuning. In this study, an empirical analysis was performed for the hyperparameters of the MLP models, including the loss functions and decision thresholds. The MLP models were trained to classify labels for the benefit of the minority class, zeros. These models were integrated with loss functions that penalize the final loss when the instance label is one. This way, they can update the weights according to the minority class, zeros. Plus, an experimental decision threshold of %65 enabled models to make classifications only when they were confident. It should be noted that experimental decision thresholds were also assessed for the classical machine learning models. However, the authors did not record any improvement for the models with such an arrangement. In this context, it can be said that MLP models are more inclined to learn the instances from the minority class, zeros, owing to the cost-sensitive loss functions. Among the MLP models, the one with cost-sensitive exponential loss produced the best results. It predicted three (3) true negatives and 320 true positives (Figure 3).

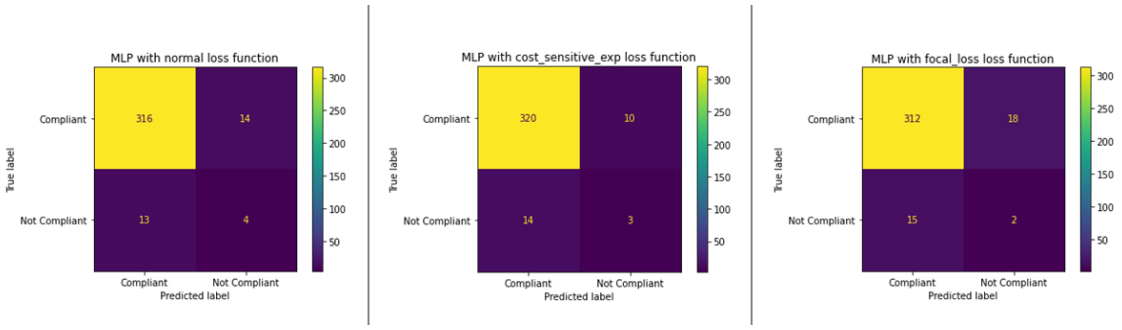

Figure 3. Confusion matrices of the MLP models with different loss functions

Table 2. Classification Results

Classifier	Accuracy	F1 Score	Brier Score	TP	FP
Support Vector Machine	0.951	0.975	0.048	330	0
Multinomial Naive Bayes	0.951	0.975	0.047	318	2
Logistic Regression	0.951	0.975	0.047	330	0
K-Nearest Neighbors	0.951	0.975	0.052	330	0
Decision Tree	0.905	0.949	0.095	310	4
Random Forest	0.954	0.976	0.045	330	1
MLP with Normal Loss	0.922	0.959	0.105	316	4
MLP with Focal Loss	0.905	0.95	0.127	312	2
MLP with Cost Sensitive Exponential Loss	0.931	0.964	0.103	320	3

4. Discussion

This study utilized different classification metrics and confusion matrices to interpret the results. Among the classification evaluation metrics, the Brier Score Loss appears to be the most reasonable metric to compare the results. This is because the Brier Score evaluates the probability of the class labels that the models make and in calculating the loss. Such a loss value holds excellent insights into the confidence level of the models in making predictions. Hence, looking at the Brier Score Loss, it is evident that the Random Forest Classifier outperformed the other models in classifying the building energy benchmark label. However, considering the estimations on the minority class, the MLP models, especially the one with cost-sensitive exponential CE loss, exhibited the most balanced distribution between classes (Figure 3). In this sense, it is evident that MLP models are powerful in handling imbalanced datasets.

This study contains some limitations. The first one is the lack of computational power. When developing MLP models with optimal hyperparameters, more than 300 million scenarios can be tested over several hyperparameters, including the learning rate, initialization type, activation functions, number of epochs, momentum, hidden layer size, and loss functions. ANN is a nascent field that offers independence in performing empirical studies. This makes it possible to improve the model's predictive capacity via hyperparameter tuning. However, in this study, only a limited number of scenarios were tested on the model's hyperparameter

settings. The authors believe the results could display remarkable improvements with sufficient computational power. The other limitation is the size of the dataset. Available datasets are separated into training, validation, and test sets in machine learning studies. At first, the model is trained using a training set, and its performance is observed over the validation set. This part generates the validation curve (Figure 7), which is very helpful in analyzing the change in the loss calculated for each epoch. Once the optimal hyperparameters are determined, the model is trained once more using these hyperparameters. The trained model's weights are then used to analyze the model's performance over the test set. This is the final process where the analysis report with the accuracy curve (Figure 7) is published using the test results. In our case, the original dataset had around 3300 instances. This number was reduced to around 1400 instances to balance the class distribution slightly. Due to the limited data and, more importantly, the size of the minority class (zeros), this study only divided the datasets into training and test sets. Then, the test set was both used in model validation and testing. This assumption forms an internal consistency for the results as it can represent the training and test sets fairly. However, when new instances are appended to the model, the predictions might not cover the actual labels, and thus the generalization capacity of the model might suffer. This suggests that the dataset should be big enough to be divided into training, validation, and test sets to provide an external consistency for the model's results.

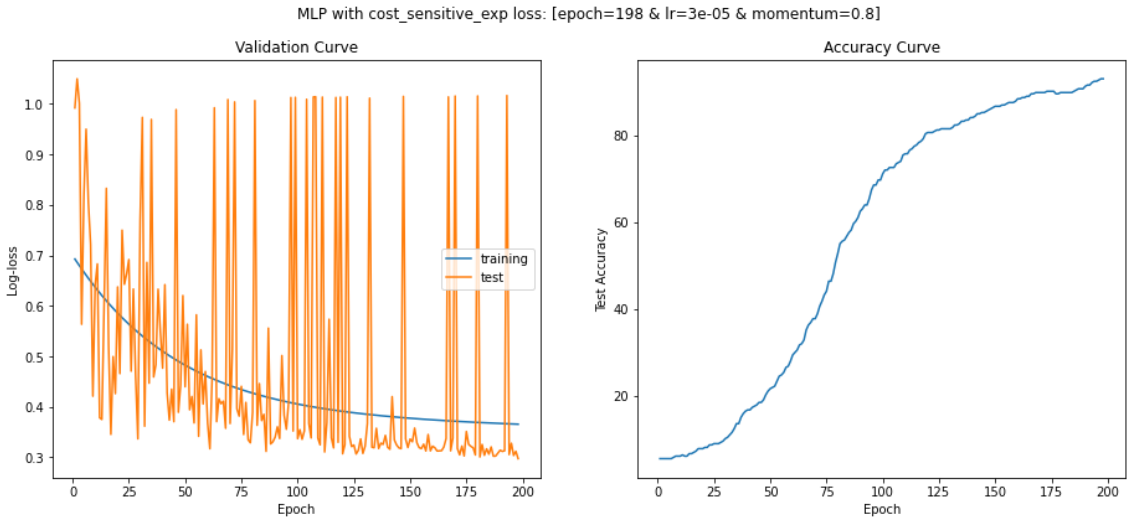

Figure 4. Validation and accuracy curves of the MLP model with cost-sensitive exponential CE loss

There is a significant fluctuation in the validation curve in Figure 4. Most features in the dataset do not have the Gaussian distribution, and they include many outliers. This complicates the accurate prediction of the labels with the parametric approach, analyzing the distribution parameters (e.g., mean and variance) and making estimations accordingly. Moreover, the size of the test set is significantly small. When the variance of the test set is high due to the small number of samples, the trained model cannot fit the test set, and steady patterns in the loss reduction cannot be observed for the test set. This explains the oscillation in the validation curve. Creating synthetic data to increase the number of samples or employing non-parametric methods, which do not require information derived from the feature distributions, might solve this problem.

5. Conclusion

Building energy benchmark labels were predicted via several classification models in this study. Six classical machine learning models were tested with their optimal hyperparameters settings, which were automatically found via the grid search. The MLP models were then manually examined to decide on the best hyperparameters. Here, the most significant part was to analyze the power of different cost functions in handling imbalanced data. To do this, three MLP models were integrated with different loss functions. Overall result interpretation was performed using different classification metrics. Here, the Random Forest was the best model from the point of the standard evaluation metrics (Table 2), where the MLP models were the winning models in truly predicting instances from the minority class, zeros (Figure 3). This study showed that the cost-sensitivity concept in ANN is a promising study field, and significant improvements in loss reduction can be achieved through empirical testing. Even though the number of accurate predictions for the minority class by the cost-sensitive loss functions is not great, the assumption of this study can be comprehensible and enhanced with a large dataset. Therefore, a possible future work of this study could be working with a large dataset to decrease the variance between training and test sets and increase the generalization power of the classification. Another future work could be utilizing an advanced computational configuration to conduct extensive hyperparameter tuning for the MLP models. This study also showed that machine learning models could make accurate building energy benchmarking. Therefore, this study underlines the importance of data-driven methods in district energy performance assessment.

References

2020 Building Energy Benchmarking. (n.d.). Retrieved from City of Seattle: http://www.seattle.gov/energybenchmarking

Alpaydın, E. (2020). Introduction to Machine Learning (Fourth Edition ed.). The MIT Press.

Dong, H.-W., & Yang, Y.-H. (2018). Training Generative Adversarial Networks with Binary Neurons by End-to-end Backpropagation. http://arxiv.org/abs/1810.04714

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning (Second Edition ed.). Springer New York, NY.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. http://arxiv.org/abs/1502.01852

Hong, T., Chen, Y., Luo, X., Luo, N., & Lee, S. H. (2020). Ten questions on urban building energy modeling. Building and Environment, 168. https://doi.org/10.1016/j.buildenv.2019.106508

Hou, Y., Fan, H., Li, L., & Li, B. (2021). Adaptive learning cost-sensitive convolutional neural network. IET Computer Vision, 15(5), 346–355. https://doi.org/10.1049/cvi2.12027

Jeewajee, A. K., & Kaelbling, L. P. (2020). Adversarially-learned Inference via an Ensemble of Discrete Undirected Graphical Models. http://arxiv.org/abs/2007.05033

Khan, S. H., Hayat, M., Bennamoun, M., Sohel, F., & Togneri, R. (2015). Cost Sensitive Learning of Deep Feature Representations from Imbalanced Data. http://arxiv.org/abs/1508.03422

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal Loss for Dense Object Detection. http://arxiv.org/abs/1708.02002

Marion, P., Fermanian, A., Biau, G., & Vert, J.-P. (2022). Scaling ResNets in the Large-depth Regime. http://arxiv.org/abs/2206.06929

Ohlsson, K. E. A., & Olofsson, T. (2021). Benchmarking the practice of validation and uncertainty analysis of building energy models. In Renewable and Sustainable Energy Reviews (Vol. 142). Elsevier Ltd. https://doi.org/10.1016/j.rser.2021.110842

Ritchie, H. (2018). Sector by sector: where do global greenhouse gas emissions come from? Retrieved from Our World in Data: https://ourworldindata.org/ghg-emissions-by-sector

Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding Machine Learning. Cambridge University Press. doi:9781107298019

Takase, T., Oyama, S., & Kurihara, M. (2018). Effective neural network training with adaptive learning rate based on training loss. Neural Networks, 101, 68–78. https://doi.org/10.1016/j.neunet.2018.01.016

Wu, J., Huang, W., de Boer, N., Mo, Y., He, X., & Lv, C. (2022). Safe Decision-making for Lane-change of Autonomous Vehicles via Human Demonstration-aided Reinforcement Learning. http://arxiv.org/abs/2207.00448