electronics

MDPI

Article

Deep-Learning-Based Natural Ventilation Rate Prediction with Auxiliary Data in Mismeasurement Sensing Environments

Subhin Yang, Mintai Kim and Sungju Lee *

Department of Software, Sangmyung University, Cheonan 31066, Republic of Korea
* Correspondence: peacfeel@smu.ac.kr

Abstract: Predicting the amount of natural ventilation by utilizing environmental data such as differential pressure, wind, temperature, and humidity with IoT sensing is an important issue for optimal HVAC control to maintain comfortable air quality. Recently, some research has been conducted using deep learning to provide high accuracy in natural ventilation prediction. Therefore, high reliability of IoT sensing data is required to achieve predictions successfully. However, it is practically difficult to predict the accurate NVR in a mismeasurement sensing environment, since inaccurate IoT sensing data are collected, for example, due to sensor malfunction. Therefore, we need a way to provide high deep-learning-based NVR prediction accuracy in mismeasurement sensing environments. In this study, to overcome the degradation of accuracy due to mismeasurement, we use complementary auxiliary data generated by semi-supervised learning and selected by importance analysis. That is, the NVR prediction model is reliably trained by generating and selecting auxiliary data, and then the natural ventilation is predicted with the integration of mismeasurement and auxiliary by bagging-based ensemble approach. Based on the experimental results, we confirmed that the proposed method improved the natural ventilation rate prediction accuracy by 25% compared with the baseline approach. In the context of deep-learning-based natural ventilation prediction using various IoT sensing data, we address the issue of realistic mismeasurement by generating auxiliary data that utilize the rapidly changing or slowly changing characteristics of the sensing data, which can improve the reliability of observation data.

Keywords: sensor data; environmental data; pattern analysis; semi-supervised learning; ensemble learning; natural ventilation prediction

Citation: Yang, S.; Kim, M.; Lee, S. Deep-Learning-Based Natural Ventilation Rate Prediction with Auxiliary Data in Mismeasurement Sensing Environments. *Electronics* **2023**, *12*, 3294. https://doi.org/10.3390/electronics12153294

Academic Editors: Juan M. Corchado, Byung-Gyu Kim, Carlos A. Iglesias, In Lee, Fuji Ren and Rashid Mehmood

Received: 29 June 2023
Revised: 27 July 2023
Accepted: 28 July 2023
Published: 31 July 2023

1. Introduction

The prediction of natural ventilation using environmental data, such as differential pressure, wind, temperature, and humidity, through IoT sensing, is a crucial aspect for optimizing Heating, Ventilation, and Air Conditioning (HVAC) control and maintaining a pleasant indoor air quality. Among the energy consumptions in building operation, the HVAC system can account for up to 50% of the total energy usage [1]. Therefore, accurate prediction of natural ventilation is a vital factor in achieving efficient energy use in building operation. That is, accurate NVR prediction can reduce carbon emissions using efficient energy by controlling the HVAC system under satisfying occupant thermal comfort. Prediction of accurate NVR can affect not only the improvement of indoor air quality and occupant thermal comfort but also build an efficient operation strategy to adjust scheduling of mechanical ventilation systems. Thus, it can ultimately increase energy efficiency. Initially, methods for predicting natural ventilation depended on basic mathematical equations, analyzing the relationships between natural ventilation and related parameters, such as differential pressure, wind, temperature, and humidity [2–5]. To reduce prediction errors, more precise methods have been developed by fine-tuning the atmospheric layer and conducting empirical observations [6–9]. Ref. [6] analyzed the influence of window

parameters on natural ventilation, while [7] reproduced and measured the ventilation phenomena inside buildings induced by thermal buoyancy. Ref. [8] investigated the impact of natural ventilation on indoor air quality and thermal comfort conditions, and [9] conducted experimental research to evaluate the performance of low-energy cooling systems and their effects on indoor air quality and thermal comfort. In recent years, computational fluid dynamics (CFD) has been employed to analyze not only natural ventilation but also atmospheric flow, aiming for a more accurate prediction of natural ventilation.

Numerical analytical methods for predicting natural ventilation are useful in various fields, including building environment configuration, design for efficient heating and cooling, and energy conservation [10–13]. Refs. [10,11] proposed computational fluid dynamics (CFD) modeling methods for efficient natural ventilation design by analyzing factors such as temperature and humidity within the space. Ref. [12] conducted CFD modeling and analyzed mild–cold climates to assess improvements in indoor temperature and environment. Additionally, ref. [13] explored optimal ventilation design using CFD by analyzing the internal airflow and thermal characteristics of structures. However, it is widely recognized that the numerical analysis approach is challenging for analyzing the dependent relationships among various parameters. To overcome this limitation, a method utilizing machine learning to design a predictive model considering the complex relationship between these various parameters has been introduced [14–17]. In particular, ref. [14] collected insolation, indoor temperature, outdoor temperature, indoor/outdoor temperature difference, indoor humidity, outdoor humidity, indoor/outdoor humidity difference, indoor/outdoor differential pressure, wind direction, and wind speed data based on ten types of IoT sensors. Note that the variable parameters for predicting NVR are selected based on related works, and they are well-known parameters that affect NVR [14]. By applying each of the eight machine learning methods and analyzing the complex interrelationships among the parameters, it has been reported that the amount of natural ventilation can be predicted with high accuracy [14].

Machine-learning-based natural ventilation prediction models rely heavily on empirical observation data, as they are designed by analyzing the complex interrelationships among various environmental parameters. Therefore, to achieve high accuracy in predicting natural ventilation, it is essential to ensure high-quality data. However, collecting accurate data in IoT sensing environments for predicting natural ventilation is challenging, as the data change irregularly depending on time and location. For instance, parameters such as indoor/outdoor differential pressure, wind direction, and wind speed between indoor and outdoor air change rapidly, representing many extreme points in a box plot [14]. Therefore, it is crucial to design natural ventilation prediction models that account for the realistic issue of mismeasurement of sensing data.

In this study, we address the issue of inaccurate sensing data by proposing a method for creating auxiliary data and designing a deep learning model that utilizes the generated auxiliary data. To generate auxiliary data from sensing data, we first employ a DNN (Deep Neural Network) model to generate rapidly changing features and a time series analysis model to generate slowly changing features. Specifically, we generate the auxiliary data by predicting the natural ventilation rate using the observed data and then repredicting each observation datum using the predicted natural ventilation rate. Also, the prediction model is designed by incorporating the key characteristics of the observed data, such as differential pressure, wind direction, and wind speed, which are known to have a relatively significant impact on the natural ventilation rate. Finally, to enhance the reliability of the observed data, we employ a bagging-based ensemble approach for effectively combining the auxiliary data and important data.

This study makes the following contributions:

1. In the context of deep-learning-based natural ventilation prediction using various IoT sensing data, we address the issue of realistic mismeasurement by generating auxiliary data that utilize the rapidly changing or slowly changing characteristics of sensing data, which can improve the reliability of observation data.

2. After constructing three models to apply the characteristics of the reliable features that affect the auxiliary data (i.e., predicted and important features), an ensemble model was designed to improve the generalization performance of the deep learning model for predicting natural ventilation.

In the experimental setup, the data were collected using each IoT sensor for a period of 32 days. To establish a mismeasurement environment for sensor data, a single-error environment was created, and the performance of the predictive model was evaluated using the collected data. Based on the experimental results, we confirm that the proposed method improved the prediction accuracy performance from 0.637 up to 0.868 compared with the deep learning model, using only observation data with a 30% error in the mismeasurement environment. Also, we extracted Shapley Additive Explanations (SHAP) to verify the feature contribution of environmental data, auxiliary data using semisupervised learning, and important data for each scenario model.

2. Background

Predicting the NV (i.e., air quality, airflow, or amount of natural ventilation) by utilizing environmental data such as differential pressure, wind, temperature, and humidity through IoT sensing is an important issue for optimal HVAC control while maintaining comfortable air quality. To predict NV, many studies have been conducted using analytical, experimental, computational fluid dynamics, and machine learning models. In the studies in [2–5], air quality and airflow were calculated using simple equations, such as the mass balance equation for several parameters. The indoor air velocity and ventilation volume were predicted through modeling of empirical analysis using basic equations, such as induced airflow velocity and airflow pattern airflow using pressure coefficient. Refs. [6–9] studied how to satisfy thermal comfort with optimal nature ventilation rate by empirically analyzing the effect of the ventilation openings size in combination with calculating the airflow according to the various building scales. It is important to calculate airflow and natural ventilation to determine the size of ventilation openings and evaluate thermal comfort in the building scales, since the lack of air circulation in a building affects internal thermal comfort and is related to energy consumption. Refs. [10–13] studied how to predict the amount of natural ventilation in indoor airflow based on the CFD model as a traditional method for analyzing complex sensing data such as air velocity by dividing the air layer more precisely. They proposed a way to provide fresh air to the building through the optimal indoor air flow rate according to the effect of outdoor wind speed based on CFD analysis. Recently, research has been conducted to predict the amount of natural ventilation based on machine learning for data convergence analysis using various IoT sensing data that affect indoor and outdoor airflow [14]. In the studies by [14–17], the amount of natural ventilation was predicted using machine learning, and the predictability of reducing energy demand and the efficient ventilation of buildings was investigated. To design the prediction model for the amount of natural ventilation, various machine-learning-based models were presented through the process of analyzing the correlation between indoor and outdoor environmental data to compare prediction performance. However, in machine learning methods, accuracy performance can be degraded if the data collected through IoT sensing is incorrectly measured or omitted depending on the different sensors and environment. Therefore, reliable IoT sensing data are required to accurately predict the amount of natural ventilation in a machine-learning-based analysis model. In this study, considering the uncertain data environment, we propose a way to enhance the reliability of features by generating auxiliary data, such as the regular and irregular features according to environmental variables occurring indoors and outdoors. Finally, the accuracy of natural ventilation prediction is improved by using the ensemble technique with the observed, predicted, and important features. Table 1 shows the summary of related works about natural ventilation prediction. Note that we represent simplistic and complicated methods which use simple equations, such as the mass balance equation, for

several parameters and complex equations, such as calculating the airflow according to the various building scales, respectively.

Table 1. Comparison of Natural Ventilation Prediction Studies.

Year	Simplistic/Complicated	Characteristic of Methods	Major Subject	Consideration of Mismeasurement Sensing Features
2 (2007) 3 (2008) 4 (2013) 5 (2017)	Simplistic	Analytical model through basic equations	Airflow	No
6 (2017) 7 (2018) 8 (2019) 9 (2020)	Complicated	Combining empirical models for small and large scale	Airflow	No
10 (2017) 11 (2019) 12 (2020) 13 (2021)	Complicated	Analysis simulation based on CFD for complex data	Airflow	No
14 (2021) 15 (2021) 16 (2022) 17 (2022)	Complicated	Prediction based on machine learning	Ventilation rate	No
Proposed	**Complicated**	**Prediction considering mismeasurement data**	**Ventilation rate**	**Yes**

3. Materials and Methods

3.1. Problem Definition

In this study, we propose a machine learning method to improve the prediction accuracy of the NVR by considering the mismeasurement issue in IoT sensing data. The overall concept of this study is depicted in Figure 1. In IoT sensing environments, mismeasurement of data can occur due to sensor errors, leading to a decrease in the overall NVR prediction accuracy. To address this, auxiliary data are generated to enhance the reliability of sensing data. By training a machine learning model that combines sensing data and auxiliary data with correlations to NVR, an improved NVR prediction accuracy is obtained, thereby enhancing the reliability of the sensing data.

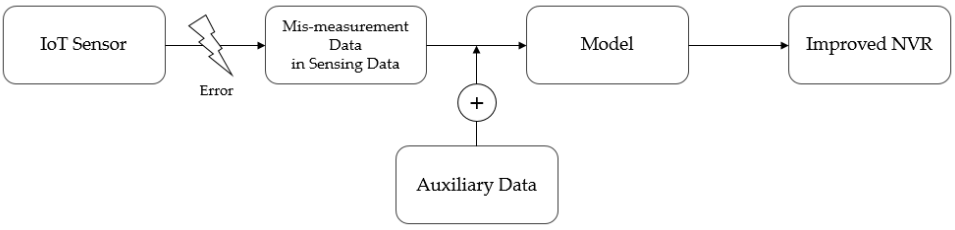

Figure 1. Overall concept of problem definition.

3.2. Experimental Environments

In this study, we predict the NVR using IoT sensing environmental data measured both indoors and outdoors. The dataset was collected from an office located on the first floor of a university research building in Daejeon, Korea from 2:00 p.m. to 6:00 p.m., between 1 October and 19 November 2019. The experimental space was restricted to a 98.72 m^3 office. In addition, there were usually two occupants entering and leaving the space with unrestricted access for practical field measurement of all environmental variables. The experimental space was composed of four façades, and two tilting windows were left open for measurement of sensor values, while the rest of them were kept closed to maintain a constant open area. A window-mounted slit was installed to measure the amount of

natural ventilation, and the indoor and outdoor environment sensors as well as the slit installation locations are depicted in Figure 2. We installed indoor/outdoor environmental sensors approximately 2 m away from the tilted windows, as shown in Figure 2, to ensure that all environmental variables were not affected by each other. Additionally, the height of each sensor was approximately 1.2 m from the ground, because the targeted space was an office. To conduct experimental tests, we used Intel(R) Xeon(R) CPU E5-2620 v4, 64GB RAM, and RTX 3090 GPU. Also, Python 3.8 in Anaconda was used.

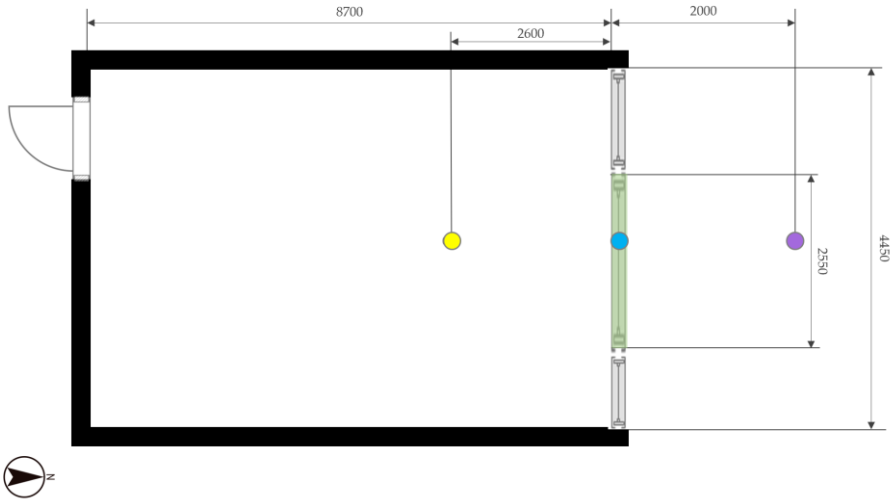

Figure 2. Space floor plan for acquisition environmental data: purple dots and yellow dots represent indoor and outdoor environmental variable measuring sensors, respectively, blue dots represent sensors for measuring natural ventilation, and the green square indicates the location of the chamber installation.

In order to construct an NVR prediction model, we exploit the IoT dataset consisting of indoor and outdoor features such as Sr, Tin, Tout, Td, RHin, RHout, RHd, Pd, Wd, and Ws (i.e., solar radiation, indoor air temperature, outdoor air temperature, difference of in/outdoor air temperature, indoor relative humidity, outdoor relative humidity, difference of in/outdoor relative humidity, pressure difference, wind direction, and wind speed), as shown in Table 2. Furthermore, Pd, Wd, Ws, and Td are well-known factors that have a significant impact on NVR. On the other hand, Sr, Tin, Tout, RHin, RHout, and RHd are relatively less influential factors on NVR. Note that we represent high influence levels, which are well-known variable parameters related to NVR (i.e., Pd, Wd, Ws, and Td). Also, other variable parameters are represented as medium level (i.e., Sr, Tin, Tout, RHin, RHout, and RHd).

To collect the accurate NVR, we exploited two airflow sensors on the open area of the slit installed in the open tilting window for measuring the airflow velocity. That is, two airflow sensors measure the amount of natural ventilation flowing into the room, and then the NVR was calculated by averaging the values of the two sensors to reduce the error of difference of two individually measured sensors, as shown in Equation (1). In Equation (1), the amount of natural ventilation and the airflow velocity are represented by A_c and V_s, respectively, where A_c is each opening area of the chamber, and vs. is each air velocity at opening area. Note that the sampling interval was 1 min, with a total of 241 data collected per day from 2:00 p.m. to 6:00 p.m. For the experimental setup, a total of 7712 sample data were collected in 32 business days.

$$NVR = \frac{\sum_{i=1}^{n}(A_{c_i} \times V_{s_i})}{n} \quad (1)$$

Although Wd and Ws are factors that greatly affect NVR prediction, it is difficult to accurately measure Wd and Ws due to their irregular and rapidly changing characteristics. Therefore, in this study, we quantized Wd and Ws into two and two (i.e., inside and

outside the building) and four (i.e., calm, light air, light breeze, and gentle breeze) levels, respectively, as shown in Tables 3 and 4. In this study, we only focus on deep-learning-based NVR prediction in a mismeasurement environment, and Wd and Ws are treated as IoT features. To accurately prediction model with Wd and Ws, we analyze Wd and Ws, including the use of quantization methods, the advance approaches of which could make prediction using artificial intelligence, such as deep learning and fuzzy logic theory [18] in future work.

Table 2. Indoor-outdoor environmental data.

Type of Variables	Variable Feature	Symbols	Units	Range	Influence
Input (Periodic)	Solar radiation	Sr	Mj/m^2	0~0.015	Medium
	Indoor air temperature	Tin	°C	15.5~27.7	Medium
	Outdoor air temperature	Tout	°C	1.7~28.1	Medium
	Difference of in/outdoor air temperature	Td	°C	−2.1~16.4	High
	Indoor relative humidity	RHin	%	16~71	Medium
	Outdoor relative humidity	RHout	%	26~93	Medium
	Difference of in/outdoor relative humidity	RHd	%	−53~12	Medium
	Pressure difference	Pd	mbar	0~0.36	High
Input (Nonperiodic)	Wind direction	Wd	True-north-based azimuth divided in 16 angles	0~359	High
	Wind speed	Ws	m/s	0~5.31	High
Target	Natural ventilation rate	NVR	m^3/m	0~3.96	-

Table 3. Quantization for wind direction (Wd).

	Grade	Degree	Building Direction
Wind Direction	0	271°~359° 0°~89°	Inside building
	1	90°~270°	Outside building

Table 4. Quantization for wind speed (Ws).

	Grade	m/s	Kind of Wind
Wind Speed	0	0~0.2	Calm
	1	0.3~1.5	Light air
	2	1.6~3.3	Light breeze
	3	3.4~5.4	Gentle breeze

3.3. Overview of the Proposed Method

In this study, we propose a way to apply a reliable deep learning model for predicting NVR accurately by using auxiliary data in consideration of the mismeasurement of the IoT sensing environment. Figure 3 shows the overall application of the deep learning model. The combined features of observation and auxiliary (i.e., generation and importance features) are used for reliable training with complementary characteristics, and then a bagging-based ensemble model to enhance the prediction model of NVR in the mismeasurement data is designed.

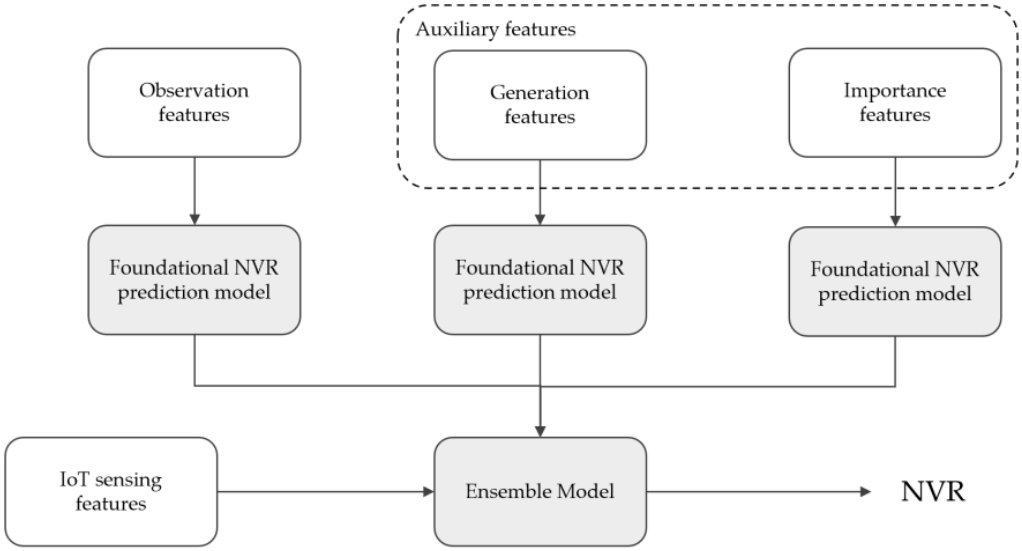

Figure 3. Overview of the proposed method ensemble prediction of NVR.

3.4. Foundational NVR Prediction Model Based on Deep Learning

In this study, the foundational NVR prediction model is designed based on a DNN (Deep Neural Network) with four hidden layers. Note that refs. [19–22] have reported that artificial neural networks with two or more hidden layers are commonly referred to as "deep", and using two to five hidden layers can provide higher prediction accuracy than using a large number of hidden layers, if the number of input features is not large. Furthermore, since the number of features in the input layer is ten for NVR prediction, the deeper hidden layers are used, and cost-increasing and overfitting problems may occur. Therefore, based on the pre-experimental test with grid search to find the best set of hyperparameters, we designed the DNN model for foundational NVR prediction using four hidden layers. To avoid an overfitting problem, a deep learning model using a small number of hidden layers may be designed in insufficient knowledge capacity, and thus dropout methods are applied to design a deep learning model maintaining sufficient knowledge capacity. To find the optimal number of drop layers, the pre-experiment was conducted by applying one to three dropout layers, and we confirmed that using three dropout layers can provide the best prediction performance. In addition, for optimization of model hyperparameters, grid search was empirically conducted to determine the parameters that improve model performance. Therefore, in this study, as shown in Figure 4, we designed the DNN model with four hidden layers and three dropout layers. Each of the four hidden layers is set to 200, 100, 100, and 50 nodes, respectively, considering the relatively small number of input features. Additionally, each hidden layer contains 2200, 20,100, 10,100, and 5050 parameters, respectively. The output layer, comprising a single node, is composed of 51 parameters. Consequently, the entire model is comprises 37,501 parameters. Finally, we deduced the optimized model for obtaining maximized model performances among all possibilities from grid search. In addition, all the dropout rates were set to 0.2, respectively. The ReLU activation function is used to prevent the gradient vanishing problem, and early stopping is applied to finish training at an appropriate time in 20 to 200 epochs. Finally, zero-mean normalization is applied to reduce the difference in scales between each IoT sensing feature.

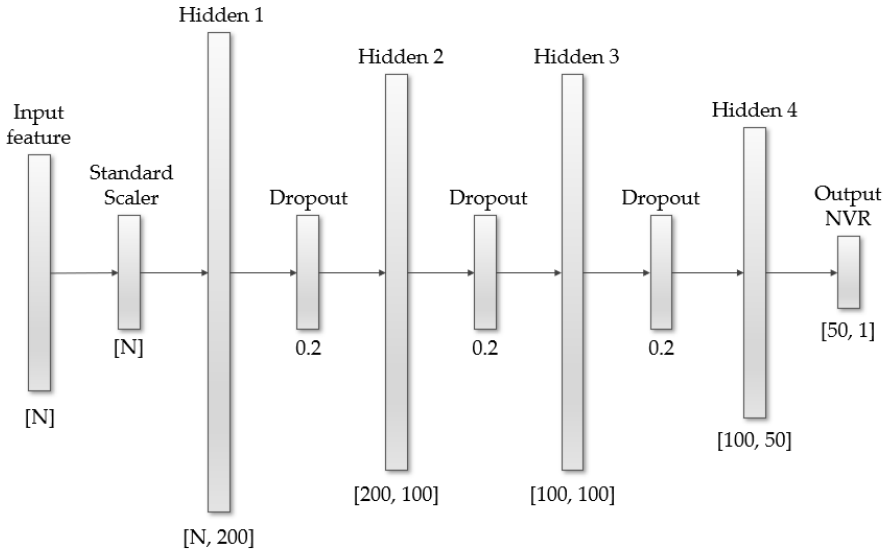

Figure 4. Foundational NVR prediction model.

3.5. ReLU (Rectified Linear Unit)

ReLU is the most used activation function in deep learning, primarily for addressing the gradient vanishing problem. ReLU is well known to provide good performance for training deep learning [23]. In this study, for predicting NVR, the ReLU activation function is applied to each hidden layer using Equation (2). ReLU returns the input value as the output, if the input value is greater than zero. Otherwise, ReLU returns zero as the output if the input value is less than or equal to zero. These characteristics of ReLU help mitigate the vanishing gradient problem, which can occur during backpropagation and hinder the training process.

$$f(x) = max(0,\ x) \tag{2}$$

3.6. Standard Normalization

The standard normalization normalizes the mean and variance of input features to 0 and 1, respectively [24]. For transformation to x'_i with reduced scale difference of each input feature by considering the distribution and scale difference of each IoT sensing data feature, the standard normalization is applied by dividing the difference between the input feature xi and the mean x_{mean} of the input features by the standard deviation x_{std}, as shown in Equation (3).

$$x_i^{'} = \frac{x_i - x_{mean}}{x_{std}} \tag{3}$$

Note that x_{std} is calculated using Equation (4) for the number N of samples.

$$x_{std} = \left[\frac{1}{N-1}\sum_{I=1}^{N}(X_i - X_{mean})^2\right]^{\frac{1}{2}} \tag{4}$$

3.7. Auxiliary Data Generation

3.7.1. Periodic and Nonperiodic Sensing Data

Training the input feature data with auxiliary feature data can help improve the NVR prediction accuracy in uncertain IoT sensing environments. In this study, we exploit two types of auxiliary feature data considering the generation and importance of IoT sensing data. To generate the auxiliary feature data, periodic and nonperiodic sensing data are leveraged using a DNN and LSTM, respectively. We believe that LSTM has the advantage of good generation accuracy in slightly changing environments. However, to consider rapidly changing environments, we apply the DNN method instead of LSTM. LSTM is used based on previous sequential data, while a DNN is used based on other features

within a same train sample. The choice of the two generation methods for the ten input data points of the sensing data depends on the presence or absence of periodicity in each feature. According to the properties of each environmental dataset, we, respectively, leveraged the generative models consisting of a DNN and LSTM for building auxiliary data with periodicity and nonperiodicity. These two generation methods are used to enhance the accuracy of the auxiliary data, and the generated auxiliary data help compensate for the uncertainty of the sensing data. Among the ten input features, the data with periodicity are Sr, Tin, Tout, RHin, RHout, RHd, and Td, while the data without periodicity are Pd, Wd, and Ws.

Figure 5 shows the daily and hourly changes in Sr, Tin, Tout, and Td. Figure 5 separately depicts the daily changes in the sensing data on the left and the hourly changes on the right. The left graph of Figure 5 shows the changes in the sensing data according to the day over six days (10/1, 10/2, 10/3, 10/4, 10/5, and 10/7), while the right graph of Figure 5 shows the changes in the sensing data according to the time of day (from 14:00 to 18:00). Among all the environmental variables, Sr, Tin, Tout, and Td showed gradual changes in the measured dataset according to both daily and hourly trends. Namely, they tended to have similar periodicity day by day. In addition, Sr and Tout decreased as time increased, but Td increased. In the case of Tin, it showed mostly stable and lower tendencies without rapid changes.

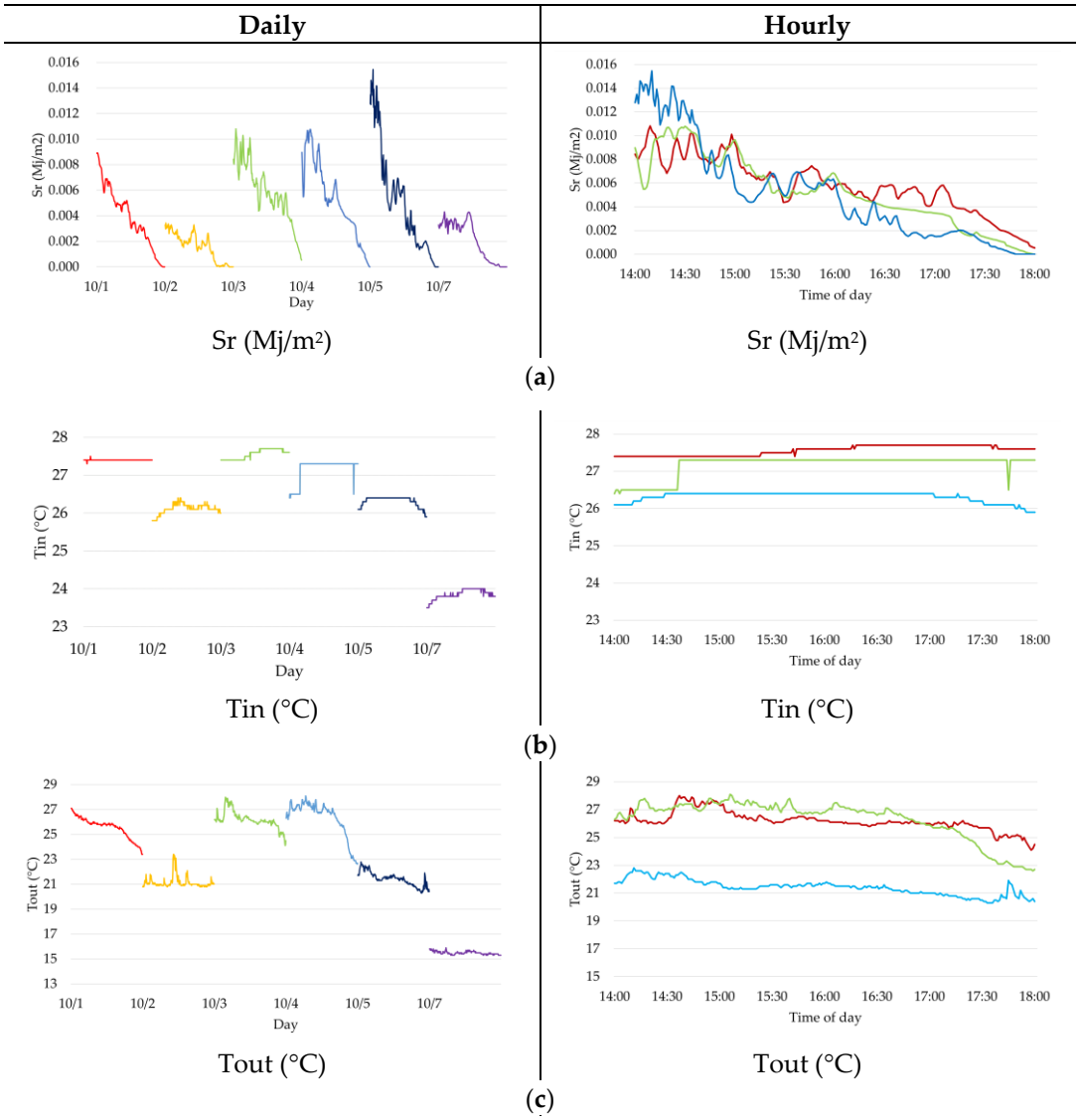

Figure 5. *Cont.*

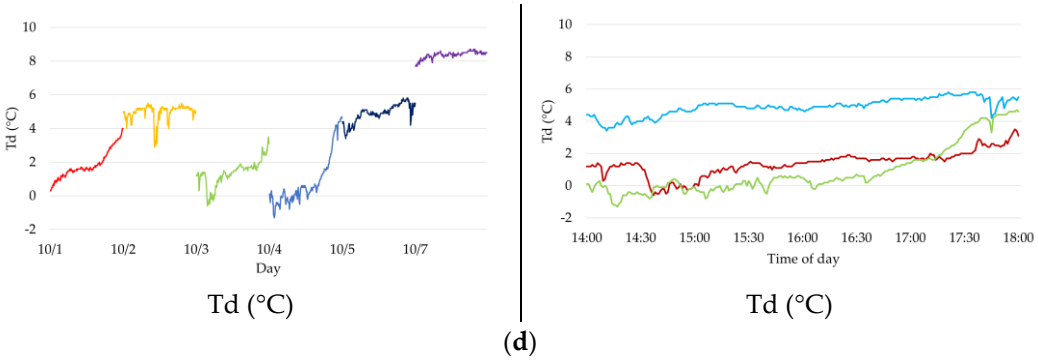

Figure 5. Daily and hourly periodicity of (**a**) Sr, (**b**) Tin, (**c**) Tout, and (**d**) Td (each date is represented in a different color).

Figure 6 shows the periodicity of RHin, RHout, and RHd. The left side of Figure 6 shows the daily changes in the sensing data, and the right side of Figure 6 shows the gradual changes in the sensing data over the course of a day. Relative humidity-related variables also had similar tendencies as the periodicity of Tin. They did not change much, showing gradual changes without high increases or decreases. Thus, LSTM was finally utilized by generating auxiliary data with periodicity, since Sr, Tin, Tout, Td, Rhin, RHout, and RHd were classified into the features having the periodicity and tendencies of the sequential dataset.

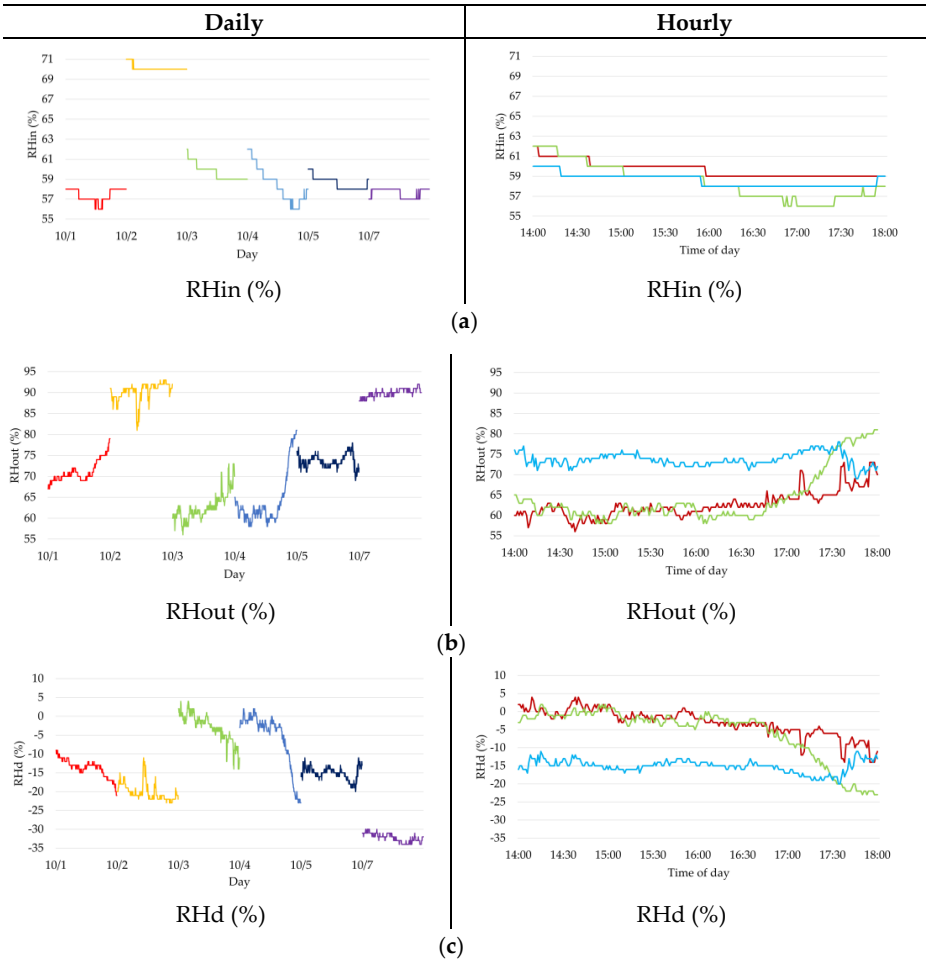

Figure 6. Daily and hourly periodicity of (**a**) RHin, (**b**) RHout, and (**c**) RHd (each date is represented in a different color).

On the other hand, Figure 7 shows the Pd, Wd, and Ws daily and hourly changes according to time, demonstrating a pattern without periodicity. The daily changes in Pd, Wd, and Ws in Figure 7 fluctuate rapidly regardless of time. Furthermore, the hourly changes in the sensing data throughout the day show nonperiodicity for the same time periods changing drastically. Thus, the DNN was used for generation of auxiliary data without periodicity, because Pd, Wd, and Ws referred to the nonperiodic dataset without lower changes in periodicity and tendencies. Note that in this study, Wd and Ws change dynamically, as shown in Figure 7b,c. Wd and Ws are quantized into two and four (i.e., calm, light air, light breeze, and gentle breeze) levels.

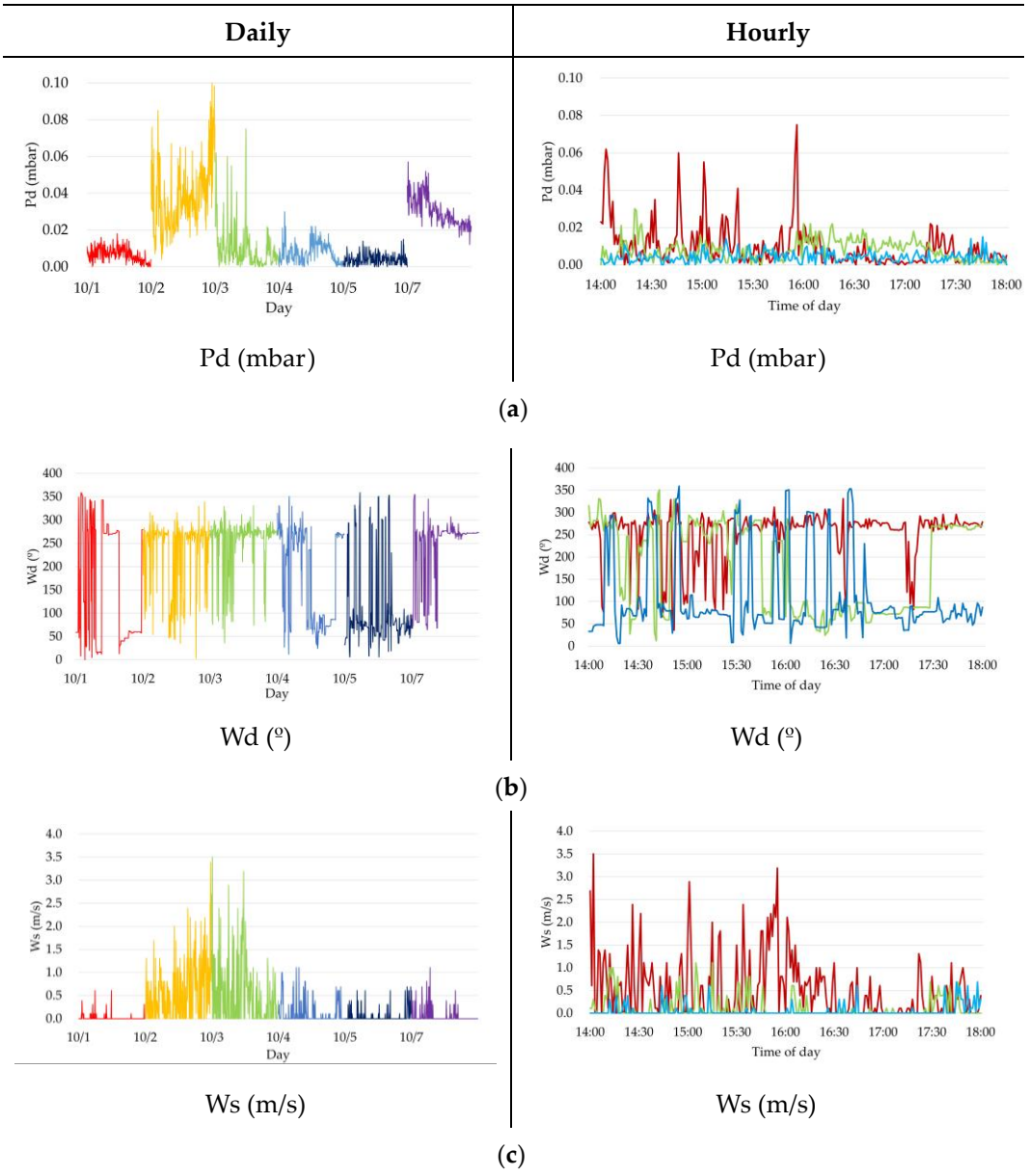

Figure 7. Daily and hourly periodicity of (**a**) Pd, (**b**) Wd, and (**c**) Ws (each date is represented in a different color).

3.7.2. Generating Auxiliary Data Based on SSL

- In this study, we propose a method for generating auxiliary data based on SSL (semisupervised learning) to address the issue of mismeasurement in sensing data. SSL is a type of ML (machine learning) technology that aims to overcome the limitations of both

supervised and unsupervised learning. Supervised learning requires a large amount of training data to classify test data, while unsupervised learning does not require labeled data but struggles to accurately cluster data. To overcome these challenges, SSL learns and labels data with a small amount of training data [25]. By iteratively learning and adding predicted features from a deep learning model to the training data, SSL achieves improved accuracy compared with general ML approaches [26]. Generating auxiliary data based on SSL helps prevent the deterioration of reliability in mismeasurement data.

- Figure 8 shows the process of generating auxiliary data based on SSL. To generate SSL-based auxiliary data, a model trained on observation features, which represent all the features of environmental data, predicts a specific feature, called Prediction F_i. Subsequently, the predicted feature, Prediction F_i, is added to the training data to generate more refined auxiliary data. To ensure accurate generation of auxiliary data, the method distinguishes between data that changes rapidly and gradually over time. It applies specific auxiliary data generation techniques for each type of data. Data with gradual changes and identifiable patterns are used to generate prediction regular features through LSTM-based prediction, while rapidly changing data generate prediction irregular features through prediction using the FC-DNN model. By combining these two types of auxiliary data, we ultimately create generated auxiliary features that complement the mismeasurement data.

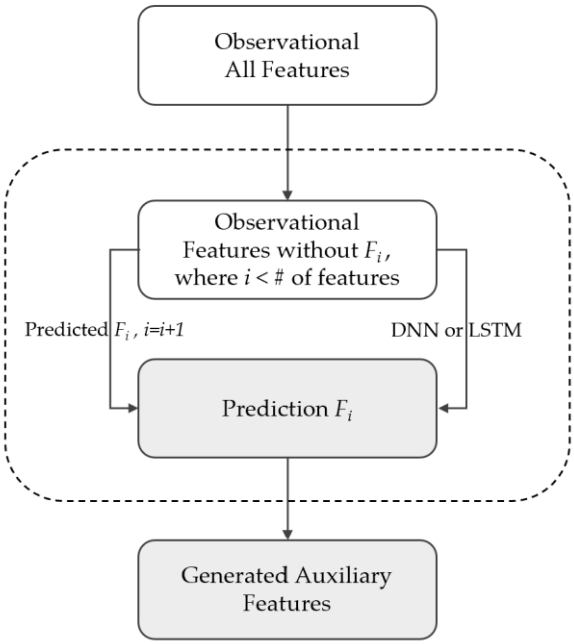

Figure 8. Auxiliary data generation methodology based on SSL.

3.7.3. Auxiliary Data with Important Features

- Along with the generation of sensing data, this study selects important features after analyzing the correlation between the NVR and environmental variables. We finally use them as auxiliary data. Considering the correlation between target data and input data in the prediction model is one of the ways to improve the model's predictive performance [26]. Therefore, we improve the prediction accuracy by using the main features that have a relatively greater impact on NVR through prior research results and correlation analysis as auxiliary data. In prior research on NVR prediction, it has been reported that the factors affecting NVR are four elements near the building (Pd, Wd, Ws, and Td) [27]. We use the heatmap method to compare the correlation of the prior research results with the sensing data and analyze the impact between each feature, as shown in Figure 9 in the heatmap. The relationship between the

four important factors highlighted in the previous research and NVR are all positive correlations, appearing as 0.84, 0.13, 0.36, and 0.38, respectively. Even excluding these four factors, Tout and Tin are high correlations because they are related to Td. Based on the results of previous research, we reidentified how much wind-related variables and pressure differences could be affected to variations in NVR based on correlation analysis. As shown in the results, in/outdoor temperature differences were significantly influential to changes in NVR, because they are highly related to making pressure differences changeable. Therefore, we select Td as the important feature and exclude Tin and Tout. Also, the environmental data assumptions vary depending on the building's measurement conditions (e.g., direction, building scale, and location). In this study, considering the differences in measurement conditions, we select four factors as the importance features based on the results of previous research and use them as auxiliary data.

	Pd	Tin	RHin	Tout	RHout	Ws	Wd	SR	Td	RHd	NVR
Pd	1.000000	-0.332209	-0.391909	-0.360095	-0.316617	0.371923	0.009786	-0.128548	0.329794	-0.054506	0.848179
Tin	-0.332209	1.000000	0.716484	0.898999	0.309282	-0.041381	-0.025039	0.270567	-0.685121	0.461324	-0.378411
RHin	-0.391909	0.716484	1.000000	0.718428	0.699613	-0.190659	0.016545	0.323527	-0.614474	0.284350	-0.352625
Tout	-0.360095	0.898999	0.718428	1.000000	0.128573	-0.029686	-0.080041	0.343475	-0.934939	0.706170	-0.417259
RHout	-0.316617	0.309282	0.699613	0.128573	1.000000	-0.280230	0.074918	0.059447	0.036716	-0.486092	-0.253711
Ws	0.371923	-0.041381	-0.190659	-0.029686	-0.280230	1.000000	0.086727	-0.046028	0.015851	0.142814	0.369632
Wd	0.009786	-0.025039	0.016545	-0.080041	0.074918	0.086727	1.000000	-0.066233	0.112844	-0.080287	0.132437
SR	-0.128548	0.270567	0.323527	0.343475	0.059447	-0.046028	-0.066233	1.000000	-0.352089	0.315931	-0.180277
Td	0.329794	-0.685121	-0.614474	-0.934939	0.036716	0.015851	0.112844	-0.352089	1.000000	-0.800807	0.387442
RHd	-0.054506	0.461324	0.284350	0.706170	-0.486092	0.142814	-0.080287	0.315931	-0.800807	1.000000	-0.090863
NVR	0.848179	-0.378411	-0.352625	-0.417259	-0.253711	0.369632	0.132437	-0.180277	0.387442	-0.090863	1.000000

Figure 9. Heatmap between features of environmental data (the darker the color, the higher the value).

3.8. NVR Prediction Model Scenarios

In this study, we designed an ensemble model based on the bagging method using data generated by semisupervised learning and selected key feature data. Bagging is a method of creating a strong classifier by combining multiple weak classifiers, generating multiple bootstrap samples, modeling them, and then combining them to improve the accuracy of the final prediction model [28]. In this study, we created three sub-DNN models and improve prediction accuracy through the bagging ensemble model. Table 5 shows the learning feature data for each scenario, with three submodels and an ensemble model configured for each scenario. In Table 5, S1 uses all the features of the observed data for learning. S2 uses the same number of input dimensions as S1 but learns using auxiliary data that recreate the features of the observed data. S3 learns four important features composed through the selection process, and S4 learns using all features of S1's input features and the auxiliary data of S2 and S3.

Table 5. Features by Learning Scenario.

Scenario	Title	Features
S1	Observed features	X
S2	Auxiliary features	$PRED_X$
S3	Important features	Pd, Wd, Ws, and Td
S4	Ensemble of S1, S2, and S3	$S1_{features}$, $S2_{features}$, $S3_{features}$
X: Sr, Tin, Tout, Td, RHin, RHout, RHd, Pd, Wd, and Ws		

3.9. Proposal Algorithm

In this study, we propose a method to obtain an improved NVR prediction accuracy using auxiliary data and a deep learning model, even in environments where mismeasurements can occur. Algorithm 1 represents the proposed method as an algorithm, obtaining the improved accuracy $PRED_E$ through the input of sensing data X and the proposed method. During the generation of auxiliary data, in Step 1, the predicted value $PRED_{S1}$ of DNN_{S1}, which is trained with the sensing data $Train_X$, is generated for use in the learning process. In Step 2, depending on the periodicity of the auxiliary data to be generated, either DNN or LSTM is selected as the model, and train data are formed by excluding one of the features of the auxiliary data, X_i, and using X and $PRED_{S1}$. The process is repeated C times, resulting in 10 auxiliary datasets, $PRED_X$. Additionally, in Step 3, $Important_X$ is generated based on its importance to the NVR. The generated auxiliary data are split into $Train_{PREDx}$ and $Train_{Importantx}$ using the train_test_split function, and two DNN models, DNN_{S2} and DNN_{S3}, are fitted with each dataset. Finally, In Step 4, an ensemble model, Enet, is created to fuse the three scenario models, DNN_{S1}, DNN_{S2}, and DNN_{S3}, and the datasets $Train_X$, $Train_{PREDx}$, and $Train_{Importantx}$ are set accordingly. Subsequently, outlier data are inserted into the test dataset to create $Test_X'$, $Test_{PREDx}'$, and $Test_{Importantx}'$, and the predicted results $PRED_E$ are obtained using Enet. Through this process, we enhance the reliability of sensing data.

Algorithm 1. Proposed NVR Prediction Methods with Auxiliary Data

Input:
Observation Train Data: $Train_X$
Observation Test Data: $Test_X$
Observation Data Columns: C
Output:
Natural Ventilation Rate: NVR

Step 1	DNN_{S1}.fit($Train_X$)
	$PRED_{S1}$ = DNN_{S1}.predict(X)
Step 2	For i in C:
	If X_i ! = Periodicity:
	$Model_{Xi}$ = DNN
	else:
	$Model_{Xi}$ = LSTM
	$Model_{Xi}$.fit($Train_X$ - $Train_{Xi}$ + $PRED_{S1}$)
	$PREDx_i$ = $Modelx_i$.predict($Test_X$ − $Test_{Xi}$)
	$PRED_X$ + = $PREDx_i$
	$Important_X$ = X[[Important]]
Step 3	$Train_{PREDx}$, $Test_{PREDx}$ = train_test_split($PRED_X$)
	$Train_{Importantx}$, $Test_{Importantx}$ = train_test_split($Important_X$)
	DNN_{S2}.fit($Train_{PREDx}$)
	DNN_{S3}.fit($Train_{Importantx}$)
Step 4	Enet = Ensemble_create(DNN_{S1}, DNN_{S2}, DNN_{S3})
	Enet.fit($Train_X$, $Train_{PREDx}$, $Train_{Importantx}$)
	$Test_X'$, $Test_{PREDx}'$, $Test_{Importantx}'$ = insert_outlier($Test_X$, $Test_{PREDx}$, $Test_{Importantx}$)
	$PRED_E$ = Enet.predict($Test_X'$, $Test_{PREDx}'$, $Test_{Importantx}'$)
	return $PRED_E$

4. Experimental Results

4.1. Evaluation Metrics

In this study, R^2score, MMscore, and ACCscore were used to evaluate metrics. The reason for evaluating and comparing the model's performance using three evaluation metrics is to consider the differences in interpretation and disadvantages of each metric. The R^2score and MMscore assess the model's performance by quantifying the statistical disparity between the predicted values and the corresponding measurements within the

prediction model. Conversely, the ACCscore evaluates the performance by determining the proportion of accurately classified instances within the classification model.

R^2score (R-squared score)

The R-squared score is a statistical method that measures how to well explain a model and predict the outcome of a given dataset [29,30]. The formula for the R^2score calculation is as follows:

$$R^2score = 1 - \frac{\sum_{i=1}^{n}(t_i - \hat{p}_i)^2}{\sum_{i=1}^{n}(t_i - \overline{p_i})^2} \tag{5}$$

The value of R^2 ranges from 0 to 1. The closer it is to 1, the better the model performance. Here, t_i stands for the target value, $\hat{p}_l$ stands for the predicted value, and $\overline{p_l}$ stands for the mean of all target values.

MMscore (Mean MAE score)

MMscore is an evaluation metric created in this study to confirm the difference between prediction and measurement of the model. It uses *MAE* (Mean Absolute Error) and *Mean* to measure performance. The expression of *MMscore* is as follows:

$$MMscore = \frac{(Mean - MAE)}{Mean} \tag{6}$$

In the case of *Mean*, it means mean for Y data among test data, and *MAE* means the average absolute error between prediction and actual measurement. In Equation (6), when the *MAE* value approaches 0, the *MMscore* reaches its maximum value of 1. Conversely, if the *MAE* tends towards infinity, the *MMscore* takes on a negative infinity value. When the *MAE* is equal to the *Mean*, the *MMscore* becomes 0, which is the same as the model predicting all values to 0. The calculation of Mean-MAE is performed using Equation (7).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right| \tag{7}$$

where $Y_{\hat{i}}$ expresses the predicted value, Y_i is the observed value, and n is the number of samples. The *MAE* scores are linearly increasing with the increase in errors [31,32].

ACCscore (Accuracy score)

To evaluate the accuracy of the numerical data, we applied an allowable error range based on the distribution of each datum. Equation (8) was used to apply the allowable error range for each datum.

$$True = \ |y - prediction| < \frac{1}{n}\sum_{i=1}^{n} y_i \times 0.2 \tag{8}$$

$$Accuracy = \frac{1}{n}\sum_{i=1}^{n} True_i \tag{9}$$

In Equation (8), if the absolute difference between the actual value y and the predicted value falls within 20% of the average value of y, it is classified as *True*. The *accuracy* is measured using Equation (9), which calculates the rate of the number of *True* values to the total number of y.

4.2. Outlier Dataset

In this study, we considered sensor mismeasurement situations during the data collection process and evaluated the model's accuracy by arbitrarily generating outliers in the test data. For data with periodicity, it changes gradually over time. However, there may be cases where abrupt outlier data occur due to sensor mismeasurement, deviating from the periodic pattern. Even for data without periodicity, outliers can occur from unseen patterns. Therefore, in this study, we defined data with different relationships with various

observation features due to mismeasurement as outlier data, and we created a portion of the test data as outlier data to assume outlier occurrence situations.

To generate outlier data, we randomly divided the data into training and test data, then within the test data, we generated outliers at some proportions to regenerate as the test set. Note that to generate the value and position within the test data, the random function provided in Python 3.8 was used. Figure 10 shows the datasets that include outlier data at each of the 0%, 10%, 20%, and 30% proportions. Once outlier data are generated, they are replaced within the sample data. For example, if we divide the training/test rate into 60%/40%, the test data are separated into 3,084 observations out of the total 7712 data. Note that the dataset with 30% outliers means that 925 outliers are included in the 3084 test samples, which is 30% of the rate. Also, 0% means that no outlier data were included.

	Sr	T in	T out	Td	RH in	RH out	RHd	Pd	Wd	Ws
1	0.003	27.4	25.8	1.6	56	70	−14	0.012	0	0
2	0.006	21.5	15.2	6.3	44	65	−21	0.03	1	0
3	0.002	16.1	5	17	33	11.1	−16	0.12	1	0
4	0.006	27.4	26.4	1	58	69	−11	0.014	0	0
5	0.002	19.5	15	4.5	23	33	−10	0.02	0	0
6	0.006	21.6	15.4	6.2	44	64	−20	0.06	1	1
7	0.008	27.4	27.1	0.3	58	67	−9	0.011	1	0
8	0	21	12.4	8.6	45	67	−22	0	0	0
9	0.002	24.2	19	5.2	33	31	2	0.032	0	1
10	0.006	22	18	4	32	41	−9	0.012	1	0

(**a**)

	Sr	T in	T out	Td	RH in	RH out	RHd	Pd	Wd	Ws
1	0.003	27.4	25.8	1.6	56	70	−14	0.012	0	0
2	0.006	21.5	15.2	6.3	44	65	−21	0.009	1	0
3	0.002	16.1	5	17	33	11.1	−16	0.12	1	0
4	0.006	27.4	26.4	1	58	69	−11	0.014	0	0
5	0.002	19.5	15	4.5	23	33	−10	0.02	0	0
6	0.006	21.6	15.4	6.2	44	64	−20	0.06	1	1
7	0.008	27.4	27.1	0.3	58	67	−9	0.011	1	0
8	0	21	12.4	8.6	45	67	−22	0	0	0
9	0.002	24.2	19	5.2	33	31	2	0.032	0	1
10	0.006	22	18	4	32	41	−9	0.012	1	0

(**b**)

	Sr	T in	T out	Td	RH in	RH out	RHd	Pd	Wd	Ws
1	0.003	27.4	25.8	1.6	56	70	−14	0.012	0	0
2	0.006	21.5	15.2	6.3	44	65	−21	0.009	1	0
3	0.002	16.1	5	17	33	11.1	−16	0.12	1	0
4	0.006	27.4	26.4	1	58	69	−11	0.014	0	0
5	0.002	19.5	15	4.5	23	33	−10	0.02	0	0
6	0.006	21.6	15.4	6.2	44	64	−20	0.06	1	1
7	0.008	27.4	27.1	0.3	58	67	−9	0.011	1	0
8	0	21	12.4	8.6	45	67	−22	0	0	0
9	0.002	24.2	19	5.2	33	31	2	0.032	0	1
10	0.006	22	18	4	32	41	−9	0.012	1	1

(**c**)

	Sr	T in	T out	Td	RH in	RH out	RHd	Pd	Wd	Ws
1	0.008	27.4	25.8	1.6	56	70	−14	0.012	0	0
2	0.006	21.5	15.2	6.3	44	65	−21	0.009	1	0
3	0.002	16.1	5	17	33	11.1	−16	0.12	1	0
4	0.006	27.4	26.4	1	58	69	−11	0.014	0	0
5	0.002	19.5	15	4.5	23	33	−10	0.02	0	0
6	0.006	21.6	15.4	6.2	44	64	−20	0.06	1	1
7	0.008	27.4	27.1	0.3	58	67	−9	0.011	1	0
8	0	21	12.4	8.6	45	67	−22	0	0	0
9	0.002	24.2	19	5.2	33	31	2	0.032	0	1
10	0.006	22	18	4	32	41	−9	0.012	1	1

(**d**)

Figure 10. An example of ten samples for the observation dataset with outlier data by proportions of outlier data (red box is mismeasurement data): (**a**) includes 0% outlier data, (**b**) includes 10% outlier data, (**c**) includes 20% outlier data, and (**d**) includes 30% outlier data.

4.3. Evaluation Metrics of Generated Auxiliary Data

The proposed method generated auxiliary data by learning the NVR, which was predicted by training the measured sensing data. Therefore, the accuracy of the NVR prediction tends to depend on the generation of auxiliary data. To validate the accuracy of the auxiliary data, we used three evaluation metrics (i.e., R^2score, MMscore, and ACCscore). The R^2score incorporates the notion of prediction and measurement discrepancies, while the MMscore utilizes the concept of errors. These two evaluation metrics are primarily employed for regression problems. However, the binary classifications Wd and Ws were excluded from consideration due to the challenges associated with their measurement using R^2score and MMscore.

Table 6 shows the evaluation metrics of the generated auxiliary data for each environmental feature data. Based on the ACCscore, the accuracy of Td, RHin, Pd, and RHout was found to be 0.978, 0.945, 0.917, and 0.912, respectively. Regarding the R^2score and

MMscore, most of the characteristics, except for Wd and Ws, which are challenging to evaluate, exhibited results of 0.9 or higher.

Table 6. Evaluation metrics of the generated auxiliary data for each environmental feature datum.

	Sr	Tin	Tout	Td	RHin	RHout	RHd	Pd	Wd	Ws
R^2score	0.933	0.996	0.992	0.986	0.995	0.981	0.974	0.938	-	-
MMscore	0.916	0.997	0.987	0.964	0.992	0.981	0.937	0.806	-	-
ACCscore	0.891	0.871	0.814	0.978	0.945	0.912	0.728	0.917	0.626	0.615

4.4. Analyzing NVR Prediction Accuracy

NVR prediction model learning was conducted in the following way: From the entire dataset, five training sets were randomly selected at rates of 0.2, 0.3, 0.4, 0.5, and 0.6, and the model was trained. The remaining datasets (rates of 0.8, 0.7, 0.6, 0.5, and 0.4) were used as test data to measure the NVR prediction accuracy relative to the learning data rate. Figure 11 shows each NVR prediction accuracy with three evaluation metrics of the foundation model (Scenario 1) according to the learning rate. It was observed that as the rate of learning data increased, the NVR prediction accuracy also increased across all three evaluation metrics. However, when the three evaluation metrics were used, there was a difference in the results, which can be interpreted as follows: When a 0.6 rate of learning data were trained, the evaluation metrics R^2score, MMscore, and ACCscore of Scenario 1 were 0.978, 0.845, and 0.745, respectively. In the case of ACCscore, as shown in Figure 11, it exhibited an accuracy trend similar to R^2 as the rate of the train set increased. The three accuracy curves for NVR prediction exhibit similar patterns based on the learning rate. However, the ACCscore consistently appears to be approximately 10% lower than both the MMscore and R^2score. Nevertheless, it is important to note that this discrepancy arises due to the contrasting calculation methods employed by ACCscore (which assesses the rate of correct answers) and R^2score and MMscore (which measure the disparity between predictions and actual measurements). Thus, the interpretation of results through all three performance indicators remains equally plausible.

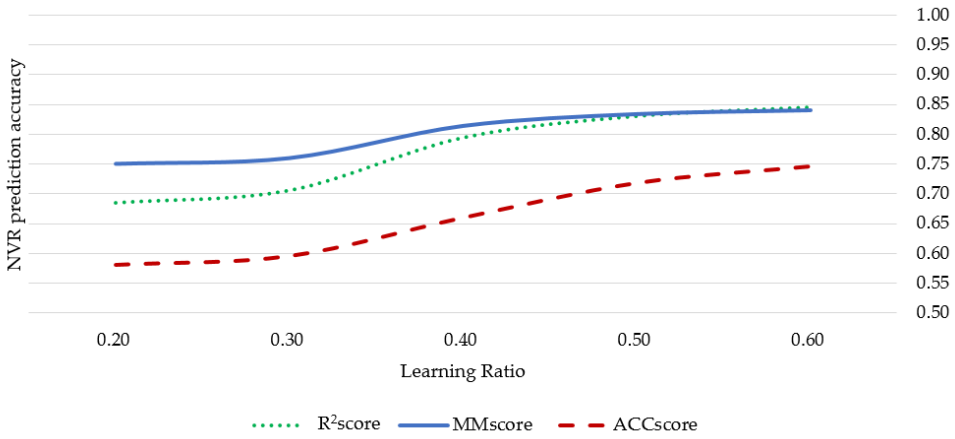

Figure 11. Foundational NVR prediction model accuracy based on learning rate using evaluation metrics (R^2score, MMscore, and ACCscore).

In this study, to consider the occurrence of outlier data due to mismeasurement, we evaluated the accuracy of the model by introducing randomly generated outlier data in datasets with proportions of 0.0, 0.1, 0.2, and 0.3 in the test data (see Figure 12). Figure 12 shows the NVR prediction accuracy for various training and mismeasurement rates with different scenarios. The *x*-axis represents the scenarios, while the *y*-axis represents the accuracy. Figure 12a–d shows the ACCscore for each scenario of training data rates of 0.2, 0.4, 0.5, and 0.6 at mismeasurement rates of 0, 0.1, 0.2, and 0.3, respectively. Overall,

as the training rate increases, the NVR prediction accuracy improves across all scenarios. Furthermore, we observed a decrease in NVR prediction accuracy with an increase in the outlier data rate. Specifically, for S2, which only trained on generated auxiliary data, a lower accuracy was observed compared with S1, which trained on actual observed data. Additionally, for S3, which had a smaller input dimension due to using only four main features, a lower accuracy was observed compared with S1. On the other hand, for S4, the proposed method, which utilized the knowledge of observed data from S1, auxiliary data from S2, and main data from S3 through a powerful ensemble model with strong generalization performance, an improvement of approximately 0.2 in accuracy was observed across all graphs, depending on the outlier data rate and training rate. When train 0.2 and outlier 0 were considered, S4 showed an improvement of approximately 0.28 compared with S1, and for outlier 0.3, an improvement of approximately 0.29 in accuracy was observed. Similarly, when train 0.6 and outlier 0 were considered, S4 showed an improvement of approximately 0.18 compared with S1, and for outlier 0.3, an improvement of approximately 0.23 in accuracy was observed. Thus, through the proposed approach, S4, which involves training on observed data, achieved a maximum improvement of 0.29 (outlier 0.3, train 0.2) compared with S1, and an improvement of approximately 0.2 in accuracy was observed across all graphs in Figure 12. Therefore, we confirmed that the proposed method can provide an accurate NVR prediction performance without degradation in accuracy in mismeasurement sensing environments.

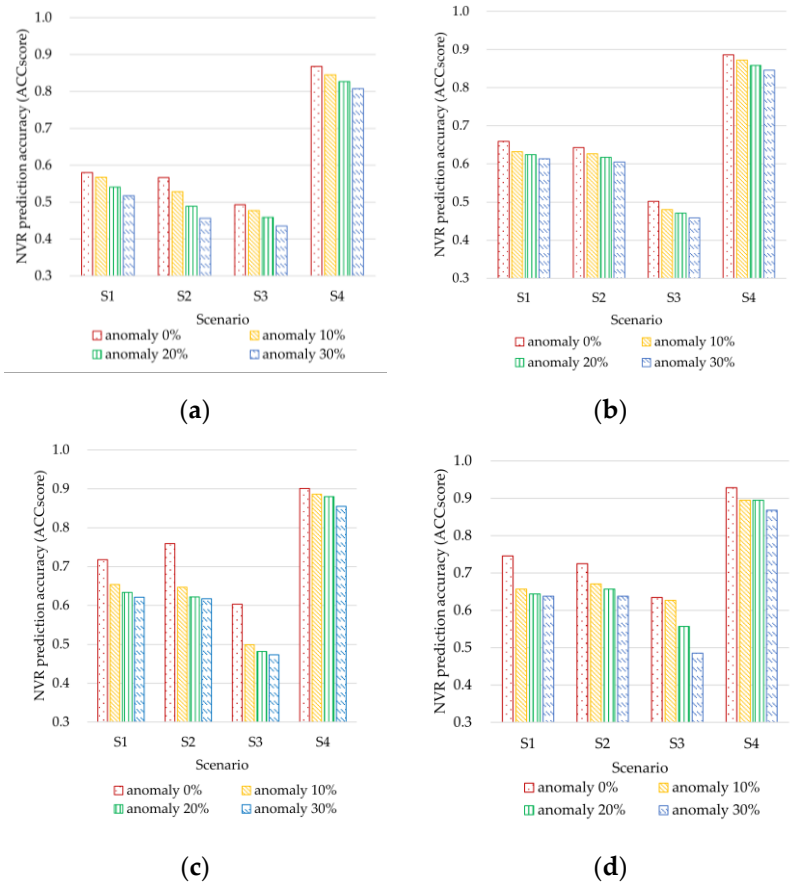

Figure 12. NVR prediction accuracy according to training and the outlier rates: (**a**) train 0.2, (**b**) train 0.4, (**c**) train 0.5, and (**d**) train 0.6.

Table 7 shows the NVR prediction accuracy according to the training and outlier rates of S1 and S4 as three evaluation metrics (R^2score, MMscore, and ACCscore). When train 0.6, outlier 0, the R^2score, MMscore, and ACCscore of S1 were 0.845, 0.840, and 0.745, and for S4, they were 0.946, 0.920, and 0.928. The proposed method, S4, improved by an average

accuracy of 0.12 over S1 in three evaluation metrics. As the training rate increases, the degree of improvement of S4 compared with S1 gradually decreases, which means that as the training rate decreases, the degree of accuracy improvement through the proposed method increases. Given that the ensemble model S4 has acquired knowledge from all three scenarios, there exists the potential to enhance prediction accuracy by enhancing the reliability of the NVR observational data, even in cases where the training rate is low. Consequently, the proposed method yielded an improvement in ACCscore of up to 0.287 (S4 and S1, train rate 0.2, and outlier rate 0.0) at an identical training rate. Moreover, when employing the training rate adjustment and proposal method, there was a significant enhancement from 0.580 (S1, train 0.2, outlier 0, and ACCscore) to 0.928 (S1, train 0.6, outlier 0, and ACCscore).

Table 7. Comparison of Scenario 1 and Scenario 4 with training data and outlier rate.

Metrics	Train Rates	Scenarios	Outlier Rates			
			0	0.1	0.2	0.3
R^2score	0.2	S1	0.684	0.603	0.681	0.628
		S4	0.931	0.901	0.898	0.874
	0.4	S1	0.794	0.744	0.757	0.724
		S4	0.941	0.927	0.914	0.877
	0.5	S1	0.830	0.756	0.760	0.736
		S4	0.943	0.933	0.914	0.905
	0.6	S1	0.845	0.786	0.768	0.743
		S4	0.946	0.934	0.922	0.908
MMscore	0.2	S1	0.750	0.744	0.740	0.724
		S4	0.860	0.851	0.847	0.842
	0.4	S1	0.814	0.808	0.786	0.780
		S4	0.918	0.911	0.898	0.871
	0.5	S1	0.833	0.802	0.802	0.795
		S4	0.922	0.905	0.898	0.890
	0.6	S1	0.840	0.805	0.799	0.774
		S4	0.920	0.896	0.888	0.870
ACCscore	0.2	S1	0.580	0.567	0.540	0.517
		S4	0.867	0.844	0.826	0.807
	0.4	S1	0.659	0.632	0.624	0.613
		S4	0.886	0.872	0.858	0.845
	0.5	S1	0.717	0.654	0.633	0.621
		S4	0.901	0.886	0.879	0.854
	0.6	S1	0.745	0.656	0.644	0.637
		S4	0.928	0.895	0.894	0.868

Furthermore, to account for possible mismeasurement scenarios in the sensing data environment, we introduced outlier data with rates of 0.0, 0.1, 0.2, and 0.3 in the test dataset to compare the model performance. As the proportion of outlier data increased, it was observed that the accuracy of all scenarios decreased. When the train rate was 0.6 and outlier rates were 0, 0.1, 0.2, and 0.3, the ACCscores of S1 were 0.745, 0.656, 0.644, and 0.637, respectively. In contrast, the ACCscores of S4 were 0.928, 0.895, 0.894, and 0.868 in the same train and outlier rates. Also, we confirmed that the other performance metrics R^2score and MMscore also have a similar trend with ACCscore. Therefore, the proposed approach can be efficiently applied to improve NVR prediction accuracy using a combination of auxiliary data in a mismeasurement sensing environment through ensemble modeling.

4.5. Analyzing the Impact of Observations and Auxiliary Data on the NVR Prediction Model

In this study, we extracted Shapley Additive Explanations to verify the feature contribution of environmental data, auxiliary data using semisupervised learning, and important data for each scenario model. We used the SHAP library in Python to extract SHAP values, and Figure 13 shows the feature contributions for each scenario. Figure 13 shows the impact

of the features used in the model learning of each scenario on NVR prediction, presenting the absolute values of the SHAP averages in descending order. In the SHAP value graph for Scenario 1, Pd, RHd, Td, Tout, and Tin have the most significant impact on NVR prediction. From the top five features, four features excluding RHd have been previously identified as important factors affecting NVR in prior prediction studies, reinforcing the significant impact of correlated features on prediction when utilizing environmental data.

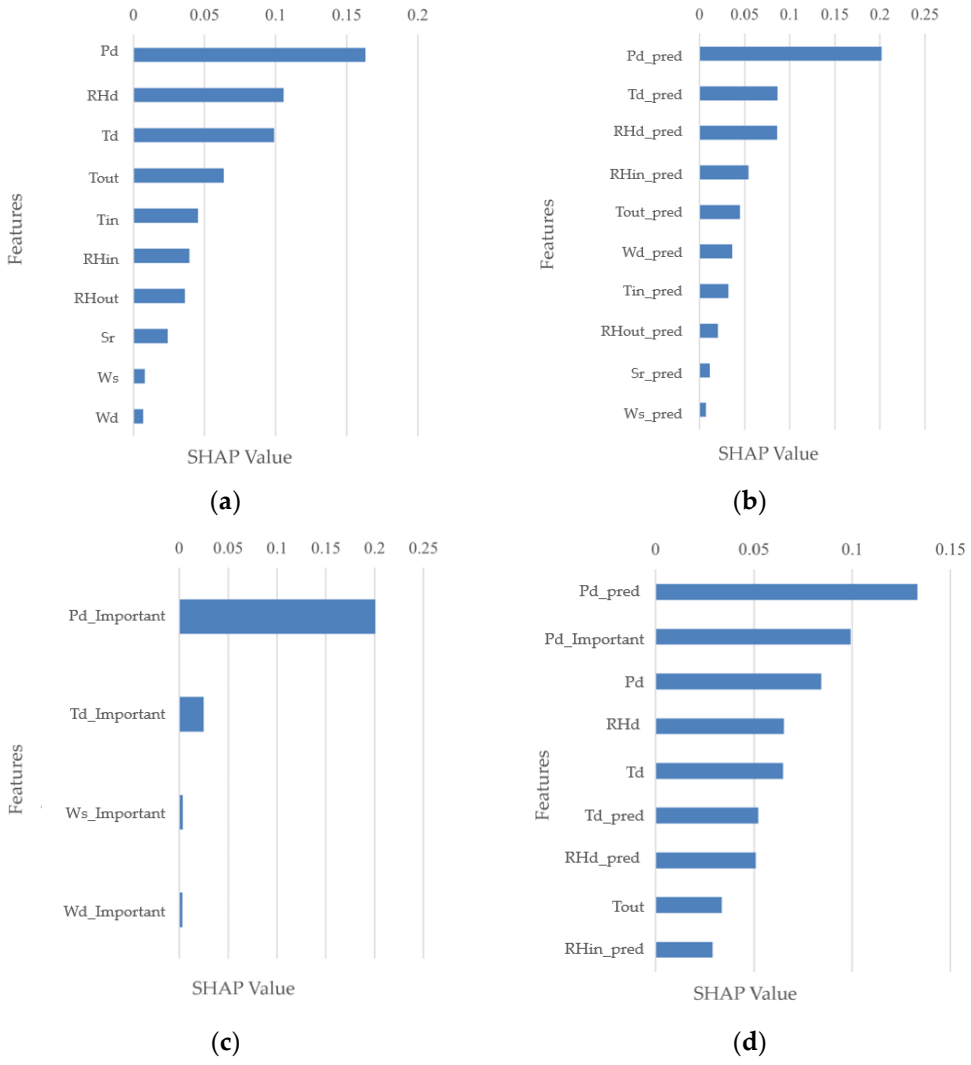

Figure 13. SHAP by scenario: (**a**) S1, (**b**) S2, (**c**) S3, and (**d**) S4 (outlier 0.3 and train 0.6).

Also, Scenario 2 places Pd, Td, and RHd, which were pointed out as important features in Scenario 1, at the top. Among the top five features, Tin and Tout are associated with Td; so, there is not much difference in ranking from Scenario 1. However, the difference in feature contribution between the two scenarios is that the features used in Scenario 2 were generated to assist in the uncertainties that can occur in the data collection environment of Scenario 1, and thus, the ranking within the features related to the importance features changed when comparing the difference in feature contribution with Scenario 1. The SHAP value of Scenario 3 shows that Pd significantly contributes to NVR prediction. Natural ventilation supplies outdoor air to the indoor space with the movement of air. Since the movement of air is induced either by a pressure difference or by the buoyancy phenomenon due to rising warm air, Pd is interpreted to have a greater influence than Td, Ws, or Wd.

Scenario 4 includes all the features of Scenario 1, 2, and 3; hence, in the case of Pd, three types of features, Pd_pred, Pd_Important, and Pd, rank at the top. Among the three Pd types, Pd_pred, which was generated as auxiliary data, shows the highest contribution, indicating

that the accuracy of NVR prediction has improved due to the learning of Pd_pred, which was generated for reliable learning, rather than the learning of sensing data Pd, which did not consider the uncertain sensing environment. In addition to Pd, auxiliary data such as Td and RHd are also among the top in feature contribution out of the total 34 features, similar to sensing data. Scenario 4, which inherited the knowledge of Scenario 2, can be interpreted as contributing to the improvement of NVR prediction accuracy by solving the realistic problem of the mismeasurement of collected data through the knowledge of auxiliary data. In Scenario 3, despite using the same data as sensing data Pd, the contribution of Pd_Important was higher. This can be attributed to the influence of Scenario 3, which recognized Pd_Important as a significant feature correlated with NVR through prior research and feature selection processes. Therefore, in the feature contribution of Scenario 4, Pd_Important exhibits a higher feature contribution than Pd.

In this study, we compared the feature contributions for NVR prediction in each scenario using SHAP and observed that temperature, humidity, and Pd-related features generally have relatively high contributions across most scenarios. Additionally, the use of auxiliary data from Scenario 2, which addresses irregularly changing sensing environments, showed the highest feature contribution in Scenario 4. This finding suggests that employing auxiliary data improves data reliability, enhances NVR prediction accuracy, and allows for better generalization performance. Consequently, Scenario 4 enables NVR prediction with enhanced generalization performance by utilizing sensing data and auxiliary data.

5. Conclusions

In this study, we proposed a way to design a deep learning NVR prediction model using generated auxiliary data considering the practical issue of the mismeasurement of data collected in the IoT sensing environment that changes irregularly according to time and space. The auxiliary data were generated based on semisupervised learning that predicts the amount of natural ventilation using the observed data and repredicts the observed data using the predicted amount of natural ventilation by using a DNN and LSTM according to irregular and regular changes in characteristics, respectively. Finally, to enhance the model against mismeasurement sensing data, a combination of observation, auxiliary, and important features (i.e., Pd, Wd, *Ws*, and Td) were trained to achieve reliable learning. Based on the experimental results, we compared the degree of improvement in each scenario by adjusting the training and outlier rates with three performance metrics. When applying the proposed approach at the same training rate, the NVR prediction accuracy improved by up to 0.287, and when combining the training rate adjustment with the proposed approach, the improvement reached up to 0.348. Furthermore, with the mismeasurement data, the NVR prediction accuracy decreased by up to 20%. In contrast, we confirmed that the proposed approach (i.e., S4) can provide more NVR prediction accuracy in the training and outlier rates than the straightforward approach (i.e., S1). Consequently, the proposed approach achieved a maximum improvement of 57% in reducing the accuracy decrease caused by outlier data. In this study, we generated generation data and importance data in addition to observation data, and we improved the accuracy of an NVR prediction model that considers the mismeasurement environment by training an ensemble model capable of comprehensively incorporating all knowledge. In future work, for a more accurate prediction model, we will study advanced deep learning techniques and analyze the accuracy performances, including accurate prediction of Wd and Ws.

Author Contributions: S.Y. and S.L. conceptualized and designed the experiments; S.Y. and M.K. designed and implemented the detection system; S.L. validated the proposed method; S.Y., M.K., and S.L. wrote the study. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by a 2022 research grant from Sangmyung University (2022-A000-0320).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fathi, S.; Srinivasan, R.; Fenner, A.; Fathi, S. Machine learning applications in urban building energy performance forecasting: A systematic review. *Renew. Sustain. Energy Rev.* **2020**, *133*, 110287. [CrossRef]
2. Khoukhi, M.; Yoshino, H.; Liu, J. The effect of the wind speed velocity on the stack pressure in medium-rise buildings in cold region of China. *Build. Environ.* **2007**, *42*, 1081–1088. [CrossRef]
3. Yan, D.; Song, F.; Yang, X.; Jiang, Y.; Zhao, B.; Zhang, X.; Liu, X.; Wang, X.; Xu, F.; Wu, P.; et al. An integrated modeling tool for simultaneous analysis of thermal performance and indoor air quality in buildings. *Build. Environ.* **2008**, *43*, 287–293. [CrossRef]
4. Dehghan, A.A.; Esfeh, M.K.; Manshadi, M.D. Natural ventilation characteristics of one-sided wind catchers: Experimental and analytical evaluation. *Energy Build.* **2013**, *61*, 366–377. [CrossRef]
5. Wang, J.; Wang, S.; Zhang, T.; Battaglia, F. Assessment of single-sided natural ventilation driven by buoyancy forces through variable window configurations. *Energy Build.* **2017**, *139*, 762–779. [CrossRef]
6. Elshafei, G.; Negm, A.; Bady, M.; Suzuki, M.; Ibrahim, M.G. Numerical and experimental investigations of the impacts of window parameters on indoor natural ventilation in a residential building. *Energy Build.* **2017**, *141*, 321–332. [CrossRef]
7. Han, D.H.; Kim, S.; Choi, J.H.; Kim, Y.S.; Chung, H.; Jeong, H.; Watjanatepin, N.; Ruangpattanawiwat, C.; Choi, S.-H. Experimental study on thermal buoyancy-induced natural ventilation. *Energy Build.* **2018**, *177*, 1–11. [CrossRef]
8. Heracleous, C.; Michael, A. Experimental assessment of the impact of natural ventilation on indoor air quality and thermal comfort conditions of educational buildings in the Eastern Mediterranean region during the heating period. *J. Build. Eng.* **2019**, *26*, 100917. [CrossRef]
9. Calautit, J.K.; Tien, P.W.; Wei, S.; Calautit, K.; Hughes, B. Numerical and experimental investigation of the indoor air quality and thermal comfort performance of a low energy cooling windcatcher with heat pipes and extended surfaces. *Renew. Energy* **2020**, *145*, 744–756. [CrossRef]
10. Muhsin, F.; Yusoff, W.F.M.; Mohamed, M.F.; Sapian, A.R. CFD modeling of natural ventilation in a void connected to the living units of multi-storey housing for thermal comfort. *Energy Build.* **2017**, *144*, 1–16. [CrossRef]
11. Villagran, E.A.; Romero, E.J.B.; Bojacá, C.R. Transient CFD analysis of the natural ventilation of three types of greenhouses used for agricultural production in a tropical mountain climate. *Biosyst. Eng.* **2019**, *188*, 288–304. [CrossRef]
12. Calautit, J.K.; O'Connor, D.; Tien, P.W.; Wei, S.; Pantua, C.A.J.; Hughes, B. Development of a natural ventilation windcatcher with passive heat recovery wheel for mild-cold climates: CFD and experimental analysis. *Renew. Energy* **2020**, *160*, 465–482. [CrossRef]
13. Rodrigues Marques Sakiyama, N.; Frick, J.; Bejat, T.; Garrecht, H. Using CFD to evaluate natural ventilation through a 3D parametric modeling approach. *Energies* **2021**, *14*, 2197. [CrossRef]
14. Park, H.; Park, D.Y. Comparative analysis on predictability of natural ventilation rate based on machine learning algorithms. *Build. Environ.* **2021**, *195*, 107744. [CrossRef]
15. Taheri, S.; Razban, A. Learning-based CO_2 concentration prediction: Application to indoor air quality control using de-mand-controlled ventilation. *Build. Environ.* **2021**, *205*, 108164. [CrossRef]
16. Hiyama, K.; Takeuchi, K.; Omodaka, Y.; Srisamranrungruang, T. Operation strategy for engineered natural ventilation using machine learning under sparse data conditions. *Jpn. Archit. Rev.* **2022**, *5*, 119–126. [CrossRef]
17. Wu, W.; Norford, L.K.; Li, N.; Malkawi, A. Model Predictive Control of Single-Sided Natural Ventilation in a Smart Building Using Machine Learning Algorithms. *IEEE Trans. Autom. Sci. Eng.* **2018**, *15*, 636–646.
18. Shakibjoo, A.D.; Moradzadeh, M.; Din, S.U.; Mohammadzadeh, A.; Mosavi, A.H.; Vandevelde, L. Optimized type-2 fuzzy frequency control for multi-area power systems. *IEEE Access* **2021**, *10*, 6989–7002. [CrossRef]
19. Koutsoukas, A.; Monaghan, K.J.; Li, X.; Huan, J. Deep-learning: Investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data. *J. Cheminform.* **2017**, *9*, 1. [CrossRef]
20. Delalleau, O.; Bengio, Y. Shallow vs. deep sum-product networks. *Adv. Neural Inf. Process. Syst.* **2011**, *24*, 1–9.
21. Hinton, G.E.; Osindero, S.; Teh, Y.W. A fast learning algorithm for deep belief nets. *Neural Comput.* **2006**, *18*, 1527–1554. [CrossRef] [PubMed]
22. Bengio, Y. Learning deep architectures for AI. *Found. Trends Mach. Learn.* **2009**, *2*, 1–127. [CrossRef]
23. Agarap, F. Deep learning using rectified linear units (ReLU). *arXiv* **2018**, arXiv:1803.08375.
24. Liu, Z. A method of SVM with normalization in intrusion detection. *Procedia Environ. Sci.* **2011**, *11*, 256–262. [CrossRef]
25. Ahfock, D.; McLachlan, G.J. Semi-Supervised Learning of Classifiers from a Statistical Perspective: A Brief Review. In *Econometrics and Statistics*; Elsevier: Amsterdam, The Netherlands, 2022.
26. Sajun, A.; Zualkernan, I. Survey on Implementations of Generative Adversarial Networks for Semi-Supervised Learning. *Appl. Sci.* **2022**, *12*, 1718. [CrossRef]
27. Larsen, T.S.; Heiselberg, P. Single-sided natural ventilation driven by wind pressure and temperature difference. *Energy Build.* **2008**, *40*, 1031–1040. [CrossRef]
28. Gaikwad, D.P.; Thool, R.C. Intrusion detection system using bagging ensemble method of machine learning. In Proceedings of the 2015 International Conference on Computing Communication Control and Automation, Pune, India, 26–27 February 2015; IEEE: Toulouse, France, 2015; pp. 291–295.
29. Joseph, R.V.; Mohanty, A.; Tyagi, S.; Mishra, S.; Satapathy, S.K.; Mohanty, S.N. A hybrid deep learning framework with CNN and Bi-directional LSTM for store item demand forecasting. *Comput. Electr. Eng.* **2022**, *103*, 108358. [CrossRef]

30. Rustam, F.; Reshi, A.A.; Mehmood, A.; Ullah, S.; On, B.W.; Aslam, W.; Choi, G.S. COVID-19 future forecasting using supervised machine learning models. *IEEE Access* **2020**, *8*, 101489–101499. [CrossRef]
31. Berhich, A.; Belouadha, F.Z.; Kabbaj, M.I. An attention-based LSTM network for large earthquake prediction. *Soil Dyn. Earthq. Eng.* **2023**, *165*, 107663. [CrossRef]
32. Vafaei, S.; Soosani, J.; Adeli, K.; Fadaei, H.; Naghavi, H.; Pham, T.D.; Tien Bui, D. Improving accuracy estimation of Forest Aboveground Biomass based on incorporation of ALOS-2 PALSAR-2 and Sentinel-2A imagery and machine learning: A case study of the Hyrcanian forest area (Iran). *Remote Sens.* **2018**, *10*, 172. [CrossRef]