

# Deep Learning Techniques to Enhance Energy Efficiency of Home Appliances by Analyzing Air Quality Levels

Jasbir Singh Saini<sup>1\*</sup>, Sunny Arora<sup>2</sup>, and Sushil Kamboj<sup>3</sup>

<sup>1,2</sup>Department of Computer Science and Engineering, Guru Kashi University  
Punjab, India 151302

<sup>3</sup>Department of Information Technology, Chandigarh Group of Colleges  
Punjab, India 140307

Email: <sup>1</sup>sainijassbir@gmail.com, <sup>2</sup>sunnyaro@gmail.com, <sup>3</sup>dr.sushilkamboj@gmail.com

**Abstract**—Energy efficiency in home appliances is a critical area of research that addresses the growing demand for reducing energy consumption. The rapid growth in artificial intelligence has prioritized the development of advanced methods to improve sustainable energy consumption, particularly by optimizing the energy efficiency of home appliances. The research introduces a novel deep learning-based framework to enhance energy efficiency in home appliances by leveraging insights from Indoor Air Quality (IAQ) metrics. Unlike conventional energy management approaches, which face challenges such as limited datasets, computational inefficiencies, and a lack of generalizability, the research incorporates advanced preprocessing and augmentation techniques. Specifically, a hybrid Synthetic Minority Over-sampling Technique - Edited Nearest Neighbors (SMOTE-ENN) approach addresses class imbalance, while Z-score normalization ensures consistent feature scaling. Among the evaluated models, the Bidirectional Gated Recurrent Unit (GRU) and the Stacked Long Short-Term Memory (LSTM) stand out, achieving exceptional validation accuracies of 99.81% and 99.64%, respectively, demonstrating superior generalization. This framework uniquely integrates IAQ data to optimize energy usage dynamically, showcasing how environmental factors such as CO<sub>2</sub>, humidity, and temperature can inform sustainable energy practices. These findings underscore the transformative potential of deep learning in fostering eco-friendly innovations for smart home energy management. They show the broader potential for integrating artificial intelligence-driven approaches into energy policies and sustainability strategies, enabling more effective reductions in residential energy consumption and combating climate change.

**Index Terms**—Deep Learning, Home Appliances, Energy Efficiency, Air Quality Level (AQL)

## I. INTRODUCTION

Received: July 16, 2024; received in revised form: Dec. 04, 2025; accepted: Dec. 04, 2025; available online: Sep. 23, 2025.

\*Corresponding Author

SINCE the 21<sup>st</sup> century began, the world has experienced significant growth in global energy consumption across almost all regions. It has been observed that energy consumption often increases due to factors such as economic growth, demographic expansion, and higher electricity usage per capita [1]. Optimizing the efficiency of household appliances plays a crucial role in mitigating this rise in energy demand because of its contribution of over 30% by residential consumers in certain nations. It also addresses both the economic and environmental impacts of rising energy consumption, which signifies the importance of sustainable practices in residential energy management [2]. Moreover, such improvements also contribute to reducing pollution, in addition to their individual contributions to climate change. In fact, an energy-efficient home integrates some advanced technologies and designs that lessen the use of energy and maintain the same level of safety, convenience, comfort, and visual attractiveness as traditional homes [3]. Figure 1 displays the percentage of energy consumed by home appliances.

Globally, policies aimed at enhancing energy efficiency provide significant benefits to both energy suppliers and consumers, yielding environmental, social, and economic advantages. Efficient energy use is essential for strengthening energy supply amid rising demand and economic expansion. It includes advancements in energy management to mitigate peak energy demands, optimize the usage of household appliances, and develop increasingly efficient appliances [5]. Particular importance is placed on improving the efficiency of heating and cooling systems, which constitute significant energy loads within residential settings. The Air Quality Index (AQI) also serves as a crucial

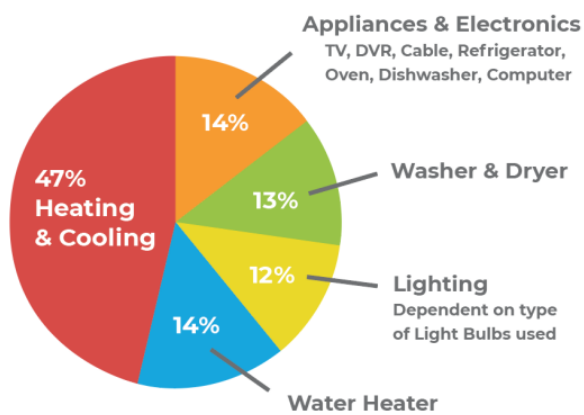


Fig. 1. Energy contributions by different home appliances to highlight their share in total residential energy consumption [4].

metric for optimizing the performance and energy efficiency of household appliances. Increased levels of pollutants, such as dust, can affect the effectiveness of air purifiers and fans, which require adjustments to maintain optimal Indoor Air Quality (IAQ). Similarly, higher CO<sub>2</sub> levels, indicative of increased occupancy, may require optimal use of heating, ventilation, and lighting systems for efficient operation [6]. Analyzing past IAQ index data enables the identification of patterns and correlations, which facilitate personalized recommendations to optimize appliance performance by considering air quality conditions. Thus, using an IAQ index will help people to predict how the appliances will perform and aid people to plan ahead to keep them working efficiently in different indoor conditions [7].

There are various conventional techniques to enhance the energy efficiency of home appliances. They involve the implementation of energy-efficiency standards, the improvement of appliance design, and the promotion of energy-efficient technologies. However, these methods also face challenges in terms of the complexity of appliance design, resistance to standards and regulations from manufacturers, and the influence of consumer behaviour [8]. To address these issues, deep learning provides a promising solution by optimizing appliance design through data-driven insights. Moreover, deep learning algorithms can identify correlations and patterns by analyzing large datasets on appliance performance, energy consumption, and IAQ that traditional methods may overlook. Hence, it enables the development of more efficient appliance designs that not only save energy but also contribute to improved IAQ [9].

Several studies have demonstrated the contribution of machine and deep learning techniques to the energy

efficiency of home appliances. Previous researchers have proposed an Artificial Intelligence-based Energy Management Model (AI-EMM) for green buildings to prioritize comfort and safety of users, along with energy efficiency. They have utilized a universal infrared communication system and a Long Short-Term Memory (LSTM) model to optimize energy consumption, emphasizing airside design optimization of the Heating, Ventilation, and Air Conditioning (HVAC) system to display both economic and environmental benefits. Green buildings benefitted from the AI-EMM compute high-performance ratio of 94.3%, reduced energy consumption of 15.7%, accuracy (97.4%), energy management level (95.7%), and prediction accuracy (97.1%) [10]. Another research has used correlation analysis to collect data to discard redundant sensors and focused on optimizing Internet of Things (IoT) system design for smart homes. It has used data analysis and prediction techniques to enhance energy efficiency by correlating heterogeneous IoT sensor data and proposed a machine learning-based intelligent service model, which is evaluated using Root Mean Squared Error (RMSE). The results indicate that the gradient-boosting regressor is the most effective, achieving an RMSE of 22.29 [11].

Additionally, various architectures of Deep Recurrent Neural Networks (DRNNs) are examined. Those are tailored for medium- and long-term energy demand predictions, specifically for heating and electricity consumption at a 1-hour resolution. In previous research, the proposed DRNN model outperforms the Support Vector Machine (SVM) and Gradient Boosting (GB) regression models by 5.4% and 7.0%, respectively, in terms of energy forecasting accuracy. A novel model is also proposed, which consists of three components, i.e., smoothing, which employs a Kalman filter for eliminating the noise from data, optimization to minimize the cost error in real-time data by using firefly and genetic algorithms, and control to manage the distribution of energy for lightning, temperature, and others, efficiently using Mamdani fuzzy logics. The previous researchers also compare their work with the existing techniques and find that their model outperforms and highlights the importance of using optimizers for energy efficiency. It improves user comfort along with the impact of adaptive controllers to overcome incorrect Proportional–Integral–Derivative (PID) controller selections [12].

Then, previous research has sourced data from Kaggle, which comprises 29 features to focus on the minimal consumption of energy using various machine learning models, such as LSTM, along with optimization techniques like genetic algorithm and Grey

Wolf Optimization (GWO) to fine-tune hyperparameters. While evaluating, GWO-LSTM highlights its superior performance, showcasing exceptional predictive capabilities with minimal errors [13]. Another research has used One Dimensional Deep Convolutional Neural Network, LSTM, and scheduling algorithm to extract features, load forecasting based on features that have been extracted, and optimized appliance operation times, respectively, to develop an energy consumption control system for smart homes. It validates the model through simulation scenarios with authentic datasets, which demonstrate its effectiveness in meeting energy demands without requiring additional energy sources. It finds that the proposed system displays advancement in smart home energy management [14]. Another research has explored deep learning techniques for air quality forecasting, focusing on Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), LSTM, and spatiotemporal networks for modelling nonlinear spatiotemporal features. It also discusses challenges such as overfitting and the practical implications for real-world deployment [15].

Although existing research has made significant strides in developing models to enhance energy efficiency in smart rooms, several challenges and limitations persist. There is often a lack of consideration for the substantial computational time required to train these models, which can impact practical implementation. Previous researchers have also encountered challenges related to the availability of limited datasets, leading to potential biases or issues with generalizability. Furthermore, data inconsistency across different sources poses a significant challenge, affecting the reliability and accuracy of the models. Addressing these challenges is crucial for advancing the field and developing more effective strategies for enhancing the energy efficiency of home appliances in smart rooms.

Moreover, even though existing studies have explored various methods to optimize energy efficiency in residential settings, the integration of IAQ as a critical factor remains underexplored. The research bridges that gap by incorporating IAQ parameters, such as CO<sub>2</sub> levels, humidity, and temperature, into a novel deep learning framework, enabling dynamic and context-aware energy management. By employing advanced models like Bidirectional Gated Recurrent Unit (GRU), Stacked LSTM, and others, the researchers capture both short-term and long-term temporal dependencies, ensuring accurate predictions of appliance performance under varying environmental conditions. The use of a hybrid Synthetic Minority Over-sampling Technique - Edited Nearest Neighbors (SMOTE-ENN) technique to address class imbalance further enhances the reliability

and generalizability of the results. This approach not only highlights the synergy between IAQ and energy efficiency but also sets a foundation for scalable and adaptive energy solutions in smart homes. Such work underscores the transformative potential of integrating IAQ metrics into sustainable energy management practices, paving the way for smarter, healthier, and more energy-efficient residential environments.

#### A. Research Coverage

The research aims to develop an automated system that utilizes deep learning techniques to identify and classify Air Quality Level (AQL) based on multiple parameters, including indoor AQI. The contribution to performing the research is as follows:

- 1) Initially, a dataset consisting of approximately 1.3 Lakh records with seven attributes, such as CO<sub>2</sub> levels, humidity, Passive Infrared (PIR), temperature, IAQ index, and AQL of rooms. The data are collected from two rooms of 415 (Data I) and 776 (Data II).
- 2) Subsequently, the data are preprocessed to check for null or missing values, followed by graphical visualization to understand the pattern of the dataset.
- 3) To address the class imbalance issue, the Synthetic Minority Over-sampling Technique (SMOTE) is employed, and the features of the dataset are standardized through scaling.
- 4) Various deep learning techniques are applied and trained with the dataset. The performances of these techniques are later examined using various standard metrics, including the learning curves, confusion matrix, and computational time.

## II. RESEARCH METHOD

The researchers define the phases that have been used to predict and classify the AQL of a room using hybrid advanced deep learning techniques, as shown in Fig. 2. The dataset collects records from 255 sensors, which have been located across 51 rooms spanning from 4 floors of Sutardja Dai Hall at UC Berkeley. It includes diverse attributes like PIR sensor, Carbon Dioxide (CO<sub>2</sub>) tiers, humidity, temperature, and luminosity, with readings recorded every five seconds time series information in the form of UNIX EPOCH TIME timestamp [16].

During the data preprocessing phase, initially, a thorough analysis is carried out to identify any missing or null values in each attribute for all 51 rooms. It is done to make sure that the data are complete and accurate, as shown in Table I. The K-Nearest Neighbors (KNN) imputer technique is employed to

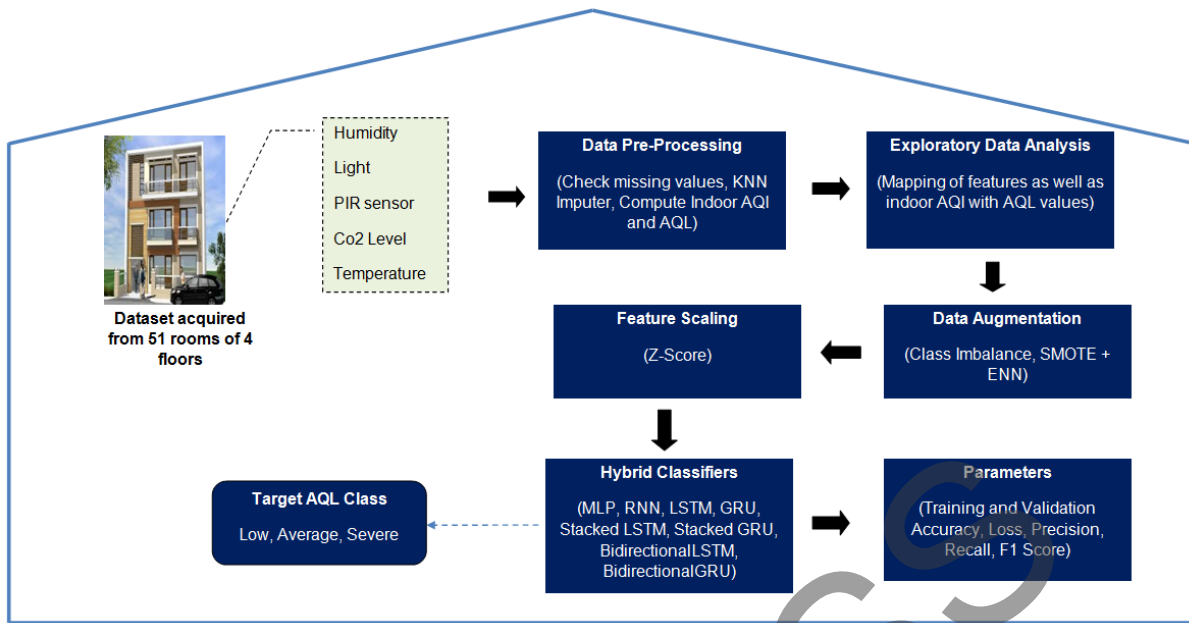


Fig. 2. Proposed system for air quality assessment using deep learning classifiers. Note: Air Quality Index (AQI), Air Quality Level (AQL), K-Nearest Neighbors (KNN), Synthetic Minority Over-sampling Technique - Edited Nearest Neighbors (SMOTE-ENN), Multilayer Perceptron (MLP), Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), and Long Short-Term Memory (LSTM).

fill in missing values based on information from nearby data points [17]. This approach helps to maintain the structure and patterns within the dataset. Afterward, using a comprehensive dataset, the AQI values for each room are calculated. This process entails collecting and analyzing several characteristics linked to each space to calculate a comprehensive measure that represents the air quality. However, it is noted throughout this procedure that certain estimated AQI values are negative, which are not suitable for meaningful interpretation and analysis. To tackle this problem, the dataset is improved by removing records with negative AQI values and concentrating only on the data from two rooms, specifically 415 (Data I) and 776 (Data II), which are chosen randomly from the original 51 rooms. Subsequently, utilizing the AQI data, the researchers derive the AQL values and establish the desired classes as Low (0-50), Average (51-100), and Severe (101-500).

Next, Exploratory Data Analysis (EDA) is used to uncover crucial information that helps understand the complex relationships between energy consumption and various environmental factors. Figure 3 illustrates the correlation between computed IAQ index (AQI) values and corresponding AQL classes, including Low, Average, and Severe. The purpose is to determine the minimum and maximum indoor AQI values recorded from both rooms 415 (Data I) and 776 (Data II). This information is crucial for improving the classification

TABLE I  
COMPARISON OF MISSING VALUES ACROSS ATTRIBUTES IN DATA I AND DATA II TO EMPHASIZE THE SIGNIFICANCE OF IMPUTING MISSING DATA FOR ACCURATE ANALYSIS.

Attributes	Data I (Room 415)	Data II (Room 776)
CO2	0	1095
Humidity	1113	1
Light	1113	1
Passive Infrared (PIR)	55875	59171
Temperature	1114	0

system and establishing distinct thresholds for different AQL. Ultimately, it can enhance comprehension and enable effective management of indoor environmental conditions.

The graphical depiction in Figs. A1 and A2 (see Appendix) portrays the distribution of attribute values, encompassing CO2 concentration, PIR, light intensity, humidity, and temperature, across varying levels of air quality: Low, Average, and Severe. Upon analyzing the data from Room 415 in Fig. A1 (see Appendix), it is observed that the highest frequency of occurrences indicating Low air quality is found within the following intervals: 450 to 500 for CO2 concentration, 23.0 to 23.5 for temperature, 58 to 60 for humidity, and 0 to 25 for light intensity. Similarly, for the Average air quality category, peak incidences manifest within the ranges of 690 to 700 for CO2, 23.5 to 23.9 for temperature, 54 to 55 and 58 to 58.5 for humidity, and

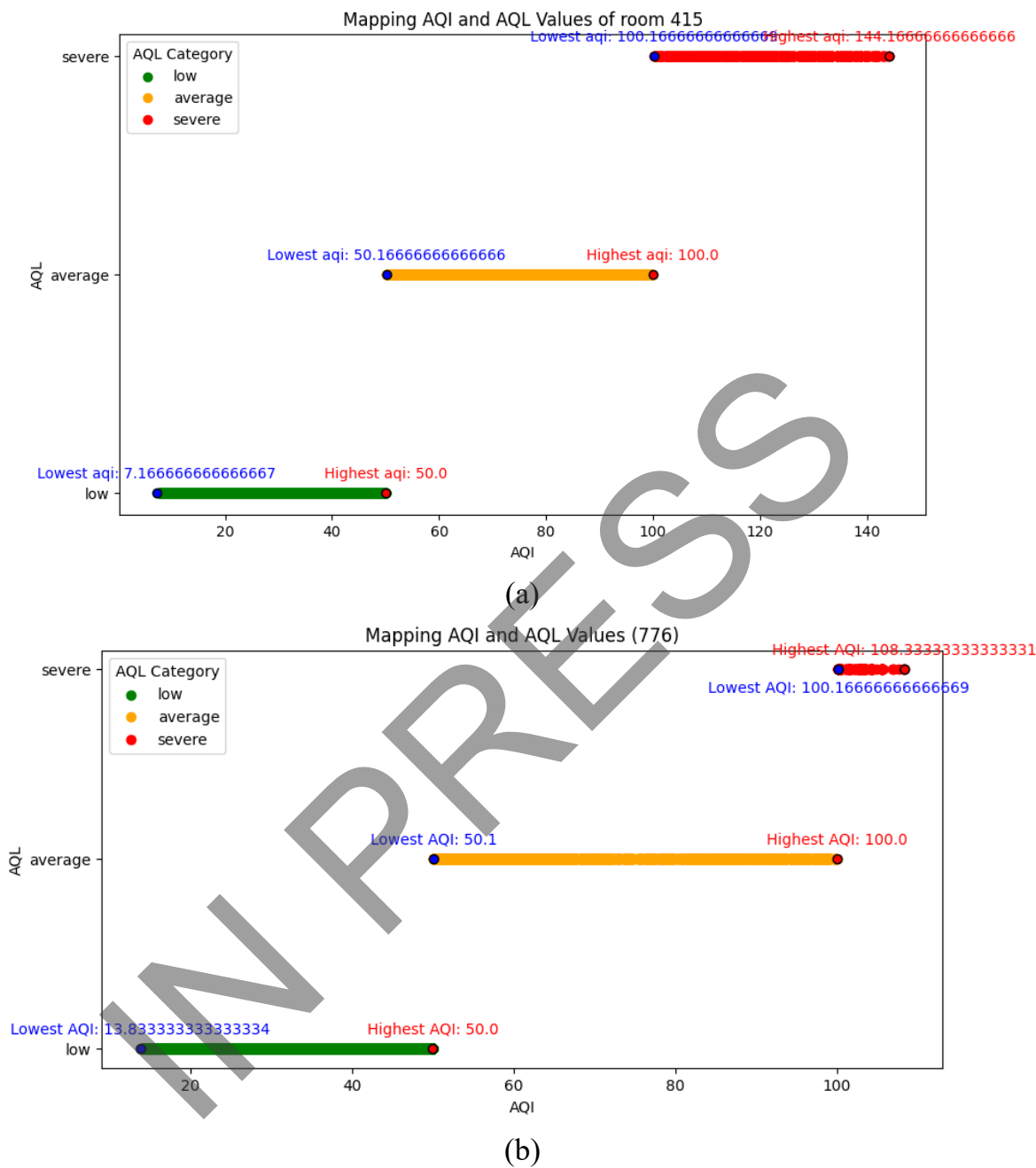


Fig. 3. Air Quality Index (AQI) values of Data I and Data II for different categories of Air Quality Level (AQL).

40 to 80 for light intensity. Conversely, in the Severe air quality classification, predominant values are recorded between 1050 to 1100 for CO<sub>2</sub> concentration, 23.65 to 23.70 for temperature, below 58 (e.g., 57.84) for humidity, and 40 to 50 for light intensity. It is noteworthy that these values are approximations and not fixed constants.

Similarly, the researchers have extracted analogous information from the data recorded for Room 776, as illustrated in Fig. A2 (see Appendix). The highest frequency of occurrences indicating low air quality is observed within the intervals of 450 to 500 for CO<sub>2</sub> concentration, 23.0 to 23.5 for temperature, 57 to 58 for humidity, and 0 to 10 for light intensity. For the

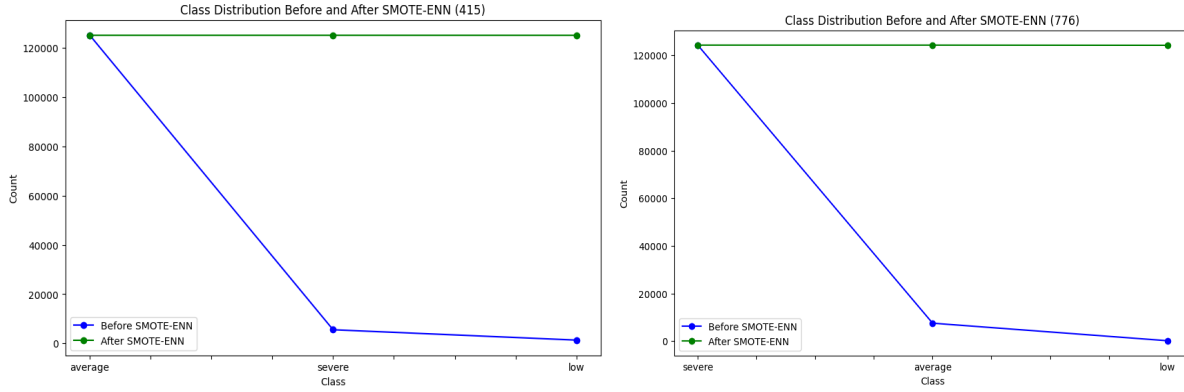


Fig. 4. Class balancing achieved using the Synthetic Minority Over-sampling Technique - Edited Nearest Neighbors (SMOTE-ENN) technique to improve the distribution of Air Quality Level (AQL) in Data I and Data I.

average air quality category, peak incidences manifest within the ranges of 700 to 720 for CO<sub>2</sub>, around 25.0 for temperature, 54 to 55 for humidity, and 60, and 90 to 100 for light intensity. Conversely, in the severe air quality classification, predominant values are recorded at approximately 700 for CO<sub>2</sub> concentration, around 25.2 for temperature, 54.5 for humidity, and 80 to 100 for light intensity. Additionally, for PIR, there are some occurrences of values apart from zero across all attributes. It is important to note that these values are approximations and not fixed constants.

These visualizations aim to enhance energy efficiency. Optimizing the detection and management systems based on these identified thresholds can facilitate proactive interventions. It can ensure that resource allocation aligns with actual environmental conditions, thereby minimizing energy consumption while maintaining air quality standards.

In the data augmentation process, a hybrid approach of SMOTE and ENN has been used to augment the data by overcoming the issue of class imbalance, as shown in Fig. 4. SMOTE generates synthetic instances for the minority class, which increases its representation in the dataset. Meanwhile, ENN removes noisy instances from both the minority and majority classes, resulting in a balanced and cleaner dataset overall [18]. By effectively addressing class imbalance and reducing noise, SMOTE-ENN enhances the performance of applied learning models. It can be represented as Eq. (1).

$$SMOTE + ENN(X, y) = ENN(SMOTE(X_{\text{minority}}, y_{\text{minority}}, k)). \quad (1)$$

Here,  $X$  refers to the feature matrix of the dataset,  $y$  represents the target vector of class labels, and  $k$  denotes the number of nearest neighbors used in both SMOTE and ENN. Many standard oversampling

techniques, like standalone SMOTE, are effective in increasing the representation of minority classes by generating synthetic samples. However, these methods can inadvertently introduce noise by creating synthetic samples near outliers or overlapping class boundaries, which can degrade model performance. Traditional undersampling methods, on the other hand, focus on removing data from the majority class to achieve balance. However, it often results in the loss of valuable information.

In the feature scaling process, Z-scores are useful to identify outliers within a dataset. Data points with Z-scores significantly greater than or less than zero are considered outliers, as they deviate substantially from the average values of the dataset. By standardizing the data using Z-scores, it becomes easier to identify and understand the significance of outliers and to compare data points across different datasets with varying means and standard deviations [19], as shown in Eq. (2). Here,  $x$  is the value of the input data point,  $\mu$  is the mean of the population, and  $\sigma$  is the standard deviation of the population.

$$z = \frac{x - \mu}{\sigma}. \quad (2)$$

The next process is classifiers. In the context of indoor energy efficiency, the Multilayer Perceptron (MLP) structure is customized to analyze various environmental parameters, including temperature, humidity, occupancy, and lighting situations, collected from sensors deployed within a building. The structure commonly includes an input layer, in which environmental data is fed into the model, followed with the aid of one or extra hidden layers, which perform nonlinear adjustments and feature extraction. Each neuron within those hidden layers applies weighted connections and activation capabilities to get the input



records. Finally, an output layer produces predictions for energy-efficient operation based on the discovered patterns [20]. The mathematical equation of MLP is represented as Eqs. (3) and (4). Here,  $n$  represents the input features,  $w_{ij}^l$  and  $z_j^{(l)}$  refer to the weight connected to the  $j$ -th neuron,  $b_j^l$  is the bias,  $x_i$  is the input feature,  $f(\cdot)$  is the activation function, and  $a_j^{(l)}$  is the output of the  $j$ -th neuron. The  $j$  refers to the neuron number within a given layer.

$$z_j^{(l)} = \sum_{i=1}^n w_{ij}^l x_i + b_j^l, \quad (3)$$

$$a_j^{(l)} = f(z_j^{(l)}). \quad (4)$$

The RNN is a type of artificial neural network that is mostly used for processing sequential data to make it applicable for multiple tasks in optimizing energy efficiency for indoor environments. Its architecture includes an input, hidden, and output layer with recurrent connections, which enable the network to maintain information about past entered data in memory during processing of current data [21]. By utilizing the sequential nature of environmental data obtained indoors, these types of networks contribute to optimizing energy use by enhancing occupant comfort and promoting sustainability in indoor spaces. The hidden state ( $h_t$ ) in RNN is defined by Eq. (5). Here,  $W_{hx}$  and  $W_{hh}$  imply the weight matrix for input to hidden and hidden to hidden connections,  $\sigma$  is activation function, and  $b_h$  is the bias vector, and  $x_t$  is the input vector at step  $t$ .

$$h_t = \sigma(W_{hx}x_t + W_{hh}h_{t-1} + b_h). \quad (5)$$

The LSTM network is one variant of RNN architectures that is designed for taking long-time dependencies in sequential primarily based information and addressing the problems related to vanished gradients. This property of LSTM makes it particularly efficient for the obligations which are associated with energy efficiency in indoor environments. The structure of an LSTM includes memory cells with self-connected devices, referred to as gates. These gates consist of an input, forget, and an output gate, which alter the flow of data through the network and manipulate it at different levels of processing [22]. In the area of efficient use of indoor energy, LSTM network excels at predicting complicated temporal styles in sensor data, which include fluctuations in temperature, tendencies of occupancy, and energy consumption profiles. The gates are presented in Eqs. (6)–(11). Here,  $W_f$ ,  $W_i$ ,  $W_o$ , and  $W_c$  represent the weight matrices for the forget, input, and output gates, and the candidate cell state, respectively. The  $b_f$ ,  $b_i$  and  $b_o$  represent bias for forget, input and output gates, respectively, whereas  $b_c$

represents bias for candidate cell state. Then,  $\sigma$  and  $\tanh$  refer to the activation and the hyperbolic tangent activation function, respectively, and  $(\cdot)$  represents matrix multiplication.

$$\text{Forget gate}(f_t) = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \quad (6)$$

$$\text{Input gate}(i_t) = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \quad (7)$$

$$\text{Output gate}(o_t) = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o). \quad (8)$$

$$\text{Candidate cell state}(\tilde{c}_t) = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c), \quad (9)$$

$$\text{Cell state update}(c_t) = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t, \quad (10)$$

$$\text{Hidden state update}(h_t) = o_t \cdot \tanh(c_t). \quad (11)$$

The other variation of RNN is the GRU structure. This architecture is similar to LST. However, it copes with the restrictions of conventional RNN. Additionally, it is also able to fetch long-term dependencies in sequential facts efficiently. The structure of GRU is based entirely on gating mechanisms, which consist of a reset gate and a forget gate. These gates control the flow of records through the network and allow the GRU to retain or forget about data selectively from previous time steps. This characteristic permits the network to capture long-time period dependencies while reducing the vanishing gradient problem [23]. Historical records may be used to train the GRU network for learning the underlying patterns in addition to the dynamics of indoor environments. It enables adaptive energy management techniques that work for conditions in real-time. It is mathematically represented by Eqs. (12)–(15). Here,  $W_z$ ,  $W_r$ , and  $W_h$  refer to the weight matrices of update, reset gate, and candidate hidden state,  $b_z$  and  $b_r$  represent the bias for update gate and reset gate. The  $\odot$  implies element-wise multiplication, and  $\tilde{h}_t$  is the current hidden state.

$$\text{Update gate}(z_t) = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z), \quad (12)$$

$$\text{Reset gate}(r_t) = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r), \quad (13)$$

$$\text{Candidate hidden state}(\tilde{h}_t) =$$

$$\tanh(W_h \cdot [r_t \odot h_{t-1}, x_t] + b_h), \quad (14)$$

$$\text{Hidden state update}(h_t) = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t. \quad (15)$$

The architecture of Bidirectional LSTM (Bi-LSTM) incorporates records from both past and future time steps. It consists of two LSTM layers, where one layer processes the input information in a forward order, and the other layer processes it in reverse order. While processing the data in both instructions, the Bi-LSTM captures data from both past and future contexts. This property of Bi-LSTM allows it to learn deeper representations of the enter records and enables

it higher to recognize the temporal dynamics of indoor environmental data. Likewise, the architecture of a Bi-GRU consists of two GRU layers, which fit precisely like a Bi-LSTM. The bidirectional nature of the model allows it to fetch information from each beyond and future contexts concurrently to learn complex representations of the enter data [24, 25]. The mathematical representation of Bi-LSTM is shown in the form of forward (Eqs. (16)–(21)) as well as backward direction (Eqs. (22)–(27)). Similarly, for Bi-GRU, the forward GRU is represented in Eqs. (28)–(31), and the backward GRU is in Eqs. (32)–(35). Ultimately, the output of both networks at timestamp  $t$  is computed using Eq. (36). The superscripts of  $f$  and  $b$  represent forward and backward.

$$\text{Forget gate}(f_t^{(f)}) = \sigma(W_f^{(f)} \cdot [h_{t-1}^{(f)}, x_t] + b_f^{(f)}), \quad (16)$$

$$\text{Input gate}(i_t^{(f)}) = \sigma(W_i^{(f)} \cdot [h_{t-1}^{(f)}, x_t] + b_i^{(f)}), \quad (17)$$

$$\text{Output gate}(o_t^{(f)}) = \sigma(W_o^{(f)} \cdot [h_{t-1}^{(f)}, x_t] + b_o^{(f)}), \quad (18)$$

$$\text{Candidate cell state}(\tilde{c}_t^{(f)}) = \tanh(W_c^{(f)} \cdot [h_{t-1}^{(f)}, x_t] + b_c^{(f)}), \quad (19)$$

$$\text{Cell state update}(c_t^{(f)}) = f_t^{(f)} \cdot c_{t-1}^{(f)} + i_t^{(f)} \cdot \tilde{c}_t^{(f)}, \quad (20)$$

$$\text{Hidden state update}(h_t^{(f)}) = o_t^{(f)} \cdot \tanh(c_t^{(f)}), \quad (21)$$

$$\text{Forget gate}(f_t^{(b)}) = \sigma(W_f^{(b)} \cdot [h_{t-1}^{(b)}, x_t] + b_f^{(b)}), \quad (22)$$

$$\text{Input gate}(i_t^{(b)}) = \sigma(W_i^{(b)} \cdot [h_{t-1}^{(b)}, x_t] + b_i^{(b)}), \quad (23)$$

$$\text{Output gate}(o_t^{(b)}) = \sigma(W_o^{(b)} \cdot [h_{t-1}^{(b)}, x_t] + b_o^{(b)}), \quad (24)$$

$$\text{Candidate cell state}(\tilde{c}_t^{(b)}) = \tanh(W_c^{(b)} \cdot [h_{t-1}^{(b)}, x_t] + b_c^{(b)}), \quad (25)$$

$$\text{Cell state update}(c_t^{(b)}) = f_t^{(b)} \cdot c_{t-1}^{(b)} + i_t^{(b)} \cdot \tilde{c}_t^{(b)}, \quad (26)$$

$$\text{Hidden state update}(h_t^{(b)}) = o_t^{(b)} \cdot \tanh(c_t^{(b)}), \quad (27)$$

$$\text{Update gate}(z_t^f) = \sigma(W_z^f \cdot [h_{t-1}^f, x_t] + b_z^f), \quad (28)$$

$$\text{Reset gate}(r_t^f) = \sigma(W_r^f \cdot [h_{t-1}^f, x_t] + b_r^f), \quad (29)$$

$$\text{Candidate hidden state}(\tilde{h}_t^f) = \tanh(W_h^f \cdot [r_t^f \odot h_{t-1}^f, x_t] + b_h^f), \quad (30)$$

$$\text{Hidden state update}(h_t^f) = (1 - z_t^f) \odot h_{t-1}^f + z_t^f \odot \tilde{h}_t^f, \quad (31)$$

$$\text{Update gate}(z_t^b) = \sigma(W_z^b \cdot [h_{t-1}^b, x_t] + b_z^b), \quad (32)$$

$$\text{Reset gate}(r_t^b) = \sigma(W_r^b \cdot [h_{t-1}^b, x_t] + b_r^b), \quad (33)$$

$$\text{Candidate hidden state}(\tilde{h}_t^b) =$$

$$\tanh(W_h^b \cdot [r_t^b \odot h_{t-1}^b, x_t] + b_h^b), \quad (34)$$

$$\text{Hidden state update}(h_t^b) = (1 - z_t^b) \odot h_{t-1}^b + z_t^b \odot \tilde{h}_t^b, \quad (35)$$

$$h_t = [h_t^{(f)}, h_t^{(b)}]. \quad (36)$$

A stacked LSTM architecture consists of multiple LSTM layers, which are stacked on top of each other. Here, each LSTM layer processes the input data sequentially, where the output of one layer is served as the input to the next layer. When multiple LSTM layers are stacked, it allows the model to capture both short-term and long-term dependencies in the data to make it more capable of capturing the complex dynamics of indoor environmental variables. Likewise, the Stacked GRU operates on the same concept as the Stacked LSTM. However, it uses GRU units. In this structure, various GRU layers are stacked on peak of every different layer to create a deep network. Every layer in a Stacked GRU architecture comprises a series of GRU units, each with its own set of parameters to analyze the patterns and relationships in the input data. The output of one GRU layer serves as the input to the subsequent layer, allowing the network to learn hierarchical representations of information across multiple stages of abstraction [26, 27].

In addition, the hyperparameter values used for training the models are also mentioned in Table II. Here, the learning rate is set to 0.001 to provide a stable balance between convergence speed and training stability. A batch size of 16 is chosen to allow efficient gradient amendments while sustaining manageable memory usage. The training process for the datasets is conducted over 15 epochs to ensure convergence without significant overfitting. The Adam optimizer is used due to its consistent learning rate capability, which speeds up convergence compared to traditional stochastic gradient descent. Then, a dropout rate of 0.5 is chosen to apply in the hidden layers so that the problem of overfitting can be addressed. The ReLU activation function is chosen for hidden layers due to its efficiency in handling non-linearity and avoiding vanishing gradients. Softmax activation is selected in the output layer to enable probabilistic multi-class classification.

Last, there are performance metrics. In the context of energy efficiency for smart home appliances based on the IAQ index and AQL, various key metrics are



TABLE II  
HYPERPARAMETERS AND THEIR SELECTED VALUES FOR  
MODEL TRAINING.

Hyperparameter	Value
Learning Rate	0.001
Batch Size	16
Epochs	15
Optimizer	Adam
Dropout Rate	0.5
Activation Layer	ReLU in hidden, Softmax in output

typically used to evaluate the performance of classification models. Accuracy (Eq. (37)) provides an overall measure of model performance by calculating the proportion of correctly classified instances [28]. It has True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Loss (Eq. (38)) quantifies the difference between predicted and actual values. It is often measured using metrics like cross-entropy, which serves as a gauge of model optimization and convergence [29]. In addition to these metrics, there are other measures, such as Precision (Eq. (39)). It is crucial to assess the ability of the model to avoid false positives to identify the equipment that is responsible for poor air quality. It is a measure of the proportion of correctly predicted positive cases among all predicted positive cases. Meanwhile, Recall (Eq. (40)) computes the ability of the model, which captures all actual positive cases that are correctly identified. Then, the F1-Score (Eq. (41)) is most useful when there is no synchronization between positive and negative instances, allowing models to balance their performance based on these metrics [30, 31].

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (37)$$

$$\text{Loss} = \frac{(\text{Actual Value} - \text{Predicted Value})^2}{\text{Number of observations}}, \quad (38)$$

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (39)$$

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (40)$$

$$\text{F1-Score} = 2 \frac{\text{Precision} * \text{Recall}}{\text{Recall} + \text{Precision}}. \quad (41)$$

### III. RESULTS AND DISCUSSION

#### A. Analysis of Models for Data I in Room 415

Table III presents the performance metrics of various neural network models, including MLP, RNN, GRU, LSTM, and combinations of these models. They are evaluated on training and validation datasets. The metrics considered are accuracy and loss, where higher accuracy and lower loss values indicate better performance.

Notably, during the training phase, the MLP model achieves the best accuracy of 99.53% with the lowest loss of 0.0146, making it the best performer in terms of training metrics compared to the other models. RNN model achieves the accuracy of 99.44% with the loss of 0.0169 in terms of training metrics. Bidirectional versions of GRU and LSTM additionally show strong overall performance, with Bidirectional LSTM slightly outperforming Bidirectional GRU in terms of lowest loss (0.0148 vs. 0.0152) and marginally better accuracy (99.47% vs. 99.46%). This result shows that the bidirectional nature of those models efficaciously captures temporal dependencies in both directions, thereby enhancing their learning capability. Meanwhile, the Stacked LSTM and Stacked GRU architectures achieve accuracies of 99.40% and 99.45%, respectively. However, the values are slightly lower than their non-stacked counterparts (LSTM and GRU, with accuracies of 99.41% and 99.49%, respectively). The results indicate that deeper networks no longer necessarily enhance performance for this dataset.

During the validation phase, the Bidirectional GRU and LSTM models achieve exceptionally high validation accuracy (99.81% and 99.72%, respectively) with low loss values (0.0081 and 0.0066, respectively). The results indicate strong generalization capabilities due to their ability to capture temporal dependencies in both directions. The MLP achieves good performance with a validation accuracy of 99.16% and a loss of 0.0200. However, they are slightly underperformed by RNN-based models with a validation accuracy of 99.75% and a loss of 0.0061 due to their limited ability to capture sequential dependencies. LSTM and GRU models perform well, with accuracy of 99.57% and 99.79% and losses of 0.0104 and 0.0082, respectively. The results underscore their effectiveness for sequential data. Meanwhile, the Stacked LSTM and Stacked GRU architectures exhibit solid performance but do not significantly surpass their single-layer counterparts, achieving accuracies of 99.56% and 99.51%, respectively.

Figures A3–A5 in Appendix illustrates the learning curves of the models for training and validation accuracy, as well as loss, over 15 epochs. It has been found that MLP and RNN exhibit a good fit in their learning curves, while the remaining models display some zig-zag movement, indicating minor fluctuations in their performance. In addition, the curve of validation loss is lower than the training loss. It indicates that the validation dataset may be easier for the model to predict than the training dataset. Likewise, the curve of validation accuracy is higher than training accuracy from the beginning of the epoch. The validation dataset

TABLE III  
ACCURACY AND LOSS METRICS FOR TRAINING AND VALIDATION PHASES OF VARIOUS MODELS APPLIED TO DATA I.

Model	Training		Validation	
	Accuracy	Loss	Accuracy	Loss
Multilayer Perceptron (MLP)	99.53	0.0146	99.16	0.0200
Recurrent Neural Network (RNN)	99.44	0.0169	99.75	0.0061
Bidirectional GRU	99.46	0.0152	99.81	0.0081
Bidirectional LSTM	99.47	0.0148	99.72	0.0066
Long Short-Term Memory (LSTM)	99.41	0.0166	99.57	0.0104
Gated Recurrent Unit (GRU)	99.49	0.0149	99.79	0.0082
Stacked LSTM	99.40	0.0173	99.56	0.0102
Stacked GRU	99.45	0.0162	99.51	0.0156

TABLE IV  
ANALYSIS OF MODELS FOR ENERGY EFFICIENT HOME APPLIANCES IN DATA I.

Model	Precision	Recall	F1-Score
Multilayer Perceptron (MLP)	0.9971	0.9915	0.9914
Recurrent Neural Network (RNN)	0.9975	0.9975	0.9974
Bidirectional GRU	0.9980	0.9982	0.9979
Bidirectional LSTM	0.9972	0.9972	0.9972
Long Short-Term Memory (LSTM)	0.9957	0.9957	0.9956
Gated Recurrent Unit (GRU)	0.9979	0.9980	0.9979
Stacked LSTM	0.9955	0.9955	0.9954
Stacked GRU	0.9951	0.9950	0.9950

may be easier or have a different distribution than the training set, which leads to higher performance.

Table IV presents Precision, Recall, and F1-Score for various neural network models, providing a comprehensive evaluation of their performance. After comparing the performance of all the applied classifiers, it has been found that Bidirectional GRU computes the highest scores for Precision, Recall, and F1-Score, with values of 0.9980, 0.9982, and 0.9979, respectively. GRU follows the result with Precision of 0.9979, Recall of 0.9980, and F1-Score of 0.9979. This performance depicts that these models can classify the instances correctly. RNN, LSTM, and Bidirectional LSTM also indicate their effectiveness by generating the balanced values of Precision (0.9975, 0.9957, and 0.9972, respectively), Recall (0.9975, 0.9957, and 0.9972, respectively), and F1-Score (0.9974, 0.9956, and 0.9972, respectively). Likewise, Stacked LSTM and Stacked GRU also perform similarly by maintaining a balance between Precision, Recall, and F1-Score with their generated values. However, it has been observed that MLP generates the least value in terms of Recall and F1 Score, with values of 0.9915 and 0.9914, respectively. The results show that the MLP requires further improvement to enhance its performance for this specific dataset.

A  $3 \times 3$  confusion matrix has also been created in Figs. A6 and A7 in Appendix. This matrix provides a class-wise distribution of correctly and incorrectly

predicted samples across the Low, Average, and Severe classes. The confusion matrix enables a deeper understanding of the strengths and weaknesses of each model by highlighting the distribution of true positives, true negatives, false positives, and false negatives across all the categories, rather than overall accuracy, which combines the performance into a single metric.

It is observed from the analysis that all models show very high accuracy in predicting the Low and Severe classes, with very minimal misclassifications. The results point out that these two classes are relatively well-separated in the feature space, allowing the models to identify them with high confidence. However, minor misclassifications are noted in the Average class. Such behavior is expected, as the Average air quality range often shares characteristics with its neighboring classes, making it harder to distinguish.

In the applied models, the inference is that RNN and the gated variants, such as GRU and LSTM, exhibit little misclassification in the Average class compared to MLP, thus confirming the advantage of sequence-based architectures in handling temporal variations in indoor environmental data. The Bidirectional GRU and Bidirectional LSTM further improve class separation by incorporating contextual information from both past and future time steps, reducing confusion between adjacent classes. Therefore, the conclusion is that the confusion matrix analysis validates the robustness of all the models, highlighting the comparative superiority of recurrent architectures in minimizing class overlaps.

Table V presents the performance metrics of models across three different classes of smart home datasets, Low, Average, and Severe, based on Precision, Recall, and F1-Score, to evaluate their effectiveness in classifying data. It has been observed that in the case of the Low class, all models have performed well, scoring perfect precision scores of 1.0000. It means that every positive prediction made by the model is indeed a TP, with slight variations among the Recall and F1-Score values. This variation indicates that models identify TP with differing success rates.

TABLE V  
CLASS-WISE ANALYSIS OF MODELS FOR DATA I BY SHOWING PRECISION, RECALL, AND F1-SCORE VALUES ACROSS LOW, AVERAGE, AND SEVERE AIR QUALITY CLASSES.

Model	Class	Precision	Recall	F1-Score
Multilayer Perceptron (MLP)	Low	10.000	0.9918	0.9958
	Average	0.9751	1.0000	0.9873
	Severe	1.0000	0.9829	0.9913
Recurrent Neural Network (RNN)	Low	1.0000	0.9980	0.9989
	Average	0.9980	0.9946	0.9962
	Severe	0.9945	1.0000	0.9972
Bidirectional GRU	Low	1.0000	0.9951	0.9975
	Average	0.9951	0.9995	0.9972
	Severe	0.9991	1.0000	0.9990
Bidirectional LSTM	Low	1.0000	0.9965	0.9982
	Average	0.9965	0.9951	0.9958
	Severe	0.9951	1.0000	0.9975
Long Short-Term Memory (LSTM)	Low	1.0000	0.9967	0.9983
	Average	0.9967	0.9904	0.9935
	Severe	0.9905	1.0000	0.9952
Gated Recurrent Unit (GRU)	Low	1.0000	0.9948	0.9973
	Average	0.9941	0.9997	0.9968
	Severe	0.9997	0.9997	0.9997
Stacked LSTM	Low	1.0000	0.9950	0.9974
	Average	0.9949	0.9917	0.9932
	Severe	0.9917	1.0000	0.9958
Stacked GRU	Low	1.0000	0.9959	0.9979
	Average	0.9959	0.9892	0.9925
	Severe	0.9895	1.0000	0.9947

Some models may be missing more TP (higher FN) than others. For this class, the highest Recall and F1-Score values are achieved by the RNN, with 0.9980 and 0.9989, respectively. It is followed by LSTM and bidirectional LSTM, with Recall scores of 0.9967 and 0.9965, respectively, and F1-Score values of 0.9983 and 0.9982, respectively. However, MLP shows a high number of FN and misses many TP by computing the lowest Recall and F1-Score values of 0.9918 and 0.9958, respectively.

In the case of the Average class of AQL, only MLP achieves a perfect recall of 1.0000, compared to other models. The result indicates that MLP correctly identifies all the true positive cases without missing any data. While examining the rest of the parameters, such as Precision, the highest value is computed by RNN with 0.9980. It is followed by LSTM and Bidirectional LSTM, with 0.9967 and 0.9965, respectively. The results define the TP prediction made. Likewise, for Recall and F1-Score, the highest values are achieved by GRU with 0.9997 and 0.9968, respectively, as well as Bidirectional GRU with 0.9995 and 0.9972, respectively. Then, the lowest values for all performance metrics are obtained by GRU in the case of Precision, with 0.9941, and by Stacked GRU in terms of Recall and F1-Score, with 0.9892 and 0.9925, respectively.

For the Severe class, only MLP computes the perfect Precision score, while the rest of the models, such as

RNN, Bidirectional GRU, Bidirectional LSTM, LSTM, Stacked LSTM, and Stacked GRU, obtain the perfect Recall scores of 100%. GRU maintains the balance relationship between the metrics by computing 0.9997. It means a very low number of FP and FN in the Severe class. In addition, Stacked GRU achieves the lowest Precision value of 0.9895, while MLP has the least Recall and F1-Score values, indicating room for improvement.

#### B. Analysis of Models for Data II in Room 776

Table VI presents a comparative analysis of various deep learning models based on their training accuracy and loss for the data collected from Room 776. In the training phase, the Bidirectional GRU achieves the highest training accuracy of 99.27%, closely followed by the RNN with 99.26% and the GRU with 99.24%. The results suggest that these recurrent-based models are particularly effective in capturing the patterns in the training data. In terms of training loss, which indicates how well the model fits the training data (with lower values being better), the Bidirectional GRU also performs the best, with the lowest loss of 0.0238. The RNN follows this result with a loss of 0.0240, and the GRU with 0.0247. These low-loss values, in conjunction with their high accuracies, suggest that these models not only learn well but also generalize effectively on the training data without overfitting.

TABLE VI  
ACCURACY AND LOSS VALUES FOR TRAINING AND VALIDATION PHASES OF NEURAL NETWORK MODELS APPLIED TO DATA II.

Model	Training		Validation	
	Accuracy	Loss	Accuracy	Loss
Multilayer Perceptron (MLP)	99.17	0.0276	99.58	0.0140
Recurrent Neural Network (RNN)	99.26	0.0240	99.43	0.0158
Bidirectional GRU	99.27	0.0238	99.41	0.0151
Bidirectional LSTM	99.14	0.0273	99.45	0.0130
Long Short-Term Memory (LSTM)	99.20	0.0255	99.47	0.0143
Gated Recurrent Unit (GRU)	99.24	0.0247	99.63	0.0118
Stacked LSTM	99.13	0.0281	99.64	0.0117
Stacked GRU	99.16	0.0264	99.34	0.0177

TABLE VII  
ANALYSIS OF MODELS FOR ENERGY EFFICIENT HOME APPLIANCES IN DATA II.

Model	Precision	Recall	F1-Score
Multilayer Perceptron (MLP)	0.9957	0.9957	0.9957
Recurrent Neural Network (RNN)	0.9943	0.9943	0.9942
Bidirectional GRU	0.9940	0.9940	0.9939
Bidirectional LSTM	0.9945	0.9945	0.9945
Long Short-Term Memory (LSTM)	0.9920	0.9973	0.9946
Gated Recurrent Unit (GRU)	0.9963	0.9963	0.9962
Stacked LSTM	0.9959	0.9964	0.9961
Stacked GRU	0.9961	0.9966	0.9949

On the other hand, models like the Stacked LSTM and Stacked GRU, while still achieving high accuracy (99.13% and 99.16%, respectively), show slightly higher loss values (0.0281 and 0.0264, respectively) compared to their simpler counterparts. It can imply that the added complexity of stacking layers does not necessarily translate to better performance on the training data and may even lead to marginally increased training loss.

From the validation data, all models show high performance, with accuracy ranging from 99.34% to 99.64%, and low loss values between 0.0117 and 0.0177. The Stacked LSTM model achieves the highest validation accuracy at 99.64%, closely followed by the GRU with 99.63%, indicating that these models generalize particularly well to unseen data. This high performance suggests that these models effectively capture the underlying patterns in the dataset. In terms of validation loss, which measures how well the model fits the validation data (with lower values indicating better performance), the Stacked LSTM again performs the best, with a loss of 0.0117, closely followed by the GRU with a loss of 0.0118. These low-loss values reinforce the superior generalization capability of these models. Interestingly, the MLP model, despite its simpler architecture, achieves a very high validation accuracy of 99.58% and a low loss of 0.0140. It is also a strong performer and can be considered a competitive model for this task. The RNN, Bidirec-

tional GRU, and Bidirectional LSTM models show slightly lower validation accuracies (99.43%, 99.41%, and 99.45%, respectively) and higher losses (0.0158, 0.0151, and 0.0130, respectively) compared to the GRU and Stacked LSTM. The results suggest that while they are still effective, they may not capture the validation data patterns as well as the GRU-based models and the Stacked LSTM. The Stacked GRU model achieves 99.34% validation accuracy at a higher loss of 0.0177. The results indicate the possibility of overfitting or difficulty in effectively training the deeper architectures.

Similar to the case of Data I, the learning curves for the models' training and validation accuracy, as well as loss, over 15 epochs are defined in Figs. A8–A10 in Appendix. According to the research findings, it has been observed that MLP and RNN models exhibit a good fit of learning curves. Meanwhile, in the case of Bidirectional GRU, a peak can be seen, which indicates that the model may not have learned enough to make accurate predictions. However, as the training progresses, the model learns to capture more complex patterns in the data, which leads to its improved performance.

Table VII analyzes the performance of various deep learning models based on their Precision, Recall, and F1-Score. GRU and Stacked GRU indicate the effectiveness in minimizing the false positives by obtaining the highest precision values of 0.9963, 0.9961, and 0.9959, respectively. The results are followed by Stacked LSTM and MLP, with scores of 0.9959 and 0.9957 respectively. Models, like Bi-Directional GRU, RNN, Bi-Directional GRU and LSTM, have very low precision values. On the other hand, in terms of Recall, LSTM, with a value of 0.9973, reflects its ability to identify true positives, albeit at the expense of lower precision, 0.9920. It ultimately computes 0.9946 as the F1-Score. Apart from LSTM, Stacked GRU and Stacked LSTM also demonstrate their efficiency in predicting actual instances, achieving top Recall values of 0.9966, 0.9964, and 0.9963, respectively.

TABLE VIII  
EVALUATION OF MODEL PERFORMANCE IN DISTINGUISHING AIR QUALITY LEVEL (AQL) (Low, Average, AND SEVERE) IN DATA II.

Model	Class	Precision	Recall	F1-Score
Multilayer Perceptron (MLP)	Low	1.0000	0.9911	0.9955
	Average	0.9912	0.9961	0.9936
	Severe	0.9961	1.0000	0.9980
Recurrent Neural Network (RNN)	Low	1.0000	0.9918	0.9958
	Average	0.9918	0.9911	0.9914
	Severe	0.9912	1.0000	0.9955
Bidirectional GRU	Low	1.0000	0.9905	0.9952
	Average	0.9903	0.9916	0.9909
	Severe	0.9918	1.0000	0.9958
Bidirectional LSTM	Low	1.0000	0.9931	0.9966
	Average	0.9932	0.9904	0.9918
	Severe	0.9903	1.0000	0.9951
Long Short-Term Memory (LSTM)	Low	0.9922	10.000	0.9961
	Average	0.9921	0.9919	0.9919
	Severe	0.9919	1.0000	0.9959
Gated Recurrent Unit (GRU)	Low	1.0000	0.9913	0.9956
	Average	0.9914	0.9977	0.9945
	Severe	0.9976	1.0000	0.9987
Stacked LSTM	Low	1.0000	0.9914	0.9957
	Average	0.9913	0.9979	0.9945
	Severe	0.9964	1.0000	0.9982
Stacked GRU	Low	1.0000	0.9900	0.9949
	Average	0.9898	0.9999	0.9948
	Severe	0.9902	1.0000	0.9950

Then, they have an F1-score of 0.9949, 0.9961, and 0.9962, respectively. The MLP model also demonstrates its robustness by achieving a good F1-Score of 0.9957, while Bidirectional GRU, RNN and Bidirectional LSTM generate the lowest values of Precision and Recall at 0.9940, 0.9943 and 0.9945, respectively. It suggests that the models may not perform as consistently well as the other models.

A  $3 \times 3$  confusion matrix for Data II is generated to analyze the class-wise performance of the models across Low, Average, and Severe classes. The matrix provides a precise view that how accurately each model predicts individual classes, moving beyond aggregate performance metrics, such as accuracy and loss. By segmenting the predictions into true positives, false positives, true negatives, and false negatives, the confusion matrix allows a more transparent evaluation of strengths and weaknesses of the models as shown in Figs. A11 and A12 (see Appendix).

The results project that all models maintain a high level of accuracy across all the three classes, with very strong performance in predicting the Low and Severe air quality categories. Like observed in Data I, in here, the Average category possesses greater challenges, as it is more prone to overlap with neighboring classes. In some cases, misclassifications occur when Average samples are inaccurately predicted as either Low or Severe.

When the models are compared, the RNN and its gated extensions, like GRU and LSTM, show better handling of the Average class compared to the MLP. It focuses on the advantage of sequence-based learning in capturing subtle temporal variations in environmental data. Furthermore, the Bidirectional GRU and Bidirectional LSTM show the improved discrimination across all classes by utilizing both past and future contextual information that helped to reduce overlaps between adjacent categories. It is concluded that the confusion matrix analysis for Data II validates that recurrent models, especially the bidirectional versions, outperform the MLP by providing more consistent and balanced classification across all AQLs. This further proves the claim that temporal modeling is crucial for accurate energy-efficient indoor environment management.

Table VIII analyzes the performance of applied classifiers based on three different classes of dataset (Low, Average, and Severe) using evaluation metrics, such as Precision, Recall, and F1-Score. For the Low class, all models demonstrate their robustness in both classifying and identifying the classes perfectly. They obtain a perfect precision score of 100% except for LSTM, as this model computes the perfect Recall score. However, there are also a few classifiers that compute the lowest Recall values, like MLP (0.9911), RNN (0.9918), Bidirectional GRU (0.9905), GRU (0.9913), Stacked LSTM (0.9914), and Stacked GRU (0.9900). These

models are failing to correctly identify a significant number of true positive instances for this particular class. In addition, the Bidirectional LSTM shows slight variations, with a Recall of 0.9931, resulting in an F1-Score of 0.9966. It depicts slightly better performance compared to others.

Likewise, in the case of the Average class, all models except Stacked GRU maintain high Precision, Recall, and F1-Score, typically above 0.99. However, models like the Bidirectional GRU, RNN, LSTM, and Bidirectional LSTM have slightly lower F1-Score of 0.9909, 0.9914, 0.9919, and 0.9918, respectively, due to their slight fall in performance in either Precision or Recall. The stacked GRU exhibits a highly sensitive nature in detecting instances of the average class, achieving the highest Recall value of 0.9999, but a lower Precision score of 0.9898. Ultimately, this leads to an F1-Score of 0.9948. Similarly, the lowest Recall values have also been computed by the models such as RNN (0.9911), Bidirectional GRU (0.9916), Bidirectional LSTM (0.9904), and LSTM (0.9919). It means that they are not able to classify the actual positive values correctly.

In the Severe class, all models again demonstrate very high performance, achieving a perfect Recall score of 1.0000. This characteristic indicates that the models can identify all instances of the Severe class correctly. In the case of the F1-Score, GRU and Stacked LSTM models stand out by computing the highest values of 0.9987 and 0.9982, respectively. Meanwhile, the other models, such as MLP and RNN, obtain slightly lower F1-Score (0.9980 and 0.9955, respectively). In addition, Stacked GRU and Bidirectional LSTM obtain the lowest Precision score of 0.9902 and 0.9903. The results indicate that these models are not able to predict the true positive classes and require room for improvement.

### C. Overall Results

Table IX provides a detailed analysis of the time frame taken by different deep learning models to process Data I and Data II. Because of its simple architecture, the MPL stands out with the shortest training time of 1 hour. The result indicates its faster computational efficiency compared to the other models. On the other hand, due to their deeper and complex architectures, the Stacked LSTM and Stacked GRU require longer training times of 1 hour and 40 minutes and 2 hours, respectively. Similarly, the Bidirectional LSTM and Bidirectional GRU models are computed in the longest training times of 2 hours and 5 minutes and 2 hours, respectively. Meanwhile, RNN, LSTM, and GRU models train in an average period of time, with

TABLE IX  
OVERALL EXECUTION TIME OF APPLIED LEARNING MODELS FOR DATA I AND DATA II TO UNDERScore THE IMPACT OF MODEL COMPLEXITY ON COMPUTATIONAL EFFICIENCY.

Model	Time frame
Long Short-Term Memory (LSTM)	1 hour, 25 minutes
Gated Recurrent Unit (GRU)	1 hour, 40 minutes
Stacked LSTM	1 hour, 40 minutes
Stacked GRU	2 hours
Multilayer Perceptron (MLP)	1 hour
Recurrent Neural Network (RNN)	1 hour, 30 minutes
Bidirectional GRU	2 hours
Bidirectional LSTM	2 hours, 5 minutes

training times of 1 hour and 30 minutes, 1 hour and 25 minutes, and 1 hour and 40 minutes, respectively.

Overall, in terms of practical application scenarios, the excellent performance of Bidirectional GRU and Stacked LSTM models in accuracy and loss can be applied to real-time energy optimization systems in smart homes. These models can dynamically adjust energy usage for heating, ventilation, and lighting based on IAQ parameters, such as CO<sub>2</sub> levels and temperature, ensuring energy efficiency while maintaining comfort. Additionally, insights from the learning curves, which highlight fluctuations in training and validation accuracy, can guide practical decisions in hyperparameter tuning during the deployment of models in real-world scenarios. The SMOTE-ENN technique for class balancing can be effectively applied in homes with uneven energy consumption patterns, such as those with varying seasonal appliance usage, ensuring that models handle such imbalances for accurate predictions. Lastly, the execution times of different models, detailed in Table IX, can inform decisions in scenarios where computational efficiency is crucial, such as real-time energy management systems requiring rapid model updates and predictions. These practical applications demonstrate how deep learning models can be directly translated into impactful and energy-saving solutions for residential buildings, contributing to sustainability and efficiency.

## IV. CONCLUSION

The research highlights significant advancements in applying deep learning techniques to improve energy efficiency in home appliances. By analyzing factors such as CO<sub>2</sub> levels, humidity, and temperature, the models demonstrate impressive results in optimizing energy consumption and promoting sustainability. Among the classifiers tested, Bidirectional GRU and Stacked LSTM outperform others in terms of accuracy and loss for data collected from Rooms 415 and 776. The results showcase the potential of artificial intelligence-driven approaches to revolutionize energy

management in smart homes. Furthermore, these models can optimize real-time decision-making in smart appliances, contributing to significant energy cost savings and a reduced environmental footprint. The research highlights the broader potential for integrating artificial intelligence-driven approaches into energy policies and sustainability strategies, enabling more effective reductions in residential energy consumption and combating climate change.

Nevertheless, the research's dependency on specific datasets and observed fluctuations during training and validation highlight limitations such as overfitting and constrained generalizability. Addressing these issues requires diversifying datasets, adjusting learning rates, increasing batch sizes, or employing regularization techniques to stabilize training and enhance model robustness. Future research should also explore advanced optimization methods, such as Adaptive Moment Estimation, Root Mean Square Propagation, and Evolutionary algorithms. Additionally, integrating IoT devices and real-time data processing can enhance responsiveness and scalability, bridging the gap between technological advancements and their practical deployment in sustainable energy systems.

#### AUTHOR CONTRIBUTION

Conceived and designed the analysis, J. S. S., S. A., and S. K.; Collected the data, J. S. S. and S. A.; Contributed data or analysis tools, J. S. S., S. A., and S. K.; Performed the analysis, J. S. S., S. A., and S. K.; and Wrote the paper, J. S. S.

#### DATA AVAILABILITY

The data that support the findings of the research are available on Kaggle. These data were derived from the following resources available in the public domain: <https://www.kaggle.com/datasets/ranakrc/smart-building-system>

#### REFERENCES

- [1] M. Santamouris and K. Vasilakopoulou, "Present and future energy consumption of buildings: Challenges and opportunities towards decarbonisation," *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, vol. 1, pp. 1–14, 2021.
- [2] T. M. Olatunde, A. C. Okwandu, and D. O. Akande, "Reviewing the impact of energy-efficient appliances on household consumption," *International Journal of Science and Technology*, vol. 6, no. 2, pp. 1–11, 2024.
- [3] U.S. Department of Energy, "Buildings & industry pillar." [Online]. Available: [www.energy.gov/eere/energy-efficiency-buildings-and-industry](http://www.energy.gov/eere/energy-efficiency-buildings-and-industry)
- [4] Energy Savers Network, "Simple home energy saving tips." [Online]. Available: <https://www.greenbuilt.org/energysaversnetwork/home-energy-saving-tips/>
- [5] W. Chen, M. Alharthi, J. Zhang, and I. Khan, "The need for energy efficiency and economic prosperity in a sustainable environment," *Gondwana Research*, vol. 127, pp. 22–35, 2024.
- [6] S. Karatzas, J. Merino, A. Puchkova, C. Mountzouris, G. Protopsaltis, J. Gialelis, and A. K. Parlikad, "A virtual sensing approach to enhancing personalized strategies for indoor environmental quality and residential energy management," *Building and Environment*, vol. 261, pp. 1–13, 2024.
- [7] A. Verma and Y. Kumar, "Study on machine learning based energy efficiency in developed countries," in *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. Palladam, India: IEEE, Oct. 7–9, 2020, pp. 895–899.
- [8] A. Abobakirov and O. Omonboev, "Energy efficient building materials in the design of buildings and structures," *Holders Reason*, vol. 1, no. 3, pp. 406–412, 2023.
- [9] M. Khalil, A. S. McGough, Z. Pourmirza, M. Pazhoohesh, and S. Walker, "Machine learning, deep learning and statistical analysis for forecasting building energy consumption—A systematic review," *Engineering Applications of Artificial Intelligence*, vol. 115, 2022.
- [10] Y. Xiang, Y. Chen, J. Xu, and Z. Chen, "Research on sustainability evaluation of green building engineering based on artificial intelligence and energy consumption," *Energy Reports*, vol. 8, pp. 11 378–11 391, 2022.
- [11] S. Park, "Machine learning-based cost-effective smart home data analysis and forecasting for energy saving," *Buildings*, vol. 13, no. 9, pp. 1–16, 2023.
- [12] J. Khan, M. Fayaz, U. Zaman, E. Lee, A. S. Balobaid, S. Ali, and K. Kim, "A hybrid machine learning and optimization algorithm for enhanced user comfort and energy efficiency in smart homes," 2024. [Online]. Available: <https://doi.org/10.20944/preprints202401.1331.v1>
- [13] L. Shree, S. Manjunatha, V. Vekariya, G. D. Benal, and H. Patil, "Efficient optimization of energy consumption at home through machine learning," in *2024 5<sup>th</sup> International Conference*



- on *Mobile Computing and Sustainable Informatics (ICMCSI)*. Lalitpur, Nepal: IEEE, Jan. 18–19, 2024, pp. 109–114.
- [14] M. Khan, J. Seo, and D. Kim, "Towards energy efficient home automation: a deep learning approach," *Sensors*, vol. 20, no. 24, pp. 1–18, 2020.
- [15] Q. Liao, M. Zhu, L. Wu, X. Pan, X. Tang, and Z. Wang, "Deep learning for air quality forecasts: A review," *Current Pollution Reports*, vol. 6, no. 4, pp. 399–409, 2020.
- [16] R. R. Chowdhury, "Smart building system," 2020. [Online]. Available: <https://www.kaggle.com/datasets/ranakrc/smart-building-system>
- [17] P. P. Kasaraneni, Y. Venkata Pavan Kumar, G. L. K. Moganti, and R. Kannan, "Machine learning-based ensemble classifiers for anomaly handling in smart home energy consumption data," *Sensors*, vol. 22, no. 23, pp. 1–20, 2022.
- [18] Y. R. Yoon, Y. R. Lee, S. H. Kim, J. W. Kim, and H. J. Moon, "A non-intrusive data-driven model for detailed occupants' activities classification in residential buildings using environmental and energy usage data," *Energy and Buildings*, vol. 256, 2022.
- [19] Y. Ding, L. Fan, and X. Liu, "Analysis of feature matrix in machine learning algorithms to predict energy consumption of public buildings," *Energy and Buildings*, vol. 249, 2021.
- [20] A. H. Alharbi, D. S. Khafaga, A. M. Zaki, E. S. M. El-Kenawy, A. Ibrahim, A. A. Abdelhamid, M. M. Eid, M. El-Said, N. Khodadadi, L. Abualigah, and M. A. Saeed, "Forecasting of energy efficiency in buildings using multilayer perceptron regressor with waterwheel plant algorithm hyperparameter," *Frontiers in Energy Research*, vol. 12, pp. 1–16, 2024.
- [21] C. Fan, J. Wang, W. Gang, and S. Li, "Assessment of deep recurrent neural network-based strategies for short-term building energy predictions," *Applied Energy*, vol. 236, pp. 700–710, 2019.
- [22] Y. Natarajan, S. P. K. R., G. Wadhwa, Y. Choi, Z. Chen, D. E. Lee, and Y. Mi, "Enhancing building energy efficiency with IoT-driven hybrid deep learning models for accurate energy consumption prediction," *Sustainability*, vol. 16, no. 5, pp. 1–22, 2024.
- [23] S. K. Singh, M. Kumar, S. Tanwar, and J. H. Park, "GRU-based digital twin framework for data allocation and storage in IoT-enabled smart home networks," *Future Generation Computer Systems*, vol. 153, pp. 391–402, 2024.
- [24] K. Attarde and J. Sayyad, "A CNN and BiLSTM fusion approach toward precise appliance energy forecasts," *International Journal of Intelligent Engineering & Systems*, vol. 17, no. 3, pp. 199–212, 2024.
- [25] D. Niu, M. Yu, L. Sun, T. Gao, and K. Wang, "Short-term multi-energy load forecasting for integrated energy systems based on CNN-BiGRU optimized by attention mechanism," *Applied Energy*, vol. 313, 2022.
- [26] M. A. Alghamdi, S. Abdullah, and M. Ragab, "Predicting energy consumption using Stacked LSTM snapshot ensemble," *Big Data Mining and Analytics*, vol. 7, no. 2, pp. 247–270, 2024.
- [27] T. Han, K. Muhammad, T. Hussain, J. Lloret, and S. W. Baik, "An efficient deep learning framework for intelligent energy management in IoT networks," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3170–3179, 2020.
- [28] I. Izonin, R. Tkachenko, S. A. Mitoulis, A. Faramarzi, I. Tsmots, and D. Mashtalir, "Machine learning for predicting energy efficiency of buildings: A small data approach," *Procedia Computer Science*, vol. 231, pp. 72–77, 2024.
- [29] L. D. Long, "An AI-driven model for predicting and optimizing energy-efficient building envelopes," *Alexandria Engineering Journal*, vol. 79, pp. 480–501, 2023.
- [30] K. Mayer, L. Haas, T. Huang, J. Bernabé-Moreno, R. Rajagopal, and M. Fischer, "Estimating building energy efficiency from street view imagery, aerial imagery, and land surface temperature data," *Applied Energy*, vol. 333, 2023.
- [31] N. Kapoor and Y. Kumar, "The efficient management of renewable energy resources for Vanet-Cloud communication," in *Nature-Inspired Computing Applications in Advanced Communication Networks*. IGI Global Scientific Publishing, 2020, pp. 228–253.

## APPENDIX

The Appendix can be seen in the next page.

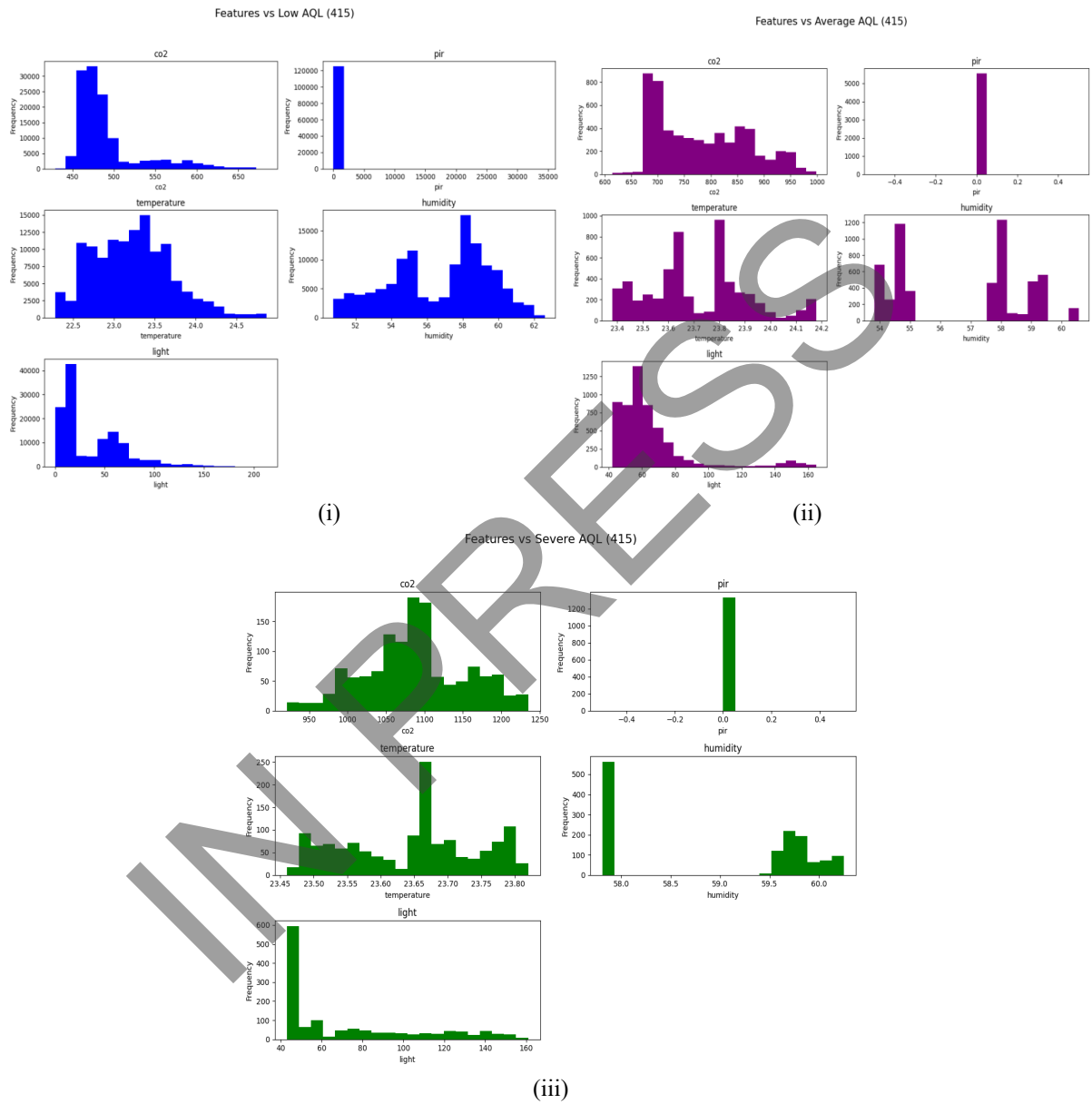


Fig. A1. Distribution of values of Data I features across various classes of Air Quality Level (AQL).

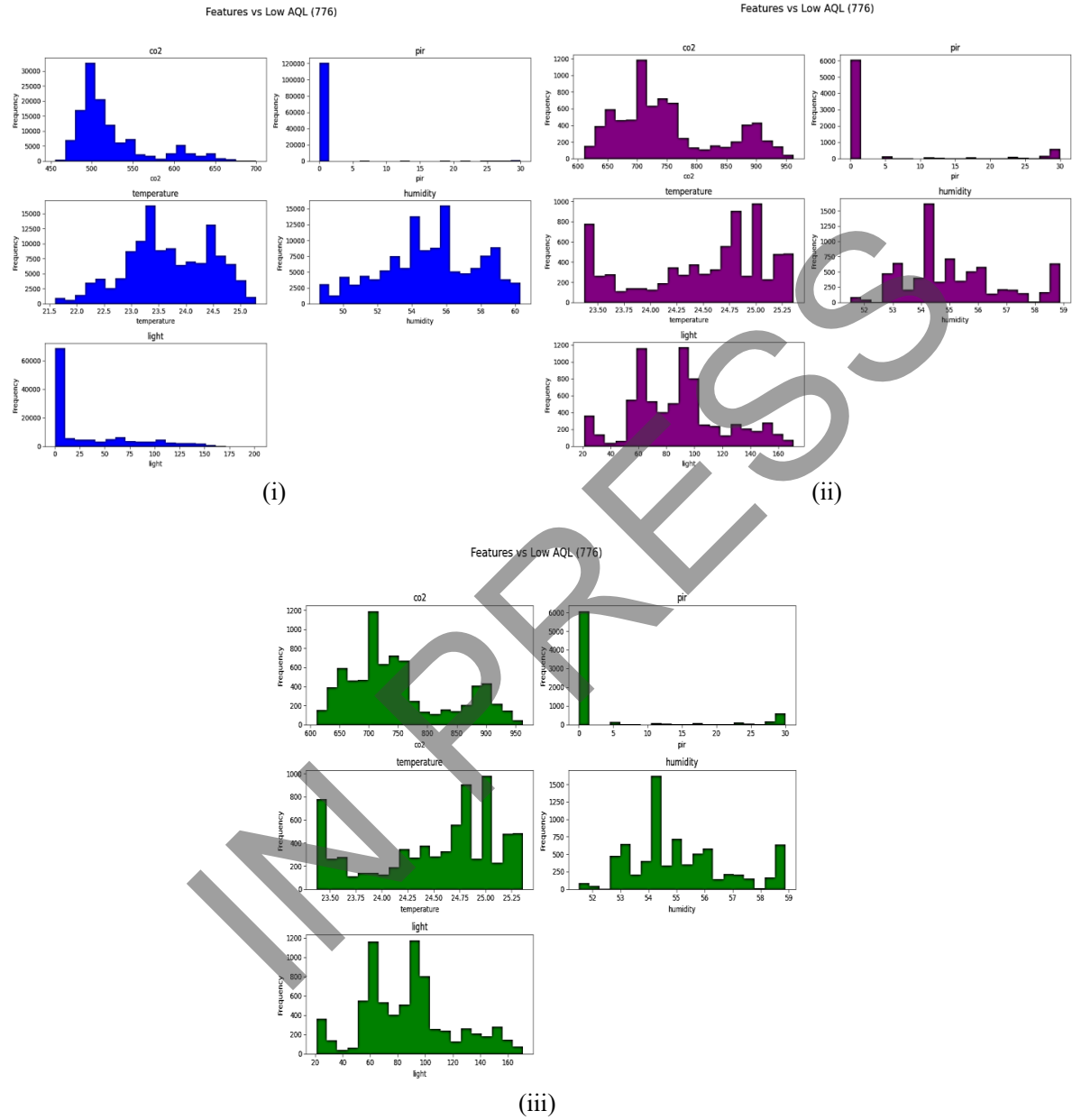


Fig. A2. Distribution of values of Data II features across various classes of Air Quality Level (AQL).

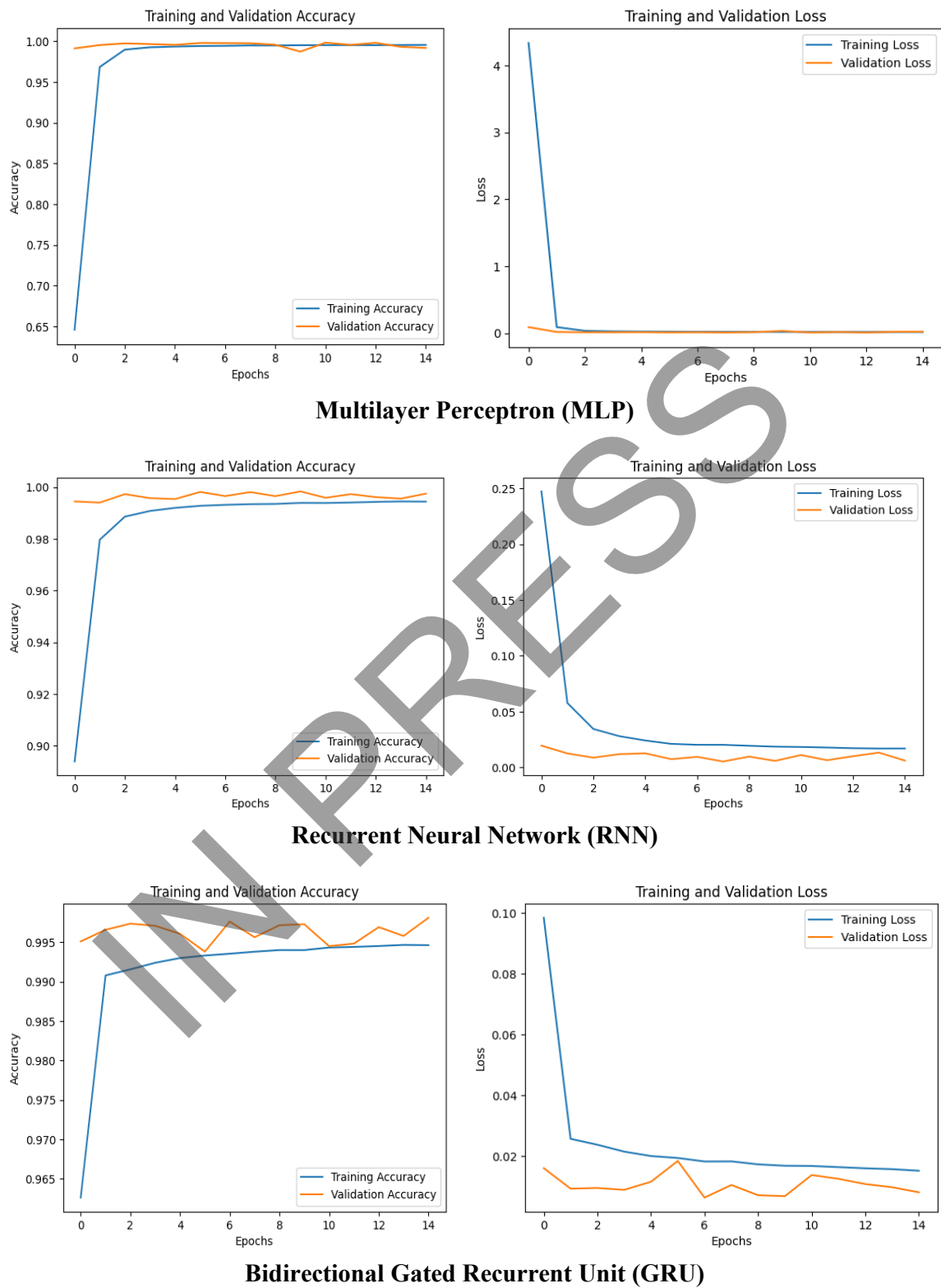


Fig. A3. Learning curves depicting accuracy and loss in the training and validation datasets for models applied to Data I to indicate convergence and generalization trends (1).



Fig. A4. Learning curves depicting accuracy and loss in the training and validation datasets for models applied to Data I to indicate convergence and generalization trends (2).

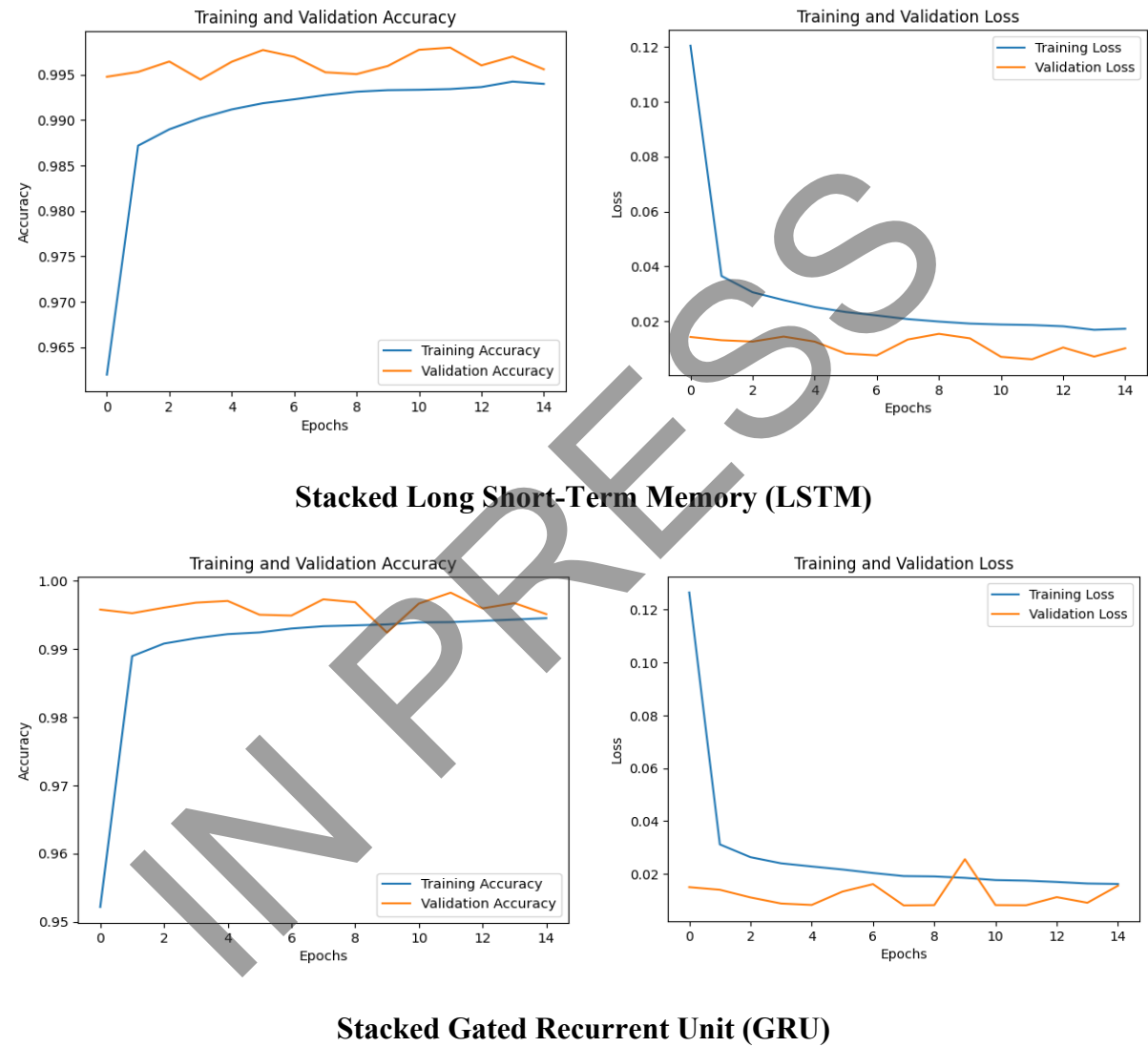


Fig. A5. Learning curves depicting accuracy and loss in the training and validation datasets for models applied to Data I to indicate convergence and generalization trends (3).

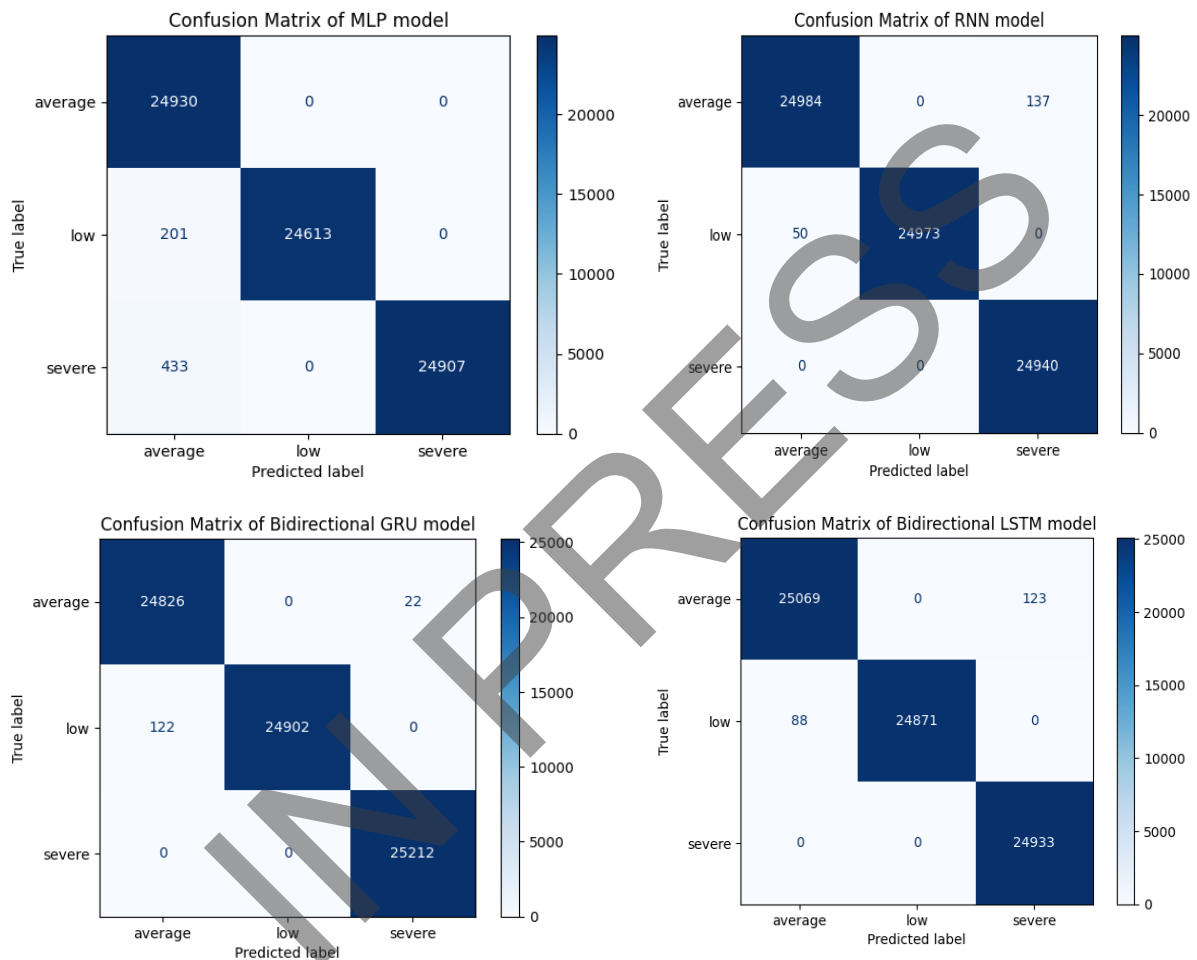


Fig. A6. Confusion matrix depicting the performance of models on Data I for three air quality classes: Low, Average, and Severe (1). Note: Multilayer Perceptron (MLP), Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), and Long Short-Term Memory (LSTM).



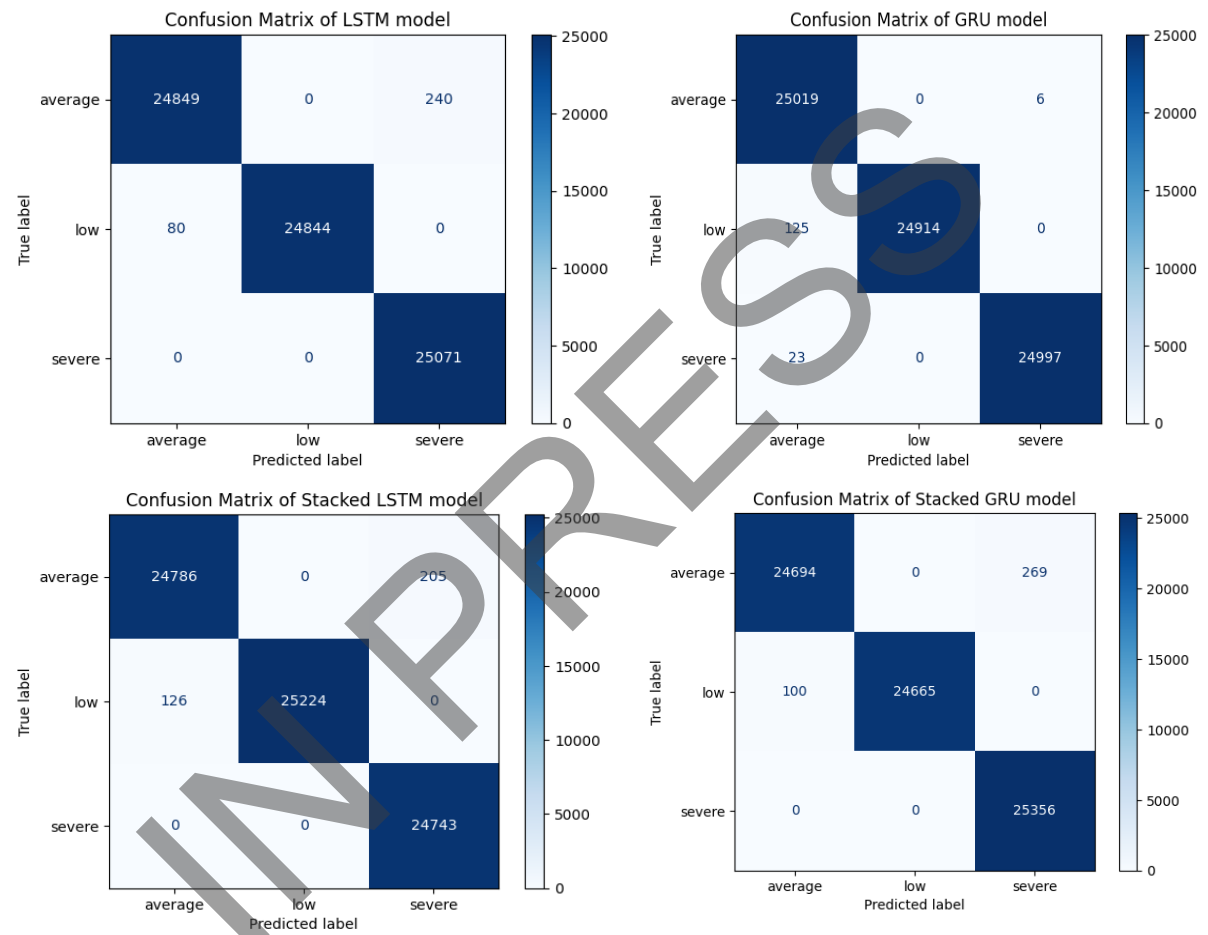


Fig. A7. Confusion matrix depicting the performance of models on Data I for three air quality classes: Low, Average, and Severe (2). Note: Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM).

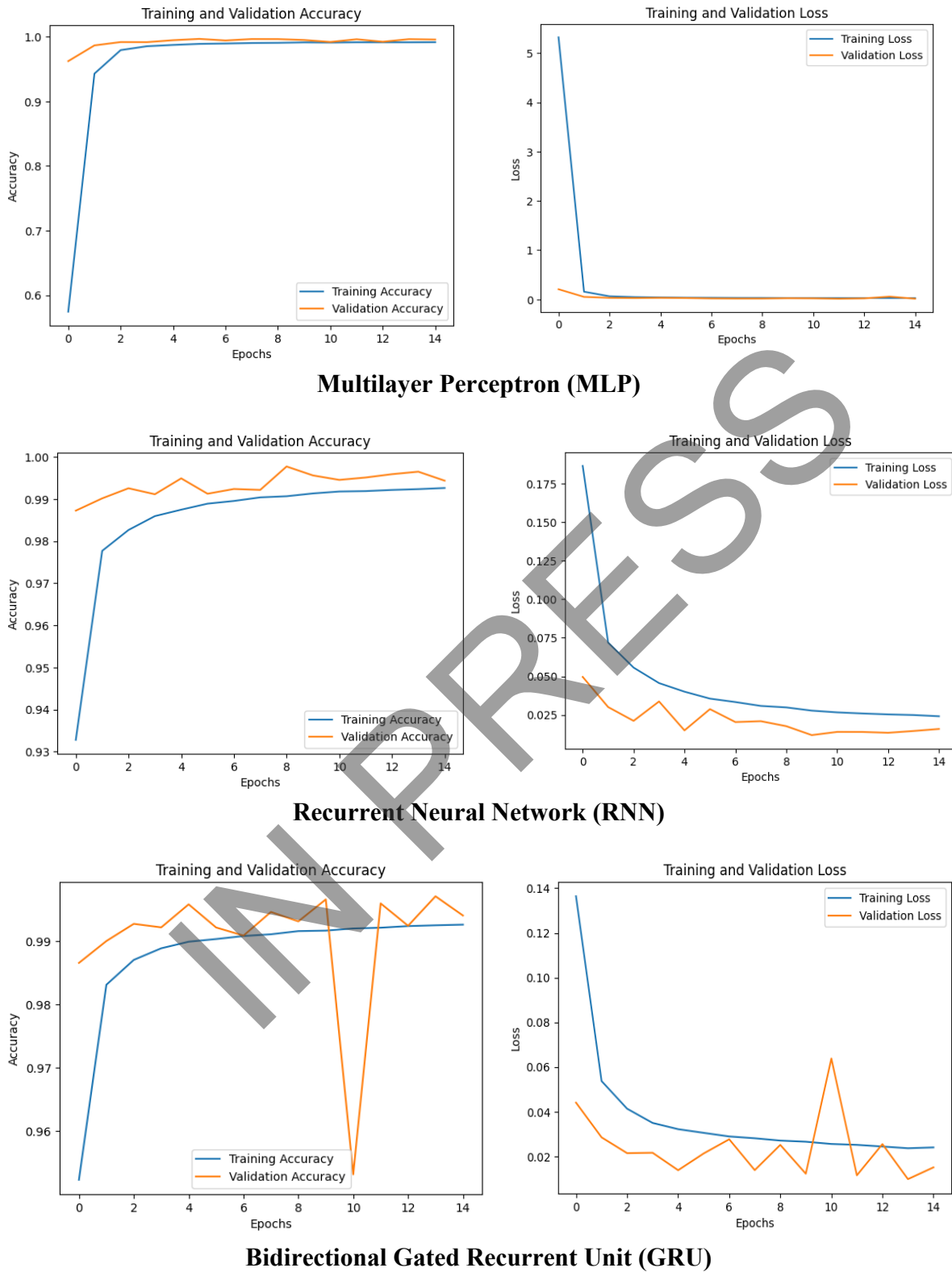


Fig. A8. Learning curves for models trained on Data II to highlight differences in performance between training and validation datasets (1).

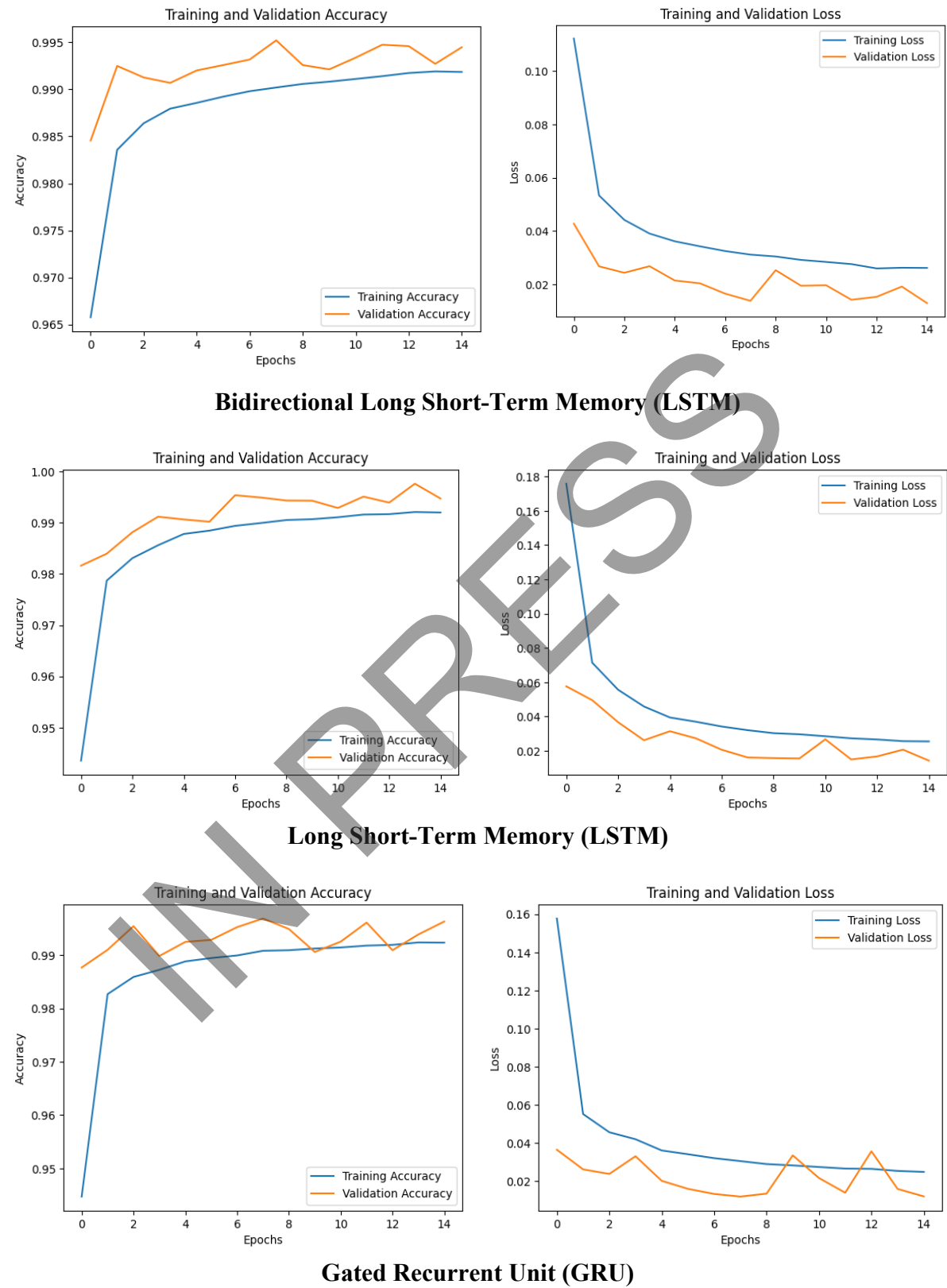


Fig. A9. Learning curves for models trained on Data II to highlight differences in performance between training and validation datasets (2).

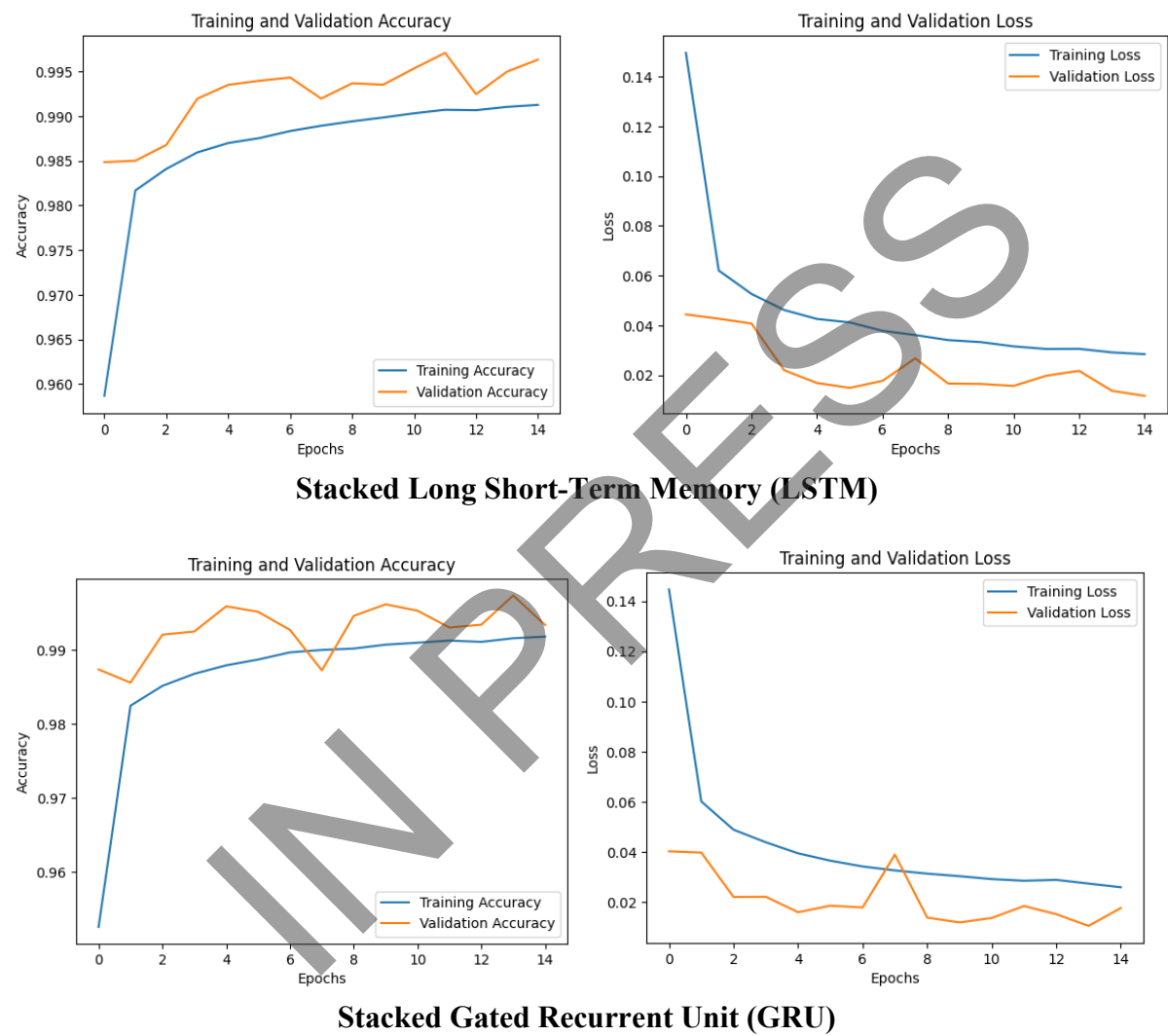


Fig. A10. Learning curves for models trained on Data II to highlight differences in performance between training and validation datasets (3).

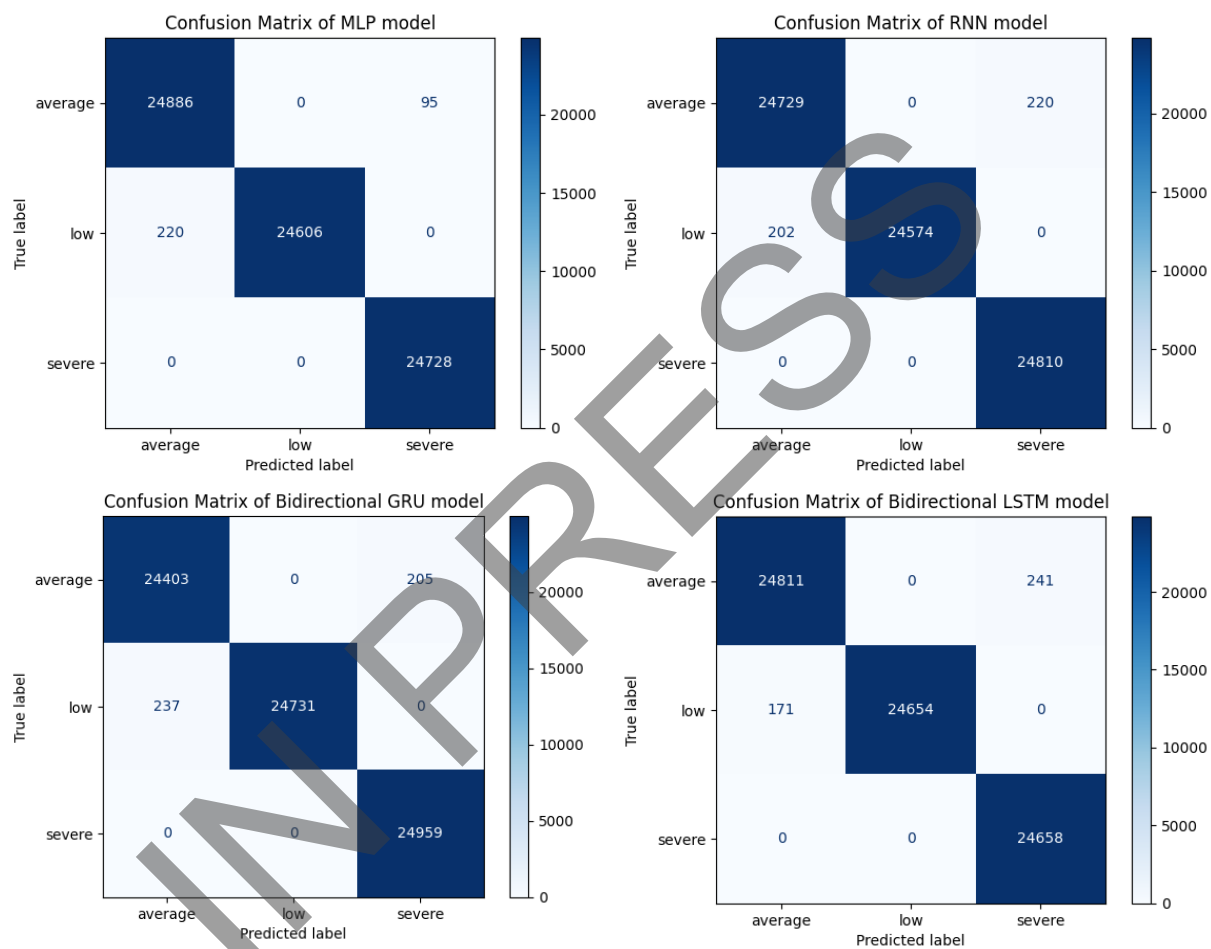


Fig. A11. Confusion matrix depicting the performance of models on Data II for three air quality classes: Low, Average, and Severe (1). Note: Multilayer Perceptron (MLP), Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), and Long Short-Term Memory (LSTM).

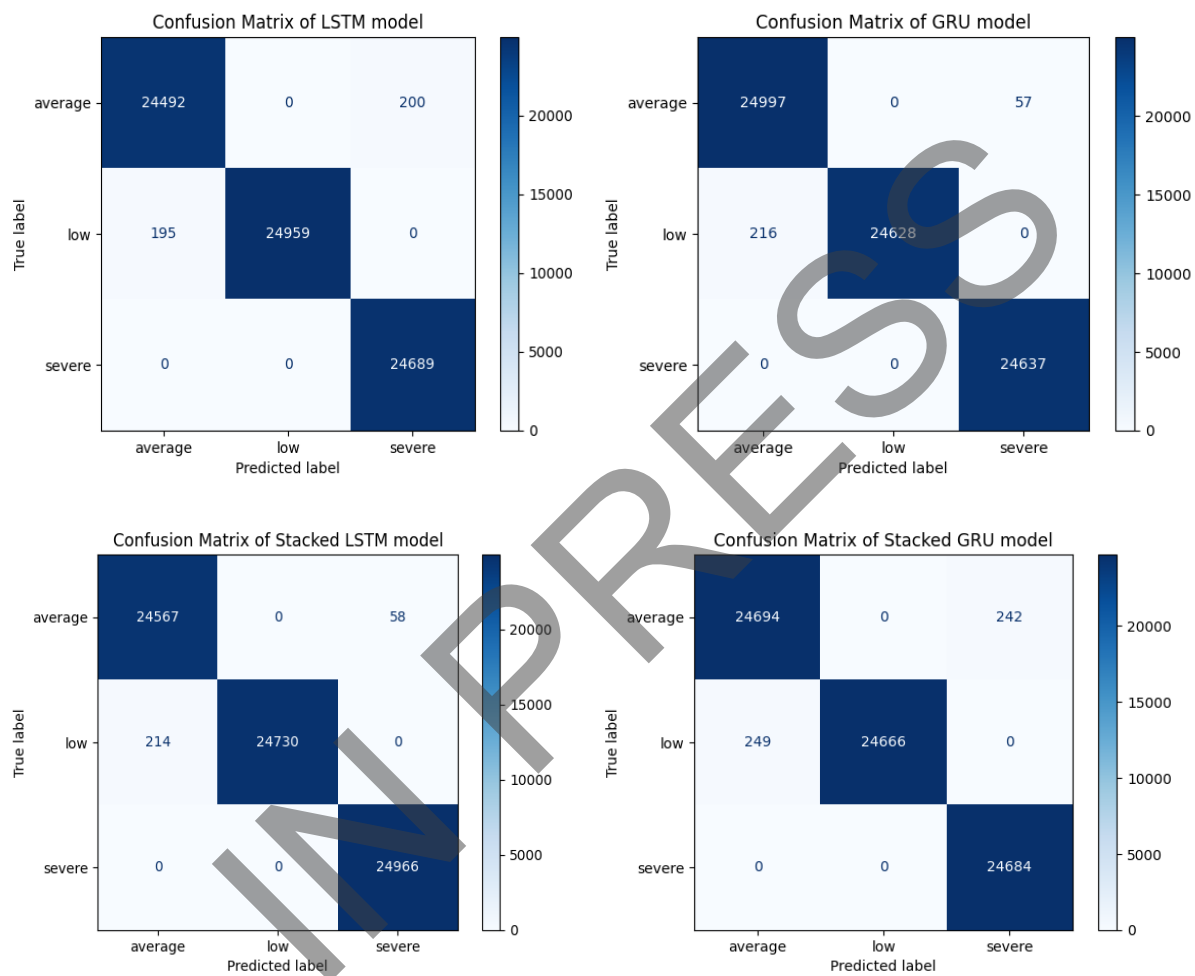


Fig. A12. Confusion matrix depicting the performance of models on Data II for three air quality classes: Low, Average, and Severe (2). Note: Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM).